ETR-2026-05 | STAGE 2 RESEARCH

Prepaying Semantics

Bithkuil as a Developmental Substrate for Representation-Efficient Relational and Compositional Learning

Revision 0.4.7
Evidence cutoff: 12 September 2026
Not published; independent review pending.

An explicit semantic developmental substrate is a hypothesis about learning cost, not a demonstrated shortcut to intelligence.

This edition preserves the ambitious integrated-system experiment and gives its components discriminating tests, controls and loss conditions.

Execution boundary: finite-reference engineering tests and tiny smoke runs completed. The main integrated trial, real model-assisted teacher run, independent replication and later-language transfer are not reported as completed.

Table of Contents

Abstract

Large language models learn remarkable abstractions from natural-language token streams, but the training substrate entangles at least two problems: learning useful cognitive relations and learning how human languages irregularly, implicitly, and context-dependently encode those relations. This paper proposes a falsifiable research program around a simple question: **does making semantic structure explicit during early model development reduce the neural capacity and training experience required to acquire generalizable relational and compositional competence?**

We introduce **Bithkuil**, a provisional machine-oriented semantic intermediate representation and developmental curriculum derived substantially from New Ithkuil. Bithkuil is not a proposal to train models on ordinary surface Ithkuil. Its initial form exposes morpheme-level semantic identities, a parsed semantic abstract syntax tree, a dependency-aligned developmental traversal through that structure, explicit default and null distinctions, scope, compositional morphology, controlled semantic minimal pairs, and a supplementary algebra for logic, mathematics, probability, causality, proof, and belief revision. Natural languages are introduced later as codecs over the learned semantic substrate. The central conjecture is that this may change a quantity we call the "semantic reconstruction tax": the difference, if any, in learning burden attributable to recovering task-relevant semantic factors from a representation that exposes them less directly. The quantity is not assumed to be positive. It may be zero, negative, temporary, or dependent on model scale and task regime.

The conjecture is deliberately separated from several easier claims that already have prior art. Large Concept Models demonstrate modeling over higher-level language-agnostic sentence representations; Coconut and related work explore reasoning in continuous latent space; TinyStories directly shows that redesigning a constrained synthetic developmental environment can move coherent language and some reasoning behavior into very small-model regimes, while Phi provides neighboring curated-data precedent [7, 8]; BitNet establishes native low-bit language modeling; and transformers can learn deep recursive formal grammars directly from strings. None of these establishes the proposed mechanism. They instead sharpen the experiment: Bithkuil must outperform matched natural-language and generic typed-representation controls on sample efficiency, parameter efficiency, compositional generalization, robustness, or later language acquisition. If a generic typed representation captures the whole effect, the Ithkuil-specific claim should be rejected while the broader representation-first result survives.

We specify two linked experimental programs. The **integrated demonstration program** intentionally combines a BitNet-style ternary-forward learner trained from scratch, Bithkuil semantic tokens, a prerequisite-aware checkpointed developmental teacher, deterministic world generators and answer oracles, competence/retention gates, and later natural-language transfer. Its purpose is to ask whether the combined system can produce an unusually strong capability-per-compute signal on commodity or modest rented hardware. The **attribution program** then dissects representation, visible sub-symbol structure, within-object ordering, cross-example curriculum topology, teacher policy, compositional supervision, numerical precision, model scale, language transfer, and context economics. Ternary precision is therefore orthogonal in interpretation but no longer required to wait until after a full-precision representation result before the first systems demonstration. KANs, external memory, and OMoC remain downstream. Stage 2 also treats the surrounding research machinery as a routed control plane: deterministic code is preferred for exact operations; local and lower-cost models handle bounded work;

stronger interactive models are reserved for difficult author-in-the-loop synthesis; and reproducible API calls are used when model assistance becomes part of an experiment. This routing is constrained by disclosure sensitivity and by a strict separation between research assistance and intervention-plane assistance so that model strength does not become a hidden treatment confound.

The intended contribution is not a claim of demonstrated efficiency, but a precise and executable experimental object for testing whether semantic developmental structure can shift the learning curves of generalizable relational and compositional competence.

This revision adds an executable finite-world reference, two codec-access arguments, a nonseparable factor-mixing control, a world-sensitive paired-counterfactual endpoint, and a concrete seed-paired SYS design. Local engineering tests and tiny smoke runs establish bounded implementation behavior; they do not establish a Bithkuil learning advantage.

1. Introduction

Contemporary language models are trained on a representational bargain inherited from human communication. Natural language is extraordinarily expressive, socially adaptive, redundant where humans need redundancy, elliptical where shared context permits ellipsis, irregular where history has accumulated irregularity, and ambiguous where people tolerate ambiguity because pragmatic inference usually resolves it. Those properties make natural language an excellent human interface. They do not imply that it is the cheapest possible developmental substrate for an artificial learner.

The distinction matters because a model trained on text must learn at least two overlapping things. It must learn useful structure about the world and about reasoning, and it must learn how that structure is scattered across lexical choices, syntax, morphology, discourse convention, pragmatics, metaphor, omission, and context. A capable transformer demonstrably *can* induce substantial hidden structure from strings. Allen-Zhu and Li, for example, show GPT-style models learning synthetic recursive context-free grammars whose hidden states recover hierarchical structure and whose attention patterns resemble dynamic-programming information flow [15]. The scientific question here is therefore not whether neural language models are capable of discovering structure implicitly. They plainly are.

The question is what that discovery costs.

TinyStories provides a particularly close empirical ancestor for this question. Eldan and Li ask whether small language models fail because coherent language intrinsically requires large scale or because ordinary corpora burden small learners with excessive breadth and diversity [8]. They change the developmental environment rather than the basic GPT-Neo-style learner: a synthetic English story corpus constrained toward a young child's vocabulary and factual world. In that regime they report coherent multi-paragraph generation below 10M parameters, some factual/reasoning/instruction behavior in very small models, and a 1M-35M experimental family trainable on a single V100 within at most 30 hours [8]. This is not evidence for Bithkuil's semantic-reconstruction mechanism or for general tiny-model parity with frontier systems. It is evidence that at least some apparent scale requirements can move dramatically when the learner's environment is redesigned.

We use the term **semantic reconstruction tax** as a name for a quantity to be measured, not as an assumption that a penalty exists. Operationally, it is the difference, if any, in learning burden between matched representations that make the same underlying world and answerable propositions available while exposing task-relevant semantic factors with different locality, regularity, and explicitness. The measured quantity can be positive, zero, negative, or regime-dependent. A negative value would mean that the supposedly less explicit representation is actually cheaper for the tested learner. A value that shrinks with scale would mean that larger models increasingly amortize implicit structure induction.

A successful experiment need not prove that some identifiable set of parameters "stores grammar" or that natural language is intrinsically defective. It need only show, within a specified regime, that models given an explicitly factored semantic developmental substrate reach prespecified generalization targets using fewer training examples, fewer parameters, fewer floating-point operations, or less subsequent language data than matched controls.

Bithkuil is our candidate intervention. It inherits an unusual opportunity from New Ithkuil, John Quijada's constructed language. New Ithkuil makes a large number of semantic distinctions morphologically explicit. Its basic formative morphology jointly encodes categories such as Configuration, Affiliation, Perspective, Extension, and Essence [1]. Its case system marks semantic roles morphologically and

includes explicit mechanisms for case scope [2]. Its verbal morphology expresses categories including valence and mood [3], while its adjunct system can assign sentence-wide semantic or attitudinal scope [5]. Its syntax documentation explicitly notes that because semantic roles are morphologically marked, word order is used primarily for pragmatic functions such as topic and focus [6]. These are properties of a language design, not evidence of machine-learning efficiency. But they make New Ithkuil an unusually rich source of candidate factors for a controlled machine semantic representation.

The project began from a stronger developmental intuition: perhaps the model should not initially learn New Ithkuil's surface language either. A machine learner need not spend its earliest capacity mastering phonotactics, allomorphy, elision, orthography, or other human-facing compression conventions when the experimental target is semantic composition. Instead, the initial Bithkuil representation exposes the parsed semantic structure directly. Every semantically consequential category receives a stable machine identity. Defaults are made explicit. Scope is represented structurally. The structure is not presented as an arbitrary bag of fields: its early traversal is deliberately ordered so that prerequisite scaffolding is established before factors that depend on it. The model first learns to compose and distinguish semantic operators in that scaffolded order; only later does it learn compressed surface realizations as codecs.

This changes the proposal from "use a more precise language" to "change the developmental information geometry." The intended analogy is closer to an abstract syntax tree than to a translation exercise. A compiler does not require its optimizer to rediscover operator precedence from punctuation statistics on every program. The program arrives already parsed into typed structure. Bithkuil asks whether some portion of machine cognition benefits from a similar move.

This paper is a position paper and experimental specification. It makes no claim that a Bithkuil-trained model already exists, that an efficiency gain has been measured, or that New Ithkuil is a complete language of cognition. It instead contributes six things:

1. a sharpened hypothesis about semantic developmental structure and model efficiency;
2. a concrete Bithkuil intervention that can be implemented without conflating surface Ithkuil with semantic structure;
3. a checkpointed developmental teacher architecture in which executable semantics, not a teacher LLM, decides exact correctness;
4. controls designed to distinguish generic structure, Ithkuil-derived factorization, visible sub-symbol semantic composition, curriculum effects, tokenization effects, teacher policy, and numerical precision;
5. a two-lane experimental program: a deliberately integrated systems demonstration followed by causal attribution and scaling work with explicit outcomes that would weaken or kill each major claim;
6. a capability- and disclosure-routed model-assistance methodology that separates exploratory research cognition from matched experimental intervention, records reproducible model-assisted procedures, and uses cross-model review to generate discriminating tests rather than pseudo-independent votes.

2. Representation Is an Upstream Systems Variable

A model's training representation is often treated as plumbing: choose a tokenizer, present enough data, optimize next-token prediction or a related objective, and allow useful latent structure to emerge. Yet representation determines which distinctions are local, which require long-range inference, which transformations can be reused compositionally, which equivalences are obvious, and which variations look unrelated until training has discovered their common cause.

Consider a simple abstract relation such as evidential status. Human languages can express observation, recollection, report, inference, conjecture, certainty, doubt, and source reliability in many ways. Some languages grammaticalize portions of this space; English often uses lexical or syntactic paraphrase. A model can infer that "I saw X," "apparently X," "they told me X," and "X must therefore be true" place different constraints on epistemic status, but the equivalence classes and distinctions are learned across many forms. An explicit semantic representation can instead make the relevant variable a first-class operand.

The important claim is not that explicit representation always wins. Explicit structure can also be wrong, brittle, expensive to construct, or mismatched to the task. A hand-designed ontology can inject the designer's blind spots. A semantic parser can leak labels. A compact representation can erase ambiguity that the model would otherwise exploit. These risks are reasons for matched controls and ablation, not reasons to leave representation unexamined.

Recent work makes the broader research direction difficult to dismiss. Large Concept Models (LCMs) operate autoregressively in a higher-level semantic representation where a "concept" is instantiated as a sentence embedding in the multilingual SONAR space [16]. This demonstrates that a generative model need not use ordinary word/subword tokens as its only modeling unit. Coconut instead feeds continuous hidden states back as subsequent reasoning inputs and reports advantages on selected logical tasks requiring search and backtracking [17]. These systems are not Bithkuil. Their representations are learned dense vectors rather than an explicit typed semantic algebra. Precisely because they differ, they help isolate the research question: perhaps moving above surface language matters, and perhaps the *kind* of structure supplied above language matters too.

Follow-up latent-reasoning work supplies a useful warning. Ozeren and Assenmacher report that latent-space methods can be sensitive to design choices and still lag language-space chain-of-thought on mathematical reasoning [18]. Wang and colleagues find that inference-time scaling recipes transfer only weakly to continuous reasoning and argue that current continuous thought representations lack useful inductive biases for discriminating correct and incorrect trajectories [19]. This suggests a productive possibility without proving it: a machine semantic language may need more than compression. It may need explicit structure that exposes distinctions useful to learning, evaluation, and controlled composition.

A related author-originated hypothesis predating this paper concerns structure *inside* representational units. Chinese orthography offers a natural motivating case because many characters contain reusable sub-character components, including semantic radicals. Si et al. explicitly exploit glyph- and pronunciation-derived sub-character structure in Chinese pretraining and report competitive downstream performance together with substantial sequence compression in some settings [33]. Haslett later reports that token boundaries misaligned with Chinese semantic radicals systematically distort model

representations, and that single-token characters can perform worse than multi-token characters when the latter expose useful sub-character structure [34]. These results do not imply that Chinese is intrinsically a superior language for AI, and they do not explain the efficiency of Chinese AI laboratories. They establish a narrower methodological point: compression and factor visibility are not the same objective. A shorter or more atomic encoding can hide reusable structure that a learner could otherwise exploit.

We reserve **H132, the Visible Subsymbol Semantic Structure Hypothesis**, for the corresponding Bithkuil experiment: holding underlying semantics and component statistics constant, a representation whose visible sub-symbol components consistently expose reusable semantic factors may improve sample efficiency or held-out composition relative to an opaque or semantically permuted encoding. The effect may be absent, reversed, or tokenizer-dependent.

The economic motivation is upstream of accelerator choice. Bithkuil is not primarily asking how to squeeze the same giant training route onto cheaper hardware. It asks whether some of the route itself is avoidable: if a learner spends compute repeatedly reconstructing factorization, prerequisites, and reusable relations that a developmental substrate could expose directly, then part of the hardware demand may be downstream of a representational choice rather than a fixed cost of intelligence. This is an author-motivating conjecture, not a demonstrated explanation of present training economics.

2.1 Equal information, unequal access cost

A lossless representation cannot manufacture task information. Let W be a finite world and query, Y its oracle answer, and R_B and R_C two invertible encodings on the supported domain. Because each encoding determines W, an unrestricted decision rule has the same Bayes-optimal risk from either encoding. This follows by composing any decision rule with the inverse codec. It does not follow that a bounded neural learner, a finite dataset or a fixed optimizer can find those equivalent rules at equal cost.

That distinction is the central mechanism, rather than an embarrassment to it. The compiler may move a relation from a distant, irregular dependency into a stable local coordinate. A bounded model class need not be closed under arbitrary input recoding: a compact function of R_B can correspond to a cumbersome function of R_C. Training also depends on which functions are easy for the optimizer to reach, not merely on which exist in the parameterized class. Bithkuil's proposed advantage is therefore a reduction in the cost of *accessing and reusing* available information. It is not an increase in Shannon information that was absent from the underlying world.

This formulation distinguishes three transfers of burden. A semantic compiler can perform work that the learner would otherwise discover. A curriculum can supply examples in an order that makes a compact rule discoverable. A factorized interface can make a learned rule reusable across combinations. These mechanisms can coexist, but their costs belong to different ledger entries. A system that saves learner compute by spending an unreported fortune on parsing has shifted the bill, not established a total-cost advantage.

For a concrete intuition, a learner can discover a relation from many descriptions of tools being used contrary to their intended purpose, or receive two explicit fields that permit that contrast immediately. The fields do not answer every question about the tool. They give the learner stable operands on which a general rule can operate. Whether this reduces the number of examples required to learn a transferable rule remains the empirical question. The experiment preserves the intended-versus-actual distinction in every codec, rather than granting only the Bithkuil arm extra facts.

2.2 A precise limit: token-renaming symmetry

Consider a lookup embedding E and a permutation pi of token IDs. Replace every input ID i by pi(i), and define E' at row pi(i) to equal the original E at row i. Keep positions, attention masks, architecture and downstream weights unchanged. Every initial embedding vector is then identical, so every subsequent activation and prediction is identical. For a tied output vocabulary, permute the corresponding output coordinates and targets as well. For the four-answer reference, reserved answer IDs remain fixed.

The same argument extends inductively through training when optimizer state is permuted with its parameter rows and all update operations are coordinate-equivariant. Gradients and optimizer updates correspond under the same permutation. In exact arithmetic the trajectories coincide. With independent identically distributed initial embeddings the systems are equivalent in distribution, not necessarily in one unpaired random draw. Floating-point reductions, non-equivariant regularizers, pretrained embeddings or token-specific initialization can break the assumptions and must be named rather than silently ignored.

The delivered reference tests the coupled statement through two AdamW updates in both floating-point and ternary-forward modes. The tested outputs and parameter-coordinate correspondence are bit-identical in the observed environment. This is an implementation invariant, not a claim that arbitrary real-language tokenizers are equivalent.

The consequence is decisive: an arbitrary factor-isomorphic notation cannot be expected to lose merely because its symbols lack Ithkuil-looking names. Donor-specific benefit must reside in the selected distinctions, scope conventions, factor organization or developmental structure, not in a magical advantage of the alphabet. This strengthens the research question. The candidate is a designed learning environment, not a naming ceremony.

3. Why New Ithkuil Is an Interesting Donor Language

New Ithkuil was designed for unusually dense and explicit semantic expression. For our purposes, the attraction is not linguistic exoticism. It is factorization.

3.1 Morphology as a factorized semantic space

New Ithkuil's formative structure exposes mandatory categories that many natural languages encode lexically, contextually, or not at all. Its basic morphology describes Configuration, Affiliation, Perspective, Extension, Essence, Version, Function, Context and other dimensions [1]. Configuration itself factors grouping structure through distinctions involving plexity, separability, and similarity. Affiliation distinguishes relationships among members by purpose, function, or benefit. Perspective distinguishes ways of construing the instantiation or collectivity of a referent.

This is valuable for machine learning because it generates controlled semantic neighborhoods. If two structures differ only in one category, a training generator can construct minimal pairs in which the semantic delta is known exactly. The learner can be asked to identify the changed factor, predict its consequences, compose it with another factor, or recover it from a paraphrase.

3.2 Semantic roles and scope

New Ithkuil's case system explicitly describes case as an indicator of semantic role [2]. It also distinguishes relationships that English often expresses with generic prepositions or noun compounds. For example, the applicative and purposive cases distinguish incidental use from dedicated purpose [2]. Case-scope morphology can specify which neighboring formative a case relation governs. Such distinctions are appealing not because English cannot express them, but because the machine representation can make the distinction local and typed rather than recover it from variable surface constructions.

3.3 Event structure, factuality, and communicative status

The verbal system includes valence and mood [3]. In New Ithkuil, mood explicitly concerns perspectives on an event and degrees of factuality. Elsewhere in its verbal and affix systems, the language provides machinery relevant to illocution, validation/evidentiality, effect, aspect, phase, level, and other semantic dimensions [4]. Adjuncts supply additional sentence-level functions, including bias adjuncts whose semantic scope covers the entire sentence [5].

Again, none of this means New Ithkuil is uniquely correct. It means the grammar contains a large manually designed inventory of distinctions that can be turned into experimental factors.

3.4 Role marking frees linear order to carry developmental structure

The syntax documentation states that semantic roles are marked by case morphology rather than primarily by syntactic position, and that word order is consequently used mainly for pragmatic relations such as topic and focus [6]. For Bithkuil, that freedom should **not** be interpreted as making order semantically or developmentally irrelevant. It makes order available for deliberate use.

A semantic object can preserve stable typed relations independent of any one human surface word order, while the learner still encounters or constructs that object through some sequence of attention. Whatever traversal is chosen determines which scaffolding is already available when a later factor is processed, predicted, contrasted, or attached. The working hypothesis is therefore that early Bithkuil should establish low-ambiguity, high-reuse structure first and introduce more dependent distinctions only after their prerequisites exist.

This makes developmental ordering part of the intervention rather than neutral transport plumbing. Ithkuil's morphology lets the experiment avoid spending linear position primarily on recovering who-did-what-to-whom. Bithkuil can instead test whether using that positional freedom as a scaffold changes learning efficiency or generalization. The benefit is not assumed; matched ordering controls must be able to make this claim lose.

Neighboring semantic-graph work gives a concrete reason not to dismiss traversal as plumbing. DeBenedetto reports that changing among valid AMR linearization orders materially changed AMR-to-text generation performance, with a drop of up to 17.5 BLEU in the studied system [31]. Gao and colleagues report structure-loss accumulation for AMR nodes and edges decoded later in a linearized sequence and improve parsing by using reverse graph linearization as an additional training signal [32]. These results do not establish that scaffold-first Bithkuil is optimal, or even beneficial. They establish a narrower methodological point: semantically equivalent graph content can cease to be learning-equivalent once forced through a position-sensitive sequence model.

3.5 Why not simply train on surface Ithkuil?

Because that would test too many things at once.

Surface Ithkuil includes a human language's phonology, morphophonology, morphological realization rules, written form, and compression conventions. If a model trained on surface Ithkuil outperformed an English baseline, we would not know whether the cause was semantic factorization, token density, regularity, vocabulary size, morphology, sequence length, or accidental benchmark affinity. Conversely, if it underperformed, we would not know whether a useful semantic substrate was being obscured by the difficulty of learning its surface codec.

The proposed first intervention therefore begins one layer below surface language: the semantic parse itself.

3.6 From donor grammar to a dependency map

The production package now contains a 36-entry grammar/source map rather than treating the donor language as a list of interesting labels. Each entry records its official source, interpretation, prerequisites, Bithkuil disposition, implementation stage, acceptance test and unresolved question. The map covers phonology, morphophonology, formative morphology, case, verbal categories, affixes, adjuncts, referentials, specialized constructions, syntax, numbers, script, appendices and the official lexicon. Four entries explicitly identify author-defined formal machinery rather than donor-language facts [1-6, 42-50].

A source dependency is not automatically a learning prerequisite. Phonological alternations may be necessary to parse surface Ithkuil yet irrelevant to the first semantic-only student. Conversely, finite identity, set membership and an explicit scope stack can be necessary to implement a reliable

interpreter even when the donor grammar explains a category without those mathematical preliminaries. The map therefore separates donor reading order, compiler dependency order and the hypothesized student curriculum. Conflating them would turn a chapter sequence into an untested theory of cognition.

The first implemented family is configuration-inspired: finite groups, cardinality, uniformity and connectedness. These are deliberately bounded operational probes, not a claim that an arbitrary graph exhausts Ithkuil Configuration or Affiliation. The second is the distinction between actual application and intended purpose. The official APL and PUR cases supply the donor contrast; the finite-world fields and answer oracles are our implementation [2]. Identity and explicit group membership come first, these two families become available next, and logical composition is introduced after its operands. Evidence-support composition follows its own exact algebra.

Scope prevents several attractive but invalid shortcuts. An affix's attachment position affects its scope; moving it is not necessarily a meaning-preserving permutation. Different concatenation constructions cannot both be flattened into Boolean conjunction. A referential form and a bare entity ID are not interchangeable claims about the full donor language. Likewise, the official lexicon's narrowly specified copular root must not become a universal equality, existence, location and membership operator in the formal engine [4, 5, 43, 47, 48]. The reference implements its own explicitly typed equality instead.

The detailed map also records source questions rather than inventing a repair. Chapter 6 contains inconsistent editorial wording around an otherwise explicit category enumeration. The current finite ABI does not depend on resolving that wording; a future surface parser must resolve it against authoritative examples before claiming donor fidelity [42]. This is a local source question, not a reason to halt the semantic experiment.

The engineering sequence is consequently semantic before phonological: define exact world types and operators; cross-check the oracle; freeze the token allocation; implement reversible renderers; test defaults, nulls and scope; introduce the student and developmental teacher; then implement broader donor morphology and later language codecs. The map preserves deferred grammar instead of pretending the first two families are the entire project.

4. Bithkuil: A Developmental Semantic Intermediate Representation

Bithkuil is a working name for an experimental machine representation. It is not presented as an official variant of Ithkuil and should not be confused with New Ithkuil as a human constructed language.

4.1 Morpheme-addressed identity

Each semantically meaningful Ithkuil-derived morpheme or grammatical value receives a stable machine identity. The identity is not the surface phonological string. Multiple surface realizations can therefore compile to the same semantic item, and one semantic item can be serialized differently by later codecs.

The purpose is to make reusable factors reusable in the literal training representation. A model need not infer from unrelated strings that several expressions instantiate the same semantic operator if the developmental representation exposes that fact directly.

4.2 Semantic AST and developmental traversal first

An early training item might be represented schematically as:

```
FORMATIVE
  ROOT: <concept identity>
  STEM: <value>
  FUNCTION: <value>
  SPECIFICATION: <value>
  CONTEXT: <value>
  CA:
    AFFILIATION: <value>
    CONFIGURATION: <value>
    EXTENSION: <value>
    PERSPECTIVE: <value>
    ESSENCE: <value>
  AFFIXES:
    - TYPE: <semantic operator>
      DEGREE: <value>
      SCOPE: <target>
  CASE_OR_ILLOCUTION: <value>
  VALIDATION: <value>
```

This is schematic. The experimental compiler must follow the current New Ithkuil grammar rather than this illustrative layout when exact morphology matters.

The AST defines the semantic object, but the developmental representation must also define **how that object is built or traversed**. Early Bithkuil should not choose field order alphabetically, historically, or merely because a serializer happens to emit it that way. A factor should normally appear only after the smaller set of relations needed to interpret, test, or constrain it is already available. Among factors whose prerequisites are satisfied, prefer the ones that are low in ambiguity and reusable across many later constructions.

The exact traversal is deliberately not fixed by this paper. It is a design variable to discover and then preregister. A first implementation should preserve the dependency structure explicitly enough to

compare a scaffolded canonical order against arbitrary, shuffled, and deliberately dependency-reversed orders using the same semantic content.

Early examples make defaults explicit instead of relying on zero marking. The learner can therefore distinguish at least:

- an explicit default value;
- a semantically absent field;
- an unknown value;
- an irrelevant or inapplicable dimension;
- information intentionally omitted from a surface serialization.

Only after the learner reliably manipulates the full representation should training introduce compressed encodings and ordinary surface forms.

4.3 Grammar before lexicon

The developmental curriculum begins with relations that can be understood substantially through their relationships to one another: identity and difference, set membership, order, quantity, arithmetic, comparison, implication, conjunction, disjunction, negation, conditional dependence, probability, evidence relation, causal direction, temporal order, and related operators.

Vocabulary is introduced as operands inside this algebra rather than as a giant memorization problem that precedes composition. A small natural-language seed lexicon can help a teacher or mapper traverse existing semantic inventories, but it is not the target representation and should not silently define the learner's ontology.

"Grammar before lexicon" therefore has an inner ordering rule: **prerequisites before dependents**. The first curriculum should be generated from an explicit dependency map rather than a topical chapter order. One plausible boot sequence is: identity/sameness/difference/presence and basic ordering; then logical composition and set-like relations; then quantity, arithmetic, comparison, and temporal order; then relational/event frames and participant roles; then scope, quantification, modality, evidence, probability, causation, intervention, counterfactuals, proof, contradiction, and revision as their prerequisites become available; only then broad lexical/world concepts and later compressed language codecs. This sequence is provisional. If two factors do not depend on one another, their order should be interleaved or experimentally varied rather than canonized by taste.

4.3.1 Prerequisite topology as an empirical object

The dependency map should not become a hand-authored catechism. It is an initial developmental hypothesis. Some candidate primitives may be prerequisites for many later concepts; some may be independent and therefore learnable in parallel; some apparent dependencies may disappear once a better representation is used. The curriculum should therefore be modeled as a **partially ordered frontier**, not merely a fixed numbered syllabus.

A first implementation can annotate each candidate primitive with properties useful for bootstrapping: its conjectured prerequisites, how independently it can be grounded, how objectively competence can be tested, how many later structures it may unlock, how cleanly it supports minimal contrasts, how broadly it transfers, and how entangled it is with concepts not yet introduced. These annotations are scheduling

hypotheses, not semantic truth. Prerequisites constrain eligibility; among currently eligible concepts, training can interleave or adapt based on measured competence and learning progress.

Most importantly, the graph itself can become measurable. For candidate primitives `p_i` and `p_j`, define a preregistered learning-cost measure `C(p_j)` such as examples, optimizer steps, or compute required to reach a held-out competence threshold. Then estimate, schematically:

```
DeltaC(p_j | p_i) = C(p_j | matched control) - C(p_j | prior mastery of p_i)
```

A positive value means prior mastery of `p_i` reduced the measured cost of acquiring `p_j`; zero means no detected prerequisite benefit; a negative value indicates interference under the tested regime. Pairwise and selected higher-order measurements can produce a directed weighted **developmental dependency graph**. That graph should be allowed to disagree with the author's initial boot sequence.

Adaptive curriculum learning itself is established prior art. Graves et al. automatically select a neural-network syllabus from learning-progress signals [35]; Matiisen et al. use a teacher that selects subtasks according to student progress and can revisit degrading skills [36]; Vakil and Amiri combine graph-derived difficulty with model competence when scheduling graph-neural-network training [37]; and Liu and Chen explicitly formulate prerequisite-connected environments as a curriculum graph for active refinement in reinforcement learning [38]. Bithkuil therefore does **not** claim to invent automatic curriculum selection or graph curricula. Its narrower hypothesis is that an explicit semantic factorization may expose the prerequisite structure of cognition-like training objects well enough to make that curriculum graph testable and revisable.

We reserve **H133, the Representation-Exposed Curriculum Topology Hypothesis**, for this interaction: holding semantic worlds and total training budget fixed, a factorized representation that identifies the primitive requirements of each example may enable a competence-gated, topology-aware scheduler to improve acquisition or held-out generalization relative to fixed prerequisite order, difficulty-only adaptation, or a scheduler supplied with a semantically permuted dependency graph. H133 fails if topology awareness adds no benefit, if the same benefit is available without factor visibility, or if adaptive scheduling merely improves local training loss without transferable competence.

The Stage 2 implementation operationalizes this idea as a **developmental teacher system** rather than a static syllabus. Each concept family is represented by a versioned stage contract containing prerequisites, generators, executable answer oracles, training distributions, development probes, sealed evaluations, regression obligations, and promotion thresholds. The learner advances only when the contract is satisfied. A teacher LLM may propose exercises, adversarial variants, explanations, or remediation, but it does not certify exact answers where executable semantics exist. This distinction lets curriculum policy adapt without turning a large pretrained model into an untracked source of ground truth.

Checkpoints become experimental witnesses rather than mere backups. A promoted checkpoint records the learner state before the next developmental intervention, allowing retention, interference, transfer, and remediation cost to be measured against its parent. When a curriculum claim needs cleaner causal evidence, sibling learners can branch from the same checkpoint and receive different pedagogical interventions while sharing the same developmental history up to the branch point.

4.4 Formal cognitive algebra beside linguistic semantics

New Ithkuil is a linguistic system, not a formal replacement for probability theory, proof theory, or causal inference. Bithkuil therefore places formal operators alongside the Ithkuil-derived semantic layer rather than forcing every cognitive relation into a linguistic morpheme.

At minimum, early experiments should provide explicit representations for:

- Boolean and predicate-like logical relations;
- arithmetic and quantitative operations;
- probability and uncertainty;
- derivation/proof objects;
- causal graphs, intervention, and counterfactual relations;
- contradiction and belief revision;
- optimization or preference relations where tasks require them.

Pearl's structural causal framework is an obvious formal donor for the causal portion [25]. The design goal is not to invent a universal algebra in advance, but to avoid asking linguistic morphology to carry formal burdens for which better mathematical systems already exist.

4.5 Grammar as curriculum generator and teacher substrate

A factorized grammar can generate nearly unlimited controlled exercises without requiring open-web text.

Examples include:

- hold a root constant and vary exactly one morphological category;
- classify the semantic delta between two structures;
- compose two operators and predict the resulting interpretation;
- decompose a complex structure into primitives;
- map paraphrases to one canonical semantic structure;
- generate distinct surface paraphrases from one semantic structure;
- detect malformed or contradictory compositions;
- infer a deliberately missing component;
- construct a semantic object one prerequisite-satisfied step at a time;
- predict the next admissible expansion of a partial structure;
- repair a structure whose dependent factor was introduced before its prerequisite scaffold;
- distinguish observation, report, recollection, inference, and conjecture when the representation supports them;
- transform causal, temporal, quantitative, or modal relations while preserving unrelated factors.

This is the central reason "grammar before lexicon" is interesting experimentally. The grammar is not merely something to learn. Once represented explicitly, it becomes a controllable generator of semantic training distributions.

For Stage 2, the canonical training artifact is therefore not one enormous `training-data.txt`. It is a **versioned teacher program**: semantic specification + curriculum topology + scenario generators + answer oracles + representation codecs + evaluation families + checkpoint/promotion policy. Concrete examples are generated products of that program. This matters scientifically because the experiment

can reproduce not only the bytes a student saw, but the rule system that created those bytes and the rule system that judged the student's response.

The teacher program also distinguishes *who may propose work* from *which execution lane should perform it*. Exact computation should remain code/solver work; high-volume bounded pedagogy can use a local or lower-cost model; difficult research synthesis can escalate to a stronger human-in-the-loop model; and calls that become part of the experimental intervention should be frozen into a reproducible model-assistance contract. The routing policy is not itself a cognitive hypothesis. It is an experimental-control mechanism intended to spend strong model cognition where it has leverage without allowing unequal teacher strength to masquerade as a representation effect.

The teacher program should separate proposal from authority. A local general-purpose model can be useful as a curriculum author, adversarial fuzzer, paraphrase generator, translation critic, difficulty designer, and failure-clustering assistant. But whenever a task admits exact evaluation, the authoritative answer should come from executable semantics. Arithmetic can be evaluated arithmetically; Boolean cases can be exhaustively checked or solver-validated; graph relations can be computed from the generated graph; Bithkuil/AST codecs can be subjected to semantic round-trip and property tests. Important generator/oracle pairs should use independent implementations or cross-checks where feasible so that one deterministic bug cannot both create a problem and certify the same wrong answer.

This creates a training loop closer to developmental experimentation than static pretraining: `teach -> checkpoint -> probe -> diagnose -> remediate/rehearse -> regression test -> promote or repeat`. The procedure remains statistical learning. Nothing in the loop makes the student symbolic by fiat; it makes the student's developmental environment and evaluation much more inspectable.

4.6 Natural language as a later codec

One of the stronger hypotheses is that after a model has learned a reusable semantic substrate, learning English or another natural language may become closer to learning a codec between surface forms and already-familiar semantic structures.

Large Concept Models provide a useful neighboring precedent because their concept space is designed to be language-agnostic and supports multilingual generation [16]. But Bithkuil predicts a different mechanism: transfer should arise from explicit shared semantic factors rather than only from alignment in a learned dense embedding space.

This claim is unusually easy to falsify. Pretrain matched models with or without Bithkuil semantic development, then expose them to the same amount of a previously unseen natural language. Measure how quickly they acquire comprehension, generation, and reasoning transfer. If Bithkuil preconditioning does not accelerate acquisition after controlling for information leakage, the codec hypothesis is weakened.

4.7 Executable semantics and the boundary of the first student

The first reference defines a bounded finite-world language with two to six entities, categorical attributes, undirected relations and explicit reports. An input object contains only its ABI identity, world and query. Gold answers, provenance, teacher commentary and evaluation receipts are outside the student-visible object. Every query is a typed abstract syntax tree, not an unrestricted string passed to a language model.

The answer domain distinguishes supported truth, supported falsity, absent support and conflict. Represent an evidence state as two bits: positive support and negative support. TRUE is (1,0), FALSE is (0,1), UNKNOWN is (0,0), and CONFLICT is (1,1). Negation exchanges the bits; conjunction and disjunction use explicitly defined support operations. This is an author-defined operational algebra. It is not Ithkuil Mood, not the donor's Validation category and not a probability estimate. It is also unrelated to the three numerical values used by ternary model weights.

Two executable evaluators check answers using different computational routes: recursive evaluation with graph search, and postorder evaluation with transitive closure. Agreement detects many implementation mistakes. Because both were produced from the same specification in this run, it is not independent epistemic validation of the specification. An external reviewer can still discover that both implemented the wrong intended task. Exhaustive small-domain fixtures and adversarial malformed inputs provide a stronger foundation than teacher confidence, but they do not eliminate that specification risk.

The neural reference is intentionally narrower than the eventual language-model program. It reads a world/query sequence and predicts one of four reserved answers using a tied readout. It is scratch-trained and transformer-based, with a fixed token table and checkpointed optimizer. It is not yet an open-ended Bithkuil generator or a natural-language conversational model. That boundary makes the first semantic experiments inspectable without abandoning the later generative and language-transfer program.

The implemented controls are reversible compact Bithkuil-inspired serialization, controlled-English serialization, a coupled arbitrary token permutation, and a nonseparable mixed-factor code. None is silently labeled full surface Ithkuil, AMR or MRS. Those broader codecs remain concrete extensions with source and round-trip acceptance obligations.

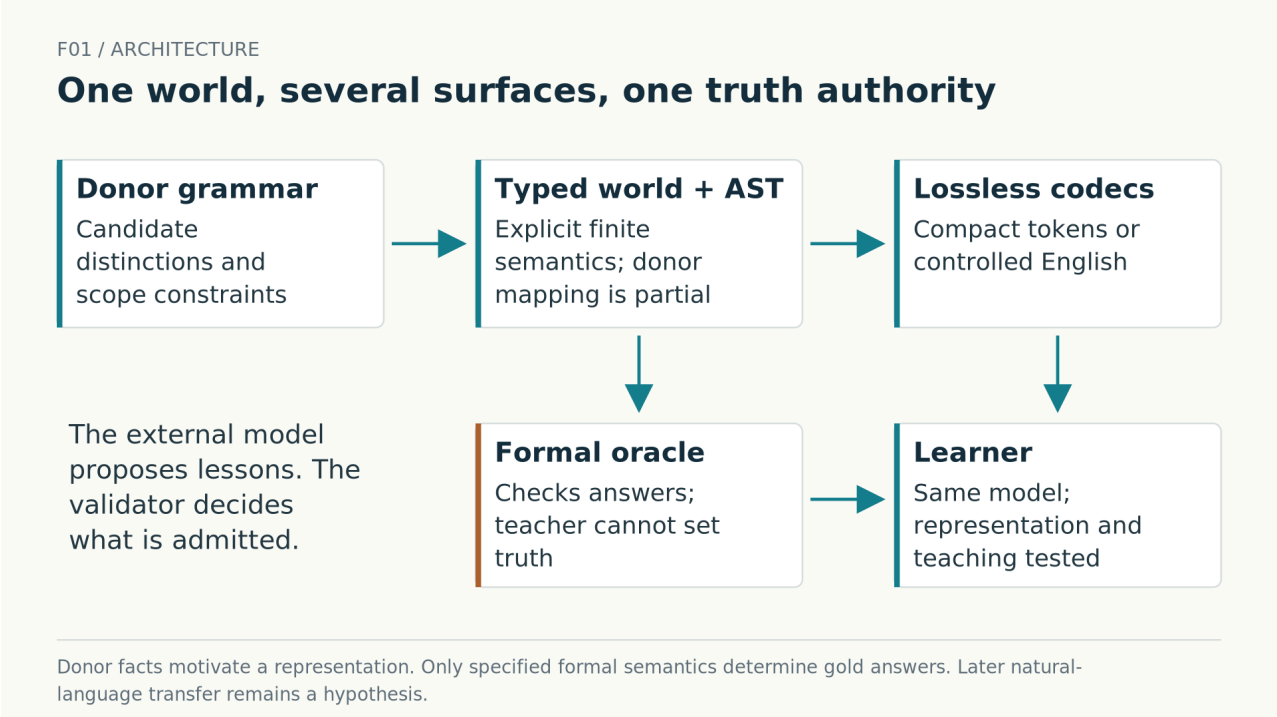

Figure 01. One world, several surfaces, one truth authority. Donor facts motivate a representation. Only specified formal semantics determine gold answers. Later natural-language transfer remains a hypothesis.

4.8 A control that changes factor access without deleting information

An ordinary permutation of independent factor names preserves compositional structure. To disturb local semantic alignment, the reference instead mixes two three-valued fields. If c is color and p is intended role, expose u=c+p and v=c+2p, with all operations modulo three. The original fields are recovered by c=2u-v and p=v-u. Every fact remains available, the number of local field values remains three, and the field count and serialization length remain unchanged.

What changes is the interpretation of a local coordinate. Neither exposed coordinate is simply color or intended role. A task about one original factor now requires combining coordinates. This tests the proposed access-cost mechanism more directly than changing an English label to an arbitrary syllable. It remains a designed operational test: mixing changes distributions and correlations, and a sufficiently capable learner may discover the inverse easily. The expected effect can shrink with scale, disappear after exposure, or reverse for tasks naturally aligned with the mixed coordinates.

This control also locates an important negative outcome. If aligned factor visibility helps only when the target directly names the field but not on held-out compositions or interventions, the gain may be elementary input convenience rather than reusable relational competence. That is still an engineering result, but a weaker scientific conclusion. The primary tasks must therefore require consequences of relations, not merely copying a field value.

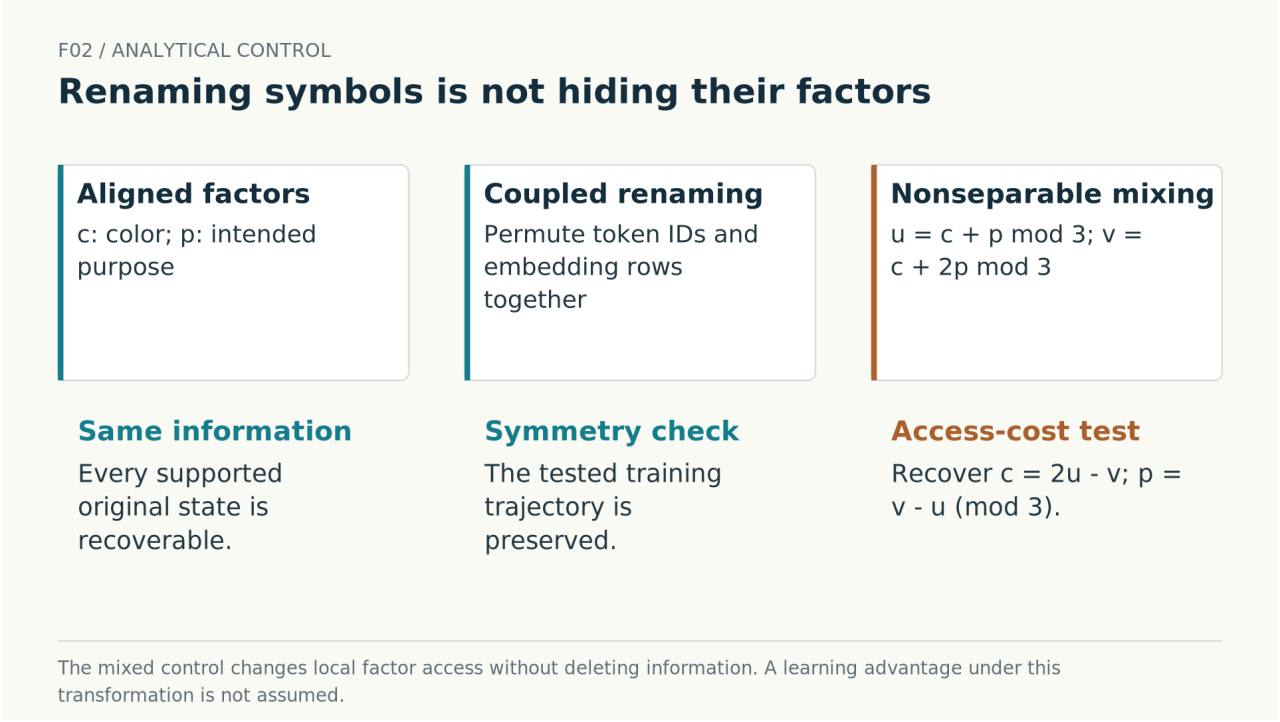

Figure 02. Renaming symbols is not hiding their factors. The mixed control changes local factor access without deleting information. A learning advantage under this transformation is not assumed.

5. The Semantic Reconstruction Tax Hypothesis

The strongest version of the thesis can be stated compactly:

A language model trained from surface text must devote learning capacity to recovering semantic regularities that an explicitly structured developmental representation can provide directly. If that reconstruction burden is substantial, explicit semantic preconditioning should shift the relationship among training experience, parameter count, compute, and generalizable reasoning.

We can express the intuition schematically, without pretending the terms are currently measurable as cleanly separable physical quantities:

```
training burden ~= world/reasoning structure
                 + surface-language structure
                 + inference needed to align the two
                 + irreducible noise/optimization cost
```

Bithkuil attempts to reduce the third term during early development and perhaps part of the second. This does not imply that natural-language knowledge disappears. A useful system still has to learn languages, culture, idiom, factual knowledge, and domain-specific information. The proposed gain is that those later inputs arrive to a model that already possesses more of the relational machinery needed to integrate them.

5.1 Predictions

If a positive semantic reconstruction tax exists in the tested regime, at least some of the following should occur under matched experiments:

1. **Sample efficiency:** Bithkuil-preconditioned models reach fixed compositional-reasoning targets with fewer training examples.
2. **Parameter efficiency:** smaller Bithkuil-preconditioned models match larger surface-language baselines on held-out structural tasks.
3. **Compute efficiency:** fixed-capability thresholds are reached with fewer training FLOPs.
4. **Systematic generalization:** novel combinations of known primitives are handled more reliably than memorized surface patterns predict.
5. **Paraphrase invariance:** semantically equivalent surface expressions map to more stable internal outputs or task behavior.
6. **Transfer efficiency:** a new natural language is acquired faster after semantic preconditioning.
7. **Scaling displacement:** capability versus parameter/token curves differ in slope or intercept, rather than showing only a one-off benchmark bump.

5.2 What would count against it

The thesis should lose force if, under strong controls:

- ordinary token-language models reach the same generalization targets at the same cost;
- curriculum quality explains the entire effect;
- generic typed serialization explains the entire effect and Ithkuil-derived factorization adds nothing;
- sequence/tokenization overhead erases semantic-density advantages;
- teacher-generated semantic labels leak solutions or benchmark structure;

- advantages vanish outside tasks isomorphic to the semantic operators used in training;
- later natural-language acquisition is not accelerated;
- observed gains fail to persist across model sizes, seeds, or task families.

A negative result would still be valuable because it constrains a broad family of representation-first claims that are currently easy to discuss but rarely isolated experimentally.

5.3 Where large gains could come from, and where they cannot

The ambitious version of the thesis concerns combinatorial reuse. A domain with n factors and k states per factor has k to the n possible joint assignments. A learner that treats every joint state as an unrelated atom has no basis for predicting an arbitrary unseen labeling of those states. A task generated by a short reusable composition rule can instead have a description whose size grows with the rule and its factors rather than with the entire assignment table. Making the factors and their operations accessible could help a bounded learner find that shorter description.

This is a mechanism for potentially large savings, not a theorem that Bithkuil converts exponential sample complexity into polynomial sample complexity on every task. The unrestricted lookup-table class and the compositional rule class are different hypothesis classes. A fair experimental comparison gives both arms the same actual task-generating process and measures whether the interface makes its shared structure easier to learn. It must also count the prior work supplied by the designer. Otherwise the experiment gives one arm the solution class and congratulates it for not searching everywhere else.

The strongest developmental prediction is consequently not simply that compact inputs train faster. It is that early mastery of stable operators changes the *marginal cost* of acquiring later compositions: a newly introduced combination becomes learnable from substantially fewer examples because its components are already operational. Measure this with held-out operator combinations, interference branches and transfer to independently rendered descriptions. A collection of isolated field classifiers cannot establish that prediction.

A second route to large gains is reuse across descendants. If one validated developmental substrate and one competent small checkpoint are used in many later tasks, fixed compiler and curriculum costs can be amortized. Let F_B and F_C be one-time construction costs and r_B and r_C the measured per-descendant costs. B becomes cheaper only when F_B + N r_B is below F_C + N r_C. When r_C exceeds r_B, the break-even N is (F_B-F_C)/(r_C-r_B), subject to the chosen accounting units and maintenance costs. There is no break-even advantage from repetition when the recurring B cost is higher and its fixed cost is not lower.

This arithmetic preserves the orders-of-magnitude hypothesis while making it accountable. Very large gains require genuinely large recurring savings, broad reuse, or both. They cannot be established by reporting only the final tiny learner's GPU bill while excluding the teacher, ontology work, rejected generations and failed developmental lineages. Conversely, counting the entire research program anew against every reuse would obscure a legitimate amortized benefit. Both first-use and repeated-use economics should be shown.

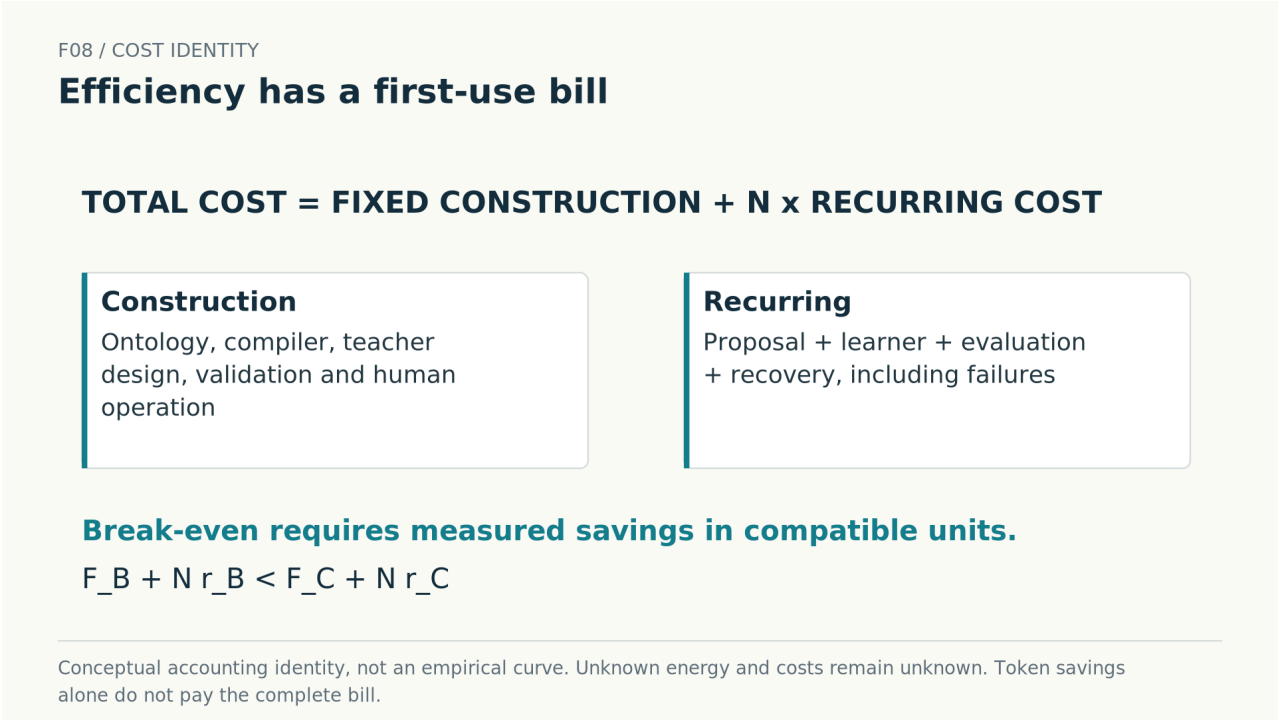

Figure 08. Efficiency has a first-use bill. Conceptual accounting identity, not an empirical curve. Unknown energy and costs remain unknown. Token savings alone do not pay the complete bill.

5.4 The tax is a contrast, not a hidden organ

No experiment here identifies a unique intrinsic quantity called semantic tax independently of its learner, representation, task distribution and budget. Define it operationally as a contrast in resources required to reach a fixed competence criterion under a declared matched design. Different criteria can yield different contrasts. A codec may help short-range factual queries while harming ambiguity-sensitive reasoning; a curriculum may help small models while offering little benefit once a larger model has learned the factors implicitly.

This does not reduce the thesis to the generic statement that representation matters. The specific hypothesis is that externally specified factor identity, scope, within-object dependency traversal and cross-example prerequisites jointly lower the developmental cost of acquiring reusable operators, with later transfer retaining enough of the saving to pay for construction. Each part has a discriminating test. What is refused is only an unmeasurable claim that a particular percentage of a network's parameters must be devoted to an identifiable linguistic tax.

6. Prior Art and the Novelty Boundary

Bithkuil sits in a crowded neighborhood. That is a strength for the experiment and a constraint on the novelty claim. Several mature research traditions already establish pieces of the broader idea that structured or non-token representations can be useful. The paper therefore should not claim to have invented semantic intermediate representations, curriculum learning, latent reasoning, or the idea that model families can inherit prior computation.

6.1 Semantic graphs already externalize meaning structure

Abstract Meaning Representation (AMR), Minimal Recursion Semantics (MRS), UCCA, semantic-role labeling, dependency semantics, and related frameworks have long represented aspects of sentence meaning in structures that abstract away from ordinary surface syntax. AMR in particular represents core sentence semantics as a labeled graph. Bai, Chen, and Zhang explicitly pretrain models over AMR graphs and report improvements on AMR parsing and generation, arguing that text-only pretraining is not ideal for learning graph structure [20]. Hajdik and colleagues show that rich MRS-derived representations can support high-quality neural text generation [21]. More recent work continues to investigate how structured linguistic representations should be integrated with large language models [22].

These lines of work are close enough that any broad claim such as "structured semantics helps neural models" would be neither new nor sufficiently discriminating. Bithkuil asks a narrower developmental question. Instead of using semantic graphs mainly as an auxiliary task, parser target, generation input, or post hoc representation for an already language-pretrained model, the strongest proposed experiment begins with a small learner whose early training world is itself expressed through a typed semantic algebra. Natural language arrives later.

That difference must be tested, not protected rhetorically. AMR and an MRS-like rich semantic representation are therefore not merely background literature; they are serious controls. They should be rendered from the same generated worlds, trained under the same curriculum, and evaluated against the same held-out semantic tasks as Bithkuil. The tested Bithkuil delta is narrower: a denser inventory of explicit grammatical-semantic factors, stable morpheme-addressed identities, explicit default/unknown/omission states, and a supplementary formal cognitive-algebra layer. If an AMR/MRS-style control produces the same gains, the result would support explicit semantic preconditioning while rejecting or sharply weakening the stronger Ithkuil-derived factorization claim.

6.2 Higher-level and latent modeling already escape ordinary token reasoning

Large Concept Models provide direct evidence that autoregressive modeling can operate over sentence-level, language-agnostic concept representations rather than ordinary token sequences [16]. Coconut demonstrates another route: reasoning states can remain continuous and be recursively fed back into the model without decoding each step into language [17]. These systems make it untenable to frame Bithkuil as the first attempt to move computation above surface tokens.

Their differences are nevertheless experimentally useful. LCM concepts are dense sentence embeddings. Coconut states are learned hidden vectors. Bithkuil proposes explicit discrete or

discretizable semantic factors whose identities and composition rules are inspectable outside the weights. Follow-up work on continuous reasoning is especially relevant because it reports that moving into latent space does not automatically solve reasoning: latent methods can remain sensitive to training design [18], and continuous-space inference-time scaling appears to lack some of the inductive structure needed to discriminate useful from incorrect trajectories [19].

This motivates, but does not establish, the Bithkuil wager: perhaps escaping natural-language serialization is not enough. A useful developmental substrate may need explicit structure that exposes what is being combined, what changed, what scope applies, and what kind of uncertainty or relation is present.

6.3 TinyStories is the closest developmental-environment ancestor

TinyStories is unusually important here because it attacks a variable upstream of model size. Its motivating question is whether small-model incoherence reflects an intrinsic scale requirement or the excessive breadth and diversity of ordinary corpora [8]. Its intervention is deliberately developmental: synthesize a much narrower English world with a roughly 1,500-word child-oriented vocabulary while forcing diversity through randomized lexical combinations and story features. The resulting models show coherent constrained-domain language and some reasoning/instruction behavior at sizes far below the regimes that motivated the paper, with the 1M-35M experimental family trainable on one V100 within at most 30 hours [8].

That makes TinyStories a stronger ancestor than a generic ``small model'' citation, but not evidence for Bithkuil's central mechanism. TinyStories still asks the learner to acquire cognition-like regularities through ordinary English strings; it reduces environmental breadth without explicitly supplying semantic roles, scope, epistemic status, formal relations, or prerequisite structure. Bithkuil asks whether reducing *representational reconstruction* after already controlling environmental complexity moves the curve again.

TinyStories therefore becomes both a calibration target and a hard baseline. Before interpreting a Bithkuil failure, the implementation should demonstrate that its small-model harness can reproduce a known constrained-development regime. In the causal experiment, TinyStories-style restricted natural language should sit below the same deterministic semantic worlds used by structured-English and semantic-IR conditions. If Bithkuil beats only a broad or badly matched language baseline but not a TinyStories-style restricted one, the result is primarily an environmental-simplification effect.

There is also an accounting distinction. TinyStories used GPT-3.5 and GPT-4 to manufacture the training environment [8]. The paper does not report the total teacher-side generation cost. Bithkuil's early algebraic worlds can in principle be generated deterministically, but that is an economic hypothesis only after generation, validation, learner training, and evaluation costs are measured separately. A fair end-to-end comparison must not count only the cheap student's GPU bill while ignoring the cost of constructing its childhood.

6.4 Curriculum is an independent causal variable

The Phi family and TinyStories demonstrate that data selection and pedagogical structure can radically change what small models learn at a given parameter budget [7,8]. Formal work provides an even cleaner warning against conflating representation with curriculum. Abbe, Bengio, Lotfi, and Rizk show that a degree-ordered curriculum can improve learning of Boolean monomials in their

generalization-on-the-unseen setting [23]. More broadly, Bengio and colleagues formalized curriculum learning as meaningful ordering of training examples and reported generalization improvements in the studied neural-learning settings [29]. Dekker, Otto, and Summerfield later found that human compositional generalization depends on training organization and showed that a modified neural model could reproduce sensitivity to those curricula in their experimental paradigm [30].

Accordingly, Bithkuil cannot compare a carefully staged semantic curriculum against an ordinary shuffled web-text baseline and attribute the full difference to representation. Content-matched curriculum controls are mandatory. The experiment must cross *what is represented* with *the order and supervision through which it is learned*.

There are two ordering variables here, and they must not be collapsed: the order in which concepts/examples enter the curriculum, and the order in which factors of one semantic object are exposed or constructed. The first is developmental sequencing across training experience; the second changes the local conditioning structure of each training item. Bithkuil now treats both as experimental variables.

6.5 Orthographic compositionality is a natural precedent, not the claim

Chinese writing provides a useful natural example of why representational atoms should not automatically be treated as semantically atomic. Si and colleagues show that Chinese pretrained models can exploit sub-character encodings derived from glyph or pronunciation structure; their sub-character tokenizers preserve competitive downstream performance while shortening some tokenized sequences substantially [33]. Haslett finds that when token boundaries fail to align with semantic radicals, GPT-4, GPT-4o, and Llama 3 show systematic distortions on radical and semantic categorization tasks; in the reported comparison, single-token characters can underperform multi-token characters because the longer segmentation exposes useful internal form structure [34].

These findings are not evidence that Chinese orthography causes the recent efficiency of DeepSeek, Moonshot, Z.ai, or any other research ecosystem. Architecture, data, optimization, hardware constraints, distillation, mixture-of-experts design, reinforcement learning, and engineering practice are major confounds. The paper therefore treats the author's older Chinese-character intuition as a source of **experimental design**, not a retrospective causal explanation.

The useful abstraction is that **compression which destroys reusable factorization may be worse than a slightly longer representation that exposes it**. Bithkuil should therefore optimize neither token count nor semantic density in isolation. The more defensible target is useful reusable structure per unit of learning cost. This motivates H132 and a controlled synthetic rendering experiment in which visible compositional symbols are compared with opaque and semantically scrambled controls.

6.6 The actual novelty target

The most defensible novelty target is the conjunction of the following features as one falsifiable developmental intervention:

1. a morpheme-addressed semantic inventory derived substantially from New Ithkuil's unusually explicit factorization;
2. a canonical machine semantic AST that exposes those factors before surface compression;

3. an explicit dependency map and scaffolded developmental traversal through that AST rather than treating serialization order as arbitrary plumbing;
4. explicit treatment of default, absent, unknown, irrelevant, inapplicable, and intentionally omitted states;
5. a formal cognitive-algebra layer for logic, mathematics, probability, causality, proof, and belief revision rather than forcing all cognition into linguistic morphology;
6. a grammar-before-large-lexicon curriculum built from controlled one-factor semantic deltas and compositional exercises;
7. delayed natural-language acquisition as a codec/translation problem over an already trained semantic substrate;
8. matched controls designed specifically to separate generic structure, Ithkuil-derived structure, visible sub-symbol factorization, within-object ordering, curriculum topology, token length, teacher leakage, numerical precision, and scale.

Even this conjunction is presented as a proposed experimental object, not as an established patent or priority claim. A fuller novelty search may narrow it further.

6.7 Current competitors sharpen the mechanism

Shai and colleagues' 2026 work on factored representations is particularly relevant counterpressure. Their transformer experiments study learning latent factor structure, and their analysis distinguishes compact factored predictive representations from a larger joint space. This establishes a competing route: a learner can discover factors without receiving a Bithkuil-like explicit interface. It also exposes a condition under which forced factorization is inappropriate, because dependencies can require joint information [39]. Bithkuil therefore predicts an advantage in *cost or robustness under specified conditions*, not an exclusive ability to represent factors.

The correct response is a correlation stress test. Train on varying degrees of factor dependence while holding the oracle task fixed. Compare independent-factor assumptions, explicit joint relations and the generic typed control. If a factorized substrate systematically suppresses dependencies the task needs, that is a design failure, not an anomalous data point to exclude. A successful substrate should make both factors and their relations available, rather than confuse decomposability of representation with statistical independence of the world.

Rajaraman and colleagues provide theoretical support for the importance of curriculum in a specific compositional reasoning setting. Their 2026 analysis separates direct reasoning from an interactive teaching arrangement with strong oracle structure [40]. That result motivates explicit attention to the teacher's powers. It does not prove that a binary verifier, a small local LLM or the present finite-world policy inherits the same learning guarantee. The proposal must specify which information the teacher supplies and price that intervention.

Recent work on compositionality also distinguishes structure in inputs from structure in labels and learning dynamics [41]. The inspected abstract is sufficient to motivate separate audits of input factorization and target supervision, but not to import uninspected numerical claims. A model can exploit structured labels even when the proposed input mechanism contributes little. This is why answer-only and factor-supervised arms have separate supervision budgets.

The updated continuous-reasoning literature reinforces the difference between a suggestive mechanism and a dependable interface. Coconut's current revision remains a precedent for hidden-state reasoning

rather than explicit typed semantics [17]. Wang and colleagues' published 2026 analysis reports that available diverse trajectories do not straightforwardly become gains under process/outcome reward selection in continuous space [19]. Bithkuil's exact algebra and oracle are a different proposed intervention on that bottleneck, not a demonstrated solution to it.

Finally, automated curriculum and curriculum refinement are established neighboring research programs [35-38]. Bithkuil's novelty cannot be a scheduler that selects lessons from a graph. The candidate contribution is the coherent coupling of a donor-informed semantic ABI, accessible factor structure, two distinct ordering mechanisms, a truth-constrained adaptive teacher, checkpoint-local experimental branching and a reproducible systems test. Whether that coupling creates unusual capability per resource remains open to the experiments below.

7. Experimental Program

The program should first establish whether the integrated system helps against a credible matched baseline, then determine what contributed. A positive SYS result is a systems finding even before component attribution; it must not be presented as proof of every bundled mechanism.

7.0 Calibration: reproduce the TinyStories-scale regime before testing Bithkuil

TinyStories remains the closest external calibration ancestor, but Stage 2 no longer requires a full-precision TinyStories reproduction to occur before any Bithkuil development. Instead, maintain two related calibration duties. First, reproduce or faithfully approximate at least one published TinyStories-scale reference point when practical, expanding toward the released 1M, 3M, 9M, and 28M families [8]. Second, and more important for the integrated demonstration, train a **matched constrained-language baseline on the same ternary student architecture and checkpointed teacher harness** used by the Bithkuil treatment. The former checks historical plausibility; the latter prevents the systems result from being dismissed as merely a BitNet effect. If neither lane can produce sensible constrained-language behavior, repair the harness before interpreting Bithkuil results.

Record learner architecture, parameter count, optimizer, context, tokenizer, examples/tokens, wall time, FLOPs where estimable, device, and peak memory. Keep **learner-training cost** separate from **corpus/curriculum-generation cost** and **evaluation cost**. The paper's GPT-4 grading can be reproduced as a secondary historical comparison if available, but Bithkuil's primary scientific endpoints should remain deterministic held-out semantic and compositional tasks rather than dependence on a frontier-model judge.

This calibration also gives the scale experiment a grounded starting ladder. The first question is not whether a 300M Bithkuil model can do something interesting. It is whether representational differences are already visible where TinyStories shows that meaningful developmental phenomena can be studied cheaply.

7.1 Phase 0: build the world before training the learner

The first implementation phase requires no paid large-model training. Build a deterministic corpus generator and evaluation harness around small synthetic semantic worlds. Each world should have a known underlying relational structure and multiple renderings derived from the same world state.

The matching target is **world-state information equivalence**, not identical representational accessibility. Every condition must preserve the same underlying entities, relations, truth conditions, and answerable propositions. The treatment is allowed to change how locally, regularly, or explicitly those facts are exposed, because that accessibility is precisely what the experiment is testing. Rather than pretending the serializations are informationally identical in every practical sense, report their byte/token length, vocabulary size, entropy, explicit field count, factor locality, serialization depth, and traversal policy.

At minimum, each semantic item should be renderable into:

- a TinyStories-style vocabulary-bounded simple-natural-language rendering from the common world state;

- pedagogically structured natural language with the same vocabulary/content budget;
- an ordinary-language rendering where useful as a broader ecological control;
- a generic typed semantic representation;
- an AMR-style graph and, where practical, an MRS-like rich semantic control derived from the same generated world;
- an arbitrary factor-isomorphic symbolic notation whose labels have no Ithkuil semantics;
- the Ithkuil-derived Bithkuil representation;
- the canonical semantic AST serialization;
- selected surface-New-Ithkuil realizations where exact grammar is available and useful as a control.

The underlying world state, train/test split, task targets, and gold answers must remain identical across renderings. Representation-specific byte/token length, vocabulary size, entropy, explicit-field count, serialization depth, and traversal policy should be measured rather than assumed away. A representation that exposes a factor directly is not disqualified for being more accessible; that accessibility is the independent variable whose learning consequences we want to measure.

This phase also produces the exact grammar map needed for Bithkuil. The current schematic AST is not enough. New Ithkuil's morphology, case, verbs, affixes, adjuncts, referentials, specialized constructions, syntax, numbers, and lexicon must be mapped into a machine-readable inventory with explicit treatment of defaults, scope, allomorphy, and composition. The compiler should preserve the distinction between the *semantic factor* and the *surface morphophonological realization*.

7.1.1 The developmental teacher is a program, not a corpus

Stage 2 treats the training environment as a versioned executable system. At minimum it contains:

- the canonical semantic AST/IR and token ABI;
- a conjectured prerequisite graph and eligibility frontier;
- deterministic semantic-world and task-family generators;
- representation renderers/codecs;
- independent answer oracles or cross-checks for exact domains;
- training, development/probe, and sealed-test generator families;
- stage-specific mastery, retention, transfer, and integrity thresholds;
- rehearsal/remediation policy;
- checkpoint lineage and promotion receipts;
- a complete cost ledger for generator, learner, and evaluation work.

A general-purpose local LLM may sit beside this machinery as a **pedagogue**, not above it as an oracle. It may propose lesson families, paraphrases, adversarial cases, or explanations and may cluster student failures. Every accepted example retains generator/proposer provenance and validation state. Exact claims are rejected if the executable oracle disagrees.

7.1.2 Checkpointed developmental training

The initial student architecture remains fixed within one developmental lineage. "Growing" the learner first means increasing competence, not silently adding width or layers between curriculum stages. A typical lineage is:

```
GENESIS
 -> CP-001 primitive distinction/identity
```

```
-> CP-002 order/equivalence
-> CP-003 set/relation structure
-> CP-004 Boolean/compositional operations
-> ...
```

Each transition executes a stage contract:

1. verify prerequisites and token/codec integrity;
2. generate and independently validate the training tranche;
3. train for the bounded tranche or until a preregistered stop condition;
4. checkpoint weights, optimizer state, curriculum/version identities, seeds, exposures, and cost;
5. run current-stage mastery probes;
6. run regression probes over earlier critical competencies;
7. run held-out transfer probes;
8. diagnose failures and generate targeted remediation if policy permits;
9. promote the checkpoint only when the contract passes.

The model may therefore stop and resume between conceptual stages without treating every stage as a new model. Separate 3M/10M/30M/etc. lineages can be compared later without confounding ordinary cognitive development with architectural surgery.

To keep that claim honest, the **token ABI and embedding-table shape should normally be frozen at lineage genesis**. Primitive IDs that are introduced only at later stages can be preallocated but withheld from training until they become eligible. Otherwise, adding token rows between checkpoints quietly changes parameter count and architecture. If a genuinely new primitive requires an ABI expansion beyond reserved capacity, start a versioned descendant lineage or treat the architectural change as its own intervention rather than calling it ordinary curriculum progression.

7.1.3 Three evaluation pools and generator-family holdout

Fresh random instances from the training generator are not enough. Use three evidence pools:

- **TRAIN:** generated material the learner may consume directly;
- **DEV / PROBE:** unseen cases that may influence remediation or later curriculum decisions;
- **SEALED TEST:** examples and, where feasible, entire generator families that cannot influence curriculum design, thresholds, or remediation before milestone evaluation.

The sealed lane should deliberately test transfer under changed operands, surface ordering, compositional depth, distractors, incomplete or contradictory premises, novel combinations of mastered primitives, and cross-representation rendering. A learner that masters only the statistical fingerprints of one generator has not demonstrated the generalization claimed here.

7.1.4 Checkpoints as experimental witnesses and branch points

Checkpoint lineage enables more than recovery. For each capability `c` and parent/child checkpoints `CP_n` and `CP_{n+1}`, record a developmental delta such as:

```
Delta_c = score_c(CP_{n+1}) - score_c(CP_n)
```

This exposes acquisition, forgetting, interference, and unexpected transfer. Cases solved by the parent but missed by the child are high-value remediation data.

When the order of two lessons or teacher policies is itself under test, branch sibling learners from the same parent checkpoint. For example, one branch may learn temporal ordering before causal direction while another receives the reverse intervention. Shared ancestry reduces noise from unrelated developmental history and turns the checkpoint tree into a causal experimental surface.

7.1.5 Integrated demonstration lane versus attribution lane

The first attention-worthy prototype is intentionally a **systems demonstration**, not a complete factorial proof. Its experimental unit combines:

ternary student + Bithkuil semantic ABI + developmental teacher + deterministic oracles + checkpointed competence gates

with at least one credible matched ternary baseline. The minimum baseline uses the same student architecture/precision family, underlying generated worlds, checkpoint harness, and cost accounting while replacing the Bithkuil developmental representation with a constrained natural-language or otherwise simpler control. A cheap factor-isomorphic/opaque control should be added if resources permit.

A large and transferable advantage in this integrated lane is sufficient to justify escalation and deeper attribution. It is **not** sufficient to claim which component caused the advantage. Access to the LLM pedagogue must be matched between the primary SYS treatment and baseline wherever its assistance could alter lesson quality; treatment-specific pedagogue assistance would be another bundled intervention that must be reported explicitly. The existing P0/P1/P2/P3 experiments remain the forensic program for separating representation, ordering, curriculum policy, supervision, and numerical precision after a systems signal exists.

7.1.6 Capability-routed research and model-assistance control plane

The developmental system should not use one model indiscriminately for every operation. Stage 2 instead treats model cognition as a routed resource. A task is classified by at least four variables: cognitive difficulty, disclosure sensitivity, reproducibility requirement, and whether it belongs to the research/engineering plane or the experimental intervention plane.

The **research/engineering plane** includes literature synthesis, architecture, code generation, debugging, manuscript refinement, exploratory failure analysis, and adversarial experiment design. These operations may use heterogeneous models and reasoning strengths because they do not directly define the student's treatment. Prefer deterministic code when it can solve the task exactly; then local or inexpensive models for bounded work; then stronger hosted reasoning when the lower tier fails an acceptance criterion or the synthesis stakes justify escalation. Model/provider names are configuration rather than protocol identity: the durable scientific object is the functional role plus the recorded provider/model/version and procedure used for a particular run.

The **experimental intervention plane** includes any model assistance that can change lessons, curriculum selection, remediation, feedback, accepted training examples, paraphrase distributions used for training, or evaluation behavior. Here opportunistic routing is constrained. Primary compared conditions must receive a matched `model_assistance_contract` or model assistance must itself be declared as an intervention. The contract records the allowed role, provider/model/version or preregistered equivalence

class, reasoning configuration, prompt/protocol identity, budget ceiling, and required validators. This prevents a stronger hidden teacher from contaminating a representation comparison.

Disclosure sensitivity is separate from cognitive difficulty. A low-difficulty task containing protected research context should remain local or on an explicitly approved trusted lane even if an inexpensive external model could solve it. Where a third-party reviewer is scientifically useful, construct a **declassified microtask** containing only the minimum necessary projection, replace sensitive identifiers with nonce labels where feasible, and preserve a private reversible mapping to the canonical internal object. The returned analysis remains non-authoritative and must pass the same semantic or formal validators before it affects accepted experimental material.

Cross-model review is used to create procedural diversity, not vote-count evidence. Reviewers should propose competing mechanisms, identify supporting and contradicting observations, state predictions under each explanation, and design the cheapest discriminating experiment. Agreement among models sharing training ancestry or prompt framing is not counted as independent empirical corroboration.

This creates an outer research flywheel around the developmental teacher:

```
result/failure
  -> structured evidence packet
  -> routed mechanistic/adversarial analyses
  -> competing explanations
  -> discriminating experiment design
  -> implementation + formal QA
  -> sibling checkpoint branches where causal
  -> new evidence
  -> bounded curriculum/specification revision
```

The flywheel may substantially reduce human coordination cost or increase scientific throughput, but those process gains are not evidence for the semantic-reconstruction tax. They are part of the laboratory used to test it.

7.2 P0: representation isolation

The first **attribution** experiment should hold architecture, optimizer, initialization family, training examples, underlying semantic worlds, cross-example curriculum order, within-object traversal policy, teacher policy, and total information exposure as constant as practical while varying representation. It may follow, rather than precede, the integrated systems demonstration described above.

The critical comparison is not merely English versus Bithkuil. It is a ladder:

TinyStories-style constrained language -> structured constrained language -> AMR/MRS-style semantic control -> generic typed IR -> arbitrary factor-isomorphic IR -> Bithkuil IR -> canonical semantic AST

Surface Ithkuil can be included as an additional condition, but it should not be privileged. A Bithkuil or AST advantage over surface Ithkuil would support the decision to expose semantic structure before human-facing compression. A Bithkuil advantage over ordinary English but not generic typed IR would reject the Ithkuil-specific claim while preserving a broader representation effect. A gain shared by every explicit structure condition would implicate factorization more generally.

Primary endpoints should emphasize held-out composition rather than training reconstruction. Candidate endpoints include exact semantic-state reconstruction, systematic recombination of known

primitives, semantic-delta prediction, scope resolution, counterfactual and causal transformations, malformed-composition detection, proof-step validity, and transfer to structurally novel combinations.

7.3 P1: developmental ordering

Bithkuil now makes two distinct but related order claims. Test them separately. We reserve **H131, the Scaffolded Semantic Traversal Hypothesis**, for the within-object claim so it cannot be silently conflated with H019, the broader Curriculum-Topology Hypothesis.

Across the curriculum, hold representation and within-object traversal fixed and compare prerequisite-ordered training against matched shuffled, frequency-driven, and difficulty-matched curricula. The prerequisite curriculum should begin with relations whose semantics can be established through interaction among a small primitive set, then expand the operand vocabulary while preserving those operations.

Inside each semantic object, hold the object, fields, examples, and cross-example curriculum fixed while varying how the same structure is traversed. At minimum compare: (1) a dependency-aligned canonical order, (2) one fixed arbitrary order, (3) per-example shuffled order, and (4) a deliberately dependency-reversed order. Where practical, include a graph-native or otherwise less order-dependent control receiving the same structural information.

For a causal sequential learner, a serialized object $S = (s1, ..., sn)$ is learned through conditionals of the form $p(si \mid s<i)$. The same complete semantic object can therefore present easier or harder local learning problems depending on which prerequisites are already available in the prefix. Bithkuil's ordering hypothesis is that a scaffolded traversal can reduce unnecessary ambiguity and make later composition easier for a finite learner. This is a hypothesis, not a property declared true by the representation. The AMR linearization results in [31,32] motivate taking this ablation seriously, but they do not choose the winning Bithkuil order in advance.

A positive representation result that disappears under curriculum matching is a curriculum result, not a representation result. A gain that appears only under dependency-aligned traversal is an ordering result. If arbitrary or reversed traversals perform equally well, the claim that canonical semantic order is developmentally important should be weakened or dropped.

The fixed prerequisite curriculum is only the first H019 test. A later gated branch should ask whether progression improves when the learner advances on **competence** rather than elapsed examples, and whether the initial dependency map can be corrected from measured transfer. This branch is where H133 becomes distinguishable from ordinary curriculum ordering: representation is useful not only because it presents factors explicitly, but potentially because it tells the scheduler which factors an example depends on.

7.4 P1C: visible sub-symbol semantic structure

H132 tests whether reusable structure exposed *inside* representational units changes learning. Use the same generated semantic worlds and compare at least five renderings:

1. opaque atomic symbols whose internal form carries no reusable semantics;
2. alphabetic/random labels built from semantically meaningless components;
3. compositionally constructed symbols whose visible components correspond consistently to semantic factors;

4. an information-preserving, nonseparable factor-mixing control using the same local component inventory and sequence lengths, with marginal/correlation statistics measured explicitly; independent factor renaming is reserved for a symmetry control;
5. the explicit typed Bithkuil/AST representation.

Where practical, add a sixth condition in which a composite state is aggressively collapsed into one opaque token after the learner has already mastered its factors. This separates the developmental value of exposed composition from the later efficiency value of learned compression.

The critical comparison is condition 3 versus condition 4. If semantically aligned composition improves held-out recombination, one-factor delta recognition, novel component combinations, or later codec acquisition after accounting for measured surface statistics, that supports the claim that visible reusable structure matters. If both perform similarly, any gain is more likely due to generic regularity, segmentation, or sequence economics. If condition 1 wins, exposed factorization may itself be a burden for the tested learner.

Do not use real Chinese versus English as the primary causal test. Real languages differ in too many dimensions. Chinese is motivating prior art; the synthetic renderer is the discriminating experiment.

7.5 P1D: empirical prerequisite topology and competence-gated curriculum

Run this branch only after the basic representation/curriculum harness is trustworthy. Hold the selected representation, semantic worlds, evaluation set, and total budget fixed while comparing:

1. a fixed author-conjectured prerequisite order;
2. a competence-gated scheduler that may choose among currently prerequisite-satisfied frontier concepts;
3. an adaptive difficulty/learning-progress scheduler without semantic dependency information;
4. the same topology-aware scheduler supplied with a matched but semantically permuted dependency graph;
5. an empirical-topology condition allowed to update candidate prerequisite edges only from preregistered transfer/learning-cost measurements.

The primary outcome is not which scheduler minimizes training loss fastest. Measure examples/steps/compute to held-out competence thresholds, systematic recombination, transfer to newly unlocked concepts, and retention of earlier skills. For candidate edge `p_i -> p_j`, estimate the change in `p_j` acquisition cost after controlled prior mastery of `p_i`, with matched extra-training controls. Only stable effects across seeds and selected higher-order combinations should modify the candidate dependency graph.

This creates a possible **self-characterizing bootloader** without pretending the first dependency map is correct. We supply the initial conjecture; experiments may confirm, delete, reverse, or add edges. If the adaptive and permuted-graph conditions are equivalent, H133 is weakened. If difficulty-only adaptation matches topology-aware scheduling, the result supports adaptive curriculum learning without evidence that explicit semantic dependencies add value.

7.6 P2: compositional supervision

Cross the best representation conditions with and without explicit compositional exercises. The treatment may include:

- semantic minimal pairs differing in one factor;
- compose/decompose objectives;
- canonicalization of paraphrases to one semantic object;
- generation of varied surface forms from one semantic object;
- missing-factor inference;
- contradiction detection;
- operator substitution while preserving unrelated fields;
- causal interventions and counterfactual edits;
- proof or derivation checking.

This phase tests whether any gain arises from the representation itself or from training objectives that force the model to use its factorization.

7.7 P3: numerical precision interaction

The Stage 2 systems demonstration may begin with BitNet-style ternary/1.58-bit forward-path training from scratch rather than waiting for a full-precision representation result. This is an author-level sequencing decision motivated by the intended commodity-hardware demonstration, **not** a claim that ternarity removes semantic ambiguity or automatically makes training proportionally cheaper. BitNet b1.58 demonstrates that ternary weights can match same-scale full-precision language models in reported perplexity and downstream performance while improving inference-side efficiency [9]. The 2B4T technical report extends native low-bit evidence to a larger open model trained on four trillion tokens [10].

P3 therefore becomes an **attribution** experiment after the integrated signal is characterized: cross the strongest representation/control pair with BF16/FP16 and ternary/1.58-bit training under matched teacher policies and budgets. Estimate separate main effects for representation and precision plus their interaction. Ternary weights are not symbolic truth values; the intended conjunction is high precision about semantic distinctions with low precision in the learned numerical substrate. A null or negative precision interaction leaves any representation/teacher result interpretable and may simply narrow the economic stack.

7.8 P4: scale and slope

A single endpoint win is weak evidence for a scaling claim. Repeat the strongest conditions across several deliberately small model sizes and training budgets. Fit capability against parameters, examples/tokens, and training compute. The interesting outcome is a reproducible change in intercept or slope: a fixed target reached earlier, a smaller model matching a larger control, or improved extrapolation as semantic complexity grows.

Use TinyStories as the first practical scale landmark rather than jumping immediately to hundreds of millions of parameters. A provisional ladder near 1M, 3M, 9M, and 28M parameters is attractive because it overlaps a published regime in which coherent constrained-language behavior and scaling patterns

were already measurable [8]. Exact architecture matching and confirmatory sizes remain Stage-2 preregistration decisions.

Precommit the possibility that the advantage is largest only for small learners. If the Bithkuil or generic-IR benefit systematically shrinks with model scale, report that as a primary result gradient rather than treating it as a failed study. It would identify a developmental regime in which explicit structure is useful while supporting the alternative that larger models increasingly amortize structure induction from ordinary strings. Later experiments that mix natural-language data with the semantic curriculum should likewise report mixture composition and repetition rather than treating raw token count as a complete exposure statistic.

7.9 P5: natural-language transfer

After semantic preconditioning, train matched models on the same bounded quantity of English and at least one additional natural language not used in the semantic-development phase. Measure acquisition curves, not merely final loss.

The codec hypothesis predicts that a model with a reusable semantic substrate should need less language-specific data to map new surface forms onto familiar relations. Stronger tests deliberately choose languages whose morphosyntax differs substantially from English. The experiment should include comprehension, generation, semantic parsing, paraphrase invariance, compositional reasoning, and retention of previously learned semantic tasks.

A transfer advantage could still arise from generic structure or multilingual leakage, so the same generic-IR controls remain necessary.

7.10 P6: context and working-state economics

The later architecture makes strong claims about semantic compression, reusable cognitive objects, and reduced context burden. Those should not be inferred from elegance. Measure them.

For matched tasks, record serialized tokens/bytes, semantic-state count, active-context length, KV-cache footprint, retrieval/reference overhead, accuracy, and latency. If a canonical semantic object can be content-addressed and reused, compare the complete cost of reference plus retrieval against simply repeating the text. A shorter symbolic representation that requires expensive reconstruction downstream may not be an economic win.

7.11 Teacher contamination and corpus provenance

Teacher models may help propose Ithkuil mappings, generate paraphrases, create exercises, search adversarial cases, cluster failures, or suggest remediation, but their outputs cannot silently become ground truth. For exact domains, gold world states, semantic relations, train/test membership, and answer keys must be generated or validated executably. Prefer deterministic or grammar-based renderers for structured conditions so that one treatment is not quietly receiving a stronger teacher.

The teacher system follows the rule **LLM proposes; formal machinery disposes**. Which LLM proposes is a routed implementation choice on the research plane, but a frozen experimental variable on the intervention plane. A local model, an API-hosted model, or another approved provider may occupy the pedagogue/critic role if the applicable disclosure and role contract allows it; compared cells may not

receive unequal effective teacher assistance invisibly. The selected general-purpose model is a fallible pedagogue whose accepted outputs carry provenance and validator receipts. The canonical semantic AST/IR and executable domain semantics outrank its narration. Where feasible, do not use the exact same code path to generate a task and certify its answer: arithmetic, Boolean, graph, codec, and other important primitive families should receive an independent evaluator, solver, exhaustive checker, round-trip invariant, or second implementation. A deterministic system that self-certifies the same bug is not reliable supervision.

If an LLM is used to produce natural-language paraphrases, generate them from the common world state under a blinded, condition-independent procedure. Preserve teacher identity/version, prompt or generation protocol, generated artifact, validation method, and acceptance/rejection state in provenance. Test the resulting corpora for target leakage and systematic complexity differences before training.

The strongest early tasks should be solvable from the generated world state without depending on a teacher's answer. This prevents a powerful teacher from smuggling target benchmark behavior into the student curriculum.

Once a DEV/PROBE failure has influenced remediation, that probe belongs to development history and cannot be cited as untouched evaluation. Milestone claims should rely on the sealed lane, preferably including generator families held out from both the student and the curriculum controller. This protects the system from iterating until it fits its own worksheet factory. Capability routing does not relax this boundary: SEALED examples, family implementations, outputs, and failure clusters stay unavailable to routed reviewers and external microtasks until the permitted milestone opening. Once opened and analyzed, they may inform a later version or lineage but cannot be used to repair and then re-label the same milestone as untouched.

The identities, versions, and hashes of sealed generator families should be committed before serious training begins and kept inaccessible to the curriculum controller until the permitted milestone opening. A secret random seed is weaker than a held-out generative procedure because the training generator may still imprint family-specific shortcuts.

Checkpoint promotion should also be transactional. A stage does not pass merely because training loss fell. Promotion policy should jointly require current-stage mastery, bounded regression on critical prior skills, semantic/token-map integrity, zero unresolved oracle disagreements on accepted exact tasks, and the preregistered transfer criterion. Threshold values remain Stage 2 preregistration decisions rather than numbers chosen after outcomes are visible.

For economic claims, preserve distinct research/control-plane, generator, learner, and evaluation ledgers; §7.12 defines the combined accounting. TinyStories is a useful warning: its learner training is deliberately cheap, but its synthetic childhood was generated by GPT-3.5/GPT-4 [8]. Bithkuil should not claim a cheaper developmental route merely because upstream intellectual or generation work is easy to omit.

7.12 Research and model-assistance cost accounting

Stage 2 expands the economic boundary beyond the v0.4.5 generator/learner/evaluation ledger. The complete program-level accounting is:

```
C_program = C_research + C_generator + C_learner + C_eval
```

C_research includes human-supervised model reasoning, literature/synthesis work, architecture, implementation assistance, adversarial review, experiment design, debugging, and interpretation. It can be subdivided into subscription-bundled interactive use, metered API use, local inference, third-party inference, human time, and ordinary implementation compute. The four top-level buckets are mutually exclusive for any one accounting view: production curriculum generation/validation belongs in C_generator, student optimization in C_learner, and probe/sealed scoring in C_eval. A shared operation must be split by a declared allocation rule or reported separately rather than counted twice. A subscription or local model may have a low marginal cash price without being literally costless; broad economic claims should distinguish marginal, amortized, and sunk-resource views when that distinction matters.

This accounting has two purposes. First, it prevents the experiment from claiming an inexpensive student while hiding an expensive intellectual childhood-construction process. Second, it lets the program test whether reusable semantic specifications, Skills, validators, generators, and governance behave like capital: high initial method-construction cost may be amortized across many stages, lineages, languages, or subsequent research programs.

For causal treatment comparisons, common research-plane overhead should not be confused with condition-specific exposure. Any model assistance that directly changes one condition belongs in that condition's intervention ledger and must be matched or explicitly manipulated. The scientific claim remains capability per measured experience/compute under a declared accounting boundary, not capability per conveniently omitted expense.

7.13 Result gradients

Precommit how outcomes change the theory:

- **Bithkuil > all matched controls:** strengthens both explicit-semantic-development and Ithkuil-derived-factorization hypotheses.
- **TinyStories-style constrained language closes most of the gap to Bithkuil:** environmental simplification, not semantic factorization, explains most of the observed efficiency; narrow H001/H003 accordingly.
- **Bithkuil/typed IR still separates from TinyStories-style constrained language under matched worlds and budget:** strengthens the claim that representation contributes beyond corpus simplification.
- **Bithkuil ~= generic typed IR > natural language:** supports explicit factorization but rejects or sharply narrows Ithkuil specificity.
- **prerequisite curriculum > shuffled, representations and within-object traversal similar:** supports cross-example curriculum topology rather than representation.
- **dependency-aligned traversal > fixed arbitrary, shuffled, and reversed traversal with representation/curriculum matched:** supports the semantic-ordering hypothesis.
- **semantically aligned compositional symbols > matched semantically permuted symbols:** supports H132, visible reusable sub-symbol structure, rather than generic regularity alone.
- **compositional and permuted symbols perform similarly:** weakens H132 and points toward segmentation/regularity or other representation effects instead.
- **traversal order matters only for the causal transformer but not a less order-dependent control:** narrows the result to architecture-specific learning dynamics rather than a universal property of semantic representation.

- **competence-gated topology-aware curriculum > fixed prerequisite order:** supports adaptive developmental scheduling beyond a hand-authored syllabus.
- **topology-aware > difficulty-only adaptive curriculum:** supports H133's claim that explicit dependency information adds value beyond generic learning-progress scheduling.
- **true dependency graph ≈ semantically permuted dependency graph:** weakens H133 and suggests adaptation/difficulty, not semantic prerequisite topology, explains the gain.
- **empirically updated graph improves later acquisition across seeds:** supports treating developmental topology as a measurable object; individual edges still require their own evidence.
- **compositional objectives cause the gain:** supports supervision design more than representation identity alone.
- **surface Ithkuil underperforms AST/Bithkuil:** supports separating semantic structure from human-facing compression.
- **all structured conditions lose after length/information matching:** weakens the reconstruction-tax hypothesis for the tested regime.
- **advantage shrinks monotonically with model scale:** rejects a scale-invariant efficiency claim but identifies a developmental regime in which explicit structure helps smaller learners; this is a primary discriminating outcome, not post hoc damage control.
- **language transfer is null:** weakens the codec hypothesis without invalidating within-representation efficiency findings.
- **integrated ternary Bithkuil teacher >> matched ternary constrained-language baseline on sealed transfer and cost-to-competence:** justifies escalation and forensic attribution, but does not by itself identify which bundled component caused the gain.
- **integrated system wins only on same-generator exercises but not sealed generator families or cross-representation transfer:** treat the result as curriculum/generator overfit, not evidence of broadly reusable cognition.
- **integrated system reaches high current-stage mastery while prior competence erodes:** the teacher has a retention/interference defect; do not promote the developmental checkpoint as successful.
- **LLM pedagogue outputs materially determine gold answers or leak sealed targets:** invalidate affected runs until supervision provenance is repaired.
- **primary conditions receive unequal routed model assistance:** invalidate the causal comparison or reclassify teacher/model assistance as an explicit intervention.
- **cross-model critics agree without discriminating evidence:** treat the agreement as analysis convergence, not independent confirmation; run the discriminating experiment.
- **declassified external analysis changes materially when restored to full context:** restrict the conclusion to the declassified projection and do not generalize it to the protected task.
- **research/control-plane cost dominates learner savings:** the semantic-learning result may remain valid, but narrow any end-to-end economic claim and report amortization assumptions explicitly.
- **low-bit interaction is null or negative:** leaves representation findings intact and demotes the combined efficiency stack.

The experiment should kill the smallest justified claim, not the entire research program at once.

7.14 The first concrete SYS design

The locally locked SYS-01 plan instantiates the integrated trial without pretending its broad attribution program has already run. Eight seed pairs compare the compact Bithkuil-inspired representation with the same-world controlled-English representation. Both use the same ternary-forward student, the same

topology-guided teacher framework and the same permitted local-pedagogue assistance. Holding that framework constant makes C a credible systems comparator. It also means this first contrast does not independently estimate the effect of the shared teacher or ternary precision.

The reference-scale architecture has six layers, width 192, six attention heads, context 384 and 512 preallocated token rows, totaling 2,828,736 trainable parameters. Each seed receives 4,096 steps of sixteen examples, or 65,536 semantic exposures. Repeated examples and rehearsal count. Every 64 steps produces a complete learner checkpoint, followed by DEV mastery, retention and transfer decisions. The architecture does not grow during a developmental lineage. A future scale ladder uses distinct lineages rather than silently changing the model behind a competence claim.

The primary effect is the B-minus-C difference in macro accuracy over eight world-sensitive family-by-stage strata, averaged across paired training seeds. A proposed practical signal requires at least five percentage points of average gain, a paired 95% interval above zero, no severe family-level regression and valid integrity/custody boundaries. These are design choices, not constants extracted from prior literature. The package records their exact values so they cannot migrate after outcomes are visible.

Teacher responses can diverge because the two representations produce different errors. That divergence is part of the integrated adaptive-system effect. The subsequent matched-example replay freezes the same accepted world/query sequence for both representations, asking what remains without adaptive curriculum divergence. The primary comparison is thus not falsely called a pure representational effect, nor is it postponed until every interaction has been eliminated.

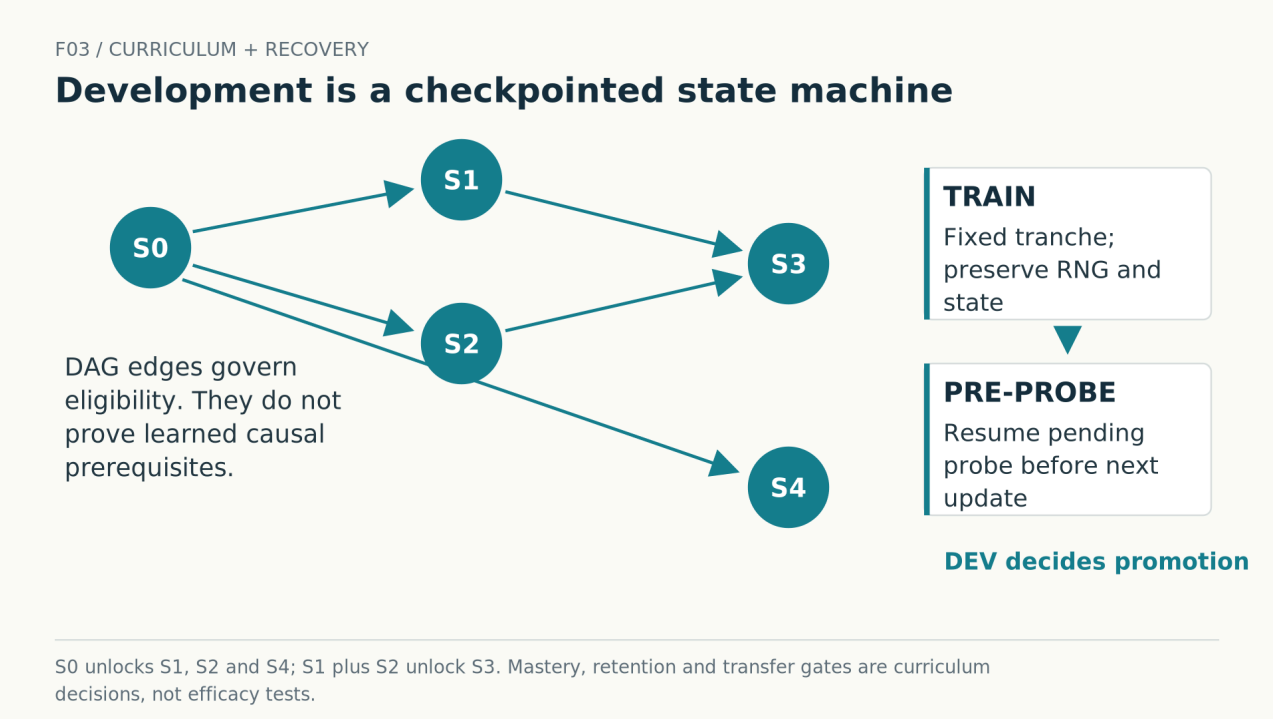

Figure 03. Development is a checkpointed state machine. S0 unlocks S1, S2 and S4; S1 plus S2 unlock S3. Mastery, retention and transfer gates are curriculum decisions, not efficacy tests.

7.15 World-sensitive counterfactual pairs

A semantic benchmark can look relational while permitting a query-only shortcut. Equality of two explicitly named IDs and the cardinality of an explicitly listed group are sometimes answerable without inspecting the supplied world. Such tasks can calibrate primitives, but they should not carry the main claim of learned world relations.

The reference consequently searches for a legal change to the world that preserves the query and flips the correct answer. A changed attribute, relation edge or report-support bit becomes a stored intervention witness. Stages one through four enter the primary set only with such a witness. Stage zero remains a separately reported diagnostic. The evaluator predicts on both members and records whether both answers are correct. A deterministic query-only predictor necessarily gives the same output to the unchanged query and cannot solve an opposite-label pair.

Witness construction is deliberately bounded. It searches the declared one-field/edge/report-bit intervention class; failure to find a witness is not a theorem that no more complex intervention exists. Rejection changes the benchmark distribution and is logged. This converts a plausible shortcut concern into an executable endpoint rather than a disclaimer appended to a favorable result.

The design uses 512 ordinary cases in each of ten family-by-stage cells, totaling 5,120 per final checkpoint. Excluding stage zero leaves 4,096 primary ordinary cases and 4,096 corresponding counterfactual pairs. Those thousands of items provide precision about each trained system's behavior. They do not create thousands of independent training replicates: the primary uncertainty remains across eight paired seeds.

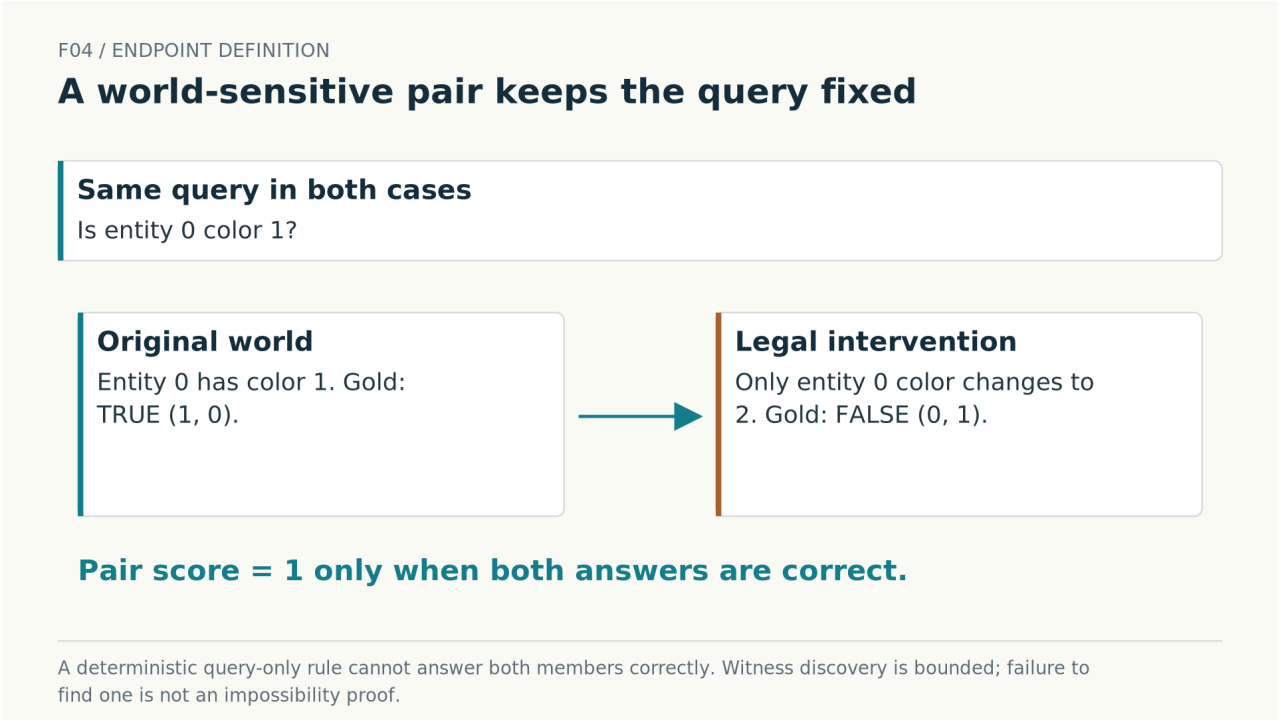

Figure 04. A world-sensitive pair keeps the query fixed. A deterministic query-only rule cannot answer both members correctly. Witness discovery is bounded; failure to find one is not an impossibility proof.

7.16 Engineering evidence obtained in this revision

This production run implements and tests a finite reference. Twenty-nine test methods pass, including oracle/codec properties, malformed-input rejection, source/config boundaries, token-renaming symmetry, checkpoint corruption, ordinary interrupted recovery, recovery from the pre-probe boundary, explicit condition branching, local-only assistance boundaries and paired-counterfactual evaluation. The property suite includes 360 generated world/query objects across three families and five stages, with eight codec/order round trips per object, in addition to exhaustive small graph and support-algebra fixtures. These are software invariants on the tested domain, not learned generalization results.

Four scratch-training smoke runs exercise Bithkuil/CNL crossed with ternary-forward/floating-point mode. Each has 53,408 parameters and executes four optimizer steps on sixteen semantic examples. The compact runs consume 958 nonpadding input tokens and the controlled-English runs consume 1,982 for the corresponding generated examples. No run is long enough to establish the main developmental effect, and none meets the full promotion requirement. These token counts demonstrate a concrete serialization difference, not lower total training cost or lower energy.

A frozen tiny Bithkuil/ternary checkpoint was evaluated through forty ordinary cases and thirty-two counterfactual pairs. The small evaluation is retained to verify the scoring and immutable-model paths, with `scientific_efficacy_claim: false`; it is not used to choose an effect size or claim superiority over an unscored comparator. Earlier development smoke runs remain historical artifacts, while the manuscript summary refers only to the final-code locked run family.

The code supplies a loopback-only local-model adapter with append-only proposal events and identity hashes. No real Qwen weights, local server or paid model endpoint were executed in this session. Two important boundaries remain empirical rather than textual: an independent operator has not yet regenerated the package, and an independent custodian has not yet prepared unseen family implementations. The main trial remains unrun. This precise boundary makes the implementation useful now without converting readiness into a result.

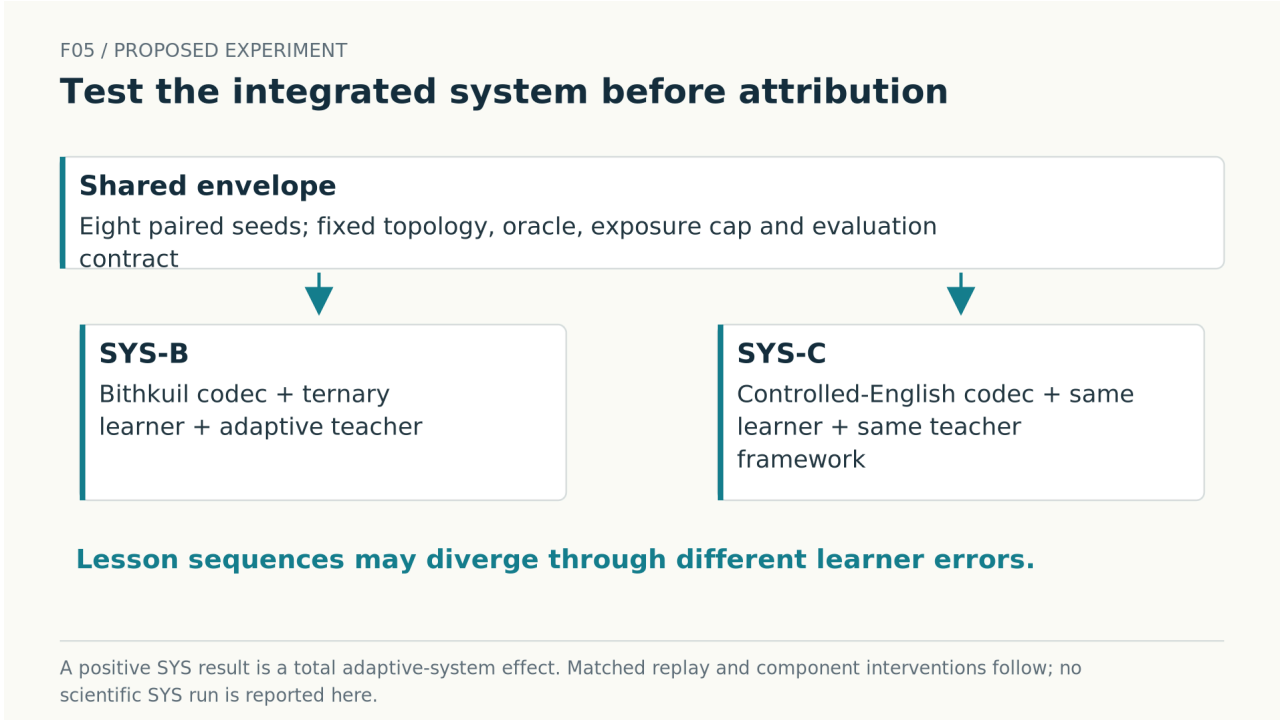

Figure 05. Test the integrated system before attribution. A positive SYS result is a total adaptive-system effect. Matched replay and component interventions follow; no scientific SYS run is reported here.

Figure 07. What was actually executed. Executed on the recorded CPU environment. These receipts validate a small implementation path, not comparative efficacy, native low-bit efficiency or independent replication.

7.17 Reproducibility as a tested transfer of competence

A runbook is successful when a receiving operator can regenerate the behavior without inheriting the author's unstated decisions. The handoff therefore specifies semantic and token ABIs, actual model shape, exact dependencies observed, config identities, generators, dual oracles, accepted assistance fields, counters, stage gates, seeds, parent hashes, split commitments, commands, negative fixtures and recovery points. A source lock binds executed engineering results to the final reference code.

The clean-room acceptance procedure starts from source in a fresh directory, not from delivered weights. It requires all negative fixtures to fail for the right reasons, a regenerated smoke lineage, tensor-identical recovery in the same pinned environment, an opened local engineering seal, deliberate tamper detection and an explicit explanation of what would still be needed for confirmatory custody. A hash-locked receiving-platform wheelhouse remains necessary for portable binary reconstruction; a list of package versions alone is not that wheelhouse.

This requirement is integral to the thesis. If the supposed saving depends on a uniquely skilled operator continuously repairing the curriculum or interpreting ambiguous grammar, hidden human expertise is part of the system. The relevant quantity is not only whether the original author can produce a good run, but whether the developmental substrate has externalized enough expertise for a second operator to reproduce it at an accountable cost.

8. Downstream Architecture: What Becomes Interesting Only If the Core Signal Exists

Bithkuil arose inside a larger Entif research lineage concerned with explicit semantic objects, content-addressed cognition, modular computation, memory, and orchestration. Those ideas are relevant motivation and downstream design space, but they must not become circular evidence for the representation thesis.

8.1 Model inheritance and developmental families

Function-preserving model expansion has substantial prior art. Net2Net transfers a trained network into a larger network while preserving its function [13]. Knowledge Inheritance uses pretrained models to improve the training efficiency of larger successors, while HyperCloning initializes a larger language model from a smaller pretrained one and reports training savings [14]. Progressive model-family training likewise reports reduced aggregate compute when models are expanded rather than independently trained from scratch [24].

If Bithkuil produces a small model with robust semantic competence, these methods suggest an obvious next experiment: can the cognitive substrate survive model growth? The strong version is not merely weight inheritance. It is **semantic-interface inheritance**: larger descendants preserve measurable competence at the same named semantic operations while acquiring greater lexical/world knowledge or representational resolution.

That is currently a hypothesis, not a result.

8.2 KANs as exposed local functions

Kolmogorov-Arnold Networks replace scalar edge weights with learned univariate edge functions [11]. That makes them interesting for a future Bithkuil setting in which operands are already semantically typed and local transformations can be meaningfully isolated. But current small-language-model evidence counsels against treating KANs as a magical wholesale replacement for transformer feed-forward blocks: recent experiments find useful auditability but no consistent benchmark, quality, or latency advantage over strong MLP baselines [12].

The scientifically cleaner question is therefore conditional: if Bithkuil yields stable typed operands, do some relations benefit from inspectable, locally mutable learned functions? Until that precondition exists, KAN belongs downstream.

8.3 Content-addressed cognition and external memory

Entif's Cognitive Tiles, Tapestries, and Rosetta work provide a pre-existing project lineage for representing reusable semantic objects outside opaque parametric state [26-28]. In that lineage, immutable content-addressed tiles can be composed into larger semantic working sets, with provenance and explicit relations preserved rather than overwriting source evidence. These are project-design and provenance sources, not independent evidence for the Bithkuil hypothesis. The important scientific boundary is that an architecture designed around such objects does not prove that models learn better when developed on Bithkuil.

If the representation experiment succeeds, however, a stable semantic object creates a practical bridge between parametric learning and external memory: the same canonical structure can be learned, referenced, retrieved, versioned, exchanged, or compiled into a working context. This could allow some knowledge to remain external while the parametric model specializes in reusable relational operations. Existing memory-augmented and conditional-memory work makes this a credible architecture question, but not yet a Bithkuil result.

8.4 OMoC as downstream orchestration

Ontological Mixture of Concepts (OMoC) predates the present Bithkuil discussion in the Entif research lineage [28]. Its relevant future question is whether semantically identified cognitive functions, memories, or reusable structures can be activated and composed conditionally rather than every task invoking every capability.

Bithkuil should precede that question experimentally. There is little value in building elaborate semantic routing around primitives that have not yet demonstrated useful developmental or computational behavior. If P0-P6 reveal stable operators, reusable semantic states, or separable competence, OMoC becomes a concrete orchestration experiment. If they do not, OMoC must find its justification elsewhere.

8.5 Rosetta as lifecycle and provenance substrate

Rosetta is useful here because its project lineage provides machinery for stable identity, provenance, immutable source/evidence separation, ambiguity preservation, and governed extension [27]. A Bithkuil experiment can use those capabilities to identify semantic objects, source corpora, hypotheses, experimental runs, and derived artifacts without claiming that Rosetta itself proves the representation thesis.

The correct integration posture is therefore conservative: Bithkuil begins as an experimental representation/profile/Pack or bridge around Rosetta, not a silent redefinition of Core semantics. A successful experiment may later justify a standardized interoperable projection. A failed experiment should leave Rosetta's core meaning intact.

9. Limitations and Failure Modes

9.1 Hand-designed semantics can move the problem rather than solve it

An explicit representation is not free information. Someone must choose its factors, define their meanings, map examples into them, and decide what distinctions deserve primitive status. Bithkuil could merely relocate difficult inference from the learner into the corpus compiler or human designer. That may still be economically useful, but it is a different claim from discovering a universally superior cognitive representation.

The experiment must therefore account for representation-construction cost separately from learner training cost.

9.2 Ithkuil is a donor, not an oracle

New Ithkuil was designed as a human constructed language with extraordinary semantic precision and compression goals. Its categories are intellectually rich, but they are not guaranteed to align with optimal machine-learning factors. Some distinctions may be too fine, some may be culturally or linguistically specific, and important machine-relevant variables may lie outside the grammar.

This is why generic typed and factor-isomorphic controls are central rather than ceremonial.

9.3 Explicitness can destroy useful ambiguity

Natural-language ambiguity is not always a defect. It can defer commitment, preserve multiple interpretations, support creative analogy, or allow learning from partially specified evidence. A naive semantic compiler that forces one interpretation too early could make the training data cleaner while making the learner epistemically worse.

Bithkuil therefore needs explicit representations of unknown, underspecified, ambiguous, conflicting, and inapplicable states. When multiple interpretations are live, the corpus should preserve alternatives rather than collapse them into one teacher-selected label.

9.4 Sequence length, traversal, and vocabulary economics can reverse the expected gain

A representation can be semantically dense but token-inefficient under a particular tokenizer, or structurally elegant while requiring many serialized fields. Conversely, ordinary language contains powerful compression learned through centuries of use. Claims about efficiency must use measured byte/token/context/compute costs for each condition.

A scaffolded traversal can also make local next-step prediction easier without producing more reusable cognition. Lower training loss or perplexity is therefore not sufficient evidence for the ordering hypothesis. The ordering condition must improve preregistered held-out composition, semantic transformation, robustness, transfer, or another competence measure that cannot be explained merely by an easier local sequence factorization.

Likewise, shorter tokenization is not automatically better. The Chinese sub-character literature shows why token boundaries can expose or conceal useful structure [33,34]. A Bithkuil condition that wins only because it uses fewer tokens has demonstrated an engineering advantage, not necessarily better semantic development; a condition that uses more tokens but exposes reusable factor boundaries may still be cognitively cheaper. Report both costs rather than collapsing them into one notion of compression.

9.5 Synthetic worlds can overfit the theory

A generated benchmark built from Bithkuil primitives could trivially reward Bithkuil. The evaluation suite therefore needs several layers: tasks generated from the same formal system, held-out compositions, independently specified formal tasks, natural-language transfer, and later external benchmarks that were not used to design the semantic inventory.

9.6 Architecture dependence

A transformer may benefit from one serialization while a recurrent, state-space, graph-native, or other architecture behaves differently. The first experiment should use a conventional small transformer for comparability, but positive results must eventually be tested for architecture dependence before being generalized to "machine cognition."

9.7 Defining cognition operationally

The word *cognition* can become an escape hatch for vague claims. Here it should be operationalized as families of measurable competence: relational composition, systematic generalization, inference under explicit uncertainty, causal/temporal transformation, proof-like manipulation, transfer, and efficient acquisition. Fluency, factual recall, and benchmark scores can be measured too, but none is allowed to stand in for the whole construct.

9.8 The strongest efficiency claims may fail

Orders-of-magnitude sample or parameter improvements are intentionally preserved as bold hypotheses because they are worth testing, not because they are already probable. Smaller improvements, null results, or regime-specific reversals are all plausible. The experiment is valuable precisely because it converts a seductive architecture story into something that can lose.

9.9 A deterministic teacher can be deterministically wrong

Executable supervision is stronger than an unverified LLM answer key, but it is not infallible. A shared implementation defect can corrupt both generated examples and expected answers. High-value primitive families therefore require independent checks, property tests, or alternate implementations where practical. Any discovered oracle defect must taint all downstream curriculum artifacts and checkpoints that depended on it until lineage is re-evaluated.

9.10 The integrated demonstration intentionally underdetermines attribution

The first Stage 2 systems lane bundles ternary training, Bithkuil representation, developmental ordering, deterministic supervision, and checkpointed promotion. A spectacular result in that lane would be strategically and scientifically interesting, but it would not prove that Bithkuil morphology, ternarity, or any single teacher mechanism caused the effect. The paper must distinguish **demonstration of a regime** from **attribution of a mechanism**. P0-P3 remain necessary if the project later makes component-level causal claims.

9.11 Capability routing can hide experimental asymmetry

A sophisticated router can make one condition better supported than another without anyone explicitly choosing a biased baseline. If the Bithkuil treatment receives stronger curriculum generation, more capable remediation, or more expensive reasoning than the primary control, the resulting comparison no longer isolates the intended treatment. Intervention-plane assistance therefore belongs in the preregistered condition contract, not in an invisible infrastructure layer.

9.12 Cross-model agreement is not independent evidence by default

Different model APIs can still share training sources, architectures, benchmark culture, prompt framing, or common generated context. Multiple agreeing model reviews may increase confidence that an argument is legible or robust to some procedural variation, but they are not equivalent to an independent empirical replication. Their strongest use here is to generate distinct failure theories and discriminating experiments.

9.13 Declassification is lossy

Removing protected project context from an external microtask can alter the problem. A reviewer may miss a dependency that was deliberately withheld, or perform better because distracting context disappeared. The package must preserve both the canonical private task and the declassified projection so the external result is interpreted only at the scope actually reviewed.

9.14 Falsification must have local consequences

The twenty-entry hypothesis registry distinguishes the broad reconstruction-tax idea, generic factor visibility, donor specificity, two ordering mechanisms, teacher assistance, low precision, transfer, total economics and truth/custody boundaries. These are not twenty aliases for one desired answer. A generic typed schema matching Bithkuil is a loss for donor specificity, even if it is a win for representation-first learning. A token-isomorphic control matching Bithkuil is expected under the symmetry argument, not a failure of semantic structure. A topology intervention failing does not negate a successful within-object traversal result.

Likewise, a system win that disappears when assistance is matched loses the claimed representational interpretation. A compact codec that reduces context but not sample complexity can remain useful while the sample-efficiency hypothesis weakens. A source-independent custodian exposing failure on new graph families downgrades generalization even when in-domain accuracy is high. An unresolved oracle

discrepancy invalidates affected exact-answer evidence regardless of how plausible the model's explanations sound.

The strongest efficiency hypothesis is not protected by endless relabeling. If competent baselines repeatedly match or beat the proposed substrate across predeclared scales and realistic total-cost accounting, the tested reconstruction-tax mechanism should be rejected or sharply narrowed. What remains would be a transparent engineering language and experimental harness, not proof that the original ambition was secretly achieved in another form.

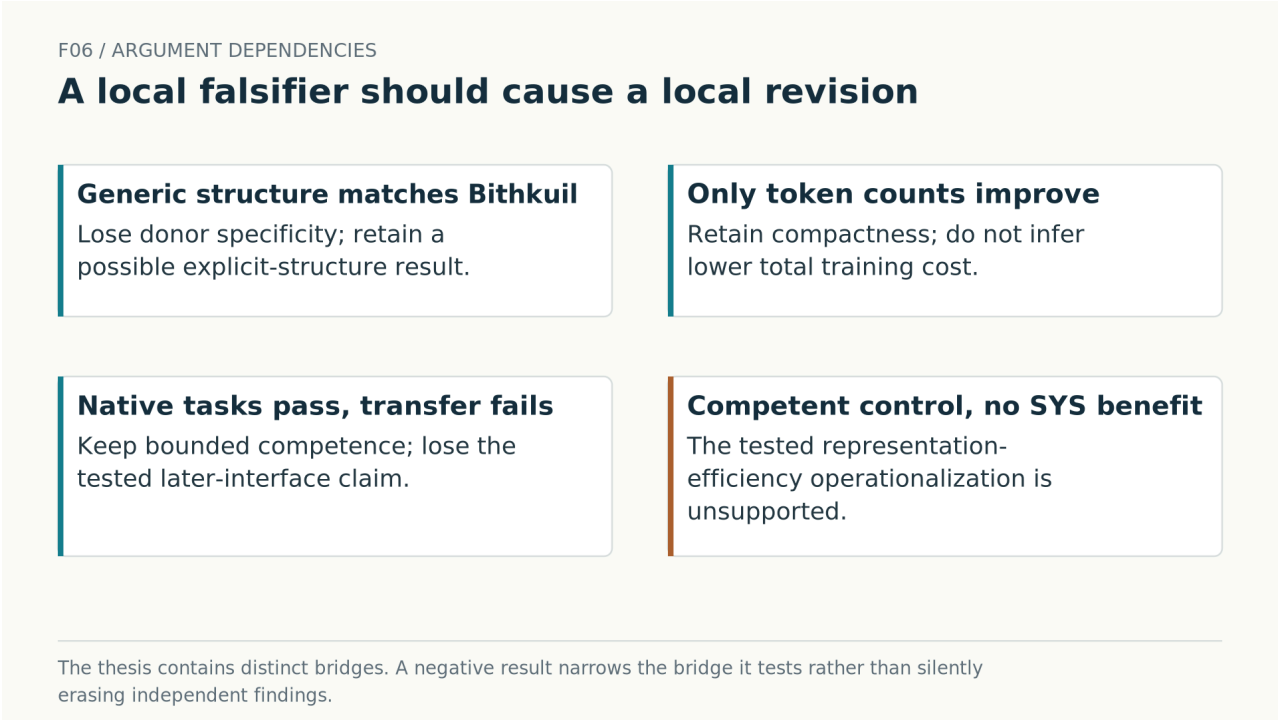

Figure 06. A local falsifier should cause a local revision. The thesis contains distinct bridges. A negative result narrows the bridge it tests rather than silently erasing independent findings.

10. Implementation Roadmap and Conclusion

Stage 2 begins by building the **teacher before asking the student to impress us**.

First, complete the machine-readable semantic map of current New Ithkuil and the Bithkuil token ABI. The map must identify semantic factors, defaults, scope/dependency rules, surface realizations, and legal composition while preserving the canonical semantic AST as the authority beneath human glosses and surface morphology. In parallel, define the minimal formal cognitive algebra for identity, relation, order, logic, set structure, quantity, arithmetic, comparison, time, evidence, causation, counterfactuals, proof, contradiction, and revision.

Second, compile those primitives into a versioned **developmental teacher system** and its capability/disclosure router. Define stable model roles, research-versus-intervention planes, a model-assistance ledger, a declassification transform for bounded third-party microtasks, and a stage-contract model-assistance policy before serious comparative runs. Exact work remains on deterministic/formal machinery wherever possible; probabilistic model output remains proposal/analysis until validated.

Third, implement the routed assistance layer itself before it becomes experimentally consequential. A local model can occupy the high-volume pedagogue/critic role; stronger interactive models can remain author-supervised research infrastructure; API calls used in a scientific procedure should freeze provider/model/version, reasoning setting, protocol/prompt identity and output/validation provenance. Preserve role contracts so future models can substitute without rewriting the experimental architecture.

Fourth, instantiate the first **versioned developmental stage contracts** inside that teacher system. Each contract names prerequisites, generators, executable oracles, train/dev/sealed evaluation families, rehearsal obligations, model-assistance contract, mastery/retention/transfer gates, and the checkpoint it may promote from. Build independent cross-checks for exact domains before trusting large volumes of generated examples. A local Qwen-class model may propose pedagogical material and diagnose failures, but accepted training/evaluation objects must retain provenance and pass the applicable semantic validators.

Fifth, implement checkpointed fixed-architecture developmental lineages. Start from random weights, teach a bounded primitive tranche, stop, checkpoint, run mastery/regression/transfer exams, remediate if permitted, and promote only when the stage contract passes. Preserve parent and child checkpoints as experimental witnesses. When curriculum order or teacher policy is under test, branch sibling learners from the same parent checkpoint rather than rebuilding unrelated histories. Freeze or preallocate the token ABI/embedding-table shape for each lineage so curriculum progression does not quietly add parameters. When branches compare teacher policies, use matched randomization/exposure budgets and preserve branch-specific seeds.

Sixth, build the deterministic semantic-world generator and its multiple renderers. Every renderer must expose the same underlying world and answerable propositions while allowing representation-specific accessibility to differ. The generator must support TinyStories-style constrained language, structured language, generic/factor-isomorphic IRs, Bithkuil, the canonical AST, and the H132 visible-composition controls. Lock three evaluation planes: TRAIN, DEV/PROBE, and SEALED TEST, with generator-family holdout for milestone claims where feasible.

Seventh, run **tiny engineering lineages** on the local machine to validate the complete teacher/compiler/oracle/checkpoint loop. The present executable smoke model has 53,408 parameters and the proposed reference-scale lineage has 2,828,736. The earlier roughly 3M-10M debugging range remains an escalation proposal, not permission to change architecture within a lineage; larger local lineages follow only after measured throughput and competence justify them. Record tokens/examples, optimizer steps, wall time, peak memory, research/control-plane cost, generator cost, learner cost, evaluation cost, retention, transfer, model-assistance exposure, and remediation cost. Do not infer hardware savings from nominal ternary bit width; measure the implementation actually used.

Eighth, run the **integrated demonstration lane** with BitNet-style ternary-forward training. The treatment combines Bithkuil semantic development, prerequisite-aware checkpointed pedagogy, deterministic supervision, and competence gates. The minimum credible control uses the same ternary student architecture, comparable generated worlds, checkpoint harness, and total accounting with a constrained-language or simpler representation. This first result is allowed to be a systems result. If it is large, robust, and transferable, the next order of magnitude of compute has earned consideration. Paid external compute still requires separate authorization.

Ninth, if the integrated result is interesting, execute the attribution program rather than retroactively claiming causality. P0 isolates representation; P1A/P1B distinguish cross-example curriculum from within-object traversal; P1C tests visible sub-symbol semantic structure; P1D measures topology-aware curriculum control; P2 isolates compositional objectives; P3 estimates the full-precision/ternary interaction. Repeat the strongest conditions across model sizes to determine whether Bithkuil shifts an intercept, changes a slope, helps only small learners, or disappears once ordinary models have enough capacity.

Tenth, test the most consequential transfer claim: **cognition before English**. After a learner demonstrates semantic competence in the Bithkuil substrate, expose it to a bounded natural-language mapping curriculum. Compare how quickly it learns to understand and express previously mastered relations in English and other surfaces relative to matched learners without the same developmental substrate. A dramatic reduction in language exposure required for cross-representation reasoning would be substantially more interesting than perfect performance on Bithkuil-native worksheets.

The larger thesis can now be stated without requiring faith in Ithkuil, BitNet, Rosetta, or any particular teacher policy:

Representation, developmental order, and pedagogy are part of the learning problem. If a learner repeatedly spends capacity reconstructing the same semantic distinctions from variable surface forms, receives dependent structure before the scaffold needed to interpret it, or trains without a teacher capable of distinguishing mastery from memorized exercise patterns, then making those distinctions explicit and controlling development may change the cost of acquiring reusable cognition.

Modern transformers may simply amortize these costs and erase the advantage. Generic typed IR may capture everything useful. TinyStories-style environmental simplification may explain nearly all of the gain. The integrated teacher may overfit its own generators. Ternary training may complicate optimization without helping learning efficiency. Those are all legitimate outcomes, and the Stage 2 design now makes them observable rather than philosophical disputes.

Bithkuil is valuable because it turns the question into an executable developmental laboratory. New Ithkuil supplies a rich donor inventory of semantic distinctions. Formal mathematics and logic provide exact relations and answer oracles. The AST and token ABI preserve stable semantic identity. The

teacher system generates and validates experience. A capability/disclosure router allocates model cognition without silently changing treatment identity. Checkpoint lineage records what changed when. Natural languages arrive later as codecs. And the attribution experiments remain available if the combined system produces a result strange enough to deserve dissection.

The immediate question is therefore no longer only "What does a small neural learner stop having to learn implicitly when semantic structure is explicit?"

It is also:

How much reusable cognition can a fixed small neural substrate acquire when its developmental world, semantic representation, teacher, tests, and progression are all designed to make learning rather than reconstruction the scarce resource?

That is the Stage 2 experiment.

10.1 What this revision changes

The original proposal asked whether a developmental semantic substrate could move the cost of learning. This revision makes the first part of that question executable, identifies a symmetry that forbids an easy but misleading explanation, and introduces a world-sensitive endpoint that distinguishes relational use from query-only competence. It adds a source-grounded grammar dependency map, a concrete integrated trial, a reference implementation and an operator handoff. It does not replace the larger thesis with those small engineering accomplishments.

The ambitious trajectory remains: acquire operators and relations cheaply in a controlled developmental world, preserve and recombine that competence across stages and descendants, then learn natural-language interfaces without paying the whole developmental cost again. The critical bridge is reusable competence that survives a change of combination, generator family or codec. The first systems trial is designed to put that bridge under load. Component attribution then determines whether the load is carried by factor accessibility, ordering, teacher adaptation, numerical precision or an unanticipated interaction.

The practical conclusion is stronger than a literature summary and narrower than a victory announcement. There is now a specific system to build from, an exact early experiment to run, and an explicit set of observations that would strengthen, split or defeat the thesis. The next scientific result should be obtained by executing that object, not by making its motivating language more cautious or more triumphant.

Project-lineage note

References [26]-[28] document antecedent Entif/Rosetta design ideas and attribution. They are included to prevent project-lineage drift, not counted as independent scientific evidence for Bithkuil.

References

[1] John Quijada. *A Grammar of New Ithkuil, Chapter 3: Basic Morphology*. Official New Ithkuil grammar. Source: https://www.ithkuil.net/newithkuil_03_morphology.htm Inspection: Official grammar relevant sections read live.

[2] John Quijada. *A Grammar of New Ithkuil, Chapter 4: Case Morphology*. Official New Ithkuil grammar. Source: https://www.ithkuil.net/newithkuil_04_case.htm Inspection: Official grammar relevant sections read live.

[3] John Quijada. *A Grammar of New Ithkuil, Chapter 5: Verb Morphology*. Official New Ithkuil grammar. Source: https://www.ithkuil.net/newithkuil_05_verbs.htm Inspection: Official grammar relevant sections read live.

[4] John Quijada. *New Ithkuil Affix Inventory* and *Chapter 7: Affixes*. Official New Ithkuil materials. Source: https://www.ithkuil.net/newithkuil_07_affixes.htm Inspection: Official grammar relevant sections read live.

[5] John Quijada. *A Grammar of New Ithkuil, Chapter 8: Adjuncts*. Official New Ithkuil grammar. Source: https://www.ithkuil.net/newithkuil_08_adjuncts.htm Inspection: Official grammar relevant sections read live.

[6] John Quijada. *A Grammar of New Ithkuil, Chapter 11: Syntax*. Official New Ithkuil grammar. Source: https://www.ithkuil.net/newithkuil_11_syntax.htm Inspection: Official grammar relevant sections read live.

[7] Suriya Gunasekar et al. "Textbooks Are All You Need." arXiv:2306.11644, 2023. Source: https://arxiv.org/abs/2306.11644 Inspection: Primary landing/abstract checked; no independent replication.

[8] Ronen Eldan and Yuanzhi Li. "TinyStories: How Small Can Language Models Be and Still Speak Coherent English?" arXiv:2305.07759, 2023. Source: https://arxiv.org/abs/2305.07759 Inspection: Primary landing/abstract checked; no independent replication.

[9] Shuming Ma et al. "The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits." arXiv:2402.17764, 2024. Source: https://arxiv.org/abs/2402.17764 Inspection: Primary landing/abstract checked; no independent replication.

[10] Shuming Ma et al. "BitNet b1.58 2B4T Technical Report." arXiv:2504.12285, 2025. Source: https://arxiv.org/abs/2504.12285 Inspection: Primary full-text relevant methods/results/limitations sections read.

[11] Ziming Liu et al. "KAN: Kolmogorov-Arnold Networks." arXiv:2404.19756, 2024. Source: https://arxiv.org/abs/2404.19756 Inspection: Primary landing/abstract checked; no independent replication.

[12] Felippe Alves and Renato Vicente. "Kolmogorov--Arnold Networks for Small Language Models." arXiv:2607.15525, 2026. Source: https://arxiv.org/abs/2607.15525 Inspection: Primary full-text relevant methods/results/limitations sections read.

[13] Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. "Net2Net: Accelerating Learning via Knowledge Transfer." arXiv:1511.05641, 2015. Source: https://arxiv.org/abs/1511.05641 Inspection: Primary landing/abstract checked; no independent replication.

[14] Mohammad Samragh et al. Scaling Smart: Accelerating Large Language Model Pre-training with Small Model Initialization. arXiv:2409.12903v2, 2024. Source: https://arxiv.org/abs/2409.12903v2 Inspection: Official arXiv abstract and metadata read; v2 2024-09-20.

[15] Zeyuan Allen-Zhu and Yuanzhi Li. "Physics of Language Models: Part 1, Learning Hierarchical Language Structures." arXiv:2305.13673, v4, 2025. Source: https://arxiv.org/abs/2305.13673 Inspection: Primary landing/abstract checked; reconciled v4 (2025) superseding inherited v3.

[16] LCM Team et al. "Large Concept Models: Language Modeling in a Sentence Representation Space." arXiv:2412.08821, 2024. Source: https://arxiv.org/abs/2412.08821 Inspection: Primary landing/abstract checked; no independent replication.

[17] Shibo Hao et al. Training Large Language Models to Reason in a Continuous Latent Space. arXiv:2412.06769v4, revised 2026-08-23; originally 2024; COLM 2025. Source: https://arxiv.org/abs/2412.06769v4 Inspection: Official current abstract and revision history read; v4 2026-08-23.

[18] Enes Özeren and Matthias Aßenmacher. "Reinforcement Learning for Latent-Space Thinking in LLMs." arXiv:2512.11816, 2025. Source: https://arxiv.org/abs/2512.11816 Inspection: Primary landing/abstract checked; no independent replication.

[19] Minghan Wang, Thuy-Trang Vu, Ehsan Shareghi, and Gholamreza Haffari. Towards Inference-time Scaling for Continuous Space Reasoning. Findings of ACL 2026, 26842-26856. DOI: 10.18653/v1/2026.findings-acl.1338. Source: https://aclanthology.org/2026.findings-acl.1338/ Inspection: Official ACL abstract and publication metadata read; arXiv precursor same lineage.

[20] Xuefeng Bai, Yulong Chen, and Yue Zhang. "Graph Pre-training for AMR Parsing and Generation." *Proceedings of ACL 2022*, pp. 6001-6015. DOI: 10.18653/v1/2022.acl-long.415. Source: https://aclanthology.org/2022.acl-long.415/ Inspection: Official ACL abstract and metadata read.

[21] Valerie Hajdik, Jan Buys, Michael Wayne Goodman, and Emily M. Bender. "Neural Text Generation from Rich Semantic Representations." *NAACL-HLT 2019*, pp. 2259-2266. DOI: 10.18653/v1/N19-1235. Source: https://aclanthology.org/N19-1235/ Inspection: Official NAACL abstract and metadata read.

[22] Jiahuan Zhang et al. "SR-LLM: Rethinking the Structured Representation in Large Language Model." *ACL 2025*. Source: https://aclanthology.org/2025.acl-long.172/ Inspection: Primary landing/abstract checked; no independent replication.

[23] Emmanuel Abbe, Samy Bengio, Aryo Lotfi, and Kevin Rizk. "Generalization on the Unseen, Logic Reasoning and Degree Curriculum." *Journal of Machine Learning Research* 25(331):1-58, 2024. Source: https://www.jmlr.org/papers/v25/24-0220.html Inspection: Primary landing/abstract checked; no independent replication.

[24] Kazuki Yano et al. "Efficient Construction of Model Family through Progressive Training Using Model Expansion." arXiv:2504.00623, 2025. Source: https://arxiv.org/abs/2504.00623 Inspection: Primary landing/abstract checked; no independent replication.

[25] Judea Pearl. *Causality: Models, Reasoning and Inference*. 2nd ed., Cambridge University Press, 2009. Source: https://doi.org/10.1017/CBO9780511803161 Inspection: Cambridge publisher search metadata checked; book not reread; direct page fetch failed.

[26] Entif.ai. *Cognitive Tiles and Swarm Gnosis: A Tile-First Knowledge Framework*. Internal project design artifact, 2025. Project-lineage/provenance reference only. Inspection: Supplied project-lineage citation; not independent evidence.

[27] Entif.ai. *RFC ENTIF-0001: Rosetta 2.0 Unified Protocol and Architecture Specification*, v2.0.2. Internal/historical Rosetta specification, 2025. Project-lineage/provenance reference only; current Rosetta authority supersedes historical semantics where applicable. Inspection: Supplied project-lineage citation; not independent evidence.

[28] Entif.ai. *Entif.ai 2.0 Architecture & Roadmap Blueprint*. Internal project design artifact, 2025. Project-lineage/provenance reference for representation-first architecture and pre-existing downstream orchestration concepts only. Inspection: Supplied project-lineage citation; not independent evidence.

[29] Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. "Curriculum Learning." *Proceedings of ICML 2009*, pp. 41-48. DOI: 10.1145/1553374.1553380. Source: https://ronan.collobert.com/ Inspection: Coauthor publication page abstract and metadata read; ACM direct fetch failed.

[30] Ronald B. Dekker, Fabian Otto, and Christopher Summerfield. "Curriculum learning for human compositional generalization." *Proceedings of the National Academy of Sciences* 119(41):e2205582119, 2022. DOI:

10.1073/pnas.2205582119. Source: https://pmc.ncbi.nlm.nih.gov/articles/PMC9564093/ Inspection: Primary article abstract and full-text navigation consulted; no human-study replication.

[31] Justin DeBenedetto. "Linearization Order Matters for AMR-to-Text Generation Input." *Proceedings of the 2024 UMR Parsing Workshop*, pp. 1-7, 2024. Source: https://aclanthology.org/2024.umrpw-1.1/ Inspection: Primary landing/abstract checked; no independent replication.

[32] Bofei Gao, Liang Chen, Peiyi Wang, Zhifang Sui, and Baobao Chang. "Guiding AMR Parsing with Reverse Graph Linearization." *Findings of EMNLP 2023*, pp. 13-26. DOI: 10.18653/v1/2023.findings-emnlp.2. Source: https://aclanthology.org/2023.findings-emnlp.2/ Inspection: Official ACL abstract and metadata read.

[33] Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. "Sub-Character Tokenization for Chinese Pretrained Language Models." *Transactions of the Association for Computational Linguistics* 11:469-487, 2023. DOI: 10.1162/tacl_a_00560. Source: https://aclanthology.org/2023.tacl-1.28/ Inspection: Official TACL abstract and metadata read.

[34] David A. Haslett. "Tokenization Changes Meaning in Large Language Models: Evidence from Chinese." *Computational Linguistics* 51(3):785-814, 2025. DOI: 10.1162/coli_a_00557. Source: https://aclanthology.org/2025.cl-3.3/ Inspection: Primary landing/abstract checked; no independent replication.

[35] Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. "Automated Curriculum Learning for Neural Networks." *Proceedings of ICML 2017*, PMLR 70:1311-1320. Source: https://proceedings.mlr.press/v70/graves17a.html Inspection: Official PMLR abstract and metadata read.

[36] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. "Teacher-Student Curriculum Learning." *IEEE Transactions on Neural Networks and Learning Systems* 31(9):3732-3740, 2020. DOI: 10.1109/TNNLS.2019.2934906. Source: https://arxiv.org/abs/1707.00183 Inspection: Official original preprint abstract and metadata read; IEEE publication is same lineage.

[37] Nidhi Vakil and Hadi Amiri. "Curriculum Learning for Graph Neural Networks: A Multiview Competence-based Approach." *Proceedings of ACL 2023*, pp. 7036-7051. DOI: 10.18653/v1/2023.acl-long.389. Source: https://aclanthology.org/2023.acl-long.389/ Inspection: Official ACL abstract and metadata read.

[38] Zhenya Liu and Yuxin Chen. "Active Curriculum Refinement for Reinforcement Learning." arXiv:2608.26469, 2026. Source: https://arxiv.org/abs/2608.26469 Inspection: Primary full-text relevant methods/results/limitations sections read.

[39] Adam Shai et al. Transformers learn factored representations. arXiv:2602.02385v1, 2026. Source: https://arxiv.org/abs/2602.02385 Inspection: Primary full text: conditional independence, product/direct-sum distinction, hidden-factor and noise experiments.

[40] Nived Rajaraman et al. Learning to Reason with Curriculum II: Compositional Generalization. arXiv:2606.27721v1, 2026. Source: https://arxiv.org/abs/2606.27721 Inspection: Primary full text: learning setup, verifier assumptions and theorem scope.

[41] Devon Jarvis, Richard Klein, Benjamin Rosman, and Andrew M. Saxe. Compositionality and systematicity emerge from iterated learning in deep linear networks. PNAS 123(19):e2509739123, 2026. DOI: 10.1073/pnas.2509739123. Source: https://www.raillab.org/publication/jarvis-2026-compositionality/ Inspection: Author lab abstract read; full text fetch blocked by browser check; no full-text claim.

[42] John Quijada. A Grammar of New Ithkuil, Chapter 6: More Verb Morphology. Source: https://www.ithkuil.net/newithkuil_06_more_verbs.htm Inspection: Relevant full text: illocution, validation, case frames; editorial count discrepancy recorded.

[43] John Quijada. New Ithkuil Lexicon, version 1.0. Source: https://www.ithkuil.net/newithkuil_lexicon.pdf Inspection: 570-page PDF selectively consulted: cover and printed p.5 copula visually verified; containment entries text-inspected only, failed page screenshot recorded.

[44] John Quijada. A Grammar of New Ithkuil, Chapter 2: Morpho-Phonology. Source: https://www.ithkuil.net/newithkuil_02_morpho-phonology.htm Inspection: Official chapter navigation and structure consulted.

[45] John Quijada. A Grammar of New Ithkuil, Chapter 1: Phonology. Source: https://www.ithkuil.net/newithkuil_01_phonology.htm Inspection: Official chapter navigation and structure consulted.

[46] John Quijada. A Grammar of New Ithkuil, Chapter 13: Numbers. Source: https://www.ithkuil.net/newithkuil_13_numbers.htm Inspection: Official chapter navigation and structure consulted.

[47] John Quijada. A Grammar of New Ithkuil, Chapter 9: Referentials. Source: https://www.ithkuil.net/newithkuil_09_referentials.htm Inspection: Official referential chapter consulted.

[48] John Quijada. A Grammar of New Ithkuil, Chapter 10: Specialized Constructions. Source: https://www.ithkuil.net/newithkuil_10_spec_constructions.htm Inspection: Concatenation, carrier root and affix-to-root constructions read.

[49] John Quijada. A Grammar of New Ithkuil, Chapter 12: The Writing System. Source: https://www.ithkuil.net/newithkuil_12_script.htm Inspection: Official chapter navigation consulted; deferred surface codec.

[50] John Quijada. A Grammar of New Ithkuil, Appendices. Source: https://www.ithkuil.net/newithkuil_appendices.htm Inspection: Official inventory navigation consulted; not an exhaustive conformance certification.