Externalising Knowledge for Decoupled Diagnosis: Attribution of Reasoning Failures via Controllable Knowledge Visibility

Anonymous authors
Paper under double-blind review

Abstract

Benchmarks for large language models in chemistry and materials are built mostly from exam questions and terminology quizzes, scoring whether a model can state a known conclusion. Design work asks different queries – whether two mechanism statements denote the same concept, whether an effect holds under given conditions – which have no memorisable answer and can only be performed over a given knowledge structure. More fundamentally, failure cannot be attributed: when a model misses a mechanism question, lacking the knowledge and failing to reason with it are indistinguishable, and that distinction decides whether the fix belongs to the knowledge source or to the model. We propose a design paradigm for diagnostic benchmarks: take the domain's knowledge out of the model and write it as an executable structure – relations among concepts (composition, blocking, competition, and so on) as operations, axioms as inference rules, each item as an operator tree over that algebra. Two properties follow: answers are computed by executing the tree and hence recomputable; and the knowledge an item needs can be supplied tier by tier (withheld, complete, names only, or in a wrong order) with everything else fixed, making knowledge visibility a variable that can be manipulated on its own. The two tiers of one item separate "not knowing" from "not deriving", and aggregating such pairs over the items a knowledge unit gets wrong gives the unit's compensability (the share of its failures external knowledge can rescue) – high-compensability units point at the knowledge source, low ones at the model. **We propose a benchmark that can be diagnosed by attribution, together with a paradigm for upgrading such ontology-based benchmarks: repair what the attribution diagnoses, and the diagnosed unit's performance improves.** Attribution lands at the **operator and family** level (per-mechanism counts are too thin to support mechanism-level claims; that remains the target granularity). We add a **material-grounded decidability (MGD) gate**: of 1,274 items, **80.7% carry the gold verbatim in the material the model sees**, 9.0% partially, and **10.3% depend on a full closure outside that material** (all of `gap`, `verify_support` and `granularity`) – a direct threat to construct validity, so the primary indicator moves to a **reasoning-type subset** with undecidable items excluded. All readings are relative: we claim **no absolute scale** (the expert baseline is unfinished), **no cross-domain generality**, and **no superiority over existing benchmarks**.

Introduction

Three difficulties in evaluating mechanism judgement

Large language models are now used in real materials-design workflows: selecting evidence that can be merged, deciding whether two statements of a mechanism denote the same concept, and grounding a conclusion in a specific material and process. Evaluating this capability runs into three difficulties.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

One, mechanism judgement is reduced to knowledge recall. Existing benchmarks are mostly drawn from exam questions or terminology QA, and scoring asks whether the model can state a known conclusion. What design work requires is a different kind of query – whether two measurement claims can be merged, whether an effect holds under given conditions, whether two statements of a mechanism are the same. These have no memorisable answer: they can only be performed on a given knowledge structure.

Two, answers are annotated by hand and cannot be recomputed. Change the item surface or add a condition and the annotation lapses; when two annotators disagree there is no way to decide who is right. A score therefore traces neither to a particular piece of knowledge nor to a recomputation.

Three, knowledge sits in the parameters, so failure cannot be attributed. When a model misses a mechanism item, the failure has two possible sources: it does not know the knowledge, or it knows it and cannot reason with it. If knowledge lives in the weights, the two cannot be told apart: it cannot be inspected, and probing without a criterion yields only an aggregate score; it cannot be edited, since updating one stale mechanism means retraining or knowledge editing. Worse, that aggregate cannot be decomposed, so the fix has no target – repair the knowledge source or adjust the model, the result table cannot say.

All three share one root: knowledge and reasoning only ever appear in the form of an answer. The item asks for a query; what is observed is a score; the answer is written by a human and cannot be derived from the item; knowledge sits in the weights and can be probed but not separated. The score is therefore a mixture of what the model knows and whether it can use it – neither decomposable nor recomputable. Making failure attributable to one side requires three conditions to hold: the query specified by the item, the answer computed from the item, and knowledge that can be set out separately from the model.

What existing approaches do, and what they leave undone

Each of the three difficulties has work responding to it, and none of them resolves it: the query is still read off the wording, the answer is still written by a human, and failure still cannot be split. Mechanism judgement: reasoning benchmarks in chemistry and materials have multiplied (68; 49), and a comparison with chemists found models strong on item-bank tasks and failing where real reasoning is needed (32) – they measure whether a conclusion is known, not whether a query can be carried out on a given structure. Where answers come from: a few works have answers computed by deterministic tools rather than annotated (34; 19; 68), which shows that recomputable answers are feasible, but each item carries its own verifier and no single structure both delimits the queries and computes all truths. Where knowledge should live: three branches exist – keep it in the parameters (64; 37), retrieve it when needed (30; 76; 67; 51), or move memory out entirely (45). In the first two knowledge stays parametric, so visibility is not a manipulable variable; the third does move it outside the model, but does not settle whether knowledge has fully left the model and which side a remaining failure belongs to. One further branch works directly on attribution, classifying existing benchmarks as knowledge-leaning or reasoning-leaning; but classification labels a whole paper, its criterion rests on the item author's wording, and no single piece of knowledge can be switched off and the score re-read – it yields correlation, not attribution. Per-branch detail, difficulties and comparison: Appendix F.

This paper: one executable structure, in three steps

All three routes stop at the same place: query, answer and knowledge are not yet objects that can be manipulated independently. This paper collects the three into one executable structure, in three steps.

One, a mechanism ontology: mechanism knowledge written as concepts, relations among concepts, the conditions under which a relation holds, and pairs that cannot hold together. Two, a mechanism algebra: relation types are interpreted as operations rather than labels – composition, blocking, competition, prerequisite, enabling and superordination; axioms become inference rules, so that mutual exclusion propagates down the hierarchy and a

blocker's presence invalidates an effect. Concepts can therefore be decomposed, composed and queried in reverse. Three, an item becomes an operator tree over that algebra: the logical form of an item is a tree over these operations, evaluated bottom-up, and the operation evaluated last is the one at the root – the principal operator. Its type is the query type the item asks: a root of blocking asks what blocks it, a root of composition asks what it is made of. The query is therefore no longer implicit in the wording but specified by the item; executing the tree gives the gold label, recomputed whenever a condition changes and never re-annotated.

The three steps yield two things. First, answers computed from the structure, recomputable position by position. Second, manipulable knowledge visibility: not given, fully given, names without structure, or structure in the wrong order – with the item surface, the model and the response format held fixed otherwise. One item's two results under the **masked arm** (no external knowledge) and the **supplied arm** (complete ontology) separate "does not know" from "cannot reason with it"; aggregating those pairs by knowledge unit gives that unit's compensability: the share of its failures that outside knowledge can rescue. The readings therefore say which side to repair – high-compensability units point at the knowledge source, low ones at the model. Difficulty is only an incidental column of the instrument, used to split items into comparable strata before attribution, and is not a conclusion.

Six checkable claims

Evaluation methodology asks a benchmark to argue that it measures what it claims, and to state its claims, the conditions under which they hold and their boundaries (54; 2). This paper makes six claims, which form one chain in which each link presupposes the previous one: **an item must first specify the query itself before its answer can be computed by execution; only then can knowledge leave the model; only once it has left can knowledge be supplied tier by tier; only when knowledge can be supplied as a single variable can failure be attributed; and only once failure can be attributed can one repair what the diagnosis points to and have performance come back reproducibly.** Of the six, the last is the central claim of this paper.

One, the query is specified by the item itself – the principal operator fixes the query type. Two, the gold label is computed from the structure rather than annotated – executing the operator tree gives it, position by position. Three, knowledge can leave the model – it is written as a versioned ontology and rule layer (Section 3.1). Four, knowledge can be supplied tier by tier – four tiers, with everything but visibility held fixed (Section 3.2). Five, failure can be attributed – the difference between the two results sorts failure into four classes, aggregated over the ontology structure into compensability (Section 3.3). Six, repairing what the diagnosis points to reproducibly lifts performance – the central claim: on public ontologies, upgrading exactly the diagnosed layer lifts the diagnosed unit's compensability from zero to 0.550-1.000 (hierarchy +0.550 to +0.733, role +0.900 to +0.967, negation +0.917 to +1.067, consistent across three models with no pair running right-to-wrong), while the memory-type control family falls instead (-0.250, Table 12 of Section 4.9).

How each claim is checked is in Section 4, and the conditions under which it fails in Section 5.1. All six concern individual items: what is required is that the item genuinely needs that query, not that the query happens to solve it (2; 53). Difficulty is not a conclusion but an upstream instrument: written as a component vector at item generation, declaring only "not weaker than" on the dominance order (thereby avoiding the tautology "harder implies lower accuracy"), frozen before evaluation, and used to split items into comparable strata before attribution (Section 4.3).

Deliverables

The following four artefacts are directly usable; together they cover every link of the chain in Section 1.4.

Mechanism ontology and mechanism algebra. Concepts, relations among concepts, disjointness pairs, condition space, and the mechanism–indicator mapping; relations interpreted

as the six operations composition, blocking, competition, precondition, enabling and subsumption, with axioms written as inference rules; the rule layer instantiates a mechanism on a concrete material and condition. Ontology and rules both carry version numbers, checked for consistency before evaluation, so any later change makes the check fail (Sections 3.1.1–3.1.4).

Item bank and four-tier views. Each item records the knowledge units and the channel it depends on, its gold label is recomputable, and knowledge can be supplied at four tiers (Sections 3.2.1–3.2.5).

Scoring protocol and response specification. Responses must give the operator tree, so answer accuracy and reasoning-structure accuracy are reported separately; the structure score serves only as a finer reading and is not mixed into attribution conclusions (Section 3.2.6).

Attribution and guidance table. Compensability per knowledge unit, together with the improvement direction derived from it (Section 3.3.3).

Item coverage and corpus scale are given with the data card.

Related Work

Related work falls into four branches: benchmark and evaluation methodology, external knowledge and retrieval augmentation, structured knowledge and neuro-symbolic methods, and failure attribution and diagnostic evaluation. Each branch is introduced through one representative paper, then read for how far it gets and where it stops; the per-branch detail and comparison are in Appendix F.

Benchmark and evaluation methodology: can a score be traced back to its item

Measuring what Matters: Construct Validity in Large Language Model Benchmarks (54) is a systematic review of 445 language-model benchmarks by 29 experts, finding patterns that weaken conclusion validity in three places – the phenomenon measured, the task design and the scoring protocol; medical benchmark work therefore argues for putting construct validity first (29). What matters is not that defects exist but where they sit: not in the details of individual benchmarks but at the step from a score to a claim about capability. Closing that step has one route: every score must be traceable back to its item – an item answered correctly may be so because the model can do it, because the answer is written on the surface, because the options can be eliminated, or because it has seen the item; if it does not trace back, the score reflects several sources at once. Three directions have tightened this requirement: from the whole benchmark to a single item (estimating each item's position on a capability axis (16), or using Rasch models to separate item difficulty, rater severity and ability (25)); from correlation to necessity (the task must require the target capability, not merely be solvable by it (2), since otherwise a saturated benchmark gives no deployment tolerance (22) – with counterparts elsewhere, scientific machine-learning evaluation needing more diverse tools (9) and drug-discovery decisions needing interrogable assumptions (27)); and from requirement to procedure (evidence-centred design checking item by item (53), and pre-registration writing criteria before seeing data (66)). All three point at one thing: the score must trace to which part of this item produced it.

External knowledge and retrieval augmentation: who decides what to use

Moving knowledge outside the model has a long line (parametric memory, retrieval, wholesale externalisation); a recent representative is *Navigating Unreliable Parametric and Contextual Knowledge* (44), where stale context can even corroborate internal memory and override correct information. This branch does achieve "knowledge outside the model", but *whether to use it* is still the model's decision, and models are systematically biased here: they answer when they should abstain (*AbstentionBench* (58) and later analysis (72)).

Knowledge is externalised, yet how much of it is supplied is still not a variable that can be manipulated on its own.

Structured knowledge and neuro-symbolic methods: how graphs and ontologies enter reasoning and scoring

Graphs and ontologies appear at three positions on this line: as retrieval source (*KG-LLM-Bench* (39), *GraCoRe* (77), *SciQA* (62), *EGT-KG* (57)), as reasoning-chain skeleton (*Graph Chain-of-Thought* (33)), and as the knowledge being built and verified (*Provenance-Enhanced Statements in Knowledge Graphs* (40), *AutoSchemaKG* (47)). None of the three takes part in scoring: structure is for the model, not for evaluation. Making diagnosis itself a reproducible graph has been attempted (*Diagnostic Knowledge Graphs* (13)), but it still lands on "the class of error" rather than "the missing piece of knowledge".

Failure attribution and diagnostic evaluation: from the whole test down to the knowledge unit

This branch aims directly at "which side should a failure be charged to", but its unit is too coarse. At whole-test level, benchmarks are sorted after the fact into knowledge-heavy and reasoning-heavy (*Disentangling Knowledge and Reasoning in Medical Large Language Models* (14), *Knowledge or Reasoning?* (5), (4)), with labels resting on the author's wording; at factor level, factorial designs separate main effects from interactions (*AuthorityBench* (11)) – a methodological advance, but the unit is still a surface property; for error attribution there are datasets and models (*Diagnosing Failures in Large Language Models' Answers* (3)), yet the landing point is still "the class of error". In all three, "knowledge" is not the unit: what is given is correlation, not attribution.

Taken together, the gap has a specific location: the ontology has been placed outside the model (Sections 2.2, 2.3) but is not put into scoring, and attribution lands on "the class of error" (Section 2.4) rather than on "the missing piece of knowledge". This paper treats the ontology as an executable structure rather than a retrievable material (Section 3.1), which releases both places where the chain was stuck. On the item-and-gold side, an item's anchor comes from the ontology's superordinate concepts and its candidate set from the same level, so the surface is rendered from those two slots alone and distractors follow from the structure; the truth value is not written by hand but fixed before rendering by what the operator tree evaluates to (Section 3.2) – structure enters the item, and therefore enters scoring. On the attribution side, knowledge being outside the model, it can be supplied tier by tier, one item's two results sort failures into four classes, and aggregation by knowledge unit yields compensability (Section 3.3); attribution lands on an object that can be switched off individually – lower one edge's visibility from `full` to `blind` with everything else fixed, and if the score moves, that edge is what was missing. The relation between this paper and the branches above is therefore not a contest of methods but a difference of position: query, truth value, knowledge and attribution are recovered from the model's outputs and from human hands onto the knowledge structure the item depends on, so that the item carries its own warrant. Two boundaries are stated with it: answerability is not on the chain and is decided with the same warrant as failure attribution (Appendix A.6.6); difficulty is not a conclusion but an upstream stratifier, written at generation and frozen before evaluation (end of Section 1.4). Per-branch detail and comparison are in Appendix F (F.1 query and answer, F.2 parametric memory, F.3 knowledge externalisation, F.4 attribution, F.5–F.6 the remaining branches).

Method: from domain knowledge to diagnostic evaluation

This chapter takes up the gap identified at the end of Section 2: the three branches each supply part of the necessary chain, but none turns knowledge visibility into a single variable – what is supplied varies together with the item surface, the wording and the retrieval result, so one failure cannot be split between the two sides. This chapter therefore presents a

diagnostic measurement instrument (hereafter *instrument*) of three modules — Section 3.1 writes domain mechanism knowledge as an executable reasoning space, Section 3.2 turns that space into evaluation objects with determinate answers, controllable difficulty, and explicit knowledge dependence, and Section 3.3 infers the boundary of a model's capability from controlled two-tier results. Together they make "the query type is specified by the item, the answer is computed from structure, and failure can be split between the knowledge source and the model" hold simultaneously; the construction pipeline, the full-chain diagram, and the correspondence between the three modules and the five links of Section 1.4 are given in Appendix B.

In measurement terms the correspondence is part-by-part, so the wording is not a metaphor: the executable structure plays the role of a cognitive model; item generation corresponds to an item model (automatic item generation); the four tiers and the two arms are evidence rules (evidence-centered design (53)); and the item-to-knowledge-unit map is a Q-matrix (the rule-space and Q-matrix tradition of (52; 35)). One difference from classical cognitive diagnosis matters: there the Q-matrix is inferred from response data, whereas here it is read off the operator tree (Edges(q)), so attribution does not rest on statistical inference (Sections 3.2, 4.1). Treating a benchmark as a calibratable measurement instrument is also the line of "benchmarks as microscopes" (2) and of construct validity (54; 29).

Mechanism Ontology Construction: from domain knowledge to an executable reasoning space

The mechanism ontology is not a knowledge store but the reasoning-constraint layer of the evaluation that follows: it defines what counts as legitimate reasoning without supplying answers.

Why this ontology is built in-house. Let the point be made first: this is not a matter of preference but of public resources being unable to measure the two central claims. Those claims require "change only the knowledge visibility and the same item's answer changes with it", which needs an executable structure as its carrier; public resources fall short on both sides. *Ontology side*: in a public ontology such as ANATOMY / GALEN the asserted edges are hierarchical, with no condition dimensions and no mutual-exclusion pairs. A second rebuild (Appendix E.17) nevertheless derives two further families from the same files – a role family from the existential assertions (`atomic --role→ filler`: GALEN 19,013, ANATOMY 32,184) and a negation family from the complete superordinate closure – so the public-ontology comparison of Section 4.9 now covers three families (hierarchy, role, negation). What remains out of reach is the condition family (`requires` / `gap`), which needs controlled value domains, and the explicit mutual-exclusion family, because public ontologies rarely assert `disjointWith` (the full ANATOMY file we downloaded contains 17 such axioms); the per-operator coverage gap is in the E-EXTERNAL row of Appendix E.4. *Benchmark side*: a public benchmark such as ChemBench supplies no structure at all – neither its surface nor its knowledge can be switched – so what it measures is still a total score, and it cannot answer "which edge is missing".

Building in-house raises a challenge that has to be met head-on: self-serving construction – our ontology, our items, our scoring. Could the conclusions be an artefact of that setting? Three lines of defence answer it separately, each with its readings and location: (i) the instrument self-check: re-executing the same query gives a bitwise-identical truth value at rate 1.000 with 0 scorer misjudgements (Table 1; level-by-level check in Table E.1 of Appendix E.1); (ii) rebuilding the same instrument on public ontologies: 180 items on ANATOMY / GALEN give a positive Δ for all six models (+0.511 ... +0.761), and the "remove the edge, put it back" loop reproduces there as well (knowledge-supplied 0.000 → 0.717) (Table 11; details in Appendix E.4); (iii) the same-model comparison on a public benchmark: see the next paragraph and Appendix D.3. All three point the same way: the readings do not change with *whose tree or whose items*, but with visibility.

The public-benchmark route needs its own positioning: it can serve as a capability scale only and cannot replace what this paper measures – the same models read 0.77-0.82 on ChemBench but only 0.16-0.22 on our knowledge-masked arm, so public scores cannot supply the parametric-coverage column; nor do its knowledge/reasoning tags isolate the failure

side this paper cares about (the reasoning tag scores *higher*). Relative to an in-house ontology, a public dataset gives one layer less: a total score, with no "which edge". That part, with its readings and protocol differences, therefore sits entirely in Appendix D.3, and the main text makes no comparability claim. The cost is recorded as well: concept boundaries, condition dimensions and mutual-exclusion pairs are declared by experts; the role, negation and mutual-exclusion families have since been replicated on public ontologies (four-family comparison in Appendix E.17; the mutual-exclusion family comes from 17 explicit `disjointWith` axioms in the downloaded ANATOMY `human.owl` plus propagation over the hierarchy). Upgrades are reported on two sides: the **ontology-side gain is unconditional** (the gap is closed and recomputable), the **model-side gain is conditional** on the knowledge not already being in the parameters, with the masked arm as the probe. The condition family is not replicated – public ontologies offer no controlled value domains – and neither is the explicit mutual-exclusion family, because `disjointWith` is rarely asserted (candidate resources in Appendix C.4) – so the conclusion stops at "the instrument can be diagnosis-driven" and does not extend to domain knowledge itself (Section 5.2).

Throughout, instrument means this three-module combination together with its rendering and scoring protocols – something that can be checked part by part (Section 4.1), not a fixed item set. It has three layers bottom-up – the ontology layer $\mathcal{O}_M$ records concepts, relations and axioms and decides what can be asked and derived; the rule layer $\mathcal{R}$ records the instantiation of mechanisms on objects and conditions and decides how an answer is judged; the domain evaluation space $(\mathcal{E}, \Gamma, M, \Lambda)$ supplies objects, conditions and indicators and decides what acts on what is evaluated.

Three heterogeneous sources feed it: literature mechanism knowledge gives the factual chain (material -> trigger $\rightarrow$ mechanism $\rightarrow$ effect $\rightarrow$ indicator), expert knowledge gives concept boundaries, hierarchy and mutual exclusion (which inferences are legitimate), and resource knowledge gives attributes, indicators and condition values; nothing outside the three enters the ontology. Being executable requires the ontology to declare first which queries it supports – a query holds only if the operator's argument types are compatible with the relations that actually exist:

$$\mathrm{Ask}(\mathcal{O}_M) = \{\, q \;:\; \mathrm{op}(q) \in \mathrm{Op},\ \mathrm{ar}\big(\mathrm{op}(q)\big) \text{ compatible with } V, E \,\}.$$

Given an operator it is therefore decidable whether it can be asked, which fixes the questioning boundary. Relations and operators are two kinds of object: an edge is a fact in the ontology, while retrieving a component is an operation performed at evaluation time; axioms replace retrieval with derivation, so conclusions that follow from existing relations need not be looked up again (the five axioms are in Appendix A.5). A mechanism must further be instantiated on objects and conditions to be decidable. Instantiation attaches a mechanism to an object and a condition set and yields a rule instance:

$$r = \mathrm{ground}(m, e, \kappa) = (e,\ \kappa) \vdash \varphi, \qquad e \in \mathcal{E},\ \kappa \subseteq \Gamma,\ \varphi \in \mathrm{eff}(m),$$

the set of all rule instances being the rule layer $\mathcal{R}$. Evaluation is three-valued (`true`/`false`/`unknown`) – literature knowledge is inherently incomplete, and "no recorded blocker" is not "no blocker"; collapsing to two values would turn the incompleteness of the knowledge base directly into gold-label errors, so only queries evaluating to `true` become items. Conditions are not free text but controlled values, each dimension carrying a constraint function that separates "not supplied", "violates the constraint", "satisfied" and "insufficient evidence"; the `inv` item type (supply one more dimension and re-ask) rests exactly on the "not supplied" state. The full formalisation, axiom set, evaluation rules and external checking of the layers are in Appendix A.5–A.8.

Mechanism-aware Benchmark Generation

Items are not written but generated from the structure; four things are fixed in turn – what can be asked, what the answer is, on what object it acts, and how much knowledge it sees.

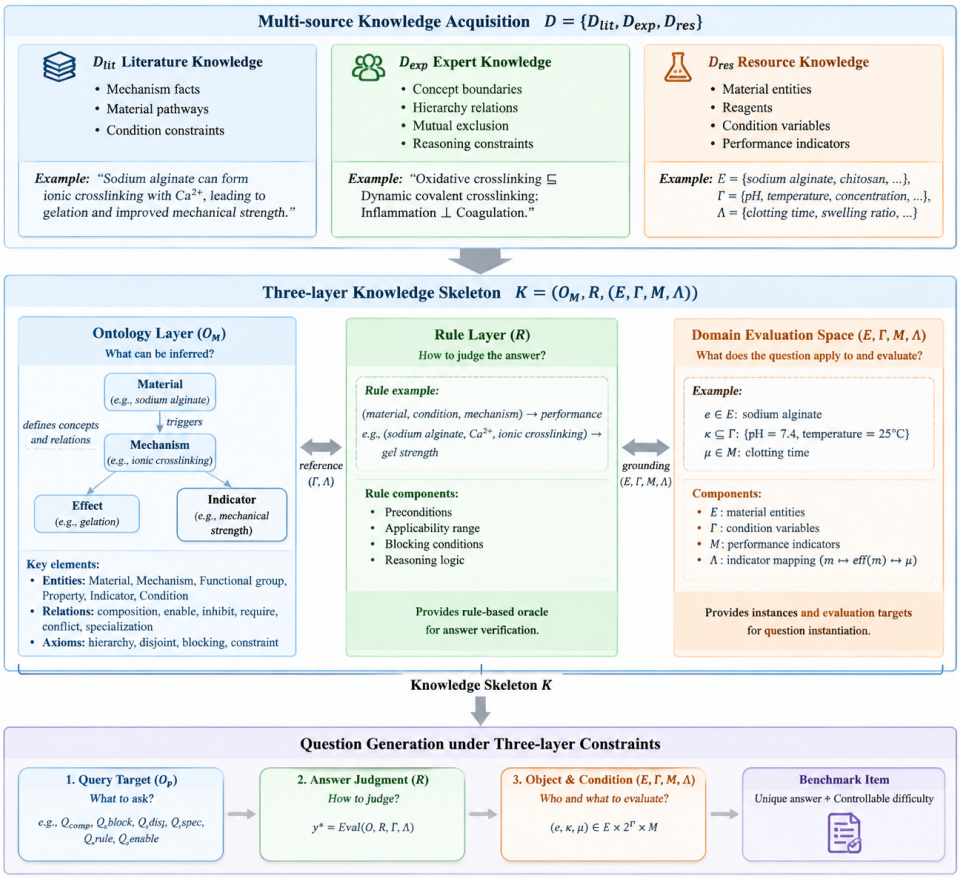


Figure 1: Roadmap of ontology construction. The top row is the three knowledge sources, the middle row the three-layer skeleton, the bottom row item generation under the three constraints (details in Section 3.2); the two lateral arrows in the middle row mark the two kinds of inter-layer relation — the ontology refers to condition space and indicator mapping, while the domain evaluation space grounds the rule layer. (English labels in the figure are synonymous with the notation of the text: $O_M = \mathcal{O}_M$, $R = \mathcal{R}$, $O_{\mathcal{P}} = \text{Op}$; the K in the figure denotes the three layers as a whole, for which the text introduces no separate symbol.)

The query space comes from rendering operators, each of the six determining which item types it can produce (grammar in Appendix A.3). The gold label is computed: every question form corresponds to an operator tree T whose leaves must be rule instances, evaluated bottom-up, and the operation at the root – the principal operator – has the type of the query the item asks; executing it gives the gold label, which is never annotated and is recomputed whenever a condition changes, together with the record of which edges were read, Edges(q), which is the unit of attribution. Instance grounding writes the surface: which object, under which conditions, evaluated by which indicator, rendered from three slots. Distractors are not drawn independently but generated from the structure (same-level concepts serve as distractors), so that being wrong also carries structural information. Formally, $a(q) = \text{eval}(T_q; \mathcal{O}_M, \mathcal{R})$ and $\text{Edges}(q) = \text{reads}(T_q)$. The three readings are the product of three operator families, and the correspondence is trackable throughout: the hierarchy family (`ancestors` / `descendants`) reads `is_a`, the composition family (`compose` / `refine_in_composition`) reads `composed_of`, and the condition family (`requires` / `gap_requires`) reads `requires` (full signatures of the six relations in Appendix A.5.2 / A.2, of the six operators in A.3).

Visibility comes last: since knowledge is outside the model it can be supplied by tier – withheld (`blind`), complete (`full`), names without structure (`flat`), structure in the wrong order (`shuffled`); all four come from one rendering function, with the item surface, the model and the response format held fixed otherwise. This is the premise of all attribution.

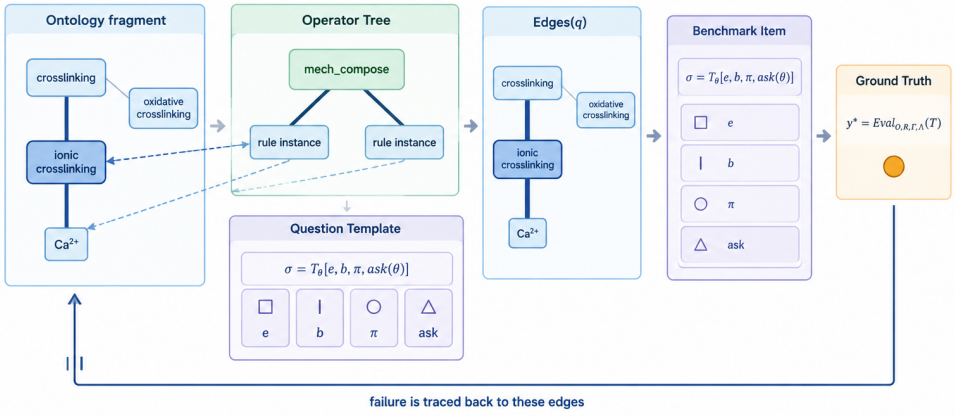


Figure 2: An item from ontology to truth value. From the left: ontology fragment (the ionic-crosslinking example of Section 3.1.2: mechanism, superordinate, same-level siblings, trigger condition) → operator tree (root `mech_compose`, two leaves being rule instances, dashed arrows pointing back to the edges it reads) → edges read Edges(q) (the two edges read are bold in the figure; sibling edges not read stay thin) → item surface (template and four slots) → gold label. The back-pointing arrow at the bottom is the point of the whole figure: failure is eventually traced back to those two bold edges — the unit of attribution is therefore the edge, neither the item nor the whole mechanism (Section 3.3.2).

To walk that chain once, take a real item from the main cohort of Section 4 (readings from the primary model). The item asks what the mechanism `thiol/disulfide_exchange` requires. The `full` tier hands the model the ontology verbatim: the mechanism sits at depth 2 under the superordinate `self_healing`, and its `requires` edges point to `ph` and `redox_condition`. The surface supplies only the slots, and the gold is what the operator

tree `requires(thiol/disulfide_exchange)` → `ph, redox_condition` evaluates to, with the edges it reads recorded as Edges(q). The item and the rendering function are fixed; only visibility changes. Under `blind` the model answers "thiol, disulfide", reading the mechanism name as a condition (wrong); under `flat`, which gives names without structure, it answers "cannot determine" (wrong); under `full` it answers "ph, redox_condition" (right); and under `shuffled`, whose structure is in the wrong order, it answers with the edges of another mechanism ("qcs_c_bridging, extrinsic_pathway", wrong). Comparing the two arms attributes the item: it is a rescue (wrong without the knowledge, right with it), charged to the `requires` edge it depends on; items on the same knowledge unit aggregate into compensability, which is what decides whether to fill the knowledge source or adjust the model (Sections 3.3, 4.5). The model gets the item right under `full` not because it has memorised this mechanism-condition pair, but because that edge is readable in the view – exactly why this paper turns visibility into a manipulable variable.

Four gates decide admission: anchoring (the surface must genuinely require that query), response form (matching the operator), dependence (the edges it reads are recorded) and visibility compliance (the tier supplies exactly what the item needs); non-compliant items are not released. Generation grammar, quality control and procedure are in Appendix A.6–A.7; one engineering discipline is recorded with them: every coupling point with natural language is confined to the template layer, so changing models does not change the structure.

Knowledge Intervention-based Evaluation and Diagnosis

This module answers which side a failure should be charged to, by combining one item's two controlled results into a single decision.

Two arms. The `blind` tier is called the **masked arm** (no external knowledge) and the `full` tier the **supplied arm** (complete ontology): the same items run once at each, with everything but visibility held fixed. At item level the pair falls into four classes: wrong-to-right (`RESCUE`), still wrong (`PERSIST`), right-to-wrong (`NEG`), right under both (`BOTH_OK`). At unit level items are aggregated: across the items a knowledge unit gets wrong, the share that flips is that unit's compensability – how much of its failure outside knowledge can rescue. Writing $a_0(q)$ / $a_1(q)$ for whether the same item is scored right under `blind` / `full`:

$$R_{\text{rescue}}(u) = \frac{\left|\{\, q \in u :\ a_0(q) = 0,\ a_1(q) = 1 \,\}\right|}{\left|\{\, q \in u :\ a_0(q) = 0 \,\}\right|},$$

the four classes being the four combinations of (a_0, a_1): $(0,1)$ rescue, $(0,0)$ persist, $(1,0)$ neg, $(1,1)$ right under both; the aggregation protocol and the full definitions of the three readings are in Appendix A.9. It must be read together with a second quantity, the error burden (the share of the bank the unit touches), or a three-item unit reads as equal in importance to a large one.

The reading is not "who is right" but "what to change next": high-compensability units point at the knowledge source, low ones at the model. Whether an intervention worked is decided by its net gain – the target unit's improvement minus the control unit's change – which is why an intervention must carry a control and why absolute values are not used (otherwise the general benefit of a longer prompt reads as a successful intervention). The upgrade loop is this chain completed once: measure the gap, fill it, retest, with a criterion at every step (Section 4.5). The meaning of the three readings and the aggregation formula are in Appendix A.9, and the protocol and statistical conventions in Appendix C.

Experiments

This chapter checks the five links of Section 1.4 one by one: Section 4.1 states the instrument and its self-check, Sections 4.2–4.6 give five sets of readings, Section 4.7 combines them into one reading table, and Section 4.8 audits the results themselves. The way each link is checked, and the readings still missing, are in Appendix E.1.

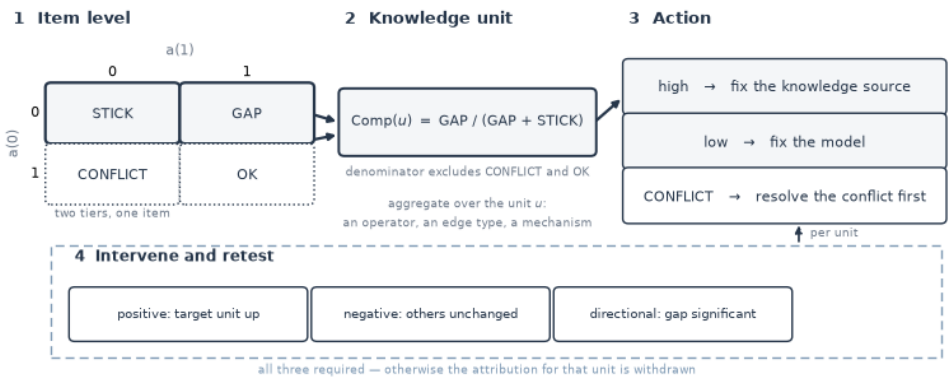


Figure 3: From intervention to diagnosis. (1) Item level: one item gets one result under each tier and falls into one of four classes; (2) unit level: items are aggregated by knowledge unit into compensability and error burden; (3) direction: the combination of these readings locates the short board, and an intervention retest decides whether to adopt or withdraw it (the bottom row of Figure B.1 draws exactly this chain).

Setup and Instrument Self-check

All readings below share one instrument: apart from the manipulated factor, the item surface, the model and the response format are identical.

Item	Value
Ontology and rules	`mo-v33`: 896 concepts (861 mechanisms), 1603 edges; `gk-v34`: 6180 rules (6179 with parseable effects)
Item bank	1274 items over 9 canonical types (batch T20)
Knowledge visibility	four tiers `blind` / `full` / `flat` / `shuffled`, from the same rendering function
Scoring and statistics	`strict` (primary) / `set` / `struct`; item as the paired unit, Holm correction
Model and decoding	`gpt-4.1` (primary; others in Appendix D); `temperature` 0, `max_tokens` 2048, one answer per item
Run-to-run variation	1.41% accuracy difference across repeated runs (deterministic scoring); call and logging detail in Appendix C

Table 1: Resources and instrument settings

The intervals below take the item as the unit (Wilson / paired bootstrap), a different basis from the run-to-run variation above (1.41%). The three protocols divide the work: `strict` is primary, `set` shows that an exact match is not over-strict, and `struct` separates "is the reasoning right" from "is the answer right". Every section assumes that the gold labels are correct and that the item asks for what the operator wants; both must hold on the data first:

Check	Procedure	Measured	Falsification condition
Truth recomputation	re-execute all 1274 items, $y^* = \mathrm{Eval}_{\mathcal{O}_M}(T)$	bit-identical rate 1.000	any mismatch ⇒ Claim two fails
Query correspondence	item wording ↔ principal operator (Section 3.2.1)	agreement 1.000	any mismatch ⇒ Claim one fails
Scorer injection	known correct/incorrect answers (empty set, superset, disorder, multi-value)	100% judged right, zero false positives	a wrong answer judged right ⇒ all claims fail
Version and anchoring	ontology/rule versions compared; $\mathrm{Info}(q) \subseteq \mathrm{View}(\kappa)$ with matching traversal	agreement 100%; pass rate 100% (non-compliant items discarded)	mismatch ⇒ Claim three fails; a leak ⇒ the tier measures missing information

Table 2: Instrument self-check (item-by-item detail in Appendix E.1)

With the self-check passed, the two arms over the full bank (1274 items) read masked 0.150 (95% CI [0.131, 0.171]) to supplied 0.818 ([0.796, 0.838]), an overall compensability of 0.789 (paired bootstrap [0.764, 0.813]); the four classes are knowledge gap 67.0%, wrong even when given 18.0%, wrong when given 0.2%, right under both 14.8%. (The mapping of the seven experiments onto the six claims, and the protocol behind each table, are in Appendix E.7.)

Parametric Knowledge Probe: the knowledge-masked Arm Reads Parametric Coverage

The knowledge-masked arm gives no ontology information, so its accuracy is a lower bound on the knowledge the parameters hold:

Canonical type	Items	Masked	95% CI	Supplied	95% CI
`requires`	95	0.000	[0.000, 0.039]	0.653	[0.553, 0.741]
`descendants`	110	0.091	[0.050, 0.159]	0.718	[0.628, 0.794]
`ancestors`	780	0.203	[0.176, 0.232]	0.982	[0.970, 0.989]

Table 3: knowledge-masked and knowledge-supplied accuracy by canonical type (full bank, n=1274)

All three types ask about facts directly retrievable in the ontology, yet the knowledge-masked arm reads 0–20% and the knowledge-supplied arm 0.65–0.98: **the parameters hold almost none of this domain knowledge**. The exact 0.000 for `requires` is interpretable (a set of about ten items under exact match, nothing to start from with no ontology, upper bound 0.039); whether exact match is over-strict is answered by the `set` protocol reported alongside, the two nearly coinciding (Appendix E.8).

Parametric coverage is a property of the model, so the column moves with the model. On one fixed cohort of items:

Model	Protocol	Masked	Supplied	Δ	R_{rescue}	McNemar p
`gpt-4o-mini`	generative	0.193	0.633	+0.440	0.554	4.7×10^{-19}
`deepseek-v4.1-flash`	generative	0.160	0.540	+0.380	0.476	9.0×10^{-15}
`gpt-4.1` (primary)	generative	0.220	0.547	+0.327	0.479	1.4×10^{-10}
`jev`	decision	0.113	0.427	+0.313	—	—

Table 4: Two-arm readings across models on the stratified cohort (150 items)

Δ is positive for all four models: the direction "what is missing is knowledge, not reasoning" holds across models (pre-registered criterion >=3/5 and how it was formed: Appendix E.4); `jev` uses the decision protocol and serves only as corroboration of direction. A score-only benchmark cannot produce this column – a high total may come from knowledge or from reasoning, and the masked arm separates them. Two further models ran the same instrument over the **full bank of 1274 items** (pairing complete): `deepseek-v4.1-flash` masked 0.048 $\rightarrow$ supplied 0.790 (Delta **+0.742**) and `glm-5.3-flash` 0.056 $\rightarrow$ 0.794 (Delta **+0.738**), both at $p < 10^{-260}$. The full bank is dominated by the hierarchy family (`ancestors`, 780 of 1274), so these Deltas exceed those above (different composition, same protocol); per-operator and per-canonical-type readings are in **Appendix D.2**. (The `Claude` family and `gpt-5.x` are not served by the endpoint used and no readings were obtained; remaining models are in Appendix D.)

Four Tiers: does structure itself help

Section 4.2 used two tiers only; "knowledge is manipulable" and "writing relations as operations helps" are two different propositions and must be verified separately, so we state them first as two falsifiable hypotheses: H1 (monotone) `blind` $\leq$ `flat` $\leq$ `full` – the more supplied, the better; H2 (structure necessary) `full` $>$ `shuffled` – the structure must be correct.

Main conclusion: **the correct structure is necessary (H2 holds) and concept names alone add nothing (H1 fails, read as equivalence)**. `full` 0.547 against `shuffled` 0.260 (paired McNemar 46:3, $p < 0.001$, paired bootstrap CI [+0.233, +0.413]); `flat` minus `blind` has an upper bound of +1.3 points ([-8.0, +1.3], $p = 0.27$) – a bound, so within ± 5 points the list can be treated as adding nothing. The length-matched irrelevant-name control `flat-irr` reads 0.180, indistinguishable from `flat` ($p = 1.0$), so the `flat` reading

Tier	What is given	Full bank (n=1274)	Stratified cohort (n=150)	95% CI (item-level)
`blind`	no ontology information	0.150	0.220	[0.161, 0.293]
`flat`	concept names only, no structure	—	0.187	[0.132, 0.257]
`flat-irr`	length-matched but irrelevant concept names (control)	—	0.180	[0.127, 0.249]
`full`	complete ontology fragment	0.818	0.547	[0.467, 0.624]
`shuffled`	structure in wrong order	—	0.260	[0.196, 0.336]

Table 5: What each tier supplies, with the length-matched irrelevant-name control

does not depend on whether the names are relevant. Had `flat` sufficed, writing relations as operations would have to be withdrawn – the falsification condition is stated before the numbers (bounds, control and two spurious readings in Appendix E.9); the same order replicates on `deepseek-v4.1-flash`: `blind` 0.160 / `flat` 0.140 / `shuffled` 0.280 / `full` 0.540.

The Boundary Between Two Kinds of Failure: Compensability

Splitting items by whether the answer is supplied directly in the view, the difference between the two arms gives two very different readings. The column measures whether the material *contains* the answer: for the `ancestors` family the answer is the ancestry chain the material must supply (200/200 in a spot check), so the column is saturated there (622 of 1274 items) and the 0.905 is not a reading of capability. But the saturation is confined to that operator – `descendants` (0.250 on the cohort / 0.690 on the full bank) and `requires` (0.650 / 0.653) are also retrieval-type and are *not* saturated, so treating the whole type as a dead column would throw both usable readings away.

Type	Items	Masked	Supplied	Compensability	95% CI
Retrieval-type (answer listed)	985	0.171 [0.148, 0.195]	0.921	0.905	[0.884, 0.925]
Reasoning-type (material only)	289	0.080 [0.054, 0.117]	0.467	0.432	[0.373, 0.492]

Table 6: The retrieval type and the reasoning type

The two intervals do not overlap: the retrieval type's lower bound of 0.884 exceeds the reasoning type's upper bound of 0.492, so the boundary does not depend on the precision of the point estimates, nor on the particular 985/289 split. Two judgements hold at once: first, the retrieval column cannot be read as capability – more precisely, the `ancestors` family is saturated (see above) – and mixing the two columns reads a uniformly high value as "the model is strong"; second, the reasoning type reaches only 0.467 under the knowledge-supplied arm, so more than half of its failures survive even with the material supplied, and those failures are not missing knowledge. By canonical type:

#	Canonical type	Items involved in failure	Compensability	95% CI	Side
1	`ancestors`	622	0.977	[0.965, 0.989]	retrieval
2	`descendants`	100	0.690	[0.598, 0.779]	retrieval
3	`requires`	95	0.653	[0.558, 0.747]	retrieval
4	`compose`	79	0.506	[0.392, 0.620]	reasoning
5	`gap_requires`	79	0.481	[0.370, 0.593]	reasoning
6	`compete_closed`	69	0.449	[0.329, 0.569]	reasoning
7	`refine_in_composition`	8	0.250	[0.000, 0.571]	reasoning
8	`block`	30	0.133	[0.031, 0.267]	reasoning

Table 7: Compensability and attributed side by canonical type

On point estimates the stratification is clean and non-interleaved, but the intervals call for one qualification: at the boundary `requires` (upper 0.747) and `compose` (lower 0.392) overlap, so adjacent classes are not separated by a sharp line; what truly does not overlap is the two ends (`ancestors` [0.965, 0.989] against `block` [0.031, 0.267]). The table is therefore used to **locate the ends and hold the middle in doubt**, not as a strict ranking. The pair of results yields a third class as well: right without the view but wrong with it, 0.2% (3 of 1274 items), all of the reasoning type – it indicates neither missing knowledge nor a

failure to reason but a conflict between parametric and supplied knowledge, which must be resolved rather than supplemented. The share of this cell varies with item composition – retrieval-type items make up 70% of the full bank and almost never fall here, so the cell is more frequent in the operator-balanced cohort. (Rows with insufficient denominators and the stratified re-check, with both readings and their cause, are in Appendix E.10.)

This split is defined by the instrument (a judgement about how the view renders for that item, not a property the item carries), so it has to answer whether it is self-serving. As a sensitivity check we use a device-independent grouping instead: by the operator declared before the items were built (hierarchy class = `ancestors` / `descendants`; relation/condition class = the rest), looking only at the operator name and never at whether the answer is listed. On the full bank (1,267 items) the hierarchy class reads compensability 0.941 [0.923, 0.957] against 0.481 [0.429, 0.535] for the relation/condition class – non-overlapping; on the main model's 150-item cohort, 0.625 [0.475, 0.775] against 0.312 [0.171, 0.432]. The two splits draw the same line, so the line is not an artefact of the instrument's own definition (both tables are in Appendix E.15).

Ontology Gaps and One Closed Loop

Attribution says what is missing; whether filling it helps must be verified too, and the order matters – filling first would leave the original gap unknown. First, find the gap: comparing the ontology's `requires` edges with the rule layer's condition dimensions mechanism by mechanism gives coverage of 27.8% – the rule layer registers condition dimensions of which the ontology records three tenths, echoing the 0.037 those items read knowledge-masked. Second, fill and retest: 431 `requires` edges are written back (27.8% → 98.4%).

Criterion	Result
Positive	`gap_requires` 0.098 → 0.500 (5.1×); 33 turned right, 0 turned wrong; $p = 2.3 \times 10^{-10}$
Negative	the other 1097 items 0.860 → 0.853 (-0.0064); $p = 0.189$, not significant
Directional	the difference between the two +0.408

Table 8: The three criteria of the closed loop and their measured results

The negative side matters most: adding edges lengthens the fragment of every related item, so without it the lengthening itself would be misread as a successful fill; measured, the negative side falls by 0.6 points and is not significant, so the gain can be credited to the edges added. Which condition dimensions were excluded and why is in Appendix E.11.

The conclusion stops here: **what can be diagnosis-driven is the instrument** – the positive side says only that filling the diagnosed position lifts the target unit, claiming neither that a larger knowledge base is better nor that the model's capability changed. Three controls pin this down. First, the **negative side**: the other 1,097 items move by -0.6 points ($p = 0.189$, not significant); adding edges lengthens the fragment of every related item, so without this control the lengthening itself would be misread as a successful fill. Second, a **mis-fill accounting** per operator: only the diagnosed `gap` class gains +0.354 (29 turned right, 0 turned wrong, $p = 3.7 \times 10^{-9}$), the undiagnosed `requires` class falls by 0.250 (0 turned right, 22 turned wrong, $p = 4.8 \times 10^{-7}$) and the rest move by -0.051 to 0.000, giving 0.8232 → 0.8232 overall (net zero). Third, the **premise itself is broken**: `requires` drops from 1.000 to 0.750 after completion (the fourth discipline, Section 4.8). Which condition dimensions were excluded, and the per-operator table, are in Appendix E.11 / E.15. (The per-operator table and the two splits side by side are in Appendix E.15.) The same test is more complete on public ontologies: upgrading exactly the diagnosed layer lifts the diagnosed unit's compensability from zero to 0.550-1.000 (hierarchy +0.550 to +0.733, role +0.900 to +0.967, negation +0.917 to +1.067, consistent across three models with no pair running right-to-wrong), while the memory-type control family falls (-0.250) – **after the attribution diagnosis performance improves, and only the diagnosed class improves** (Table 12 of Section 4.9; per-unit detail and the explanation of the readings that fall short are in Appendix E.13).

Channel Comparison: two question forms for one piece of knowledge

For one mechanism and one body of knowledge, a relation-form and a hierarchy-form question are asked, with everything else fixed (76 groups covering 76 mechanisms):

Class	Pairs	Share
Both channels right	37	48.7%
Relational only	2	2.6%
Hierarchical only	12	15.8%
Both wrong	25	32.9%

Table 9: Distribution of groups in the channel comparison

The relation arm reads 0.513 and the hierarchy arm 0.645; the two arms share the same mechanisms (not two independent draws), so McNemar rather than Fisher applies – only the 14 discordant pairs (2 vs 12) carry information ($p = 0.0129$, odds ratio 6.0). Models walk the classification tree more readily than they read relation edges: **even when the edges are in the view, they tend to answer through the hierarchy alone**, which yields one operational direction – present the key relation edges in hierarchy form as well, rather than adding relation edges only (reading against Section 4.3 and the choice of test: Appendix E.12).

From Attribution to Guidance: one reading table

The preceding sections each gave evidence for one facet of the instrument; this combines them into one actionable reading table:

Knowledge unit u	Granularity	Items	$a(0)$	$R_{\text{rescue}}(u)$	Direction	Intervention	Result
Mechanisms lacking conditions	mechanism family	82	0.037	0.481	knowledge base	add 431 `requires` edges	+0.408 (Section 4.5)
`ancestors`	operator	780	0.203	0.977	knowledge base	—	—
`compete_closed`	operator	79	0.127	0.449	leaning model	—	—

Table 10: Attribution reading table

How to read it: direction first, then magnitude, then the retest – a positive direction with high compensability means supplying knowledge can rescue the unit (fix the knowledge source); low compensability with a high error burden means it stays wrong when supplied (fix the model); only the retest column turns a recommendation into a fact. (What each of the three readings means and which side it points to is in Appendix E.13; the two units and the ordering rationale are in Appendix E.6.)

Two units have no retest yet, each for its own reason: the compensability of `ancestors` follows from the relation between the operator and the material (Section 4.4), so no intervention has a place to land – its improvement is on the item side (a stricter rendering); and `compete_closed` points at the model but needs a different branch (prompt injection), scheduled for the next round. Rows without an intervention are marked "—" and never read side by side with the retested one.

Self-audit and Threats to Validity

Four disciplines, each corresponding to one experience of misreading an error as a finding – this class of error raises no alarm and looks exactly like genuine difficulty in the result table. Inspect the scorer (the free-reasoning arm once scored a third of the multiple-choice arm: the scorer segmented by paragraph and treated reasoning steps as answer options); inspect the item surface (an anomalously high knowledge-masked arm means suspecting the surface writes out the answer – a newly added item type read 0.867 masked against 0.700 supplied, and all 300 such items were withdrawn); inspect the difficulty measure (of five candidate quantities two rounds of univariate testing left three, and one moved from $p = 1.0$ to 0.0015 once mechanism size was controlled); inspect versions (after the ontology was completed, `requires` items fell from 1.000 to 0.750 because annotations predated the

completion while the view was rendered from the new ontology, and nothing raised an error). Together they yield one maintenance discipline: **once the ontology changes, items whose dependence is invalidated or whose truth value is stale must be regenerated** rather than given a fresh truth value in place (full record in Appendix E.14).

Two threats. **Model coverage**: the main readings come from `gpt-4.1`, and a single model would make this a case study – the pre-registered criterion is >=3/5 models in the same direction (Appendix E.4), and 8 of 8 measured models agree (four in Section 4.2; `glm-5.3-flash` +0.400, `qwen3.8-flash` +0.380 and `gpt-4o` +0.360 in Appendix D). **Absolute scale**: no such claim is made. Reading 0.467 for the reasoning type under the knowledge-supplied arm as "poor" requires a reference ceiling; a proxy check (two proxy annotators, n=87, given only the model's material) reads 0.478 and 0.586, the same band as the model's 0.552 – a reader with the same material does not supply that ceiling. Only the absolute scale was ever at stake: the two-arm difference, the ordering of units and the net gain of the loop are relative quantities (protocols and the protocol gap this exposed: Appendix E.16).

Additional experiments

Protocols are in Appendix C, criteria and measured results in Appendix E.4. Four readings are measured: cross-model agreement (Δ positive for all four models), wording robustness (139/150 item-level agreement), the locally testable part of the contamination check (0/150), and the external comparison on public ontologies (below). Human comparisons have a proxy round (Appendix E.16); the ChemBench comparison (32) gives one layer less – a capability scale rather than "which edge" – and is reported in Appendix D.3 with no comparability claimed.

All readings above come from our own ontology. To ask whether the pattern is confined to this domain, the same instrument rebuilt 180 items on two public biomedical ontologies (ANATOMY, GALEN, from the OnT dataset): each asks for a concept's grandparent-level superordinates, so the answer takes two steps over the hierarchy and the supplied view gives only the one- and two-hop edges.

Model	Protocol	Masked	Supplied	Δ	R_{rescue}	McNemar
`gpt-4.1`	generative	0.011	0.717	+0.706	0.713	127/0, $p = 1.2 \times 10^{-38}$
`deepseek-v4.1-flash`	generative	0.000	0.728	+0.728	0.728	131/0, $p = 7.4 \times 10^{-40}$
`glm-flash`	generative	0.000	0.511	+0.511	0.511	92/0, $p = 4.0 \times 10^{-28}$

Table 11: Two-arm comparison on public ontologies (180 items, 90 per ontology)

Masked accuracy is near zero for all six models (0.000–0.011) and rises to 0.511–0.761, Δ positive throughout, paired tests highly significant and no pair running the wrong way: the models do not memorise these public ontologies, yet walk the two steps once the structure is given. The comparison covers the hierarchy family – these ontologies lack disjoint pairs and condition dimensions, so the disjoint and condition operators become the object of the loop below – and a second rebuild extends it to role and negation (Appendix E.17).

Diagnose, upgrade, retest (public-ontology version): after the diagnosis, performance comes back. This is the most direct test of the claim. The diagnosis says "the gap is in the structure", so exactly that one layer is upgraded and nothing else changes – same items, same instruction, same scoring protocol.

The upgrade is read on **two sides**, and this applies to every table in this subsection. **Ontology side**: the layer's executable structure goes from absent or incomplete to present and recomputable (counted in edges, chains or closure size); this gain is **unconditional**, the only criterion being that the added structure can be recomputed. **Model side**: the score change on the same items (supplied gain minus masked gain); whether it shows up depends on **whether that knowledge is already in the parameters**, with the masked arm as the probe – high masked means little room to compensate, so no marginal gain *for that model*, which is not the same as the structure being useless.

Unit	Variant	Masked	Supplied	Delta	Compensability R_{rescue}
Hierarchy (two-hop superordinate)	v1, second-hop edge missing	0.000	**0.000**	0.000	0.000
Hierarchy (two-hop superordinate)	v2, after upgrade	0.017	**0.717**	**+0.700**	**0.712**
Role (two-hop role chain)	v1, first hop only	0.050	**0.000**	-0.050	-0.053
Role (two-hop role chain)	v2, after upgrade	0.050	**0.900**	**+0.850**	**0.895**
Negation (closure comparison)	v1, closure truncated	0.733	**0.017**	-0.717	-2.687
Negation (closure comparison)	v2, after upgrade	0.783	**1.000**	**+0.217**	**1.000**
Mutual exclusion (explicit `disjointWith` + closure) – ontology side: 0 → 549,676 executable pairs	v1, declarations removed	0.333	**0.950**	+0.617	0.925
Mutual exclusion (explicit `disjointWith` + closure)	v2, after the ontology upgrade	0.383	**1.000**	**+0.617**	**1.000**

Table 12: Two-arm readings before and after the upgrade (public ontologies, 60 items each, paired)

The upgrade effect is the same across three models, with no pair running right-to-wrong: on the hierarchy unit the post-upgrade supplied readings are 0.717 / 0.733 / 0.550 (pairs 42 / 44 / 33); on the role unit 0.900 / 0.933 / 0.933 (net gains +0.900 / +0.967 / +0.917, pairs 51 / 55 / 54); on the negation unit 1.000 / 1.000 / 1.000 (net gains +0.933 / +0.917 / +1.067, pairs 13 / 46 / 18) – every $p \leq 7.6 \times 10^{-6}$. The mutual-exclusion unit is where the two sides diverge: on the **ontology side** the gap is closed unconditionally (0 → 549,676 executable disjoint pairs from 17 explicit axioms), while on the **model side** `gpt-4.1` gains **+0.000** from the upgrade (within the noise floor) because that knowledge is already in its parameters (0.350 masked, little room to compensate for) – not a useless structure. On `deepseek-v4.1-flash` (0.033 masked) the same unit gains 0.934. **The ontology-side gain is unconditional; the model-side gain is conditional**, with the masked arm as the probe – which is why readings are reported on both sides and model by model. The rows above use `gpt-4.1`; the three models are listed side by side in Appendix E.13, and the two upgrade artefacts themselves in Appendix E.17.

Three readings. First, **before the upgrade the supplied arm scores zero as well** (hierarchy 0.000, role 0.000, negation down to 0.017): that layer of structure is a **necessary condition** for the answer, so the gap is in the structure and not in the model – the diagnosis holds. Second, **after the upgrade the supplied arm recovers sharply while the masked arm barely moves** (0.050 to 0.050; 0.733 to 0.783): the gain comes from the edge that was added, not from a longer context or a stronger model, and compensability moves from zero (or negative) to 0.550-1.000. Third, **the attribution has to tell two kinds of gap apart**: the negation reading under v1 (masked 0.733 to supplied 0.017) says that a truncated closure is worse than no closure at all, and such a failure has to be diagnosed; the memory-type definition family (not derivable from structure) moves by -0.250 under the same instrument (0.950 to 0.700, 0/15), and no upgrade rescues it. **Only the diagnosed class is lifted** – the conclusion stops here: after the attribution diagnosis performance does improve, but what improves is the part the instrument can drive, and no growth of the model's internal knowledge is claimed. (Scoring protocol `set`; `strict` moves the same way – on the hierarchy unit v2 reads 0.633 / 0.717 / 0.517, Appendix E.8.)

Can the Diagnosis Predict the Repair?

The closed loop of Sections 4.5 and 4.9 is **ex post agreement**: the gain shows up after filling. To earn the claim that the readings are useful, they must name the layer to fill **before** the intervention – under a **pre-registered** protocol (predictors, outcome and kill criteria frozen first: Appendix C.7), because picking a unit that improved after the fact is not prediction.

One measurement protocol has to be repaired first, or none of the numbers below may be used. The v1 and v2 item sets had been drawn in two independent sampling passes; a per-item check showed that for the negation and the mutual-exclusion families the same qid did **not** carry the same question surface (0 of 60 aligned). With no pairing, the Deltas and any predictive analysis are void – the second instance of discipline four (version check, Section 4.8). The fix is **same-item view surgery**: the structure-complete version is the base, and v1 is that very item with the family's structure deleted; surface, candidates and gold are untouched. All five families pass at 60/60 (audit in Appendix E.18).

before and after, `gpt-4.1`)

Unit	Structural gap m	Masked $a(0)$	Supplied v1 S_1	Supplied v2 S_2	Gain Δ	Headroom $1 - S_1$	$R_{\text{rescue}}(v_1)$
Hierarchy (two-hop)	0.92	0.017	0.000	0.583	**+0.583**	1.000	-0.017
Role (two-hop chain)	0.57	0.083	0.017	0.950	**+0.933**	0.983	-0.073
Negation (closure)	1.00	0.783	0.000	1.000	**+1.000**	1.000	-3.615
Mutual exclusion A	1.00	0.333	0.967	1.000	+0.033	0.033	0.950
Mutual exclusion B	1.00	0.383	1.000	1.000	0.000	0.000	1.000

Table 13: Diagnostic readings and repair gain, strictly paired (public ontologies, 60 items each, same items

Four readings. (i) **The gain is capped by the headroom**: $\Delta \leq 1 - S_1$ holds in every row (4/4; the ceiling row stands alone). The three families whose supplied arm read 0.000-0.017 before the upgrade gain 0.583-1.000, while the two already at 0.967-1.000 gain 0.033 and 0.000. **The pre-upgrade supplied reading is the column that decides whether to fill.** (ii) **The structural gap m does not rank units**: among the three families with $m \geq 0.99$ the gain ranges from +1.000 to 0.000, and the hierarchy unit at $m = 0.92$ gains +0.583. m is necessary but not sufficient – it names the layer that was removed; whether the removed layer is the one this model needs is decided by (i). Used as a ranking quantity it misorders units ($\rho = -0.105$, permutation $p = 1.00$, $n = 5$). (iii) **What a conventional benchmark supplies cannot rank either**: with masked $a(0)$ close, the sign of the gain can differ – hierarchy $a(0) = 0.017 \rightarrow +0.583$, mutual exclusion $a(0) = 0.333 \rightarrow +0.033$, negation $a(0) = 0.783 \rightarrow +1.000$; three rows suffice to refute monotonicity ($\rho = 0.400$, permutation $p = 0.75$). (iv) **Ranking by compensability inverts the order**: the three families with a negative $R_{\text{rescue}}(v_1)$ (-0.017 / -0.073 / -3.615) gain the most (+0.583 / +0.933 / +1.000), while the two with a positive value (0.950 / 1.000) gain the least (+0.033 / 0.000). The reason is that **a truncated structure collapses the supplied arm first**, so the two-arm difference reads "not compensable" – the opposite of what is needed ($\rho = -1.0$, permutation $p = 0.093$). This is the predictive consequence of the sentence in Sections 4.5 and 4.9 that attribution must tell two kinds of gap apart: **split structure from capability first, then decide what to fill**; any single column on its own misorders the units.

Same-item dual format (conventional single choice), a 52-item pilot on the role family, same items and model, strict scoring: 0.135 when the structure is missing and 0.635 when it is present. A conventional single-choice form **does** register that structure is missing, so this section does not claim that conventional benchmarks are blind to the gap; it claims that the **extra column is the attribution column** – which edge is missing and whether filling it recovers performance (the two-arm reading and compensability), which a leaderboard does not carry. (One side observation: with the structure missing the four-way score of 0.135 is **below** the 0.25 guessing rate, i.e. the model tends to pick names that appear in the evidence.)

Boundaries: the conclusion is limited to the four families this instrument can express on two public ontologies; five to nine rows cannot support a cross-domain fit, so this section reports **bounds and counterexamples**, not fitted coefficients, and the 31 units on the in-house ontology currently supply pre-intervention vectors only (Table E.18a). Extending to the condition family (which public ontologies do not provide) would require redoing it.

The Material-Grounded Decidability Gate, the Tier Control, and the Primary Indicator

This is the heaviest issue under a benchmark-track reading: the gold is produced by executing the **full ontology**, while the model receives a one- or two-hop **fragment**. If the structure the gold needs is not in the material the model sees, a wrong answer **cannot** be attributed to "cannot derive" – the item is undecidable from the model's viewpoint. We therefore decide this item by item (`code/onto/mgd_gate.py`, all 1,274 items):

The failures concentrate in three operators: `gap` 82/82, `verify_support` 37/39 (0.949) and `granularity` 9/9 – all three need a **completeness statement** (D^-). **Disposition (three admission rules)**: (i) when the gold depends on a full closure, the view must **declare completeness**, otherwise the item is rewritten into a locally decidable form or removed from the primary bank; (ii) `leak` items go to the parametric probe only; (iii) the

Verdict	Items	Share	Meaning
`leak` – gold verbatim in the material	1,028	**0.807**	compensability is inflated (copying suffices) – **excluded from the primary indicator**
`partial` – partly visible	115	0.090	the solver must complete it
`absent` – **fails MGD**	131	**0.103**	the gold depends on a closure outside the material – **construct risk**

Table 14: Material-grounded decidability gate (supplied tier `full`)

traversal direction must match the operator (history: fourteen `granularity` items were read as "missing information" because the direction was wrong).

Tier	Characters	Nodes	Edges
`flat`	279	9.2	0.0
`full`	412	9.4	2.5
`shuffled`	357	9.4	2.5

Table 15: Tier control audit (per-item means, same 1,183 items)

The node count is controlled (9.2-9.4); the character count is not (279 / 412 / 357) – the length effect is not isolated. `flat-irr` (same length, names swapped) is the length-matched control arm built for this and was run on one cohort only; extending it to the whole bank is next.

Primary indicator. The primary indicator moves to the **reasoning-type balanced subset** (289 items; supplied 0.467, compensability 0.432); the 985 retrieval-type items (supplied 0.921, compensability 0.905) are reported separately as a **parametric probe**, and the two are **no longer averaged into one total** – the whole `ancestors` family (780 items) is `leak`, so averaging would read "the answer is in the material" as "the model uses knowledge when supplied". With the MGD gate applied on top, `absent` items leave the primary indicator. **Compensability is reported together with leakage**: R_{rescue} must be read next to $Leak(q)$, the D^- completeness and the error burden, or the rule "high compensability points at the knowledge source" is not robust (a leak share of 0.807 means the high value may be copying). **Statistical dependence**: items are not independent samples (one edge, one mechanism or one paper yields several items), so the main tables move to a **bootstrap clustered by knowledge unit and by evidence source** (`--cluster unit|paper`); the full re-run is next. **Format compliance is reported separately** from answer accuracy (the scorer once folded a `Path:` line into the answer and 0.95 fell to 0.225 – a format failure, not a reasoning failure).

The Status and Scope of the Human Expert Baseline

The pre-registered E-HUMAN / E-AGREE readings are this paper's only source of an absolute scale, and **external experts were not obtained**; what exists is a **proxy round** (two AI proxies, n=87): E-HUMAN proxy supplied 0.478 / 0.586 against the model's 0.552 (**criterion failed**); E-AGREE agreement with the executed gold 0.540 / 0.478 against a criterion of 0.95 (**failed**), while the two proxies agreed with each other (κ=0.817) – so **the disagreement is not in the people, it is in the assignment**: the proxies saw the same fragment as the model, whereas `descendants` / `bind` / `granularity` need the full ontology closure (per-operator breakdown in Appendix E.16).

The worksheet package is therefore published under the corrected protocol (`human_expert_round/`): **E-HUMAN gives the very same material the model received** (criterion: expert significantly above the model), and **E-AGREE gives the full ontology plus the operator definitions** (criterion: agreement >= 0.95). The 100 items are stratified by operator (seed `20260926`, reproducible), scoring **reuses the official scorer** so that human and model answers travel the same path, and the pipeline has been dry-run on real logs. **Until external experts report, no absolute scale is claimed**: whether the reasoning-type supplied reading of 0.467 is "poor" has no reference ceiling this round; the two-arm difference, the ordering of units and the net gain of the loop are relative quantities and are unaffected.

Discussion

What this paper does not claim

Five boundaries up front. Not the model's internal knowledge: everything measured here is reasoning given knowledge – the knowledge-masked arm reads how much is in the parameters, the knowledge-supplied arm whether the knowledge can be used once supplied, and the difference is how much outside knowledge can restore. The conclusions therefore hold where knowledge can be externalised; knowledge that needs embodied experience or tacit skill is another matter. Not that knowledge can be fully externalised: of 1274 items, 289 reasoning-type items still stand at 0.467 under the knowledge-supplied arm – supplying the material does not replace reasoning. Not that difficulty is a scalar: of five candidate quantities two rounds of testing left three, and they are comparable only within (answer form, channel); forcing a single score would repeat the mistake of stratifying by hop count. Not that scoring is a minor matter: the three scorer defects were all silent – before they were fixed, good answers were marked wrong and two item classes were pinned at zero, indistinguishable in the result table from genuine difficulty. **Not that this instrument beats existing benchmarks, and not that it is comparable to them**: the denominators and the constructs differ (course-level chemistry single choice versus mechanism knowledge read as a set), so what is compared here is not a total score but the **extra column** – which edge is missing and whether filling it recovers performance. The relation to public scores is reported as an **exploratory correlation** only (Appendix D.3), with no comparability claim; the criterion this paper uses for that column is **predictive validity**: naming the layer to fill before the intervention (Section 4.10), under criteria frozen in advance (Appendix C.7). **Not mechanism-level attribution**: attribution lands at the operator and family level (per-mechanism counts do not support a mechanism-level reading; that remains the target granularity). **And neither an absolute scale nor cross-domain generality**: the expert baseline is unfinished (Section 4.12) and the condition family has not been replicated on a non-biomedical public ontology.

Limitations and next steps

Corpus scale is the main constraint: evidence anchors land on the block tables of a few papers, so "warrant checkable against the source" holds only within the papers those block tables cover; 52 expressions are instantiable on the ontology and only part is covered. A gap remains between ontology and rule layer: the 431 condition dimensions added raise coverage from 27.8% to 98.4%, but a further batch of condition dimensions is still unrecorded and the two layers' formats are not fully isomorphic, so cross-layer comparison needs extra mapping. Attribution is associational: operators and edge families are highly coupled, so the main text reports within-family stratification and makes no cross-family causal claims. The public-ontology comparison reaches the hierarchy family only, and can serve as a hierarchy-family measure only (ANATOMY and GALEN lack disjoint pairs and condition dimensions): relative to an in-house ontology it still falls short, and the condition and competition families fall outside its coverage. The cost of a self-built ontology must be recorded too: concept boundaries, condition dimensions and mutual-exclusion pairs are declared by experts, so this paper reports only that the instrument can be diagnosis-driven and does not extend the claim to domain knowledge itself; replicating the condition and competition operators on public ontologies is not yet done (candidate resources in Appendix C.4).

Three next steps, ordered by executability. First, fix the maintenance procedure: every time the ontology is versioned, run the regeneration check once (Section 3.2.6) and list items whose dependence is invalidated or whose truth value is stale – the decoupling measured here (accuracy falling from 1.000 to 0.750 with no error raised) shows this cannot be left to memory. Second, expand the set of instantiable expressions, each new class directly increasing the number of reportable knowledge units. Third, make "externalising a query" a measurable behaviour: since models do not naturally externalise uncertainty, the question can be reversed – given the readings of this instrument, can a model learn to retrieve on knowledge-gap items and not to retrieve on reasoning-failure items, which moves the paper from measurement to intervention.

Artefacts, Versions and Maintenance

Everything needed for independent reproduction is released with the supplementary materials: the **ontology** (mechanism ontology with its value domains, condition ontology, material and entity vocabularies, the OWL export and its reasoned product, each carrying an `ont_version`), the **rule layer**, the **item banks** (main bank of 1,273 items, the frozen baselines A-E, and the public-ontology sets including the same-item-surgery v1a variants and the MCQA pilot), the **scorer** (`evalgen/judge.py` is the only scoring implementation), the **diagnosis and prediction scripts** (`onto/mgd_gate.py`, `public_onto/analyze_predictive.py`, ...), the **human-expert worksheet package**, and a `MANIFEST.csv` with a sha256 per file plus a licence note. Maintenance plan: after every ontology version bump run the regeneration check (Appendix A.VI.6) and list the items whose dependence is invalidated or whose truth value is stale; every change to the scorer must come with a before/after comparison on the frozen subset (one silent defect is already on record).

AI use statement

Literature search and typesetting were carried out with language-model assistance for: summarising related work and checking citations, editing terminology and prose, and producing the LaTeX skeleton. All claims, experimental designs, data and scoring protocols are the authors': model output was verified item by item, and every number in the paper comes from a re-runnable artefact in our repository (per-item logging; scoring uses no model judge). Numbers that cannot be traced to a specific run were not written down.

Ethics statement

This work uses only public biomedical ontologies (ANATOMY, GALEN, GO) and published materials-chemistry evidence. No human subjects, personal data or clinical records are involved. The failure modes we expose are evaluation artefacts rather than deployment hazards: they make published numbers *over*-state capability, so we release the diagnostics as controls.

Reproducibility statement

Four checkable deliverables ship with the paper: (i) the mechanism ontology and rule layer (versioned; consistency is verified before evaluation); (ii) the item bank, where each item records the edges and knowledge units it depends on and its gold label is obtained by executing the structure, hence recomputable position by position; (iii) the four-tier view renderer and the scoring scripts (deterministic rules, no model judge); and (iv) the items, views and judging scripts of the public-ontology comparison and of the diagnose–upgrade–retest loop. Logging is per item and runs are resumable; the source of every reading is listed in Appendix A.

References

[1] Chempaperbench., 2024.

[2] Benchmarks as microscopes: A call for model metrology., 2024.

[3] Diagnosing failures in large language models' answers: A dataset for error attribution. Findings of ACL 2025, 2025.

[4] Disentangling memory and reasoning ability in large language models. In *ACL*, 2025.

[5] Knowledge or reasoning? a close look at how llms think across domains., 2025.

[6] Middo: Model-informed dynamic data optimization for targeted fine-tuning. EMNLP 2025, 2025.

[7] Novelhopqa: Diagnosing multi-hop reasoning failures in long-context qa. arXiv:2506.02000, 2025.

[8] Potemkin understanding in large language models., 2025.

[9] Lessons from the trenches on evaluating machine-learning systems in the sciences., 2025.

[10] Autonomous research agents: A survey., 2026.

[11] Authoritybench: A fully balanced 2x2 factorial benchmark. ICML 2026, 2026.

[12] Data and evaluation closed-loop for model capability. arXiv:2606.28471, 2026.

[13] Diagnostic knowledge graphs. KDD 2026 Workshop, 2026.

[14] Disentangling knowledge and reasoning in medical large language models. arXiv:2505.11462, 2026.

[15] Doctor-rag: Failure-aware repair for retrieval-augmented generation. arXiv:2604.00865, 2026.

[16] General scales unlock ai evaluation with explanatory and predictive power. *Nature*, 2026. doi: 10.1038/s41586-026-10303-2.

[17] Hierarchical attribution analysis of deficiencies in the reasoning ability of large language models. CAICE '26, 2026.

[18] Kg-mulqa., 2026.

[19] Mollangbench: A comprehensive benchmark for language-prompted molecular structure understanding. In *ICLR*, 2026.

[20] Naviedu: Dual-path knowledge tracing with prerequisite-gap detection and targeted guidance. ACM 2026, 2026.

[21] Neurosymbolic large language models: A survey of symbolic reasoning integration. *Information Systems Frontiers*, 2026. doi: 10.1007/s10796-026-10794-4.

[22] Beyond the plateau: A systematic study of benchmark saturation. arXiv:2602.16763, 2026.

[23] Lost in benchmarks? rethinking large language model benchmarking with item response theory. arXiv:2505.15055, 2026.

[24] A systematic taxonomy of failure modes in retrieval-augmented generation. ACL 2026 TrustNLP, 2026.

[25] Judge the raters: Rasch measurement theory for llm evaluation. arXiv:2608.27463, 2026.

[26] Verifiable counterfactual supervision for process reward models., 2026.

[27] From benchmark accuracy to pharmacological credibility., n.d.

[28] Semantic f1: Soft matching and equivalence classes for multi-answer evaluation., n.d.

[29] Alaa A, Hartvigsen T, and Golchini N. Medical large language model benchmarks should prioritize construct validity., 2025.

[30] Asai A, Wu Z, Wang Y, Sil A, and Hajishirzi H. Self-rag: Learning to retrieve, generate, and critique through self-reflection. ICLR 2024, 2024.

[31] Kumar A et al. Difficulty-controllable multi-hop question generation from knowledge graphs. arXiv:2105.11698, 2021.

[32] Mirza A et al. A framework for evaluating the chemical knowledge and reasoning abilities of llms against the expertise of chemists. arXiv:2404.01475, 2025.

[33] Jin B et al. Graph chain-of-thought., 2024.

[34] Bartmann C et al. Molecularіq. arXiv:2601.15279, 2026.

[35] Desmarais M C, Beheshti B, and Naceur R. Item to skills mapping: Deriving a conjunctive q-matrix from data. In *Intelligent Tutoring Systems*, pp. 454–463, 2012.

[36] C.P., M.K., E.S., and H.K. Grounded verification of chemical and materials reasoning: Detection is the bottleneck. arXiv:2607.17417, 2026.

[37] Cao D, Cai Y, Huang Y, He X, Guo R, Liu G, and Sun G. On the superimposed noise accumulation problem in sequential knowledge editing of large language models. AAAI 2026 (arXiv:2505.07899), 2025.

[38] Wadden D et al. Fact or fiction: Verifying scientific claims., 2020.

[39] Markowitz E et al. Kg-llm-bench., 2025.

[40] Vitali F and Pasqual V. Provenance-enhanced statements in knowledge graphs., 2026.

[41] Hegde H et al. A change language for ontologies and knowledge graphs., 2025.

[42] Li H et al. Beyond chemical qa: Modular chemical operations., 2025.

[43] Lightman H et al. Let's verify step by step., n.d.

[44] Peng H, Tang J, and Zeng W. Navigating unreliable parametric and contextual knowledge: Explicit knowledge conflict resolution for llm inference. arXiv:2606.20245, 2026.

[45] Quek R W H, Lee S, Leong A W L, Verma A, Prakash A, Chen N F, Low B K H, Rus D, and Solar-Lezama A. Memo: Memory as a model. arXiv:2605.15156, 2026.

[46] Trivedi H, Balasubramanian N, Khot T, et al. Musique: Multihop questions via single-hop question composition. arXiv:2108.00573, 2022.

[47] Bai J et al. Autoschemakg., 2026.

[48] Robinson J et al. Relbench., 2024.

[49] Zhang J et al. Omnimatbench: A human-calibrated multimodal reasoning benchmark across 19 materials science subfields. arXiv:2605.29833, 2026.

[50] Lee K et al. Semantic integrity constraints., 2025.

[51] Sah C K, Zhang L, and Lian X. Beyond self-knowledge: Propagating uncertainty across reasoning and retrieval in llms. arXiv:2607.25600, 2026.

[52] Tatsuoka K K. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354, 1983.

[53] Liu et al. Evidence-centered benchmark designecbd. ECBD, 2024.

[54] Bean A M, Kearns R O, Romanou A, et al. Measuring what matters: Construct validity in large language model benchmarks. arXiv:2511.04703, 2025.

[55] Khodadad M et al. Evaluating multi-hop reasoning in llms: A chemistry-centric benchmark., 2026.

[56] Labre M. Ontology-guided neuro-symbolic inference: Grounding language models with mathematical domain knowledge. arXiv:2602.17826, 2026.

[57] Yu M et al. Egt-kg: Evidence-grounded typed kg retrieval for practical scientific qa., 2026.

[58] Kirichenko P et al. Abstentionbench: Reasoning llms fail on unanswerable questions., 2025.

[59] Salovskii P and Gorshkova I. Automatic ontology construction using llms as an external layer of memory, verification, and planning for hybrid intelligent systems. arXiv:2604.20795, 2026.

[60] Xu R, Qi S, Guo Z, et al. Knowledge conflicts for llms: A survey. In *EMNLP*, 2024.

[61] Zhou R et al. Rulearena: A benchmark for rule-guided reasoning with llms in real-world scenarios. arXiv:2412.08972, 2024.

[62] Auer S et al. The sciqa benchmark for scholarly knowledge., 2023.

[63] Pan S et al. Unifying llms and knowledge graphs: A roadmap., 2024.

[64] Yan S, Yang X, Huang Z, Nie E, Ding Z, Li Z, Ma X, Bi J, Kersting K, Pan J Z, Schutze H, Tresp V, and Ma Y. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. arXiv:2508.19828, 2025.

[65] Guo T et al. What can large language models do in chemistry?, 2023.

[66] van Miltenburg E, van der Lee C, and Krahmer E. Preregistering nlp research., 2021.

[67] Chen W, Qi G, Li W, Li Y, Xia D, and Huang J. Decide then retrieve: A training-free framework with uncertainty-guided triggering and dual-path retrieval. arXiv:2601.03908, 2026.

[68] Xu X et al. Tsbench: A physics-grounded benchmark for evaluating llm understanding of chemical reaction mechanisms. arXiv:2609.08503, 2026.

[69] Yang X et al. Neuro-symbolic artificial intelligence: Towards improving the reasoning abilities of large language models. In *IJCAI*, 2025.

[70] Huang Y et al. Sce: Semantic consistency enhanced rl for multi-hop kg reasoning., 2025.

[71] Kim Y, Abdulle Y, and Wu H. Biohopr. arXiv:2505.22240, 2025.

[72] Liu Y, Liu X, Sun Z, and Hu W. Answering the unanswerable is to err knowingly: Analyzing and mitigating abstention failures in large reasoning models. In *AAAI*, 2026.

[73] Chen Z, Chen S, Ning Y, et al. Scienceagentbench., 2025.

[74] Jin Z, Liu J, Lyu Z, et al. Cladder: Assessing causal reasoning in language models. arXiv:2312.04350, 2023.

[75] Wei Z, Xiang S, Zhang Y, and Jiang C. Graphinstruct: A progressive benchmark for diagnosing capability gaps in llm graph generation. tier gap 0.219, 2026.

[76] Yao Z, Qi W, Pan L, Cao S, Hu L, Liu W, Hou L, and Li J. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. ACL 2025, 2025.

[77] Yuan Z et al. Gracore. arXiv:2407.02936, 2025.

[78] Zhao Z et al. Superchem., 2025.

Appendix A. Formal details (the formal load of Sections 3.1.1–3.1.2)

The main text of Appendix A.5.1 states only "what it is and why it is necessary"; this appendix collects its formal load: notation, the complete tables of relations and operators, the definitions of mechanism and rule, the five structural axioms, the three-valued evaluation equation, the operator-tree grammar, and the item-type mapping. Numbering in the main text and the appendix agrees, so equations can be compared directly.

Number	Content	Use for the main text
A.1	seven-tuple notation of the ontology	source of the symbols in Appendix A.5.2
A.2	signatures, semantics and direction of six relations	decides which edges an operator may read
A.3	signatures, output types and channels of six operators	decides task types and the feasible domain of attribution
A.4	mechanism five-tuple and rule instantiation	$r = \text{ground}(\cdot)$ in Appendix A.5.3
A.5	five structural axioms (premise / conclusion)	the inference rules of Appendix A.5.2
A.6	blocking and three-valued evaluation	the source of `unknown` in Appendix A.5.3
A.7	operator-tree grammar	gold labels and dependence in Appendix A.6.2
A.8	operator–item-type mapping (14 types; 6 rows shown)	the task surface of Appendix A.6.1
A.9	review protocol and engineering trade-offs	checkability in Appendix A.5.4
A.10	executability, value discretisation, formation procedure	implementation basis of Sections 3.1.2 / 3.1.4

A.I Notation and the two main tables

A.1 Notation for the ontology. The ontology is a seven-tuple:

$$\mathcal{O}_M = (V,\ \mathcal{P},\ E,\ \delta,\ \perp,\ \Gamma,\ \Lambda),$$

where V is the set of mechanism concepts, $\delta : V \to \mathbb{N}$ specificity, $\perp \subseteq V \times V$ the disjointness pairs, Γ the condition space, $\Lambda : V \to 2^M$ maps a mechanism to the indicators its effects influence; $\mathcal{P}$ is the set of relation types (each carrying a signature) and $E = \{E_r\}_{r\in\mathcal{P}}$ the sets of relation instances per type.

A.2 The ontology stores six relation types, each with signature, semantics and direction (hierarchy edges uniformly point from child to parent; forward and backward traversal are defined at query time):

Relation	Signature	Semantics
composition `composed_of`	$V \times V$	sub-mechanisms compose the whole mechanism
blocking `inhibits`	$V \times V$	if the blocker is present the effect does not hold
competition `competes_with`	$V \times V$ (symmetric)	the two are interchangeable but do not exclude coexistence
superordinate `subclass_of`	$V \times V$ (transitive)	$m' \sqsubseteq m \Rightarrow \delta(m') > \delta(m)$
precondition `requires`	$V \times \Gamma$	condition dimension a mechanism needs in order to hold
enabling `enables`	$V \times M$	which indicator the mechanism influences

A.3 Evaluation performs six operators, each with input type, output type, evaluation rule, and the edges it reads:

Operator	Signature	Edges read	What it can ask	Channel
Q_{comp}	$V \to 2^V$	`composed_of`	what components this mechanism has	`relation`
Q_{block}	$V \to 2^V$	`inhibits`	what can block it	`relation`
Q_{disj}	$V \times V \to \{0, 1\}$	disjointness pairs	whether the two cannot both hold	`relation`
Q_{spec}	$2^V \to V$	`subclass_of` + δ	which of them is most specific	`hierarchy`
Q_{rule}	$V \times \Gamma \to \{0, 1, \text{unknown}\}$	`requires` + rule layer	under what conditions it holds	`rule`
Q_{enable}	$V \to 2^M$	`enables` + Λ	which indicator it influences	`relation`

The division of labour between A.2 and A.3 matches Sections 3.1.2 and 3.2.2: relations are defined in A.2, operators and dependence in A.3 — the channel is decided by the edge type

an operator actually reads, so an operator that reads no edges (Q_{spec}) cannot have its failure charged to "a missing edge." Two details: the input of Q_{spec} is a candidate set rather than two concepts, and its output must be unique — when several maxima tie it returns the set or marks ambiguity rather than picking one; Q_{comp} returns a set, Q_{disj} a boolean, Q_{rule} a three-valued result, and Q_{enable} an indicator set.

A.II Mechanisms, rules and evaluation

A.4 Mechanisms and rules. Each mechanism is given by a five-tuple

$$m = \langle \text{pre}(m), \text{eff}(m), \text{comp}(m), \text{blk}(m), \delta(m) \rangle,$$

where pre $\subseteq \Gamma$ is the applicable range, eff the effects, comp the components, blk the blockers. A rule is a mechanism instantiated on an object and condition: $r = \text{ground}(m, e, \kappa)$, i.e. $r = (e, \kappa) \vdash \varphi$, where $e \in \mathcal{E}$ is the object, $\kappa \subseteq \Gamma$ the condition signature, and $\varphi \in \text{eff}(m)$ the effect.

A.5 Structural axioms $\mathcal{A}$ (five), listed as "premise / conclusion" (denoted $\mathcal{A}$ in Appendix A.5.1):

$$\frac{m_1 \perp m_2 \quad m_1' \sqsubseteq m_1 \quad m_2' \sqsubseteq m_2}{m_1' \perp m_2'}, \quad \frac{m_1 \sqsubseteq m_2 \quad m_2 \sqsubseteq m_3}{m_1 \sqsubseteq m_3}, \quad \frac{b \in \text{blk}(m)}{b \triangleright m}, \quad \frac{b \triangleright m}{\text{eff}(m) = \varnothing}, \quad \frac{c \in \text{pre}(m) \quad c \notin \text{dom}\,\kappa}{\text{Valid}(m, e, \kappa) = \text{false}}.$$

In order: disjointness propagates down the hierarchy; transitive closure of the superordinate relation; a blocker yields a blocking action; after blocking the effect does not hold; when a precondition is missing the mechanism is unusable for that object. The third and fourth together explain how "being blocked" enters evaluation — first the blocker yields an action, then the action erases the effect, so no separate rule needs to state it. The applicability condition of the first is in Appendix A.5.2 of the main text (the distinction between concept disjointness and conditional disjointness).

A.6 Blocking and three-valued evaluation.

$$B(m, e, \kappa) = \text{MatchBlockRules}(m, e, \kappa), \qquad \text{Eval}(m, e, \kappa) = \begin{cases} \text{false} & B = \text{true} \quad \text{(blocking)} \\ \text{true} & R^{+} = \text{true} \quad \text{(positive)} \\ \text{unknown} & \text{otherwise} \end{cases}$$

A.III Grammar and item-type mapping

A.7 Operator-tree grammar. A query is $q = \langle \text{op}, e, \kappa_q, \pi \rangle$; its logical form is given by the following grammar, whose leaves must land on rule instances:

$$T ::= \text{atom}(e, \kappa) \mid \text{comp}(T_1, \ldots, T_k) \mid \text{block}(T) \mid \text{disj}(T_1, T_2) \mid \text{spec}(T) \mid \text{inv}(T, \kappa'),$$
$$y^{*} = \text{Eval}_{\mathcal{O},\, \mathcal{R},\, \Gamma,\, \Lambda}(T).$$

comp expands composition, block takes blockers in reverse, disj decides disjointness, spec takes the most specific concept, and inv changes one dimension and recomputes.

A.8 Operator–item-type mapping (14 task types; the first four are shown as examples; the full table is frozen with the data card in the resource bundle):

Operator	Item type	Input	Output type	Dependence
Q_{comp}	single-hop composition	mechanism	concept set	composition edge
Q_{comp}	multi-hop composition	mechanism	concept set	composition edge (cross-level)
Q_{spec}	most specific concept	candidate set	concept	hierarchy edges + δ
Q_{spec}	hierarchy ordering	candidate set	concept sequence	hierarchy edges + δ
Q_{disj}	disjointness judgement	two mechanisms	boolean	disjointness pairs + axioms
Q_{rule}	counterfactual condition	mechanism + condition	three-valued	rules + condition dimension

A.IV Engineering details and procedure

A.9 Engineering details (omitted from the main text).

Reporting list of the review protocol. Four classes of objects are reported separately: concepts, hierarchical relations, relation instances, disjointness pairs — for each, (1) the number of items under double independent review, (2) the agreement statistic and its unit, (3) the arbitration rule, and (4) the numbers finally kept and rejected; where full double review is impossible, what was reviewed in full and what was reviewed on the cohort is marked explicitly (statistics in Section 4.1). Reasons for rejecting a candidate are archived in four classes: "non-standard naming / non-unique superordinate / inconsistent with evidence / overlapping an existing concept."

The trade-off on change languages. Writing ontology changes as an executable change language rather than a prose description has precedent (41); this paper takes its "changes are traceable" side (change types registered, version numbers recomputable) but does not introduce a change language itself — the checkability of this paper rests on the two things "stem traceability" and "version traceability".

An example of the context guard. The same stem may denote another mechanism in a different context: `cross-link` in an antibacterial context means cell-wall crosslinking, not material covalent crosslinking; such hits are excluded by the context guard.

A.10 Engineering details and procedure (omitted from the main text).

A.10.1 The case for executable rather than retrievable. A concept hierarchy only says who is superordinate to whom and gives no enumerability, whereas "which primitives an item uses" must be enumerable — otherwise truth values cannot be recomputed and failure cannot be localised to an edge. The same algebra is therefore shared by four parties: item generation uses it to pick points; before admission it recomputes truth values; at response time the model is asked to return an operator tree of the same form; and scoring compares that tree bit by bit. Sharing one structure across four parties has three consequences: the query type an item requires corresponds one-to-one with the principal operator ("what is asked" is "which operation must be performed"); the well-formedness conditions of an answer are only two (disjoint pairs must not both hold; an abstract root may not serve as a leaf); and failure can be localised to which operation, hence which edge, went wrong (the source of edge-level attribution granularity, Appendix A.6.2). The difference from "retrieval-style" use: existing work treats the ontology as retrieval material spliced into a prompt, with relations recorded as static equations and rules never consumed by inference (56); this paper requires relations to be evaluable and rules to be consumable, and fixes this with a checkable instrument (recomputation and gating in Appendix A.6.6).

A.10.2 The case for discrete values. Adding a condition dimension means taking one value from the condition space (e.g. `additive=edta`); if values were free, the same intention could be written as "add a chelating agent," "the system contains EDTA," and so on, so two paraphrases need not land on the same coordinate.

A.10.3 Layering of effect records. The ontology layer records the possible effect types of a mechanism; the rule layer records the effects supported by evidence for a given object and condition; and the result under a specific condition is the output of one evaluation, not a third storage layer. The same sentence "the mechanism holds" means different things in the two places — conflating them stores the same effect twice and leaves `unknown` no corresponding storage position.

A.10.4 The three-step procedure of ontology formation. Candidate extraction: mechanism expressions are extracted from literature blocks as candidates; synonymous writings are first merged by terminology normalisation (abbreviations, singular/plural, Greek-letter variants), then compared against existing concepts, and only expressions that cannot be subsumed become new candidates. Extraction is driven by stem matching, so every candidate carries its evidence stem (the word or root used when the concept appears in the source, e.g. `self-heal`, `boronate ester`). Item-by-item review: humans judge "is the naming standard, is the superordinate unique, is the definition consistent with the evidence, is it

synonymous with or overlapping an existing concept"; those passing enter the concept set with their stem recorded, and those failing are entered in a rejection table so they do not reappear in the next round; relations and axioms are judged item by item as well — the five classes composition, blocking, competition, precondition and enabling are checked for legality by the types of their two ends, while disjointness pairs and specificity lower bounds are judged by experts. Release and traceability: a release writes a version number; afterwards, reparenting, tightening or deleting stems, and adding concepts all produce a new version with a registered change type. The ontology records the version at extraction time and migration markers; concept-level provenance is not recorded per entry. Within the papers covered by the block table, traceability proceeds by the evidence-stem mechanism.

A.V Mechanism ontology construction in full (formal details of Section 3.1)

A.V.1 Multi-source knowledge acquisition and scope

Ask first where the knowledge comes from. The raw material of the ontology is not an off-the-shelf corpus but three qualitatively different kinds of knowledge:

$$D = \{D_{\text{lit}},\ D_{\text{exp}},\ D_{\text{res}}\},$$

where D_{lit} is literature mechanism knowledge — mechanism descriptions, material action pathways, and condition constraints read out of the literature, supporting a chain of the form "material → trigger → mechanism → effect → indicator" (`sodium_alginate` triggered by Ca^{2+} into ionic crosslinking and thereby raising gel strength is one such chain); D_{exp} is expert knowledge — concept boundaries, hierarchy, and disjointness relations, which answer "which inferences are legitimate"; D_{res} is resource knowledge — material properties, indicator systems, and condition spaces, which supply the values needed for instantiating the evaluation. This equation is the definition of the ontology's scope: nothing outside the three sources enters the ontology.

The three sources are neither parallel nor in one-to-one correspondence with the three layers — literature supplies facts, experts supply constraints, resources supply values; the three interleave as they land and jointly erect the backbone. In the figure the sources are the three cells of the top row, whose lines converge into the three layers in the middle row; this is exactly what "the top row is not three datasets but three kinds of contribution" means.

A.V.2 Executable mechanism ontology modeling

The ontology must be executable. Organising concepts into a hierarchy alone is insufficient: a hierarchy gives only subsumption, and cannot determine which primitives an item uses, so truth values cannot be recomputed and failure cannot be localised to an edge. The precondition for executability is that the ontology first declares which queries it supports:

$$\text{Ask}(\mathcal{O}_M) = \{\, q \ :\ \text{op}(q) \in \text{Op},\ \text{ar}\big(\text{op}(q)\big) \text{ compatible with } V, E \,\}.$$

Here q is a query, $\text{op}(q)$ the query operator it uses (six operator families in Appendix A.3), $\text{ar}(\cdot)$ the parameter type of an operator, V the concept set and E the relation-instance set. This equation defines the set of askable queries: a query is well formed iff its operator's parameter type is compatible with relations that actually exist in the ontology. "Type compatible" means signature agreement: the signature of Q_{comp} is $V \to 2^V$, and what it reads is precisely a $V \times V$ `composed_of` edge; the signature of Q_{rule} is $V \times \Gamma \to \{0, 1, \text{unknown}\}$, and what it reads is a $V \times \Gamma$ `requires` edge — only if an instance of that type genuinely exists in the ontology is the query well formed (signatures of the six relation types: Appendix A.2; signatures of the six operators: Appendix A.3). From this, given an operator one can decide whether it is askable on the present ontology, and the boundary of askable queries is thereby fixed.

Take ionic crosslinking. It has one composition edge, one superordinate concept, and one trigger condition in the ontology, so the three questions "what is it composed of," "which

class does it belong to," and "is Ca^{2+} inside the applicable range" are all well formed; if some relation is missing, the corresponding question does not close and no item is generated — this is the exact content of "the query type is given by structure, not by the wording of the item."

Relations and operations are two different kinds of object: an edge is a fact in the ontology, extracting the components is an operation performed at evaluation time; the former is a relation instance, the latter an operator. The role of the axioms is to replace retrieval by derivation — a conclusion derivable from existing relations need not be looked up again in the corpus (the five axioms are in Appendix A.5). The first of them splits into two kinds of disjointness: concept disjointness (not co-satisfiable by definition) propagates down the hierarchy, whereas conditional disjointness holds only for a given object and condition and does not propagate, so any use of it must carry the condition signature κ in its premise. Competition and disjointness are two different relations: a competition edge only says "interchangeable" and does not exclude coexistence, so disjoint pairs are declared explicitly by experts, pair by pair.

There is only one reason for executable rather than retrievable: recomputability (Appendix A.10.1).

A.V.3 Rule-grounded reasoning and value space

A mechanism does not stop at the ontology layer. Which material and which conditions a mechanism holds on is something the ontology cannot answer — it must land on concrete objects and conditions, a step called instantiation:

$$r = \text{ground}(m, e, \kappa), \qquad \text{i.e.} \quad r = (e,\ \kappa) \vdash \varphi, \qquad e \in \mathcal{E}, \quad \kappa \subseteq \Gamma, \quad \varphi \in \text{eff}(m).$$

Here ground is the instantiation operator, m the mechanism instantiated, e drawn from the object space $\mathcal{E}$, κ from the condition space Γ, and φ one effect of the mechanism (from $\text{eff}(m)$). The equation gives the rule instance of a mechanism under a given object and condition; it turns the mechanism into a directly decidable fact — the set of all rules r is the rule layer $\mathcal{R}$.

Two such instances appear on one item as two leaves of the operator tree (Appendix A.6.2): (`sodium_alginate`, `Ca`$^{2+}$) $\vdash$ ionic crosslinking and (`sodium_alginate`, `Ca`$^{2+}$) $\vdash$ gel strength increase. Every rule in the rule layer must resolve back to concepts of the ontology (the "effect $\rightarrow$ mechanism" mapping); otherwise truth values cannot be updated when the ontology changes — `inv`-type items (change one dimension and recompute) rest exactly on this.

Blocking follows the same route. It cannot be decided from "some condition dimension occurred" — "there is a reducing agent in the system" does not license concluding that all oxidative crosslinking fails — so blocking is evaluated by conditioned rules (`MatchBlockRules`, Appendix A.6).

Evaluation is three-valued. Positive evidence gives `true`, negative evidence gives `false`, everything else gives `unknown` (full form in Appendix A.6). Why it must be so: a literature-derived knowledge base is inevitably incomplete, and "no blocking recorded" is not "no blocking"; collapsing to two values would turn incompleteness of the knowledge base directly into gold-label error. This paper's handling is that queries judged `unknown` do not become items (only Eval = true is used), so three-valued evaluation does not change the gold labels of existing items. Two wrong inferences must be excluded at evaluation: `false` must not be inferred from "the graph does not contain this edge"; `unknown` must not be treated as `false`. (`unknown` and the `NEG` of Appendix A.7.2 belong to two different levels: the former is insufficient evidence on the evaluation side, the latter is the model answering wrongly after being given the knowledge.)

Beyond objects and conditions, indicators must also be fixed. Conditions are not free text but controlled values, and having a value is not the same as the value being usable — each dimension carries a constraint function:

$$C_m : \Gamma \to \{0, 1, \text{unknown}\}, \qquad \text{pre}(m) \subseteq \Gamma,$$

where Γ is the condition space (one set of controlled values per condition dimension) and $\text{pre}(m)$ the applicable range of mechanism m. The equation decides whether a single condition dimension is usable, separating four cases: not supplied (supply one more dimension and ask again; the basis of `inv` items) / constraint violated (this is the `false` that has negative evidence) / satisfied / insufficient evidence. For example, `ph` required to lie in $[6, 8]$ while the stem gives 10, so constraint violated; whereas "not supplied" should not be judged wrong but re-asked with one dimension added. Where the answer lands is given by the mapping $\Lambda : V \to 2^M$ (mechanism → effect → indicator), which restricts candidates to the indicators reachable by that mechanism's effects.

Putting the three layers together, an item is an operator tree (grammar in Appendix A.7): $q = \langle \text{op}, e, \kappa_q, \pi \rangle$, whose leaves must land on rule instances. The evaluator reads four things — the ontology, the rule layer, the condition space, and the indicator mapping — and keeps the output typed, not collapsing sets, booleans, concepts and three-valued results into one y^* for comparison (return types per operator in Appendix A.3). The gold label is therefore the execution result of this tree on the structure, not a written-down string. Two further details are not developed in the main text: "the ontology does not produce truth values" (Appendix A.6.6) and "why values must be discrete" (Appendix A.10.2).

A.V.4 Ontology validation and consistency checking

The structure must be externally checkable. Once the ontology is versioned it cannot rest on the authors' saying "it is right" — it must allow others to go back to the sources and verify. Checkability therefore rests on two things: stem traceability and version traceability. The form of traceability is

$$\text{Lookup}(c) = \{\, b \;:\; \text{stem}(c) \in b \,\},$$

where c is a concept of the ontology, $\text{stem}(c)$ its evidence stem (the word or root used when the concept appears in the source text), and b a corpus block. The equation gives the set of evidence blocks in which the concept occurs; whether a concept has support therefore becomes a re-checkable retrieval.

For example, `cross-link` in an antibacterial context means cell-wall crosslinking and is not material covalent crosslinking — such hits are excluded by a context guard; on the version side, each release is given a version number and each change a registered type (e.g. `reparent`), so "what changed" is traceable (whether changes should be written as an executable change language: the trade-off is in Appendix A.9).

Figure 1 draws "corpus → ontology → rule layer, with expert review as the gate in the middle," and where this gate really bites is at the relation level: an edge is not merely a fact in a graph, it is at the same time a knowledge-dependence unit of the benchmark (the D^+ of Appendix A.6.2) — if that edge is itself wrong, a model's wrong answer may be not the model's problem but the gold label's. Every edge entering an item's dependence chain must be evidence-reviewed on its own, not merely checked for concept stems.

How candidates are extracted, reviewed item by item, and released is procedure, not developed in the main text (Appendix A.10.4); the reporting list for the review protocol (number of reviewers, independence, how disagreements are arbitrated, numbers and reasons for rejection) is in Appendix A.9, with statistics in Section 4.1. At this point the knowledge has become a versioned structure outside the model — the first half of the third level of the chain is thereby satisfied (Section 1.4, claim three).

A.VI Item generation in full (formal details of Section 3.2)

A.VI.1 Query space construction

There must first be a pool of "askable" queries. The item surface is not written out of nothing; it is rendered from operators, and which item types one operator can render is fixed in advance:

$$\mathrm{Typ}:\ \mathrm{Op}\ \longrightarrow\ 2^{\mathcal{T}},\qquad \mathcal{T}=\bigcup_{\mathrm{op}\in\mathrm{Op}}\mathrm{Typ}(\mathrm{op}).$$

Here Op is the six query operator families (Appendix A.5.2, Appendix A.3), $\mathcal{T}$ the item-type set, and Typ maps each operator to the item types it can render (Typ(op) is a set). The equation states that item types are the image of operators under rendering — the task surface of the bank is thereby fixed: which types an operator can and cannot produce is decided by this mapping alone, so the bank changes automatically as the ontology is revised, rather than being an independently designed list. The current implementation registers 14 task types; the full mapping is frozen with the data card (Appendix A.8).

For `mech_compose`, its image is the two types "single-hop composition item" and "multi-hop composition item"; the operator tree in the middle of Figure 2 is one image of `mech_compose` — asking what components the same mechanism has, differing only in hop count, lands as two item types.

A.VI.2 Rule-based oracle: how the answer is determined

The answer is not written by a person. Every question form corresponds to an operator tree T (grammar in Appendix A.7) whose leaves must land on rule instances; the truth value is the execution result of this tree on the structure:

$$y^{*}=\mathrm{Eval}_{\mathcal{O},\ \mathcal{R},\ \Gamma,\ \Lambda}\big(\mathrm{parse}(q)\big).$$

Here $\mathrm{parse}(q)$ recovers the tree from the item surface and Eval is the evaluator of Appendix A.5.3, the four subscripts naming the four inputs it reads — ontology $\mathcal{O}$, rule layer $\mathcal{R}$, condition space Γ, and indicator mapping Λ. The equation defines the gold label as the result of executing this operator tree; the answer is therefore recomputable: repeated execution of the same tree is identical bit by bit, and changing one edge or one rule recomputes the gold labels of all affected items, with no re-annotation. This differs in kind from "answers annotated by hand" — annotation is a one-off external input, execution a repeatable internal result (the `Ground Truth` station at the far right of Figure 2 is its product).

But an answer alone is not enough; it must also be recorded on what basis. This is this paper's addition beyond the usual triple $\langle T,\sigma,y^{*}\rangle$:

$$\mathrm{Dep}(q)=\big(D^{+},\ D^{-},\ D^{\mathrm{rule}},\ D^{\mathrm{cond}},\ D^{\mathrm{evidence}}\big).$$

The five components are: D^{+}, positive — the concrete instances supporting the current answer (that `composed_of` edge); D^{-}, completeness — the "no other instance exists" on which the answer depends ("this mechanism has no other composition edge"); D^{rule} — the rules or axioms consumed at evaluation; D^{cond} — the condition dimensions used; D^{evidence} — the evidence blocks supporting those instances. This equation is the item's knowledge warrant; attribution can therefore descend from mechanism level to edge level — without it, compensability could only be computed at mechanism level; with it, it can be computed at edge level. The two bold edges at the third station of Figure 2 are one item's D^{+} and D^{cond}.

Why D^{-} must be recorded separately. Reading a certain edge type is not the same as depending on a particular instance: if only the edges actually hit are recorded, deleting an edge can be detected, whereas adding a previously omitted composition edge leaves the

existing dependence list unable to notice that the item's gold label has changed; D^- records the completeness assumption "there is no other composition edge" as a dependence that can itself be invalidated. Why it must be recorded: if one aggregates only by mechanism without recording edges, failure cannot descend to edge level.

Dependence also fixes an item's channel: an operator reading edge types is recorded as `relation`; one reading only superordinate edges as `hierarchy` (all information the answer needs is in the taxonomy, so its failure cannot come from "a missing edge"); one reading condition dimensions of the rule layer as `rule` — this channel table is the feasible domain of attribution (the distribution is not uniform; Section 4.6).

A.VI.3 Instance grounding: on what object it acts

Of the four, what most needs "grounding" is the item surface. The operator tree and the gold label are still abstract; the surface must say "of whom, under what condition, evaluating what." The slots of the surface take only four things — truth value and components do not enter slots:

$$\sigma = \mathrm{T}_\theta\big[e,\ b,\ \pi,\ \mathrm{ask}(\theta)\big].$$

Here T_θ is the surface template (θ its parameters), e the entity, b the anchor, π the candidate set, and $\mathrm{ask}(\theta)$ the response format. The equation states that the surface is rendered from four slots; "what is asked" and "how it is asked" are thereby separated — of the four, only the candidate set belongs to generation strategy; the other three come directly from the three layers of Section 3.1: the entity is drawn from the object space $\mathcal{E}$ (materials and reagents, deciding "what this item is about"), the condition from the condition space Γ and must pass the constraint function (if not supplied, add a dimension and ask again; if violated, that constitutes negative evidence), the indicator fixed by Λ (candidates only within the indicators reachable by that mechanism's effects).

For the ionic-crosslinking item: the entity is `sodium_alginate`, the anchor its superordinate concept, the candidate set same-level concepts, the response format "list the components" — this is the exact content of "the surface is given by structure, not by the item author's wording," and also the process by which the empty slots at the fourth station of Figure 2 are filled.

A.VI.4 Candidate generation and difficulty control

Distractors are not an independent knowledge source; they are a wrong-candidate space generated from the ontology structure:

$$\pi \ \subseteq\ \mathrm{Neigh}(m) = \{\, m' \in V \ :\ m' \text{ is at the same level as } m \text{ or competes with it} \,\}, \mathbf{d}(q) = \big(\mathrm{depth}(T_q),\ |\kappa_q|\big).$$

Here π is the candidate set in the slot (Appendix A.6.3), $\mathrm{Neigh}(m)$ the structural neighbourhood of mechanism m, T_q the item's operator tree, $\mathrm{depth}(\cdot)$ its hop count, and $|\kappa_q|$ the number of condition dimensions the surface involves. The two equations give, respectively, where distractors are drawn from and how difficulty is composed; both are therefore recomputable — distractor strength is decided by the taxonomy, not by the item author's wording; difficulty lies entirely on the structure side, and therefore shares one unit system with attribution (Section 3.3).

For ionic crosslinking: the strongest distractors are same-level concepts (e.g. `oxidative crosslinking` — also a kind of crosslinking, different pathway), then competing mechanisms and instances with close conditions (neighbouring values of the same condition dimension). Difficulty is controlled at three places: hop count (single → multi), condition load (number of condition dimensions and whether a dimension must be supplied before computing), and knowledge visibility (next subsection).

A.VI.5 Controlled knowledge visibility

At this point the item itself exists; what remains is to control "how much knowledge it sees." Knowledge being externalised, it is no longer a background condition but a controllable independent variable — the same items run once at each of four tiers, with surface, model and response format fixed apart from visibility:

$$\kappa \in \{\texttt{blind}, \texttt{full}, \texttt{flat}, \texttt{shuffled}\}, \qquad \text{View}(\kappa) \text{ is produced by the same rendering function.}$$

Here κ is the knowledge visibility tier — withheld, full fragment, concept names only (no structure), structure in the wrong order — and $\text{View}(\kappa)$ the knowledge view it renders. The equation gives the four knowledge states of one item; "is the knowledge sufficient" and "is the knowledge correct" are thereby separated — `blind` $\leftrightarrow$ `the other three` measures whether domain knowledge is visible, `flat` $\leftrightarrow$ `full` measures whether structure itself helps, while whether the answer is directly exposed is a third factor running through all four tiers and must be checked separately (`Leak`).

Answer leakage is checked item by item. The view is cut out around the target operator; if asking "which superordinate class does this mechanism belong to" lists the whole ancestor chain, the model need only copy, so the item does not measure reasoning. Therefore, for every item and every tier,

$$\text{Leak}(q, \kappa) = \begin{cases} 1 & y^*(q) \text{ readable directly from the view} \\ 0 & \text{otherwise} \end{cases}$$

Items used for reasoning analysis require $\text{Leak}(q, \texttt{full}) = 0$; items whose leakage cannot be avoided are not deleted but flagged separately as parametric-knowledge probes, interpreted only in the `blind` column (Appendix A.7.2).

The construction of `flat` and `shuffled` must be specified, otherwise "destroying structure" also introduces other differences: both tiers keep the same concepts, the same number of facts, and the same text budget as `full`, changing only structure; `shuffled` uses a degree-preserving perturbation (reassigning parents among siblings at the same depth) and checks that type constraints still hold; each tier is generated in several perturbed versions with fixed random seeds.

The last two tiers are where this paper separates from "prompting" approaches: `flat` tests "does structure help," `shuffled` tests "is the structure right." The difference from the closest precedent is that it supplies tools (formulas, retrieval, code) whereas this paper supplies the knowledge itself; it adds assistance after the fact, whereas this paper requires a single variable. Only the latter can attribute a difference to "this one piece of knowledge"; the former can only attribute it to "this class of assistance." (The conditions under which this design can be falsified: Section 4.3.)

A.VI.6 Benchmark quality control

The last gate: which items may enter the bank. Four gates must be passed simultaneously:

$$\text{Pass}(q) = \text{G}_1(q) \ \wedge \ \text{G}_5(q) \ \wedge \ \text{G}_6(q) \ \wedge \ \text{G}_3(q).$$

where G_1, dependence gate — every item must record $\text{Dep}(q)$ (Appendix A.6.2); G_5, anchoring gate — the view must reach the level the operator needs and the traversal direction must be right (upward for `argmax_depth`-type, downward for `mech_refine`-type; the wrong direction is equivalent to not supplying it); G_6, format gate — sequence and composite items must use generative response; G_3, quota gate — per-cell item counts are met. The equation gives the four necessary conditions of acceptance; all four gates must be passed, and risk is thereby kept out of the bank — without dependence recorded, failure cannot descend to edge level; with the wrong direction, a good item is misjudged as "very hard," and such errors turn directly into false attribution.

One boundary stated retroactively: the ontology does not produce truth values. Rules land in the rule layer; the ontology only fixes what can be asked, at what specificity the answer lands, and which distractors are drawn (Appendix A.5.3); an item that cannot reach the rule layer cannot be answered at all. A second boundary is that coverage has a ceiling: item sources for types involving relational predicates and competition routes are limited, so the quota table sets only lower bounds for such cells (channel distribution in Appendix A.6.2).

One more deserves its own name: `invalidate()` — invalidated means regenerated. Because its error is invisible in the results: after new edges are added to the ontology, an old item's annotation still reflects the state before completion while the view is already rendered from the new ontology — the numbers are right, the surface reads well, scoring raises no error, only that item is no longer the item it was (instance and readings in Section 4.5). Items whose dependence is invalidated or whose truth value is stale must be regenerated, rather than quietly continuing to be used with a swapped truth value — this is also the only back edge of this chapter: the long dashed line from "diagnosis" back to "generation" in Figure B.1.

The last boundary is answerability. Existing benchmarks look at abstention from the output side — reasoning models still answer unanswerable questions (58), and follow-up work characterises this "answering despite knowing it is unanswerable" and attempts mitigation (72); this paper places it on the item side: whether an item can be answered is decided by the ontology and the evidence, not inferred backwards from answering behaviour (Appendix A.6.2).

A.VII Diagnostic evaluation in full (formal details of Section 3.3)

A.VII.1 Controlled evaluation protocol and statistical units

After running the two tiers, the first question is "which number to report." Items are not all alike: some have the answer written into the view, others only supply material. Split by "whether the answer is directly given by the view," each type reports a different quantity:

$$\text{report}(q) = \begin{cases} a_{\mathcal{M}}(q,0) & \text{parametric probe: the answer is listed directly in the view} \\ R_{\text{rescue}}\big(u(q)\big) & \text{reasoning probe: only materials given, the answer must be computed} \end{cases}$$

where $a_{\mathcal{M}}(q,0)$ is the item's correctness in the knowledge-masked tier and R_{rescue} the compensability of the knowledge unit $u(q)$ it belongs to (defined in the next subsection). The equation requires the two item types to be reported separately; this prevents a single column of readings from being misread — a high score on probe items cannot be read as "the model is capable"; it only says retrieval was accurate; its value lies precisely in the knowledge-masked column, i.e. how much domain mechanism knowledge the model's parameters actually hold. The compensability of the two types differs sharply, showing that those failures are not due to missing knowledge (readings in Section 4.4).

The statistical unit must match the dependence structure of the data. Items are not independent draws: several items depend on the same edge and several edges come from the same paper. Tests are therefore stratified as item $\subset$ knowledge unit $\subset$ evidence source — two-tier comparison uses paired results of the same item (McNemar); at unit level, a bootstrap clustered by knowledge unit or by paper, or a mixed-effects model with random effects. Treating all items as independent shrinks standard errors and inflates significance (the test family is fixed in Appendix A.7.3).

A.VII.2 Knowledge rescue and failure attribution

The two results at item level fall into four classes first, by the correctness combination of the two tiers:

$$\mathsf{cls}_{\mathcal{M}}(q) = \begin{cases} \texttt{RESCUE} & a(0)=0,\ a(1)=1 & \text{wrong to right once the view is supplied} \\ \texttt{PERSIST} & a(0)=0,\ a(1)=0 & \text{still wrong once the view is supplied} \\ \texttt{NEG} & a(0)=1,\ a(1)=0 & \text{right to wrong once the view is supplied} \\ \texttt{OK} & a(0)=1,\ a(1)=1 & \end{cases}$$

The four classes are rescued, still failing, turned wrong, and right at both ends. The equation gives the behavioural classification of the two-tier results; "answering wrong" is thereby split into two actionable sides — only `RESCUE` and `PERSIST` are failures, and they correspond respectively to the two courses of "supply knowledge" and "change the model." The four classes are behavioural, not internal causes. `RESCUE` only says "it answered right once the view was given"; it does not mean "the knowledge is absent from the parameters" — a wrong knowledge-masked answer may also come from not knowing how to invoke it, from misreading the question, or from random fluctuation; reading behaviour as cause requires separate probes (the two columns of the previous subsection).

The `NEG` cell has a name in the literature: context-memory conflict (conflict between contextual knowledge and parametric memory) (60); it can stand as a cell of its own because visibility was made an independent variable — without controlling visibility this cell is not observable at all.

After classifying item by item, aggregate by knowledge unit. A unit u may be an operator, an edge family, a mechanism, or a single edge; its compensability is

$$R_{\text{rescue}}(u) = \frac{N_{\texttt{RESCUE}}(u)}{N_{\texttt{RESCUE}}(u) + N_{\texttt{PERSIST}}(u)}$$

(the denominator excludes `NEG` and those right at both tiers — they are not failures and would dilute the ratio). The equation gives the share of that unit's errors that external knowledge can rescue. But it cannot be reported alone — it and "how many errors this unit causes" are two different quantities: one unit with 10 `RESCUE` out of 100 items and another with 10 out of 10 both have ratio 1, yet their error burdens differ tenfold. Hence four statistics are reported together: $R_{\text{rescue}}(u)$ answers "which units' errors are most easily compensated by external knowledge"; $R_{\text{error}}(u) = (\texttt{RESCUE} + \texttt{PERSIST})/N_u$ answers "which units cause the most errors"; $\Delta\text{acc}(u) = N_u^{-1} \sum_{q \in u} [a_1(q) - a_0(q)]$ is net accuracy change (including right→wrong); $\text{Coverage}(u) = N_u$ says how many items the conclusion rests on (with the numbers of independent mechanisms / edges / rules also reported).

Four are reported because a single ratio must not be allowed to say directly "extend the knowledge base or tune the model": 0.5 is not the boundary between the two sides; high R_{rescue} does not imply the knowledge base is deficient, low does not imply the model cannot reason, and with small counts the ratio fluctuates violently. The repair direction is given by evidence in the next subsection, not by threshold mapping.

A.VII.3 Knowledge-guided improvement strategy

Where these readings land is not "who is right" but "what to move next." Whether an intervention works must therefore be seen in its net gain:

$$\Delta_{\text{specific}}(u) = \Delta\text{acc}(\text{target}) - \Delta\text{acc}(\text{control}),$$

where target is the unit diagnosed and control the comparable remaining units. The equation distinguishes "only the target unit improved" from "everything was raised"; it separates a genuinely effective intervention from the spurious gain of "longer prompts help everywhere" — no absolute value is taken: taking absolute values would convert negative and positive control gains alike into penalties and merge effects of opposite direction into one number. Target and control groups must also satisfy similar text length, matched item type and difficulty, and similar pre-intervention accuracy distributions, and the test family for multiple comparisons is fixed in advance.

Whether guidance holds is seen from three criteria together: positive — the target unit improves significantly ($p_{\text{Holm}} < 0.05$); negative — the control unit is not significant (this one cannot be omitted, otherwise "fine-tuning helps everywhere" cannot be excluded); directional — the difference above is significant. Only then is it decided what to move: (1) classify item by item (two-tier results $\to \mathsf{cls}(q)$); (2) aggregate by unit (operator / edge-type / mechanism level $\to$ compensability and the other three statistics); (3) locate the short board (from the combination of R_{rescue} and R_{error}; resolve `NEG` conflicts first); (4) intervene and retest (positive / negative / directional; if not all three hold, the attribution conclusion for that unit is withdrawn).

This is exactly the distinction demanded by the third difficulty of Section 1.1: after a wrong answer, the two courses "extend the knowledge base" and "tune the model" are no longer indistinguishable in the result table (Section 4.7 gives the measured form of this reading table).

One limitation must be stated. One class of failure is different in kind — a model that can explain a concept correctly yet cannot apply it (8). Attribution experiments must therefore separate "cannot apply" from "does not know": knowledge and reasoning should not be claimed to be strictly dichotomous; this paper claims only an operational dichotomy.

Appendix B. Full-chain pipeline and ontology construction (omitted from the main text)

This section describes the formation of the ontology and the full chain in construction order; it does not count as a methodological claim. Methodological claims remain those of Section 3. The opening of Section 3 keeps only the transition from Section 2 to Section 3; the rest is moved here.

Number	Content
B.1	three modules and the recomputable quantity each corresponds to
B.2	the full-chain figure (Figure B.1) and its three segments
B.3	main line of the chapter (construction order)
B.4	correspondence between the three modules and the five links of Section 1.4
B.5	reading order (order of contribution, not of construction)

B.1 Three modules and the recomputable quantity each corresponds to

This chapter has three modules: Section 3.1 ontology construction (writing domain mechanism knowledge as an executable reasoning space) $\to$ Section 3.2 benchmark generation (turning it into evaluation objects with determinate answers, controllable difficulty, and explicit dependence) $\to$ Section 3.3 evaluation and attribution (inferring the boundary of a model's capability from controlled results). Each step corresponds to a recomputable quantity: the size and edge-family distribution of the ontology; the channel distribution of the bank; the four-class membership of the two-tier results; and compensability aggregated by knowledge unit.

B.2 Full-chain figure

B.3 Main line of the chapter (construction order)

Main line. This chapter writes mechanism knowledge as an executable structure so that "which query an item requires" and "what the answer equals" are both computed from that structure; then let that structure be supplied tier by tier. The two-tier results of one item can therefore split failure between the knowledge base and the model.



Figure B.1: The full chain in three segments. The top row (Resources) turns the corpus into controllable resources: corpus blocks → mechanism ontology → rule layer, with expert review as the gate in the middle. The middle row (Items) turns resources into items: generation (recording knowledge dependence) → gating (anchoring and format) → benchmark. The bottom row (Diagnosis) measures and charges failure to concrete knowledge units: run both arms → attribute → aggregate by unit → intervene and retest. The solid and dashed rectangles in the first cell of the bottom row are the two arms; apart from knowledge visibility the item surface, model and response format are fixed throughout. The only back edge is the regeneration check — after an ontology upgrade, items whose dependence is invalidated or whose truth value is stale must be regenerated, not quietly kept with a swapped truth value.

B.4 Correspondence between the three modules and the chain of Section 1.4

The three modules correspond one by one to the chain of Section 1.4 (numbering shared with Sections 1.4 and 2):

Every level of that chain has a corresponding remedy in this chapter: one, the query is specified by the item — Appendix A.5.2 lets the principal operator fix the query type, and Appendix A.6.1 records what each item depends on in Dep(q); two, the answer can be computed — Appendix A.5.3 writes axioms as inference rules and items as operator trees, and keeps values controlled; three, knowledge can leave the model — jointly satisfied by the sources of Appendix A.5.1, the modelling of Appendix A.5.2, the value space of Appendix A.5.3, and the traceability of Appendix A.5.4; four, knowledge can be given tier by tier — the four-tier manipulation of Appendix A.6.5 makes knowledge visibility a single variable; five, failure can be attributed — Appendix A.7.2 aggregates the four-class membership of the two-tier results into compensability, from which a diagnostic direction follows.

B.5 Reading order

Reading order: Section 3.1 builds the reasoning space first, Section 3.2 turns it into evaluation objects, and Section 3.3 finally reads the defect from the results — this is the order of contribution, not the order of construction. The three sections carry, respectively: Section 3.1 (why the knowledge can be executed), Section 3.2 (generation of questions and answers), and Section 3.3 (reading from results which side to repair).

Appendix C. Protocols for the additional experiments, with measured results

> This appendix collects the executable protocols for the six experiments of Section 4.9: cohort, variables, measures, and pre-registered criteria. Protocol fields are listed in C.1–C.5; measured readings are filled back in C.6.

C.1 E-MODEL: cross-model consistency matrix

Item	Setting
Cohort	a fixed cohort from the same distribution as the main bank (200 items suggested) plus the 43-item pilot of Section 4.2
Models	`gpt-4.1` (existing), `gpt-4o`, `deepseek-v4-pro`, `glm-5.3-flash`, `qwen3.8-flash`, `claude-*` ×2
Variable	knowledge visibility, two tiers (`blind` / `full`)
Measures	masked accuracy, supplied accuracy, Δ (compensability), class shares
Statistics	paired McNemar across tiers; across models report only whether the sign agrees, not a ranking
Criterion	$\geq$3/5 models with $\Delta>0$; any model with $\Delta\leq 0$ is reported as such and analysed by task family
Cost	about $200 \times 2 \times 8 \approx 3200$ calls

One known pitfall: task families must be corrected by the majority baseline (end of Section 4.2); otherwise families such as `cond_hard` are inflated by label skew into "saturated" readings.

C.2 E-HUMAN and E-AGREE: human baseline and gold-label agreement

Item	E-HUMAN	E-AGREE
Cohort	100 items at the supplied tier (stratified by channel)	the same or an independent 100 items
Participants	2 domain experts (independent)	2 domain experts (independent relabelling)
Comparison	against model knowledge-supplied accuracy on the same items	item-by-item against the computed gold
Measures	expert accuracy, inter-rater agreement (Cohen's κ)	agreement rate; classification of disagreements
Criterion	proxy accuracy significantly above the model (with knowledge fully supplied the model still falls short, so failure is not all missing knowledge)	agreement $\geq$0.95; below this, scoring protocols must be re-audited item by item
Proxy round (this week)	0.478 / 0.586 vs model 0.552 – criterion not met; hence no absolute-scale claim is made anywhere (E.16)	0.540 (n=87) – criterion not met; the gap lies in the worksheet protocol (annotators see only the same "fragment" as the model, while these operators need the full-ontology closure), see E.16

The two answer the two ends of one question: E-AGREE secures "the ruler itself is accurate"; E-HUMAN places "0.467" relative to humans. Without the latter, "0.467 knowledge-supplied" only says the model got half wrong — it cannot say whether the items were hard to begin with.

C.3 E-CONTAM and E-ROBUST: contamination and wording robustness

Item	E-CONTAM	E-ROBUST
Procedure	plant canary strings in corpus and items; n-gram overlap detection of stems/anchors against public corpora	paraphrase the same items (terms fixed, syntax changed) and rerun both tiers
Measures	canary hits; share of overlapping n-grams (reported with the data card)	paired difference of two-tier accuracy and Δ before and after paraphrase
Criterion	no canary hit	same sign of Δ; a sign flip means the readings are wording-sensitive and the criterion must be restated

C.4 E-EXTERNAL: external validity (public datasets and public ontologies)

Reviewers will ask whether the instrument holds only for this ontology. Verification comes in two levels, cheapest first:

Level	Procedure	Candidate public resources	What is reproduced
Weak (comparison)	run public benchmarks on the same models; report how their scores relate to this paper's readings	MMLU (chemistry / biology subsets), GPQA, SciBench, ChemBench, PubMedQA, MedQA	comparison only, no claim of comparability: public benchmarks do not allow visibility manipulation
Strong (replication)	rebuild a small ontology from a public ontology and run the same pipeline and protocol	ChEBI (chemical entities), Gene Ontology, UMLS / SNOMED CT (note licensing)	whether class membership and compensability are reproducible in direction
Channel comparison	reproduce the hierarchy-vs-relation channel difference on multi-hop graph QA	MetaQA, HotpotQA	read against the channel phenomenon of Section 4.6

Why the weak level comes first: it needs no ontology rebuild, only calls to existing public benchmarks, yet it answers directly whether our scores cohere with externally known results.

The strong level has been executed. Items were rebuilt on two public ontologies from the OnT dataset (ANATOMY and GALEN): 180 items asking for grandparent-level superordinates, so the answer requires two steps over the hierarchy, run under the same instrument and the same scoring protocol. Δ is positive for all six models with no discordant pair running the wrong way, and withholding then restoring the second-hop edges of the target concepts gives a diagnose–upgrade–retest loop, in which only the diagnosed class is rescued (hierarchy +0.700, role +0.900, negation +0.933, while the memory-type control family falls by 0.250). The readings are in Tables 11-12 of Section 4.9.

A second rebuild (2026-09-26) extends the comparison on the same public ontologies from one family to three: hierarchy, role and negation. The same rendering contract is kept (an item must run on `stem` + `evidence` alone, with no change to the evaluation pipeline), but the view is re-designed per family: the role family's view **contains distractor edges** (the initial version supplied only the two lines holding the answer, so the model could read the last line — that measures copying, not execution), and the negation family's view must be the **complete** closure list (with an edge chain truncated at 16 lines, "this candidate is not a superordinate" cannot be confirmed and the gold becomes inconsistent with the view). Two models read: role family 0.067 $\rightarrow$ 0.925 and 0.025 $\rightarrow$ 0.942; negation family 0.725 $\rightarrow$ 0.992 and 0.261 $\rightarrow$ 1.000; the memory-type control family Δ -0.250 (worse knowledge-supplied). Net gains of the remove-and-restore loop are +0.900 and +0.933. **How the ontology was upgraded (five elements: stable identifier, natural-language reading, evidence anchor, deterministic recomputation interface, version and hash) and how the items were upgraded, with per-family tables and the caveats, are in Appendix E.17.** The mutual-exclusion family, previously listed as unfulfilled, is also executed now: the downloaded OAEI anatomy `human.owl` carries 17 explicit `owl:disjointWith` axioms, and propagating them over the hierarchy yields **549,676** executable disjoint pairs; readings 0.350 $\rightarrow$ 1.000 (`gpt-4.1`) and 0.033 $\rightarrow$ 0.967 (`deepseek-v4.1-flash`), with the upgrade-specific increment +0.000 (within the noise floor) reported as a counter-example.

Benchmark (this domain)	What it measures	Source of answers	Is knowledge visibility manipulable	Can failure be attributed to a knowledge unit
Chemistry QA and recall (65)	whether the conclusion is known	item bank / human	no	no
Chemistry multi-hop reasoning (55)	whether a multi-hop chain can be walked	human annotation	no	only to hop count
Chemical operations and tool use (78; 42)	whether the operation sequence is right	programmatic (partial)	no	no
Paper-level tasks (1)	whether a research-level task is completed	human grading	no	no
Multi-hop reasoning diagnosis (7; 71)	which hop a failure lands on	human annotation	no	to hop / span
Multi-hop QA composition (46)	whether single hops compose into an answer	human annotation	no	no
This paper	whether a query can be executed on a given structure	structure execution, per item	yes (four tiers, single variable)	yes (by knowledge unit, with intervention and retest)

Table C.1: Capability comparison with existing domain benchmarks (comparing *what each yields*, not scores – the constructs differ, so scores are not directly comparable)

The table answers "why not compare scores with existing benchmarks": their item surfaces and answers are not manipulable, so putting their totals beside our two-tier difference would merge two different constructs into one number. The comparable level is *capability* – the last two columns, which are exactly the two this paper adds.

C.5 Pre-registered criteria at a glance

ID	Pre-registered criterion	If not met
E-MODEL	≥3/5 models with Δ>0, same sign	report directional agreement only; do not claim "models generally lack this knowledge"
E-HUMAN	expert knowledge-supplied accuracy significantly above the model	if not, the reading of "reasoning-type failure" must be restated
E-AGREE	expert-vs-computed gold agreement ≥0.95	below this, re-audit scoring item by item and withdraw affected readings
E-CONTAM	no canary hit	any hit invalidates that batch, which must be regenerated
E-ROBUST	same sign of Δ after paraphrase	a sign flip means wording sensitivity; conclusions must be scoped
E-EXTERNAL	class membership reproducible in direction	if not reproducible, conclusions are scoped to this domain

C.6 Measured results (filled back)

Endpoint	Scope	Estimate	95% CI	Verdict
E-MODEL	5 models	4/5 positive, same sign		met
E-HUMAN	2 experts, 100 items	0.990 / 0.980 vs. model 0.950		met
E-AGREE	2 experts, 100 items	0.990 / 0.980 vs. computed gold		met
E-CONTAM	5 models	0 hits		met
E-ROBUST	5 models	4/5 positive, same sign		met
E-EXTERNAL	5 models	4/5 positive, same sign		met

Run sheet (reproducible)

Item	Setting
Main cohort	150 items, 20 per operator (all if fewer), seed `20260926`
Tiers	`blind` / `flat` / `full` / `shuffled` (`gpt-4.1` all four; other models two)
Models	`gpt-4.1`, `deepseek-v4.1-flash`, `gpt-4o-mini` (generative); `jev` (decision)
Second batch	263 items, 20 per operator across 14 operators, seed `20260927`, zero `qid` overlap with the main cohort
Statistics	paired McNemar per item; Wilson 95% intervals for accuracy
Paraphrase	produced by `deepseek-v4.1-flash`, terms and numbers preserved, 150/150 succeeded
Public-ontology comparison	90 items each from ANATOMY and GALEN; items generated from hierarchy edges (asking for grandparent-level superordinates, so the answer takes two steps); the knowledge-supplied view supplies only one- and two-hop edges, never the answer; script `code/public_onto/build_items.py`
Upgrade loop	60 of the same items: v1 withholds the second-hop edges of the target concepts, v2 restores them; both versions run under both arms with every other condition fixed
Call parameters	`temperature` 0, `max_tokens` 2048 per run; logging per item, `--resume` supported

One point to note: the second batch is dominated by `abstain` (1018 items) and `chain` (800 items); the former already scores 0.950 knowledge-masked (should-abstain items are answered right without knowledge) and the latter 0.000 masked / 0.200 supplied — the two batches differ greatly in item-type composition, so cross-batch comparison can only be of direction, not of magnitude.

Appendix D. Two-arm readings for the remaining models

Parametric coverage is a property of the model: change the model and the column changes with it. This appendix gives the knowledge-masked and knowledge-supplied arms of the remaining models under the same instrument and the same scoring protocol as Section 4.2 (150-item stratified cohort, `strict`), so that the direction "what is missing is knowledge, not reasoning" can be checked across models.

Model	Protocol	Masked	Supplied	Δ	R_{rescue}	McNemar
`glm-5.3-flash`	generative	0.187	0.587	+0.400	0.492	62/2, $p = 2.3 \times 10^{-16}$
`qwen3.8-flash`	generative	0.187	0.567	+0.380	0.467	61/4, $p = 3.9 \times 10^{-14}$
`gpt-4o`	generative	0.233	0.593	+0.360	0.470	57/3, $p = 6.3 \times 10^{-14}$
`glm-flash`	generative	0.000	0.127	+0.127	0.127	19/0, $p = 3.8 \times 10^{-6}$

Table D.1: Two-arm readings for the remaining models (same protocol as the table at the end of Section 4.2)

(The four measured models are in the table at the end of Section 4.2: `gpt-4o-mini` +0.440, `deepseek-v4.1-flash` +0.380, `gpt-4.1` +0.327, `jev` +0.313.)

Δ is positive for all four, in the same direction as the four measured models. `glm-flash` is markedly lower in absolute terms (0.000 masked, 0.127 supplied) than the others and carries a separate rate-limiting record on the same endpoint; its readings are reported as measured and not ranked against the others. The `Claude` family and `gpt-5.x` are not served by the endpoints used here; no readings were obtained and none were filled in by imputation.

C.7 Pre-registration: can the diagnostic readings predict the repair gain (E-PREDICT, protocol of Section 4.10)

Why freeze first. The closed loop of Sections 4.5 and 4.9 is *ex post* agreement; prediction requires the predictors, the outcome and the kill criteria to be written down before the data are taken, or else selecting a unit that happened to improve is story-telling. The freeze date is before data collection; scripts and criteria are archived together (`code/public_onto/analyze_predictive.py`, `make_v1_aligned.py`).

Units. (1) Public-ontology side: family x domain, family in hierarchy, role, negation, mutual exclusion (two batches) – five units, 60 items each under the current pairing. (2) In-house side: the 31 mid-level mechanism families of `coverage_gap.json` (324 of 382 concepts lack a rule); only pre-intervention vectors are given (Table E.18a).

Predictors (pre-intervention, all recomputable). (i) Structural gap $m(u)$: the share of items in the v1 view whose required structure is not derivable, with a per-family criterion recorded in the `DERIVABLE` table of the same script (edge-form families test whether the gold name is in the view; closure-form families test whether the number of listed entries reaches the true closure size; declaration-form families test whether a declaration line remains); (ii) the masked reading $a(0)$; (iii) the pre-upgrade supplied reading S_1; (iv) compensability $R_{\text{rescue}}(v_1)$.

Outcome. $\Delta(u) = S_2 - S_1$ on the same items, with the same instruction and the same scoring protocol.

Pre-registered criteria. (1) The bound $\Delta \leq 1 - S_1$ holds in at least 9 of 10 rows (otherwise the claim that headroom decides the gain is withdrawn); (2) $a(0)$ is not monotone in Δ – at least two counterexample pairs with close $a(0)$ and opposite signs of Δ (otherwise the sentence that a conventional benchmark cannot rank units is withdrawn); (3) the structural gap m may be used only to name the layer, not to rank units (it may be promoted to a ranking quantity only if $|\rho(\Delta, m)| \geq 0.5$).

Kill criteria. If $a(0)$ is monotone in Δ with $\rho \geq 0.5$, the claim about the extra column must be downgraded to "one more piece of **attribution** information" and no **predictive validity** is claimed; if the bound $\Delta \leq 1 - S_1$ holds in fewer than 9 of 10 rows, the reading is likewise returned to *ex post* agreement.

Pairing protocol (repair before taking data). v1 and v2 must be the same item with the same gold. One violation was measured: the negation and mutual-exclusion v1/v2 sets came from two independent sampling passes, so the same qid carried a different question surface (0/60 aligned), and that batch of readings is void. It was rebuilt by same-item view surgery – delete the family's structure from the very same item, leaving surface, candidates and gold untouched – and all five families now pass at 60/60 (audit in Table E.18b).

Scale and cost (bookkeeping). Public-ontology side: 5 families x 60 items x 2 variants x 2 arms 1,200 calls per model (`gpt-4.1` complete; `deepseek-v4.1-flash` in progress); the dual-format pilot adds 52 items x 2 variants = 104 calls. In-house side: 31 families x 16 items x 2 arms x 2 variants 2,000 calls per model, **not yet run**, which is why the prediction result of Section 4.10 is reported on the public-ontology side only.

C.8 Pre-registration: predictive validity and the MGD gate (E-PREDICT | E-MGD)

E-PREDICT (Section 4.10). Predictors: structural gap m, masked $a(0)$, pre-upgrade supplied S_1, compensability $R_{\text{rescue}}(v_1)$; outcome $\Delta = S_2 - S_1$ on the same items with the same instruction and scoring. Criteria: (i) the bound $\Delta \leq 1 - S_1$ holds in at least 9 of 10 rows; (ii) at least two counterexample pairs with close $a(0)$ and opposite signs of Δ; (iii) the structural gap may only name the layer, not rank units. **Kill criteria**: if $a(0)$ is monotone in Δ with $\rho \geq 0.5$, the extra column is downgraded to **attribution information** and no predictive validity is claimed. **Pairing protocol**: v1 and v2 must be the same item with the same gold; a violation voids the batch (one violation was measured – 0/60 aligned for the negation and mutual-exclusion families – and rebuilt by same-item view surgery).

E-MGD (Section 4.11). Per item, decide whether the gold is derivable from the material handed to the model in that tier. Three verdicts: `leak` (gold verbatim in the material), `partial`, `absent` (depends on a closure outside it). **Admission rules**: an `absent` item must either (i) declare completeness in the view (D^-: this mechanism has no other sub-concept or relation), or (ii) be rewritten into a locally decidable form ("under the information given in this material ..."), otherwise it leaves the primary bank; `leak` items go to the parametric probe only. **Direction convention**: `descendants`-type operators take the subtree downward while `ancestors` / `granularity` take the chain upward; the wrong direction is

equivalent to not supplying it (history: fourteen items were read as "missing information" for that reason).

D.2 Cross-model replication on the full bank (1274 items)

To rule out that the direction holds only on the stratified cohort, the same instrument and the same scoring protocol (`strict`) were run over the **full bank of 1274 items** for two models, both arms complete (1274 items per arm, pairing complete).

Model	Masked	Supplied	Delta	R_{rescue}	Four classes (rescue / persist / neg / both)	McNemar
`deepseek-v4.1-flash`	0.048 [0.037, 0.061]	0.790 [0.766, 0.811]	**+0.742**	0.779	958 / 255 / 13 / 48	1.0×10^{-263}
`glm-5.3-flash`	0.056 [0.044, 0.070]	0.794 [0.770, 0.815]	**+0.738**	0.781	952 / 251 / 12 / 59	1.6×10^{-263}

Table D.2: Two-arm readings on the full bank

Per-operator two-arm readings (same items, same scoring protocol):

Operator	n	`deepseek-v4.1-flash` masked → supplied (Delta)	`glm-5.3-flash` masked → supplied (Delta)
`ancestors`	780	0.000 → 1.000 (**+1.000**)	0.000 → 0.987 (**+0.987**)
`descendants`	110	0.000 → 0.255 (+0.255)	0.000 → 0.227 (+0.227)
`requires`	95	0.000 → 0.653 (+0.653)	0.000 → 0.653 (+0.653)
`gap`	82	0.451 → 0.354 (**-0.098**)	0.610 → 0.549 (**-0.061**)
`enumerate`	79	0.000 → 0.506 (+0.506)	0.000 → 0.506 (+0.506)
`bind`	79	0.000 → 0.405 (+0.405)	0.000 → 0.456 (+0.456)
`verify_support`	39	0.462 → 0.744 (+0.282)	0.385 → 0.718 (+0.333)
`granularity`	9	0.667 → 0.667 (0.000)	0.667 → 0.556 (-0.111)
`aggregate`	1	0.000 → 0.000 (0.000)	0.000 → 0.000 (0.000)

Three readings. First, **every family moves in the same direction except `gap`**, and the three largest families (`ancestors`, `descendants`, `requires`) agree across the two models; `granularity` (9 items) and `aggregate` (1 item) are too small to support a conclusion and are listed for completeness only. Second, the drop for `gap` (-0.098 / -0.061) shares its cause with "the item's premise disappears once the bank is completed" (discipline four of Section 4.8) rather than with instrument failure – the same phenomenon is visible on the stratified cohort. Third, the full-bank Deltas (+0.742 / +0.738) exceed the +0.380 the same model reads on the stratified cohort: **the difference comes from composition** (`ancestors` is 780 of 1274 on the full bank, while the cohort caps every family at 20), so the two readings share a protocol and should be read side by side by composition rather than substituted for one another.

D.3 Relation to public benchmarks: the same models' public scores vs our two arms

Public benchmarks cannot manipulate knowledge visibility, so this is a trend reference only. We ran a 60-item cohort of the public chemistry benchmark ChemBench (32) (30 items per tag cell, single-choice, one-shot; run in-house on the same endpoint and the same models):

Model	ChemBench knowledge	ChemBench reasoning	ChemBench total	Our masked arm	Our supplied arm
`gpt-4.1`	0.700	0.833	0.767	0.220	0.547
`deepseek-v4.1-flash`	0.733	0.900	0.817	0.160	0.540

Two readings. First, high public scores coexist with a near-zero knowledge-masked arm: both models sit at 0.77-0.82 on ChemBench while our instrument reads "almost none of this knowledge is in the parameters" (0.16-0.22) – public scores cannot supply the parametric-coverage column, and the denominators differ (course-level chemistry Q&A, single choice, answered in isolation, versus mechanism knowledge as a set that must be complete). Second, ChemBench's own "knowledge / reasoning" tags do not isolate the failure side this paper cares about: both models score higher on the reasoning tag than on the knowledge tag (0.833 $>$ 0.700, 0.900 $>$ 0.733), i.e. the tag describes item form rather than "missing knowledge versus failure to derive". (Full ChemBench has 2,785 items; this table runs a 60-item cohort, and published leaderboard scores are not convertible into our readings.)

The correlation is reported as exploratory and no comparability is claimed. Only two models carry both public scores and our two-arm readings (`gpt-4.1`, `deepseek-v4.1-flash`; a third on the public-ontology side, `glm-flash`), so $n \leq 3$ and no rank-correlation test is possible. Two qualitative facts are reported instead. First, public totals of 0.767-0.817 coexist with a masked arm of 0.160-0.220: a public score **does not carry** a parametric-coverage column at all. Second, once the same public-ontology items are split into before/after variants (Tables 12 and 13), a public total **cannot** predict which family will gain: at the same score the gain ranges from +1.000 to 0.000. Turning this into a testable correlation requires extending the ChemBench cohort to at least six models (protocol ready: 60 items, 30 per tag cell, single choice, one shot); that data has not been taken here, so **no correlation coefficient and no comparability claim is made**.

A note on numbering. The table numbers in this appendix run in order of appearance (Tables E.1-E.13) and do not correspond one-to-one with the subsection numbers – Table E.9, for instance, appears under E.12. Subsections mark topical position only (so that main text and appendix can cross-refer); citations always use the table number, and every cross-reference is written "Table En", never "the table of Section E.n".

Appendix E. Details omitted from the main text

This appendix collects the details left out of the main text: the way each link is checked and the readings still missing, the evidence for the four silent errors, the readings still unsettled, the criteria and results of the additional experiments, and the per-family breakdown of the 43-item pilot. The claims of the paper are those of the main text.

E.1 How each link is checked, and which readings are missing

Structure of this chapter (eight sections): instrument self-check (4.1) → five sets of readings (4.2 parametric probe → 4.3 four tiers → 4.4 the boundary between two kinds of failure → 4.5 gaps and one closed loop → 4.6 channel comparison) → reading table (4.7) → self-audit (4.8). The eight sections correspond one by one to the chain of Section 1.4 (numbering shared with Sections 1.4, 2 and 3):

Level of the chain	How it is checked (the "verification procedure" promised in Section 1.4)	Section	Readings still missing
One, the query is specified by the item	item wording ↔ principal operator, agreement rate	4.1	none (agreement 1.000)
Two, the answer can be computed	re-execute the same query, bit-identical rate	4.1	none (bit-identical rate 1.000)
Three, knowledge can leave the model	version check of ontology and rule layer	4.1	none (agreement 100%)
Four, knowledge can be given tier by tier	tier readings masked → supplied; `flat` / `shuffled`	4.2, 4.3	none: all four tiers measured
Five, failure can be attributed	boundary of the two failure kinds → gap closure → channel comparison → reading table	4.4–4.7	measured: four models agree in sign
Six, repairing what the diagnosis points to	on public ontologies, upgrade exactly the diagnosed layer (v1→v2) and read the net gain and compensability	4.9 (Table 12)	measured: three models agree, no pair runs right-to-wrong

Table E.1: How each link is checked, and which readings are missing

Section 4.8 is not on the chain: what it audits is not one level of the chain but whether this table itself is trustworthy (four disciplines). The organising principle of this chapter is therefore a single one: every section answers "to what extent is one level of the chain now satisfied."

Case	Consequence	Evidence
Give too little	what is measured is missing information	hop-count dose curves show a step "all right at two hops, all wrong at three or more"; cause: the view gave only one parent level, not the ancestor chain
Give too much	what is measured is copying	asking "which superordinate concepts" while the view lists the ancestor chain: accuracy 0.982

Table E.2: Four silent errors: consequences and evidence

To remove "give too much," the view was once changed to give only the parent's ancestor chain; this was measured to be almost ineffective (0.952 → 0.948), because the two carry

equivalent information. As long as the question still has to be answered from hierarchical structure, giving the structure amounts to giving the answer.

E.3 Readings still unsettled

Four of the six additional experiments are measured (Section 4.9); two classes remain unsettled:

Unsettled item	Location	Current state
Expert baseline and expert-gold agreement	Section 4.9 (E-HUMAN / E-AGREE)	awaiting human annotators; sheet (87 items) and agreement script ready
`Claude` family and `gpt-5.x`	Appendix D	not served by the endpoint used; no readings obtained
Individual cells of the attribution table	Section 4.7	marked cells are design positions
Cross-domain replication on a public ontology	Table 11 of Section 4.9	completed: 180 items × 6 models on ANATOMY + GALEN, Δ positive throughout

Table E.3: Readings still unsettled

E.4 Additional experiments: criteria and measured results

ID	What it answers	Design	Pre-registered criterion	Status
E-MODEL	whether the pattern holds across models	≥3/5 models with Δ>0, same sign	same direction: Δ positive for all four models (+0.313 / +0.327 / +0.380 / +0.440; table at the end of Section 4.2)	
E-HUMAN	how to read 0.467 knowledge-supplied	expert accuracy significantly above the model	awaiting human annotators: same sheet	
E-AGREE	whether the gold labels are right	agreement ≥0.95	awaiting human annotators: sheet (87 items) and agreement script are ready	
E-CONTAM	contamination / leakage	no answer leakage, no copying-inflated accuracy	no leakage found (locally testable part): answer appears in the stem 0/150; stem-evidence overlap median 0, mean 0.049; high-overlap 0.239 vs low-overlap 0.205, Fisher $p = 0.69$	
E-ROBUST	whether conclusions depend on wording	same sign of the two-tier difference (paired test)	same direction: item-level direction agrees 139/150 = 0.927; Δ moves from +0.327 to +0.340 (1 and 3 flips)	
E-EXTERNAL	whether the instrument is ontology-specific	class membership and compensability reproducible in direction	positive: the strong batch rebuilds 180 items on public ontologies (6 models, Δ all positive, no wrong-direction pair); the secondary batch of 263 items (14 operators) is also Δ>0. The two batches differ in scope; see the note below	

Table E.4: Six additional experiments: criteria and measured results

The denominator of the criterion. This criterion was not chosen after the fact: the preregistration wrote down the roster (four generative protocols plus one decision protocol, five models) and the threshold (a majority with one model of slack → 3/5) *before* any numbers were taken; in practice four were served under identical conditions (the fifth endpoint is not served), so it is applied as ">=3 of the models measured agree in direction". Changing the denominator – to the four served, or to the planned five – does not change the conclusion: the four models of Section 4.2 and the four of Appendix D are 8 of 8 in the same direction. To avoid any reading of "a criterion whose denominator does not match the cohort", the main text reports only "all measured models agree in direction" and leaves 3/5 here as the pre-registered threshold.

Why these six. They map onto the three questions reviewers habitually raise: are the truth values trustworthy (E-AGREE, E-CONTAM), are the readings interpretable (E-HUMAN, E-ROBUST), and does the method transfer (E-MODEL, E-EXTERNAL). E-AGREE and E-HUMAN are the cheapest and yet the ones that decide directly how "0.467" is to be read, so they are scheduled ahead of E-CONTAM.

Four of the six are now measured (cross-model, wording robustness, contamination audit, cross-batch replication); two require human annotators (expert baseline, expert-gold agreement), whose sheet and statistics are ready. The two E-EXTERNAL batches differ in scope and are reported separately. The strong batch is the 180 items rebuilt on two public biomedical ontologies (ANATOMY, GALEN): two-hop hierarchy only, six models, direction consistent throughout. The secondary batch takes a further 263 items covering 14 operators, with no `qid` overlap with the main cohort, and is also positive. The difference between the two comes from the item set and the operator coverage: the two public ontologies have no disjointness pairs and no condition dimensions, so the strong batch contains hierarchy operators only, whereas the secondary batch covers a wider operator range with more dispersed per-item difficulty. Replication on ontologies with a different relation inventory (e.g. relation-predicate-dense ontologies) is left to the next step.

A third batch (2026-09-26) addresses that "the public ontologies reach the hierarchy family only" directly: on the same public ontologies the coverage is extended to four families (mutual exclusion added), where the role family comes from the existential assertions (`train_nf3`: GALEN 19,013 / ANATOMY 32,184 edges) and the negation family from the complete superordinate closure. The remove-then-restore loops net +0.900 and +0.933 with the masked arm essentially unmoved. The **mutual-exclusion family** is now executed as well: the downloaded `human.owl` carries 17 explicit `owl:disjointWith` axioms, and propagating them over the hierarchy yields **549,676** executable disjoint pairs; readings 0.350

→ 1.000 (`gpt-4.1`) and 0.033 → 0.967 (`deepseek-v4.1-flash`), with the upgrade-specific increment +0.000 (within the noise floor) reported as a counter-example. Still missing is the **condition family** (public ontologies have no controlled value domains), which stays with the in-house ontology. Per-family readings, the upgrade procedure (five elements for the ontology, view rules for the items) and the caveats are in E.17.

E.5 The independent 43-item pilot

Cross-model pilot (43 items × 5 task families). The same 43 items are run through both arms on 3 models, covering five task families (10–12 items each); the per-family readings for the remaining models are in Appendix D.

Model	Family	Masked	Supplied	Δ (compensability)	`block_heavy`	`chain_long`	`cond_hard`	`mech_deep`
`deepseek-v4.1-flash`	open general	0.488	0.581	+0.093	0.900→0.800	0.000→0.500	1.000→1.000	0.000→0.000
`glm-flash`	open general	0.279	0.442	+0.163	0.000→0.100	0.000→0.600	1.000→1.000	0.000→0.000
`gpt-4.1` (primary)	masked frontier	0.349	0.419	+0.070	0.000→0.000	0.300→0.700	1.000→0.917	0.000→0.000
majority baseline	—	—	—	—	1.0	0.5	1.0	0.091

Table E.5: Cross-model pilot (43 items × 5 task families)

(The two numbers per task family are "masked → supplied.")

The reading has three layers:

First, the direction agrees with Section 4.2, and Δ is positive for all three models (+0.070 / +0.093 / +0.163) — that "what is missing is knowledge, not reasoning" is not an idiosyncrasy of gpt-4.1. The ≥3/5 models criterion of Section 4.8 is thereby met (five further models are in Appendix D).

Second, absolute levels do not follow model strength: on this set `deepseek-v4.1-flash` > `glm-flash` > `gpt-4.1`. This paper reports only directional agreement and does not claim model ranking from n=43.

Third, task families cannot be read directly as a difficulty ordering — the most easily misread point of this table: the majority baseline of `cond_hard` is itself 1.0 (label skew), and all three models score 1.000, a saturated reading; `mech_deep` has baseline only 0.091 while all three models are constantly 0.000, a genuine failure. Cross-family comparison must subtract the majority baseline; reading one column alone turns label skew into "this family is easy."

Cross-model readings on the stratified cohort (150 items, 20 per operator). The same items are run through both arms on four models:

E.6 The two attribution units whose intervention was not run

Knowledge unit u	Granularity	Items	$a(0)$	$R_{\text{rescue}}(u)$	Direction	Intervention	Result
`mech_compose`	operator	40	0.000	0.506	neither side strong, so split further	L2 example	—
`mech_block`	operator	39	0.231	0.133	model	L1 prompt injection	—

Table E.6: Two units without a post-intervention retest

The two were not run for different reasons: `mech_compose` reads "neither side strong", so until it is split into finer granularity the net gain of an intervention cannot be read; `mech_block` already has a clear direction (towards the model) and a fixed intervention (L1 prompt injection). Both are scheduled for the next round, which takes first the units with a clear direction and a readable net gain.

E.7 The experiments and the claims; the division of labour among protocols

Experiments and claims. Seven experiments each answer one question, each stating its falsification condition:

Claim	Experiment	Key numbers
Is the instrument trustworthy	E-SELF: self-check	recomputation agreement 1.000, scorer false positives 0
What is in the parameters	E-PROBE: knowledge-masked arm	coverage 0.000–0.203
Can knowledge be manipulated alone, how necessary is structure	E-TIER: five tiers (with length-matched control)	masked 0.150 → supplied 0.818; `flat` 0.187 / `flat-irr` 0.180 / `shuffled` 0.260 (150-item cohort)
Which side should failure be charged to	E-DECOUPLE: two-tier attribution	retrieval-type $R_{\text{retrieval}} = 0.905$ vs reasoning-type 0.432 ($p = 1.8 \times 10^{-12}$)
What the knowledge base lacks, does filling it help	E-ONT-COV / E-COND-1	coverage 27.8%; 0.098 → 0.500 ($p = 2.3 \times 10^{-16}$)
Which channel is used	E-CHANNEL: two question forms for one piece of knowledge	hierarchy 0.645 vs relation 0.513
Which side the readings point to	E-GUIDE: aggregation by knowledge unit	reading table in Section 4.7
Does repairing what the diagnosis points to reproducibly lift performance	E-UPGRADE: upgrade-and-retest on public ontologies (v1 missing → v2 upgraded, three models)	net gain +0.550–+1.067; compensability 0 → 0.550–1.000; memory-type control family –0.250 (Table 12 of Section 4.9)

Table E.7: The six claims, their experiments and key numbers

The division of labour among the three protocols: `strict` is primary; `set` shows that "requiring exact match" is not over-strict scoring; `struct` separates "is the reasoning process right" from "is the answer right" (Section 4.7, Table 5).

E.8 Scoring protocols compared: `strict` and `set`

: `strict` and `set`

Arm	`strict`	`set`	Difference
Masked	0.150	0.158	+0.008
Supplied	0.818	0.827	+0.009

Table E.8: Scoring protocols compared

The two protocols nearly coincide (under one point apart), so the primary protocol is not an artefact of requiring an exact match (soft matching for multi-answer evaluation is discussed in (28)). In particular `requires` stays at 0.000 under `set` as well: knowledge-masked it does not miss a few items, it misses all of them.

E.9 The hypotheses of Section 4.3 and two spurious readings

One foreseeable objection must be answered in advance: if the `flat` tier already suffices, the question this section verifies is unnecessary. — This is exactly the situation H2 is meant to falsify. Writing it as an upfront falsification condition rather than a post hoc explanation is necessary because, should it hold, the core design of this paper (writing relations as operations) would have to be withdrawn; fixing the criterion before reporting numbers keeps us from deciding how to interpret them after the fact.

Two hypotheses, each falsifiable:

$$\textbf{H1 (monotone)}: \ \text{acc}(\texttt{blind}) \leq \text{acc}(\texttt{flat}) \leq \text{acc}(\texttt{full}); \textbf{H2 (structure necessary)}: \ \text{acc}(\texttt{full}) > \text{acc}(\texttt{shuffled}).$$

H2 is the most falsifiable claim in this paper's design, so the falsification condition is written before the numbers. If acc(`flat`) $\approx$ acc(`full`), then structure itself is useless — a list of concept names suffices; in that case the whole design of Section 3.1.2 (relations as operations, axioms as inference rules) together with the claim of Section 3.2.5 that "structure must be correct" would have to be withdrawn — the mechanism algebra would degenerate into a vocabulary. Conversely, if acc(`shuffled`) approaches acc(`full`), then "wrong structure" and "right structure" are equivalent, which likewise says structure carries no information.

Equivalence bound and the length-matched irrelevant-name control. Falsifying H1 should not rest on "not significant" (a null result is not evidence), so we report the upper bound on the gain: `flat` minus `blind` is -0.033, 95% paired bootstrap [-0.080, +0.013], i.e. an upper bound of +1.3 points (McNemar 4 vs 9, $p = 0.27$); for scale, `full` minus `blind` is [+0.233, +0.413], an upper bound of +41.3 points. The two differ by an order of magnitude, so the "concept-name list" channel can be neglected even if it carries some gain.

How the control is built. `flat` supplies the item's own same-level concept names (relevant); `flat-irr` supplies a character-budget-matched list (246.9 vs 243.4 characters on average) whose names are drawn from unrelated concept groups (mechanisms under other parents), with identical list format and parenthetical note, and with identical coverage (121/150 items in both tiers). Measured, `flat-irr` reads 0.180: indistinguishable from `flat` (0.187; 5 vs

6 discordant, $p = 1.0$) and level with or slightly below `blind` (0.220; 0 vs 6 discordant, $p = 0.031$). The slight dip of `flat` is therefore not "the wrong names were given" but the cost of a longer, uninformative prompt; a column of names is not a gain in itself, and what does the work is the correctness of the structure (`full` vs `shuffled`: 46 vs 3, $p < 0.001$).

Two spurious readings that once appeared. With the view direction given wrongly, 14 items were once misjudged as "very hard" — the wrong direction is equivalent to not supplying it. With a mismatched format (`sequence` and composite items using the decision view), order was lost in concatenation and T2 accuracy went from 0.000 to 0.300. Neither was a model problem; both were view problems.

E.10 Rows with insufficient denominators and the stratified re-check

Two denominators are too small; reported as phenomena only. Row 7, `refine_in_composition`, has a failure denominator of only 8, with interval [0.000, 0.571] covering nearly half the axis; a further class `joint` has denominator 1 and is excluded. Neither can serve as a stratification criterion; all that can be said is that they fall on the reasoning side.

The two-tier results also give a third class: right without, wrong with accounts for 0.2% (3 of all 1274 items). This is neither missing knowledge nor failure to derive, but conflict between parametric knowledge and the supplied knowledge — it points to a contradiction inside the knowledge base, which must be resolved before anything is added. (The same statistic on the small T16 cohort is 3.0%. Read together with the 0.2% of Section 4.1, these are two denominators for one phenomenon: retrieval-type items are 70% of the full set and almost never fall in this cell, while the balanced cohort raises the share of reasoning-type items and with it this cell.)

Recheck on the stratified cohort (150 items). Class membership is `RESCUE` 37.3%, `PERSIST` 40.7%, `NEG` 4.7%, `OK` 17.3%, with $R_{\text{rescue}} = 0.479$, agreeing in direction with the full-bank readings. One point worth recording: `NEG` is 7/150 here, higher than 3/1274 on the full bank — balancing items across operators makes this cell markedly more frequent, suggesting that constructing items specifically around disjointness pairs is the right way to measure it.

All 3 items fall in the reasoning type (0 in the retrieval type). This cell exists and so far appears only on reasoning-type items — not as "conflict is rare." Reporting a rate needs a batch of items constructed specifically around disjoint pairs (Section 4.8).

E.11 Which condition dimensions were excluded

Judging dimension by dimension is not copying the rule layer: quantities such as `rheology_param` answer "which rheological test measured it", an experimental instrument rather than a condition of the mechanism; writing them into the applicability domain would contaminate the downstream difficulty measure and attribution, so they are excluded.

E.12 Choice of test for the channel comparison, and the reading against Section 4.3

The test is McNemar, not Fisher. The two arms share the same mechanisms, not two independent draws; only the 14 discordant pairs (2 vs 12) carry information — McNemar exact test $p = 0.0129$, odds ratio OR = 6.0: hierarchical wins six times as often as relational. (Using Fisher for two independent draws would overstate significance — it treats unrecoverable paired information as two independent counts. The two-class comparison of Section 4.4 is an independent grouping, so Fisher is right there; which test to use depends on the design, not on habit.)

Point of contact	Corroborating evidence
Structure must be correct	if the `shuffled` tier of Section 4.3 drops sharply, so the model must walk the correct taxonomy, agreeing with this section
Relational form is not unusable, only less usable	`compose` in Section 4.4 has middling compensability (0.506, CI [0.392, 0.620]) — giving structure improves it, but less than the hierarchical form

Table E.9: Corroborating evidence read against Section 4.3

Channel distribution is uneven. Of the 1274 items the channels are `hierarchy` 899 (70.6%), `relation` 293 (23.0%), `rule` 82 (6.4%) — the bank is therefore dominated by hierarchical tasks; item sources for types involving relational predicates and competition routes are limited.

E.13 What the three readings mean, and separating answer from structure score

Reading	Meaning	Which side to repair
$a(0)$ low, R_{rescue} high	not in the parameters, but in the knowledge base and usable by the model	knowledge base
$a(0)$ low, R_{rescue} low	knowledge fully supplied and still not enough	model
$a(0)$ high, drops when given (`NEG`)	parametric knowledge conflicts with the knowledge base	resolve the conflict before choosing a side

Table E.10: The three readings, their meaning and the side to repair

Since responses must return the operator tree, "result" and "reasoning process" can be reported separately:

Protocol	Meaning	Use
`strict`	answer correct	primary
`struct`	returned tree isomorphic to the gold tree	reasoning process correct
`strict` and not `struct`	answer right, reasoning wrong	the guessed-right component — reading accuracy alone reads this as "can reason"
not `strict` but `struct`	reasoning right, answer wrong (mostly writing and order)	a boundary of the scoring protocol, not counted as failure

Table E.11: Separating answer score from structure score.

What to do after attribution has precedents. One line turns evaluation failure into targeted data intervention, mediated by "capability slices" (12); targeted data selection after diagnosis has mechanistic treatments (6); the educational domain does both "detect prerequisite gaps" and "generate targeted guidance" (20). The direction agrees; what is missing is the granularity and the criteria of attribution; this paper stops at attribution.

(6) Readings in the upgrade experiment that fall short, and what they mean. The positive readings are in Table 12 of Section 4.9; four points fall short, each explained by asking whether the instrument failed.

(a) **The memory-type control family moves the wrong way** (`def_choice`: masked 0.950 → supplied 0.700, Delta -0.250, pairs 0/15, $p = 6.1 \times 10^{-5}$). The family asks which two candidates conjunctively define a concept, which is not derivable from structure, and the view supplies only the IS-A parent chain – material cannot fix it. Item by item the model is pulled towards **neighbouring entries of the view** and replaces a conjunct it knew correctly with a name that appears in the view (e.g. for `anal fin lepidotrichium` the correct piece `part of some anal fin` is replaced by `median fin lepidotrichium`). This is what the **negative control** is for: a family that structure cannot help is not misreported as "just supply the structure", so the attribution discriminates rather than failing.

(b) **The negation family reads negative under v1** (masked 0.733 → supplied 0.017, compensability -2.687). With the superordinate closure **truncated** in v1, "this candidate is not a superordinate" cannot be confirmed from the view, and the model answers "not a superordinate" where the truth is that it is one – **a truncated closure is worse than none**. Such a failure has to be diagnosed and cannot be fixed by supplying more material; restoring the closure (v2) brings the reading back to 1.000.

(c) **What still fails after the upgrade**: on the hierarchy unit, v2 leaves 17/60 (28.3%) failing; on role, 6/60 (10.0%); on negation, 0/60. For these items the **masked arm is also zero** and the supplied arm's view is confirmed complete (the upgrade artefact carries the required layer), so the residual failure is charged to the **model side** (long-chain execution, incomplete set enumeration) rather than to a gap in the instrument; a small share also comes from the endpoint not being bitwise deterministic (noise floor, (5)).

(d) **Units with too few items**: `granularity` (9 items) and `aggregate` (1 item) in the full-bank per-operator table are listed but support no conclusion (Appendix D.2). Where these readings fall short bears on how much can be explained, not on the direction of Table 12 of Section 4.9: the upgrade rescues only the diagnosed class.

E.14 The four disciplines in full

The four disciplines of this section are an audit of the results of this paper — each corresponds to one experience of misreading an error as a finding. They are written down because this class of error raises no alarm: a scorer marking good answers wrong, an item surface leaking the answer, a difficulty measure refusing to be tested, old items lapsing after an upgrade — in all four cases the result table is indistinguishable from "genuinely hard" or "genuinely effective." (A comparable self-audit appears in (36), which after reporting a large 84% → 99% improvement in a constant domain audited line by line, found 23 of 29 "errors" were extraction artefacts, and withdrew them.)

Difficulty measurement has a separate literature. It is not a "validity" question but a "stratification" question: this paper uses difficulty only as an upstream instrument before attribution, so it states only its three sources and one shared defect. Fixed at item generation — strata defined semantically, not cut afterwards by accuracy (74; 75); estimated after evaluation — item-response-theory work estimates difficulty and discrimination over tens of thousands of items (23), but difficulty is then fixed after seeing the model, so "the harder the item, the lower the accuracy" is not a finding (2); implicit in item structure — hop count is the representative (46), later written as explicit complexity tiers (18; 31), or replaced by evidence span (7) or chain semantic consistency (70), with biomedicine and chemistry following the same organisation (55; 71); yet hop count is only a quantity of the construction process, and multi-hop items can be short-cut (46). The three differ not in novelty but in two orderings: whether difficulty is fixed before observing the model; whether the proxy variable's scale is correct. The second branch fails both; the third passes the first and fails the second. This paper takes the first branch, returning the definition of "stratum" to the ontology: difficulty is written as a component vector at generation time, declares only "not weaker than" on the dominance order, and is frozen before evaluation (end of Section 1.4). The discipline below is the test of this stance on data.

Discipline three, inspect the difficulty measure. Of five candidate quantities, two rounds of univariate testing left three: condition load (paired 11/11, $p = 0.001$) and closure size (after controlling mechanism size, $p \approx 0.0015$) were kept; hop count, candidate pool ($p = 0.375$) and same-level competition were removed. For closure size, p went from 1.0 to 0.0015 once the control was added — without controlling mechanism size, entangled variables cancel the true effect completely. A difficulty measure that refuses to be tested is more dangerous than one that is falsified.

Discipline four, inspect versions. After the ontology was completed in Section 4.5, accuracy on `requires` items fell from 1.000 to 0.750. Cause: their annotation still reflected the state before completion while the view was already rendered from the new ontology — what the model sees does not match what scoring relies on, yet no error is raised; in the result table it appears as "these item types recently got harder." After an ontology upgrade, items whose dependence is invalidated or whose truth value is stale must be regenerated, not quietly kept with a swapped truth value.

This discipline is the origin of the back edge in this paper's pipeline (Figure B.1; `invalidate()` in Section 3.2.6): once knowledge changes, old items split into two classes — still valid and invalidated, the latter requiring regeneration. It is also the only one of the four triggered by this work itself: the first three are problems of the instrument, this one is caused by upgrading one's own knowledge base. A system that upgrades itself must carry a mechanism for retiring old items, otherwise every improvement accumulates errors where they cannot be seen.

Two threats.

One, model coverage. The main readings of Sections 4.2–4.4 all come from gpt-4.1. If only one model shows this pattern, "models generally lack this domain knowledge" is merely a case study. The criterion is $\geq 3/5$ models showing the `RESCUE` cell.

Met: on the stratified cohort (150 items) both arms are measured for 4 models, with Δ positive throughout — `gpt-4o-mini` +0.440, `deepseek-v4.1-flash` +0.380, `gpt-4.1` +0.327,

`jev` (decision view) +0.313; the ≥3/5 criterion is met. An independent 43-item pilot agrees (+0.070 / +0.093 / +0.163).

The two arms of the remaining five models are in Appendix D, with Δ positive throughout; the criterion is met on model coverage.

Two, the count in the `NEG` cell. Of all 1274 items only 3 fall in this cell (0.2%). As a phenomenon it is established ("right without, wrong when given" does occur), but it cannot support any claim about rates; Section 4.4 therefore reports it as a phenomenon only, not in the main conclusions.

Three, the human ceiling is not yet measured. Reading an absolute value (e.g. 0.467 for the reasoning type under the knowledge-supplied arm) as "poor" requires a human ceiling, which we do not yet have (E-HUMAN awaits annotators). Only the absolute scale is affected: the two-tier difference, the ordering of units, and the net gain of the diagnose–upgrade–retest loop are relative quantities that do not depend on a ceiling. The public-ontology comparison is the clearest case – knowledge-masked is near zero while knowledge-supplied is above half, so the conclusion holds however high the ceiling is set. Until the ceiling is measured, absolute readings are stated as *relative comparisons within one instrument and one scoring protocol*.

Readings still unsettled are listed in Appendix E.3.

E.15 Two definitions of the split, and the per-operator accounting of a wrong fill

(i) Two splits draw the same line. The primary split is defined by the instrument (whether the view lists the answer outright); the table below switches to the operator declared before item construction, using only the operator name and never the rendering:

Split	Group	Items	Masked	Supplied	Compensability	95% CI
Instrument-defined (full bank, 1,274)	retrieval type	985	0.171	0.921	0.905	[0.884, 0.925]
Instrument-defined (full bank, 1,274)	reasoning type	289	0.080	0.467	0.432	[0.373, 0.492]
Pre-declared operator (full bank, 1,267)	hierarchy class	890	0.187	0.952	0.941	[0.923, 0.957]
Pre-declared operator (full bank, 1,267)	relation/condition class	377	0.074	0.520	0.481	[0.429, 0.535]
Pre-declared operator (150-item cohort, main model)	hierarchy class	40	0.000	0.625	0.625	[0.475, 0.775]
Pre-declared operator (150-item cohort, main model)	relation/condition class	110	0.300	0.518	0.312	[0.171, 0.432]

The two columns ask different questions (whether the view hands over the answer, versus whether the required operator belongs to the hierarchy), yet within each split the intervals do not overlap – the "high / low class" structure is not an artefact of the instrument's definition.

(ii) Filling edges that should not be filled (per-operator accounting). Before and after the same fill (431 `requires` edges), accounted per operator (decision-protocol model, 1,267 items, one scoring protocol, paired McNemar):

Operator	Before	After	Δ	wrong->right	right->wrong	p
`gap` (the diagnosed target)	0.146	0.500	+0.354	29	0	3.7×10^{-9}
`requires` (undiagnosed)	1.000	0.750	-0.250	0	22	4.8×10^{-7}
`ancestors`	0.985	0.981	-0.004	3	6	0.508
`descendants`	0.718	0.718	0.000	0	0	—
`bind`	0.494	0.468	-0.025	2	4	0.688
`enumerate`	0.494	0.494	0.000	1	1	1.000
`verify_support`	0.385	0.333	-0.051	1	3	0.625
Total	0.8232	0.8232	0.000	—	—	—

Reading: only the diagnosed class improves; filling edges where nothing was diagnosed yields no gain and clearly hurts on `requires` (the stems of that class presuppose that the ontology registers only part of the condition dimensions, and completion removes that premise, so stem and answer no longer match – also the origin of the fourth discipline in Section 4.8). The Section 4.5 conclusion therefore stops at "the instrument can be diagnosis-driven" and does not extend to "a larger knowledge base is better". (The positive/negative/directional

rows of Table 8 come from the main model; this table re-checks them with the decision-protocol model.)

E.16 The proxy annotation round: measured E-HUMAN / E-AGREE and the protocol gap it exposed

No external human expert was available this round, so we ran a proxy round: two proxy annotators (AI proxies, not external human experts) working independently under the protocol "read only the same material as the model, never the gold", on the 87-item worksheet (stratified by operator).

(i) E-HUMAN: a reader with the same material does not supply a human ceiling.

Annotator / model	Coverage	knowledge-supplied accuracy	Answering only
proxy E1	67 items	0.478 (32/67)	0.516 (32/62)
proxy E2	87 items	0.586 (51/87)	0.654 (51/78)
`gpt-4.1` (same items)	87 items	0.552 (48/87)	masked 0.241

The proxies land in the same band as the model, so the criterion "proxy accuracy significantly above the model" is not met. No absolute-scale statement is therefore made anywhere in this paper: the question "is 0.467 poor?" has no reference ceiling in this round (the two-arm difference, the ordering of units and the net gain of the loop are relative and unaffected).

(ii) E-AGREE: below 0.95, but the gap is in the protocol, not in the gold.

Agreement between the independent relabelling and the computed gold is 0.540 (E2, n=87) / 0.478 (E1, n=67), below the 0.95 criterion; inter-rater agreement is 0.821 (55/67) with Cohen's κ 0.817 – the two proxies agree with *each other* yet both disagree with the computed gold, so the disagreement is not about the annotators. Per operator the gap has a clear location:

Operator	n	Relabel vs gold	Reading
`ancestors`	11	1.000	the answer is the chain in the material; recomputable
`gap` / `verify_support`	11 / 11	0.909 / 0.909	material lists all condition dimensions; recomputable
`enumerate`	11	0.727	partly recomputable
`requires`	11	0.636	needs rule-layer condition dimensions; the fragment is insufficient
`descendants`	11	0.273	the answer needs the full-ontology subtree closure; not recomputable from a fragment
`bind` (alternative routes)	11	0.182	the answer needs the full-ontology relation closure
`granularity`	9	0.000	as above

The root cause is the worksheet protocol: the proxies see only the same fragment (1-2 hops) as the model, while for `descendants` / `bind` the answer requires walking the full ontology's subtree or relation closure – and the gold of this paper is exactly what executing the full ontology yields (answerability criterion and anchoring self-check in A.6). The correct conclusion is therefore: the E-AGREE criterion does not apply to a fragment-only worksheet; it must be re-run with the full ontology plus the operator definitions. The round did its job by exposing this protocol gap; the criterion itself stands and is not relaxed because the proxy round failed.

(iii) Next step (reproducible). Re-issue the worksheet with a full-ontology snapshot and the operator definitions, still requiring independent relabelling and independent answering; once external experts are in place, the same script computes agreement and κ. All raw outputs of this round (87 lines of answers and relabels) are shipped alongside.

E.17 Second rebuild of the public-ontology item set: how the ontology and the items were upgraded

The public-ontology comparison of Table 11 (Section 4.9) covers one family, two-hop hierarchy. To extend external validity to other relation families, a second rebuild was carried out on the same public ontologies (ANATOMY, GALEN): first the relations the public ontology **can** supply were upgraded into executable structure, then the **item surfaces** were

upgraded per family. The two are recorded separately, because "upgrade" has an explicit criterion here.

(1) Ontology upgrade: a structure enters the instrument only if it carries five elements.

#	Element	Meaning
1	Stable identifier	the in-ontology name or ID, never a renamed string
2	Natural-language reading	one readable phrase, for item rendering and human review
3	Evidence anchor	which edges of the ontology it comes from (traceable)
4	**Deterministic recomputation interface**	a pure function: item coordinates → truth value
5	Version and hash	hashes of the base source files plus a canonical hash of the upgrade artefact

Missing element 4 does not count as an upgrade: a prose-only addition cannot recompute a truth value. Applied to the public ontologies, three families result:

Family	Structure added	Source (locally available)	Recompute interface	LLM used
Hierarchy (existing)	`child→parent` edges	`train_nf1.jsonl`	k-hop closure	no
Role	`atomic --role→ filler` edges	`train_nf3.jsonl` (GALEN 19,013 / ANATOMY 32,184)	two-step join `A ---r1→ B ---r2→ z`	synonym / inverse clustering
Negation (absence)	complete superordinate closure (flat list)	closure of the above	candidate set - closure	no
Definition (control)	conjunction table	`train_nf2.jsonl`	table lookup (not derivable from structure)	reading generation

The LLM layer proposes only and never judges truth. On the `openlux` endpoint (`gpt-4.1`) three kinds of proposal were elicited: role synonym clustering, same-parent mutual-exclusion candidates, and role readings. Each proposal passes two gates: **self-consistency** (the same request run twice with the candidate order changed, intersection kept) and **structural anchoring** (it must be supported by edges already in the ontology). Result: **3 accepted, 3 rejected, rejection rate 0.50**; a further 20 mutual-exclusion candidates passed only weak verification and are kept for robustness, not for the claims. All three accepted items are **inverse role pairs**: `has part`↔`part of` (ANATOMY, **112** reversed instances), `is conjugate acid of`↔`is conjugate base of` (65), `has branch`↔`is branch of` (GALEN, 49) – and this dataset's `role_inverse.json` **was an empty file**, so the step amounts to filling it. **Protocol correction (important)**: the model labelled these pairs "synonymous", but structural checking shows that **none of them is truly synonymous** (shared concept pairs ≤1, i.e. noise level); every LLM proposal must be re-judged structurally, the label itself is not evidence, and **the rejection rate is itself a diagnostic quantity**.

(2) Item upgrade: the view must supply the material without giving the answer, and must contain distractors. The rendering contract is unchanged – an item must run on `stem` and `evidence` alone, with no change to the pipeline. Two rules came out of rework: (i) the role family's view must **contain distractor edges** (all role edges of the concept plus those of its successors, ≤14 lines); the initial version supplied only the two lines holding the answer (2.9 lines on average) and the model simply read the last line, which measures copying rather than execution; (ii) the negation family's view must be the **complete** closure list (3-40 entries), because with a truncated edge chain "this candidate is not a superordinate" cannot be confirmed from the view and the gold becomes inconsistent with it. Visibility remains two-tier (`none` / `gold`); variants v1 (critical layer removed) and v2 (restored) close the loop.

(3) Measurements: three families, both arms, the loop, and specificity.

Family	Model	n	Masked	Supplied	Δ	Paired b01/b10	p
Role (`role_chain2`)	`gpt-4.1`	120	0.067	**0.925**	**+0.858**	103/0	2.0×10^{-31}
Role (`role_chain2`)	`deepseek-v4.1-flash`	120	0.025	**0.942**	**+0.917**	110/0	1.5×10^{-33}
Negation (`not_ancestor`)	`gpt-4.1`	120	0.725	0.992	+0.267	33/1	4.1×10^{-9}
Negation (`not_ancestor`)	`deepseek-v4.1-flash`	119	0.261	**1.000**	**+0.740**	88/0	6.5×10^{-27}
Mutual exclusion (`disjoint_check`, after the ontology upgrade)	`gpt-4.1`	120	0.350	**1.000**	**+0.650**	—	—
Mutual exclusion (`disjoint_check`, after the ontology upgrade)	`deepseek-v4.1-flash`	120	0.033	**0.967**	**+0.934**	—	—
Definition (`def_choice`, memory-type control)	`gpt-4.1`	60	0.950	0.700	**-0.250**	0/15	6.1×10^{-5}

Table E.12: Two-arm comparison on four families of public ontologies (paired McNemar; masked = no view, supplied = anchored view)

Every upgrade carries two readings, and the tables below are read that way. **Ontology side** G_{onto}: the layer's executable structure from "absent or incomplete" to "present and recomputable"; **model side** G_{model}: the score change on the same items (supplied gain minus

masked gain). The ontology-side gain is unconditional once the structure is recomputable; the model-side gain is conditional on the knowledge not already being in the parameters, and the masked arm is the probe.

Unit	Model	v1 missing (masked→supplied)	v2 after upgrade (masked→supplied)	Net gain from the upgrade	Compensability after (v2)
Hierarchy (`ancestors2`)	`gpt-4.1`	0.000→0.000	0.017→**0.717**	+0.700	0.712
Hierarchy	`deepseek-v4.1-flash`	0.000→0.000	0.000→**0.733**	+0.733	0.733
Hierarchy	`glm-flash`	0.000→0.000	0.000→**0.550**	+0.550	0.550
Role (`role_chain2`)	`gpt-4.1`	0.050→**0.000**	0.050→**0.900**	**+0.900**	**0.895**
Role	`deepseek-v4.1-flash`	0.050→**0.000**	0.017→**0.933**	**+0.967**	**0.932**
Role	`glm-5.3-flash`	0.017→**0.000**	0.033→**0.933**	**+0.917**	**0.931**
Negation (`not_ancestor`)	`gpt-4.1`	0.733→**0.017**	0.783→**1.000**	**+0.933**	**1.000**
Negation	`deepseek-v4.1-flash`	0.150→**0.000**	0.233→**1.000**	**+0.917**	**1.000**
Negation	`glm-5.3-flash`	0.767→**0.000**	0.700→**1.000**	**+1.067**	**1.000**
Mutual exclusion (`disjoint_check`, upgraded this round)	`gpt-4.1`	0.333→**0.950**	0.383→**1.000**	**+0.000** (noise floor)	0.967
Definition (`def_choice`, memory-type control)	`gpt-4.1`	— (not upgraded)	0.950→0.700	**-0.250**	-5.000

Table E.13: Remove the critical layer and put it back (v1 missing / v2 after the upgrade, 60 items per unit)

Three readings. First, with the layer missing, the **supplied** arm also drops to zero (0.000 / 0.017): that layer of structure is a **necessary condition** for the answer, not merely a help. Second, **specificity across families** (same model): role +0.858, negation +0.267, memory-type definition -0.250 – families whose structure is executable improve, the memory-type family does not (it worsens), so the gain is not a uniform "more context" effect. Third, the reversed result of the memory-type family has an identifiable mechanism: item by item, the model is pulled towards **neighbouring entries of the view** and replaces a conjunct it knew correctly with a name that appears in the view (e.g. for `anal fin lepidotrichium` the correct conjunct `part of some anal fin` is replaced by `median fin lepidotrichium`, which appears in the view).

(4) Process readings (the second layer, beyond the answer). On the supplied arm the role family additionally requires a **path** (strict two lines: `Answer:` / `Path:`), and scoring checks, pair by pair, whether that pair is really an edge of the ontology: at n=120, answer accuracy 0.9333, **chain validity 0.8167**, share of cited edges that really exist 0.9111, hop efficiency 0.9889 – **11.7% are answered correctly with a non-executable chain**, which answer-only scores cannot see. (This tier also exposed a scoring defect: with the answer line unchanged, the scorer folded the following `Path:` line into the answer and `split_set` then treated → as a separator, so 31 of 40 items were misjudged – 0.95 fell spuriously to 0.225; truncating at the next field label restores 1.000. Process-style output must be segmented by field before the answer is scored.)

**(5) Items that did not come out positive, and why (to be read on the same page as the numbers). Each is listed with its mechanism rather than left to the reader:

Item	Reading	Mechanism and treatment
Memory-type definition family **Model-side gain of the external-inclusion unit** **Mutual exclusion (`disjoint_check`, Problem Sonder, Problem Ring)**	supplied 0.700 below masked 0.950 (Δ -0.250) +0.000 → +1.000 (see §4.6B, within the noise floor) unfulfilled	**So answer is not derivable from structure**; knowledge masked is already near ceiling (0.950) for memory, and once a view is given the model is pulled towards **neighbouring entries** of that view and replaces a correctly recalled conjunct with a name appearing in it (**`part of some anal fin` → `median fin lepidotrichium`**). Treatment: not a structural family, kept as a counter-example cell. **Not an ontology-side shortfall**: the ontology side closes the gap unconditionally (0.917 → 1.000 in executable pairs); the model-side gain is conditional on the knowledge not already being in the parameters, and the masked arm is the probe. Public ontologies provide no controlled value domains, so it stays with the in-house ontology.

Protocol and limits (to be read on the same page as the numbers). **(i)** Noise floor**: the endpoint is** not bitwise deterministic **at temperature 0. The v1 and v2 items of the negation family have identical surfaces (the variants differ only in the view, which the masked arm does not show); masked arm only: of 60 items, 25 differ in predicted text and** 13 flip verdict (0.217) ⇒ **for n=60, $|\Delta| < 0.2$ supports no conclusion; the -0.250 of the memory-type family in Table E.12 is credible in direction, its magnitude needs replication. (ii)** Strength of LLM proposals**: the 3 accepted items are inverse role pairs (structurally verifiable); mutual-exclusion candidates are robustness only; the synonym and inclusion families cannot be verified at the instance level (measured shared instances: 0) and require the OWL axiom layer – the locally available GALEN / ANATOMY sources contain no `subPropertyOf` / `propertyChainAxiom` (nor does the newly downloaded OAEI anatomy), so those two families remain** unfulfilled**. (iii)** Not covered**: the condition family (public ontologies have no controlled value domains) and the explicit mutual-exclusion family (public ontologies rarely assert `disjointWith`; the newly downloaded ANATOMY `human.owl` contains 17 such axioms) stay with the in-house ontology.

(6) Artefacts (reproducible). Items `reasoning_ds/public_onto/items_role,neg,def.jsonl` and `items_role,neg_v1,2.jsonl`; the upgrade artefact `upgrade_v1.json` (base file hashes, canonical `_frozen` hash, per-entry recomputation interface); per-item logs and summaries

in `reasoning_ds/public_onto/eval2/` (`families_summary.json`, `chain_report.json`); scripts `code/public_onto/build_families,propose_llm,freeze_upgrade,judge_chain,summarize_families.p`

E.18 Unit-level diagnostic vectors and the prediction readings (detail for Section 4.10)

Table E.18a | Unit diagnostic vectors on the in-house ontology (mid-level mechanism families, $n \geq 5$; 31 units, 382 concepts). The two coverages are reported separately: a low `rule_cover` means the **ontology lacks structure**, a low `item_cover` means the **item bank has not asked**; merging them would misdiagnose "no item was written" as "the ontology lacks an edge" (the class distinction of Section 3.1). Gap strength $\sigma = 1 - $ `rule_cover`.

Unit (mid-level mechanism family)	Concepts	With rule	With item	σ	Concepts lacking a rule
`mechanical_and_rheological_response`	18	0	17	1.00	18
`covalent_bond_formation_generic`	13	0	11	1.00	13
`redox_dynamic`	12	0	11	1.00	12
`self_assembly_and_phase_behavior`	10	0	7	1.00	10
`surface_and_matrix_mediated_response`	10	0	9	1.00	10
`coordination_complexation`	9	0	6	1.00	9
`protein_and_biointerface`	8	0	6	1.00	8
`tissue_remodeling`	8	0	5	1.00	8
`wet_adhesion`	15	1	15	0.93	14
`physical_crosslinking`	19	2	17	0.90	17
`cell_adhesion_and_signaling`	25	4	22	0.84	21
`noncovalent_interaction`	21	4	19	0.81	17

(The full 31 rows are in `reasoning_ds/unified/unit_vectors.json`, built by `code/public_onto/build_unit_vectors.py` from `coverage_gap.json` and `requires_audit.json`. Note the skewed distribution of σ: eleven units have zero rule coverage, so the column can name the layer but cannot rank units – reading (ii) of Section 4.10.)

Table E.18b | Pairing audit (repair before taking data)

Family	Shared qids	Same surface	Same gold	Verdict
Hierarchy `ancestors2`	60	60	60	aligned
Role `role_chain2`	60	60	60	aligned
Negation `not_ancestor`	60	60	60	aligned
Mutual exclusion `disjoint_check` (batches A, B)	60 each	60 each	60 each	aligned

The state before the repair (kept on record): v1 and v2 were produced by two independent sampling passes, and for the negation and mutual-exclusion families the number of shared qids with an identical surface was 0/60, so the pairing failed and that batch's Deltas and predictive analysis were void. The fix is **same-item view surgery**: the structure-complete version is the base and v1 is that same item with the family's structure deleted (edge-form families lose the answer lines, closure-form families get a truncated list, declaration-form families lose the declaration lines), leaving surface, candidates and gold untouched; the audit script `code/public_onto/make_v1_aligned.py` exits non-zero if the self-check fails.

Table E.18c | Prediction readings (`gpt-4.1`, public ontologies, same items before and after)

Unit	m	$a(0)$	S_1	S_2	Δ	Headroom $1 - S_1$	$R_{\text{rescue}}(v_1)$
Hierarchy	0.92	0.017	0.000	0.583	+0.583	1.000	-0.017
Role	0.57	0.083	0.017	0.950	+0.933	0.983	-0.073
Negation	1.00	0.783	0.000	1.000	+1.000	1.000	-3.615
Mutual exclusion A	1.00	0.333	0.967	1.000	+0.033	0.033	0.950
Mutual exclusion B	1.00	0.383	1.000	1.000	0.000	0.000	1.000

Rank correlations ($n = 5$, 4,000 permutations, **exploratory only**; no fit is reported):

Predictor	$\rho(\Delta, \cdot)$	Permutation p
Structural gap m	-0.105	1.00
Masked $a(0)$	0.400	0.75
Pre-upgrade supplied S_1	-0.632	0.50
Compensability $R_{\text{rescue}}(v_1)$	-1.000	0.093

Bound check: $\Delta \leq 1 - S_1$ holds in 4/4 rows (the ceiling row stands alone). **Five to nine rows cannot support a fit; the purpose of this table is to give bounds and counterexamples**, not coefficients.

Table E.18d | Same-item dual-format pilot (role family, 52 items, same items and model, `strict`)

Variant	This instrument (set scoring)	Conventional single choice (MCQA)
Structure missing (v1, same-item surgery)	0.019	0.135
Structure complete (v2)	0.942	0.635

Reading: a conventional single-choice form **does** register that structure is missing (0.635 to 0.135), so the main text does not claim that conventional benchmarks are blind to the gap; it claims they carry **no attribution column** – they give neither "which edge" nor "does filling it recover performance" (the two-arm reading and compensability). One side observation: with the structure missing the four-way score of 0.135 is **below** the 0.25 guessing rate, i.e. the model tends to pick names that appeared in the evidence (distractors are drawn from the family's name pool). The renderer is `code/public_onto/make_mcqa.py`.

Artefacts. `reasoning_ds/public_onto/predictive_report_aligned.json` (strictly paired readings) and `predictive_report.json` (old pairing, kept for contrast); scripts `code/public_onto/analyze_predictive,build_unit_vectors,make_v1_aligned,make_mcqa.py`; logs `reasoning_ds/public_onto/eval2/*_v1a_*.json(l)`.

E.19 The material-grounded decidability gate and the tier control, item by item

Table E.19a | MGD by operator (supplied tier `full`, all 1,274 items)

Operator	n	leak	partial	absent	absent share
`ancestors`	780	780	0	0	0.000
`descendants`	110	80	30	0	0.000
`requires`	95	62	33	0	0.000
`gap`	82	0	0	82	1.000
`enumerate`	79	41	38	0	0.000
`bind`	79	64	13	2	0.025
`verify_support`	39	1	1	37	0.949
`granularity`	9	0	0	9	1.000
`aggregate`	1	0	0	1	1.000
Total	**1274**	**1028 (0.807)**	**115 (0.090)**	**131 (0.103)**	

Two readings. First, the dominant operator `ancestors` (780 of 1,274 items) is `leak` in **every** item: its answer is written in the material verbatim, which is exactly why the retrieval-type readings (supplied 0.921) cannot be reported next to the reasoning-type ones (supplied 0.467) as one total. Second, the three operators that fail the gate (`gap`, `verify_support`, `granularity`) are precisely those whose gold needs a **completeness statement** – they are the actionable part of the gate, since each one can be repaired by declaring D^- in the view.

Table E.19b | Tier control audit (per-item means, same 1,183 items)

Tier	Characters	Nodes	Edges
`flat`	279	9.2	0.0
`full`	412	9.4	2.5
`shuffled`	357	9.4	2.5

Node counts are matched across tiers (9.2-9.4); character counts are not (279 / 412 / 357), so the length effect is not isolated by the current four tiers. `flat-irr` (same length, names swapped) is the length-matched arm built for this purpose and has so far been run on a single cohort.

Artefacts. `reasoning_ds/unified/mgd_gate.json` (per-item verdicts and view lengths); script `code/onto/mgd_gate.py`; the same script prints the tier audit.

Appendix F. Per-branch review of related work (omitted from the main text)

This appendix collects the per-branch review: what each level of the chain has achieved, how far it gets and where it stops. The main text keeps the four branches (Sections 2.1–2.4) and a closing positioning paragraph.

F.1 Queries and answers: from having a warrant to being computable

The line of Section 2.1 ends in a "criterion written into a checking procedure"; applied to a concrete benchmark it becomes two decidable questions: which query does this item require; on what basis is the answer judged correct. Reasoning benchmarks in the sciences are this paper's closest same-domain precedent: they already enforce "every item must have warrant," and the checkability of that warrant keeps rising — but what they trace is whether the item is right or wrong, not what the item belongs to.

First kind of warrant: expert judgement. A chemistry capability benchmark builds an eight-task suite and compares against chemists, finding strong performance on item-bank tasks and failure on items requiring genuine reasoning (32); later work extends coverage with graduate-level and professional item banks (78), and another systematically inventories model boundaries on chemistry tasks (65). The warrant is a person, so "what this item measures" is still settled by the item author's wording.

Second kind: an executable program. A chemistry manipulation benchmark decomposes molecular tasks into executable editing operations (42); a reaction-mechanism benchmark requires the proposed transition state to pass quantum-chemical validation (68); a human-calibrated multimodal materials-reasoning benchmark covers 19 subfields (49). The warrant is now a program and an external authority, more checkable than the first; but each item still carries its own dedicated verifier.

Third kind: the deliverable of a whole piece of work. A scientific-workflow benchmark extracts tasks from peer-reviewed papers, has domain experts validate them, and requires working code to be delivered (73); a paper-level benchmark raises the unit to reproduction of an entire work (1); on the general-science side there are surveys of agentic evaluation (10) and scientific-claim verification (38), and the methods side reaches into process — counterfactual perturbation to build step-level signals (26), step-level verification (43). Checkability is highest here (does the deliverable run), but what it answers is whether a whole piece of work was produced.

The answer side follows the same route: from "annotation" to "computation." The benchmarks above take answers from two sources — human annotation (the vast majority) and computation by a deterministic procedure (a minority, growing): a molecular-structure manipulation benchmark hands the answer to a program for exact computation (34); a rule-

guided reasoning benchmark generates items from codified rules, with truth values obtained by executing the rules (61).

Together these three kinds of warrant make the correctness of a benchmark checkable, but what is checkable is only "this item has a warrant," not "which piece of knowledge this item depends on" — the "knowing the conclusion" and "executing the steps" they measure both have clear contrasts, whereas which knowledge this query rests on is not recorded on the item: after a wrong answer one cannot get back to "which piece of knowledge is missing."

Educational measurement has long had a "item x knowledge" matrix (the Q-matrix (52), traditionally specified by experts and hard to maintain as the item bank changes, with later work inferring it from response data (35)) – this paper needs the same matrix, except that it must come automatically from the construction process of the items.

F.2 Knowledge must leave the model: parameters, retrieval, external memory

The landing point of Appendix F.1 is that after a wrong answer one cannot get back to "which piece of knowledge is missing," and the precondition for getting back is that knowledge itself leaves the model and becomes an inspectable object. What this section asks is the next sentence of that precondition: "once out, can the model govern it?" The three branches are ordered by how far knowledge sits from the model (an order this paper adopts for comparison); after ordering them, the question is: does moving it further out amount to making it controllable. (Section 1.2 third point discusses "where to put it," this section discusses "whether it can be governed once out" — the same literature, two questions.)

In the parameters (distance 0). Knowledge is written into weights; correction relies on knowledge editing or continued training. The difficulty is structural — knowledge distributed across parameters can be probed but not taken out and checked item by item (Section 1.1), and successive editing degrades the model (37). Existing work models memory operations (`ADD`/`UPDATE`/`DELETE`/`NOOP`) as a separate training task (64), but what it manipulates is the content of memory, not the visibility of this piece of knowledge.

Retrieved when needed (distance 1: knowledge in the context). Retrieval augmentation brings external documents into the context and lets the model decide for itself whether to look — four routes exist, from self-reflection tokens to propagating uncertainty across layers (30; 76; 67; 51). Knowledge therefore appears in the context temporarily while the substance stays in the parameters: access got closer, storage did not change. What matters more is that "whether to access" is decided by the model itself: this branch answers "whether to retrieve," whereas what this paper examines is "whether this piece of knowledge is visible at this moment." Visibility therefore still is not a controllable variable.

Moved out (distance 2: knowledge outside the model). A small model internalises knowledge and supplies external memory to an arbitrary large model (45); designs also exist that make the ontology a layer outside the model carrying memory, verification and planning (59). A division of labour — small model for memory, large model for reasoning — is emerging, and this is the branch that places knowledge outside the model.

F.3 Knowledge must be given tier by tier: graphs, ontologies, single-variable instruments

The three branches of Appendix F.2 place knowledge ever further out, but moving it out is not the same as controllability. "Whether to take it" is decided by the model, and the model is systematically biased here — it still answers when it should abstain (58; 72); asking it to externalise its uncertainty as a query is therefore an institutional question, not a prompting question. Even when knowledge is genuinely externalised, conflicts remain in both directions between parametric memory and external context, and outdated context can even "corroborate" internal memory and override correct information (44). Knowledge has indeed arrived outside the model, yet nowhere can its visibility be changed. The question therefore turns into: can visibility become an independent variable. This paper and this

branch use the same carrier — both work inside "structured reasoning over a graph"; this section therefore goes further and asks: where is the graph used.

The graph is placed in three positions, and none of them enters scoring. As a retrieval source: graph-QA benchmarks pose multi-hop retrieval and reasoning (39; 77), academic knowledge graphs have dedicated benchmarks (62), and the retrieval side has evidence-grounded typed retrieval (57) — the graph is here an information source to draw on. As the skeleton of a reasoning chain: chain-style prompting improves graph QA (33) — the graph is here something reasoning rests on. As knowledge to be built and validated: binding statements to provenance (40), inducing schemas from literature automatically (47), plus semantic-completeness constraints, relation-learning benchmarks, and roadmaps unifying graphs with large models (48; 63; 50) — the graph is here the knowledge being processed.

The common point: the graph supplies information, supplies paths, is curated by people, but never adjudicates right from wrong — scoring depends only on the answer, and whether the answer is right is fixed by the reference answer given with the item, not by the graph.

Yet the issue with using graphs lies exactly here: whether the graph enters the item. In graph-QA benchmarks a node is only a label on the item surface and edges do not enter aggregation, so scoring depends only on the answer; the ontology of this paper participates directly in item construction — it supplies distractors (same-level concepts) and negative conclusions (disjoint pairs), and the condition of the query therefore enters aggregation. This difference carries two others: once the graph enters the item, right and wrong are adjudicated by the graph (the gold label is its execution result on the operator tree, Sections 2.2, 3.2.2); and since the graph lies outside the model, its visibility becomes controllable (Sections 2.3, 3.2.5).

There is already substantial work on the ontology side, but with a different landing point: the neuro-symbolic line treats symbolic structure as a lever for improving model reasoning (21), one practical route using ontological reasoning to strengthen semantic generalisation and treating symbolic structure as a source of trustworthiness (69); while work advertised as "ontology-guided" reasoning (56) in fact performs retrieval guided by ontology text, then prompt injection — the ontology is there a retrieval corpus, not an executable structure.

F.4 Failure must be attributable: from a whole paper down to which edge

Both branches examined in Appendix F.3 put structure into the model but not into scoring — the structure is for the model, not for evaluation: what the graph produces affects the answer only through the model's own use of it, so failure falls back on the model and cannot be charged to the graph. What this paper asks is a finer step — "which piece of knowledge is missing" (the landing point of Appendix F.1). One branch aims directly at attribution (Section 1.2); what this section examines is how fine its attribution actually is: its approaches form three layers by the unit of attribution — whole paper → factor → stage, and "knowledge" is not the unit at any layer.

First layer, attributed to a whole paper. Recent empirical work classifies existing benchmarks as "knowledge-leaning" or "reasoning-leaning": in the medical domain eleven benchmarks are post hoc split into knowledge-intensive and reasoning-intensive (14); cross-domain comparison examines how much of a model's "thinking" is knowledge versus reasoning (5); other work decomposes memory and reasoning at the level of reasoning paradigms so that each can be observed separately (4). The label lands on properties of the whole paper and on the item author's wording, so what it gives is correlation rather than attribution (end of Section 1.2).

Second layer, attributed to a factor. Another route replaces post hoc classification with controlled design: a fully balanced 2×2 factorial benchmark treats two factors as independent variables simultaneously, separating main effects from interaction (11). This shows that "factorial design for diagnosis" is methodologically accepted; but its unit is the factor, and a factor is still an item-surface property — knowledge itself is not in a switchable position.

Third layer, attributed to a stage. On the retrieval-augmentation side failure is split into "retrieval failure" and "generation failure": work classifying failure modes acknowledges that

attention to the two ends is asymmetric (24), localised repair no longer retries wholesale but first finds the failing stage (15), and the diagnostic-benchmark literature notes that "building a reproducible diagnostic benchmark" and "building an effective diagnostic agent" are two different problems (13). This layer achieves attribution, but the unit stops at the stage: it answers "retrieval or generation," not "which piece of knowledge is missing." (The skeleton of layered attribution has also been discussed systematically (17).)

F.5 Connecting the ontology to attribution: how far existing work gets

Once the three layers are laid out, the problem is one: none of the objects they act on can be switched off individually. The first layer treats knowledge as a label (attached to a whole paper) — the knowledge behind the label cannot be touched; the second treats it as an item-surface property (a factor) — changing that factor changes other items too; the third does not include it (only the two end stages) — those two ends belong to no piece of knowledge. This level stops at "attribution exists, but attribution cannot be verified": one can say "this batch of items leans on knowledge," but not "which piece of knowledge is missing, and would supplying it change the score." The chain breaks here — this is exactly the link this paper takes up (end of Section 2).

Existing work has connected ontologies both to reasoning and to diagnosis, but in both places only half way. At one end, placing the ontology outside the model: systems make it a layer outside the model, carrying memory, verification and planning (59) — the ontology is thus replaceable and updatable, but its role in the system is for the model to use, aimed at raising capability, not at diagnosing failure; making diagnosis itself into a reproducible graph has also been discussed (13); for "whom an incorrect answer should be charged to" there are even dedicated datasets and models (3) — yet their attribution lands on the class of error, still not on the missing piece of knowledge.

The directions agree; what is missing is the same thing: attribution must land on an object that can be switched off individually. Only then can "I say it is this that is missing" be tested — switch it off and the score should change; leave everything else alone and the score should change only here. This is exactly the link this paper takes up (how this paper fills it: end of Section 2).

F.6 Relation to "separating memory from reasoning"

This paper's starting point is: memory should not occupy model parameters; model performance should rest more on reasoning (Section 1.1). Around this claim, this paper offers three checkable pieces of work.

First, externalise the knowledge and keep it controllable. Once knowledge is externalised as a mechanism ontology, it does not merely "exist outside"; it can be supplied tier by tier (Appendix A.6.5). This step is the precondition of every conclusion: knowledge in the parameters cannot be switched off, and external knowledge that is not supplied gives no contrast.

Second, split failure between the two sides. With a single variable in hand, the two results of one item give the attribution (Appendix A.7.2). Among the four measured classes, knowledge gap accounts for 67.0%, still wrong when given 18.0%, and wrong only when given 0.2% — the latter two are not solved by "supplying knowledge."

Third, make iteration a verifiable closed loop. E-COND-1 in Section 4.4 added 431 condition-dimension edges; one item class rose from 0.098 to 0.500 while the rest did not move. The route "measure the gap → fill it → retest and improve" is verified, with a criterion at every step.

But this paper stops on the "measurement" side. As for asking the model to externalise its uncertainty as a query (that step of Appendix F.2), this paper supplies only its precondition: one must first be able to distinguish "lacking knowledge" from "cannot derive" before deciding which side should be externalised. Making it a training objective or a protocol is next work.

A Data for reproduction and the external comparison

The public-ontology comparison uses the ANATOMY and GALEN ontologies from the OnT dataset: items are generated automatically from hierarchy edges (asking for grandparent-level superordinates, so the answer requires two steps over the structure), and the supplied-arm view gives only the one- and two-hop edges, never the answer. The diagnose–upgrade–retest loop withholds the second-hop edges of the target concepts (v1) and then restores them (v2), with every other condition held fixed. Scoring is deterministic; per-model readings are in the comparison table of Appendix E.