# Operations as Primitives: A Compositional Framework for Training Reasoning in Small Language Models

**B. K. Faruki**

*MindHYVE.ai, Inc. — HYVE Labs Research Division, Newport Beach, California*

*Correspondence: bill@mindhyve.ai*

*May 2026*

---

## Abstract

Existing approaches to training reasoning capabilities in language models organize their training curricula around philosophical categories of reasoning — deductive, inductive, abductive, and causal — treating these as the atomic units of cognition to be drilled into the model. We argue that this framing is mistaken at the level of cognitive architecture, and that this mistake imposes a ceiling on the reasoning quality achievable through supervised fine-tuning. Philosophical categories of reasoning are not primitives. They are post-hoc descriptions of complex cognitive episodes that decompose into more fundamental operations.

We propose an alternative framework in which reasoning is trained through twelve discrete cognitive operations — decomposition, premise identification, implication tracing, contradiction detection, evidence weighing, scope determination, temporal sequencing, absence reasoning, analogical mapping, confidence calibration, error recognition, and strategy selection — and we argue that the four classical philosophical reasoning modes emerge naturally as compositions of these primitives. Deductive reasoning, for instance, is the composition of premise identification, implication tracing, and contradiction detection. Inductive reasoning is the composition of evidence weighing, analogical mapping, and confidence calibration. The compositional structure is empirically verifiable but, more importantly, it explains why training on philosophical categories produces brittle reasoning while training on operations produces transferable cognitive capacity.

We further argue that a training curriculum built on operations-as-primitives requires structural diversity along three additional axes that have been under-theorized in the reasoning fine-tuning literature: (1) record-type diversity — training examples must include not only clean reasoning chains but also productive failure sequences, competing-interpretation analyses, strategy-selection cold starts, insufficient-information identifications, and adversarial bias-trap navigations; (2) system-prompt variation — the system prompt must be a meaningful control signal modulated across training records, not a static string that the model learns to ignore; and (3) domain-agnostic problem construction — reasoning operations should be trained on content that exercises the operation without requiring domain knowledge, allowing the trained operations to transfer to arbitrary professional domains at inference time through prompt-level domain alignment or lightweight secondary fine-tuning.

This paper presents the complete architectural framework as a defensive publication. We disclose the twelve operations and their compositional structure, the six record types with their full response formats, the ten

problem domains, the three-tier system prompt architecture, the seven-phase dataset generation pipeline, the LoRA fine-tuning specification targeting the Phi-4 14B model, and the proposed evaluation framework. We protect only the specific seed problems and the verbatim text of system prompt variants used in our own implementation. We do not present empirical results; this paper is a theoretical and architectural contribution intended to establish prior art on the framework and to invite empirical investigation by the broader research community.

**Keywords:** reasoning, fine-tuning, small language models, LoRA, cognitive operations, dataset architecture, defensive publication

---

# 1. Introduction

The capability gap between frontier language models and small language models (SLMs) has narrowed dramatically since 2024, particularly in reasoning-intensive tasks. Microsoft's Phi-4 14B model achieves 80.4% on the MATH competition benchmark and 56.1% on GPQA — outperforming open-source models five times its size on mathematical and scientific reasoning [Microsoft, 2024]. The DeepSeek-R1 distillation experiments demonstrated that supervised fine-tuning on reasoning traces from a frontier model can produce 14B-parameter students that achieve 93.9% on MATH-500, competitive with the frontier teacher [DeepSeek-AI, 2025]. Microsoft Research's Orca papers established that careful curriculum design — what the authors termed "explanation tuning" — could enable 13B-parameter models to match GPT-4 on specific reasoning benchmarks [Mukherjee et al., 2023; Mitra et al., 2023]. These results, taken together, suggest that the architectural ceiling on small-model reasoning is much higher than was assumed in 2023, and that the bottleneck has shifted from model capacity to curriculum design.

This shift creates an opening. If the reasoning ceiling of SLMs is set primarily by the training curriculum rather than the parameter count, then the design of the curriculum becomes the principal lever for performance. The question changes from "how large must the model be to reason well?" to "what must the training data teach the model in order to reason well?" The latter question is empirical, architectural, and — we will argue — under-theorized.

## 1.1 The Implicit Curriculum of Existing Approaches

A survey of reasoning fine-tuning approaches from 2023–2025 reveals a near-universal organizing principle: training data is structured around what we will call the philosophical taxonomy of reasoning. Datasets are partitioned into deductive examples, inductive examples, abductive examples, and causal examples. The implicit assumption is that these are the natural categories of reasoning — the atoms from which all higher-order cognition is built. By drilling the model on these atoms, the reasoning will emerge.

This assumption is rarely articulated explicitly, but it is structurally evident across the field. The four classical reasoning modes are inherited from analytic philosophy, where they have served as descriptive categories for over a century, and they have migrated into AI training pipelines as if they were natural-kind

divisions of cognition itself. We will argue that this inheritance is a category error, and that the category error has consequences for the reasoning quality of the resulting models.

## 1.2 The Operations-as-Primitives Thesis

This paper advances three principal claims:

**Claim 1 (Compositional Primacy):** The philosophical categories of reasoning — deductive, inductive, abductive, and causal — are not cognitive primitives. They are post-hoc descriptions of cognitive episodes that decompose into more fundamental operations. We identify twelve such operations and argue that the four philosophical categories emerge as specific compositions of them. Section 4 develops this argument in detail and provides a compositional table mapping each philosophical category to its constituent operations.

**Claim 2 (Curricular Implication):** A training curriculum organized around philosophical categories produces brittle reasoning because it trains the model to imitate the style of categorized reasoning rather than to execute the underlying operations. A curriculum organized around operations produces more transferable reasoning because it trains the cognitive primitives directly, allowing the model to compose them into philosophical modes — and into novel combinations — as the problem demands.

**Claim 3 (Structural Requirements):** An operations-based curriculum requires structural diversity along three axes that have been under-specified in existing approaches: record-type diversity (six types, including productive failure and insufficient information), system-prompt variation (a three-tier architecture with substantial paraphrastic variation), and domain-agnostic problem construction (problems whose content does not require professional domain knowledge but whose structure exercises operations transferable to professional domains).

## 1.3 Scope and Method of This Paper

This paper is a theoretical and architectural contribution. It does not present empirical results from a trained model. We have implemented the framework described herein and trained a reasoning-specialized model on the resulting dataset, but the empirical evaluation of that model is reserved for a subsequent publication. The purpose of this paper is to establish the framework in the public record and to invite both empirical investigation by other researchers and critical engagement with the theoretical claims.

The paper is structured as follows. Section 2 reviews the relevant prior work on reasoning fine-tuning, identifies the assumptions of the philosophical-taxonomy approach, and locates the gap that operations-as-primitives is meant to fill. Section 3 develops the theoretical case against philosophical categories as cognitive primitives, drawing on cognitive science and AI training literature. Section 4 specifies the twelve operations in detail and presents the compositional table. Sections 5 through 8 develop the curricular architecture: record types, problem domains, system prompts, and the dataset generation pipeline. Section 9 specifies the fine-tuning methodology. Section 10 presents the proposed evaluation framework. Section 11 extends the framework to domain specialization via lightweight secondary fine-tuning. Section 12 discusses limitations, theoretical implications, and open empirical questions.

What we disclose is the complete architectural framework: the twelve operations and their definitions, the six record types with their full response formats, the ten problem domains and their cognitive correspondences, the three-tier system prompt architecture, the seven-phase generation pipeline, the LoRA hyperparameter specification, and the evaluation methodology. What we protect is the specific seed problems used in our own implementation and the verbatim text of our system prompt variants. The protected material is implementation detail; the disclosed material is the architecture itself, which we believe is sufficient for any sufficiently skilled team to reproduce the framework.

## 2. Related Work

The fine-tuning of language models for reasoning has been an area of intense activity since the publication of the Chain-of-Thought prompting paper [Wei et al., 2022], which demonstrated that prompting a model to produce intermediate reasoning steps substantially improves performance on multi-step problems. The subsequent literature can be organized into four lines of work, each of which is relevant to the framework presented here.

### 2.1 Reasoning Trace Distillation

The first line of work treats reasoning fine-tuning as a distillation problem: a strong teacher model (typically a frontier API) generates reasoning traces on a large set of problems, and a smaller student model is fine-tuned to imitate those traces. Microsoft's Orca series [Mukherjee et al., 2023; Mitra et al., 2023] is the canonical exemplar. Orca-1 fine-tuned a 13B-parameter model on ~5M reasoning traces from GPT-4 and ChatGPT, with results matching or exceeding the teacher on specific benchmarks. Orca-2 introduced "cautious reasoning" — teaching the model to select among multiple reasoning strategies (direct answer, step-by-step, recall-then-generate) based on problem type. The DeepSeek-R1 distillation experiments [DeepSeek-AI, 2025] extended this approach by distilling traces from a reinforcement-learning-trained reasoning model into smaller architectures, achieving frontier-competitive performance on mathematical benchmarks at 14B parameters.

The distillation approach demonstrates that reasoning can be transferred from large models to small ones through supervised fine-tuning on traces. What it does not address is the structure of the trace dataset. Distillation papers typically describe the source of the data (which model generated it, on which problems) but spend little space on the architectural design of the dataset — the record types, the operation coverage, the prompt-level variation, the failure modes targeted. The implicit assumption is that high-quality traces from a frontier teacher are sufficient; the dataset architecture beyond "diverse, high-quality" is left unspecified.

### 2.2 Instruction Tuning at Scale

The second line of work, exemplified by FLAN [Wei et al., 2021], FLAN-T5 [Chung et al., 2022], and the Tülu series [Wang et al., 2023], emphasizes task and template diversity. The FLAN papers established that performance improves log-linearly with the number of distinct task types in the training data, and that using

ten prompt templates per task substantially outperforms a single template. This is the line of work from which we draw our argument for system-prompt variation (Section 7).

The instruction-tuning literature has produced the strongest empirical evidence for the importance of diversity in training data — diversity of task type, diversity of phrasing, diversity of format. What it has produced less of is a theoretical account of what kinds of diversity matter most for reasoning specifically. Diversity along the wrong axis is wasted effort. Our framework can be read as a hypothesis about which axes of diversity carry the highest signal for reasoning: operation coverage, record type, and prompt variation along the dimensions specified in Sections 5–7.

### 2.3 Quality-over-Quantity Curricula

The third line of work questions the assumption that more data is better. The LIMA paper [Zhou et al., 2023] demonstrated that a 65B-parameter model fine-tuned on 1,000 carefully curated examples could match or exceed models fine-tuned on hundreds of thousands of mixed-quality examples on instruction-following benchmarks. The Deita work [Liu et al., 2023] extended this finding to ~6,000 examples on more demanding tasks. These results, taken together, establish that careful curation can substitute for raw volume — sometimes by an order of magnitude or more.

The quality-over-quantity literature is the warrant for the size of our proposed dataset (~25,000 examples). We do not believe that 25,000 is a magic number; we believe it is a reasonable upper bound given a generation pipeline that targets high per-example quality, and a reasonable lower bound for covering twelve operations across six record types and ten problem domains with sufficient redundancy for stable training. A team with a more efficient generation pipeline could likely produce comparable results with fewer examples; a team with weaker quality controls would need more.

### 2.4 Synthetic Textbook-Quality Training

The fourth line of work, exemplified by the Phi series [Gunasekar et al., 2023; Li et al., 2023; Abdin et al., 2024], establishes that small models can be competitive with much larger ones when trained on "textbook-quality" synthetic data — data that is dense, pedagogically structured, and free of the noise present in web-scraped corpora. Phi-4 [Abdin et al., 2024] in particular demonstrates that a 14B-parameter model trained on synthetic textbook-quality data can outperform Llama 3.3 70B on mathematical and scientific reasoning benchmarks despite having one-fifth the parameter count.

The Phi series is the warrant for our choice of base model (Section 9). Phi-4's reasoning baseline — established by its pretraining curriculum — provides a starting point from which targeted reasoning fine-tuning yields disproportionate gains. The model is not learning to reason from scratch; it is being refined into a specific reasoning style on top of an already-strong reasoning foundation.

### 2.5 The Gap

Synthesizing across these four lines of work, we identify a gap that our framework is intended to fill. The distillation literature establishes that reasoning can be transferred through supervised fine-tuning but underspecifies the dataset architecture. The instruction-tuning literature establishes that diversity matters

but does not theorize the axes of diversity most relevant for reasoning. The quality-over-quantity literature establishes that curation can substitute for volume but does not specify the curation criteria for reasoning data. The synthetic-data literature establishes that pretraining curriculum matters at scale but addresses pretraining rather than fine-tuning.

What is missing is an architectural theory of the reasoning fine-tuning curriculum: a specification of what the dataset must teach the model in order to produce robust, transferable reasoning, structured at a level of detail that would allow a skilled team to construct an equivalent dataset without simply reusing the original. This paper attempts to provide such a theory.

## 3. The Case Against Philosophical Categories as Cognitive Primitives

The four classical reasoning categories — deductive, inductive, abductive, and causal — are products of analytic philosophy and the philosophy of science. They were developed as descriptive categories for analyzing arguments after they had been made, not as generative categories for producing reasoning in real time. The deductive/inductive distinction traces to Aristotle; the abductive category was introduced by Peirce in the late nineteenth century; causal reasoning has been refined throughout the twentieth century by writers from Hume to Pearl. In each case, the philosophical work is taxonomic: given a piece of completed reasoning, into which category does it fall?

This descriptive use is legitimate and useful. The problem arises when these descriptive categories are imported into AI training as prescriptive templates — when a training dataset is partitioned into "deductive examples" and "inductive examples" with the assumption that drilling on these partitions will produce a model capable of reasoning. We will argue that this importation rests on three confusions.

### 3.1 The Granularity Confusion

The first confusion is granularity. Philosophical categories describe reasoning at a much coarser grain than the cognitive operations that actually produce it. A single piece of deductive reasoning may involve identifying premises, tracing their implications, checking for contradictions, weighing the relative reliability of those premises against each other, and calibrating confidence in the conclusion based on the strength of the premise base. Categorizing the whole episode as "deductive" obscures the cognitive operations that constitute it.

Consider a simple deduction:

> *All members of the committee must be voting citizens. Marcus is a member of the committee. Therefore Marcus is a voting citizen.*

To classify this as "deductive reasoning" is accurate as a label. But the cognitive work involved in producing such a deduction includes: identifying that the first sentence states a universal premise (premise identification), recognizing that the second sentence instantiates the universal (analogical mapping at the level of instance-to-class), tracing the implication of the universal as it applies to the instance (implication tracing), and confirming that no other premise contradicts the conclusion (contradiction detection). Training

a model on the label "deductive" without specifying the underlying operations gives the model no procedural handle on how to produce such reasoning when it encounters a novel problem.

The granularity problem becomes acute when the philosophical category is large and heterogeneous. "Inductive reasoning" encompasses statistical generalization, analogical inference, predictive extrapolation, and pattern recognition — operations that are cognitively distinct and that can succeed or fail independently. A model trained on "inductive examples" without operation-level structure receives a confused signal about what it is supposed to be learning.

### 3.2 The Direction Confusion

The second confusion concerns the direction of inference. Philosophical categories are characterizations of completed reasoning — they describe what kind of argument has been made, after it has been made. Cognitive operations are generative — they describe what the reasoner is doing in the moment of producing reasoning.

When a human expert reasons through a problem, they do not begin by selecting a philosophical category. A physician examining a patient with ambiguous symptoms does not think "I shall now deploy abductive reasoning." She thinks "what would explain this constellation of findings?" — which is an operation (inference to the best explanation, a composition of decomposition and evidence weighing and confidence calibration). The philosophical category is a label that an outside observer could apply to her reasoning afterward. It is not a control signal that she sends to herself in the moment.

Training a model to imitate philosophical-category-labeled reasoning is therefore training the model to imitate the wrong level of abstraction. The model learns to produce text that has the surface form of categorized reasoning ("this is a deductive problem; let us proceed deductively") without acquiring the operational skills that the categorization is meant to summarize. This is a kind of stylistic mimicry that can pass shallow inspection but fails when the model encounters a problem that does not fit cleanly into one of the inherited categories — which is to say, when it encounters most real problems.

### 3.3 The Composition Confusion

The third confusion is that real reasoning episodes almost always involve operations that span multiple philosophical categories. A diagnostic reasoning task may begin with deductive elimination (ruling out conditions whose required symptoms are absent), proceed through abductive inference (selecting the best explanation for the remaining symptom set), incorporate inductive reasoning (drawing on base rates and analogical cases), and conclude with causal reasoning (checking whether the proposed diagnosis is consistent with the patient's history and presentation timeline). Categorizing this episode as belonging to any single philosophical category is arbitrary.

If the training data is partitioned into categorical buckets, the model never sees naturally composed reasoning episodes — it sees pure exemplars of each category, which are pedagogically clean but cognitively rare. The resulting model can produce category-pure reasoning when prompted but struggles when problems require fluid composition across categories. This failure mode is well-documented in

deployment: models that perform well on benchmark suites designed around clean categorical examples often fail on real-world reasoning tasks where the categorical lines are blurred.

### 3.4 What Goes Wrong in Practice

The three confusions converge on a single failure mode in fine-tuned models: brittleness. A model trained on philosophical-category-labeled data learns to produce reasoning that looks like the category it has been prompted into but lacks the operational substrate that would make the reasoning robust. Such a model will produce confident-sounding deductive prose on problems that are not actually deductive; it will produce inductive-shaped responses on problems where the inductive structure is illusory; it will fail to recognize when a problem requires a composition of operations that does not map onto any single philosophical category.

The operations-as-primitives framework is intended to address this failure mode directly. By training the cognitive operations at the level at which they actually occur — and by training their composition through record-type and domain diversity — we aim to produce models that can deploy whatever combination of operations a problem demands, without being constrained by the categorical boxes that philosophy has bequeathed to AI training pipelines.

## 4. The Twelve Cognitive Operations

We propose twelve cognitive operations as the primitives of reasoning. The operations were selected to satisfy three criteria: (1) each operation can be specified and trained independently; (2) each operation is exercised across multiple professional domains, so that training the operation on domain-agnostic content transfers to domain-specific reasoning; (3) the operations compose to produce the four classical philosophical reasoning modes plus combinations that do not map cleanly onto any single classical mode.

The twelve operations were derived through a process of cognitive task analysis on the reasoning demands of eleven distinct professional domains (general analysis, Islamic jurisprudence, education, healthcare, law, finance, marketing, retail strategy, real estate analysis, insurance underwriting, and technology engineering). The full domain analysis is implementation detail and is not reproduced here. What follows is the operation specification.

### 4.1 Operation Definitions

**Operation 1 — Decomposition.** Breaking a complex problem into well-defined sub-problems that can be addressed independently. Trains the model to recognize when a problem is too large to be solved monolithically and to construct a sub-problem decomposition that covers the original problem without gaps or overlaps. Exercised whenever a problem has internal structure that can be separated into components.

**Operation 2 — Premise Identification.** Distinguishing what has been given in the problem statement from what has been assumed by the reasoner, and distinguishing both from what remains genuinely unknown. Trains the model to be explicit about its premise base — to state what it is taking as given, to flag what it is assuming, and to mark what it does not know. The foundation operation for any rigorous reasoning chain.

**Operation 3 — Implication Tracing.** Following claims forward to their necessary consequences. Given premise p, what follows? Given premise p and rule r, what does p combined with r entail? Trains the model to extend the implications of stated and assumed premises through one or more inferential steps without losing track of the chain.

**Operation 4 — Contradiction Detection.** Recognizing when two claims, or a claim and an observation, cannot both hold. Trains the model to identify logical incompatibility — not merely surface disagreement but actual contradiction that requires one of the contradicting elements to be revised. A precondition for productive failure recovery (Section 5).

**Operation 5 — Evidence Weighing.** Assessing the relative strength of competing pieces of evidence. Direct observation vs. secondhand report; corroborated finding vs. isolated claim; recent measurement vs. dated estimate. Trains the model to articulate why one piece of evidence is stronger than another rather than silently privileging one source over another.

**Operation 6 — Scope Determination.** Deciding whether a stated rule, principle, or precedent applies to a specific case. Includes recognizing when a case falls outside the scope of a rule despite surface similarity, and when a case falls inside the scope despite surface dissimilarity. The operation underlying statutory interpretation in law, takhṣīṣ al-ʿāmm in Islamic jurisprudence, and any application of general principles to specific instances.

**Operation 7 — Temporal Sequencing.** Reasoning about the order in which events occurred, rules were enacted, or evidence was produced, and how that order changes the analysis. Includes recognizing when temporal order is decision-relevant (e.g., a later statute supersedes an earlier one) and when it is decision-irrelevant.

**Operation 8 — Absence Reasoning.** Drawing inferences from what is not present. The dog that did not bark in the night. The control experiment that produced no effect. The patient symptom that, if present, would indicate one condition; whose absence supports another. Trains the model to attend to negative evidence — a class of evidence that is systematically under-attended in naive reasoning.

**Operation 9 — Analogical Mapping.** Identifying structural similarity between cases that differ in surface features. Trains the model to extract the structural skeleton of a case and to map it onto another case with the same skeleton but different content. The operation underlying case-based reasoning in law, qiyās in Islamic jurisprudence, and pattern recognition across domains.

**Operation 10 — Confidence Calibration.** Honestly assessing how certain one is, and why. Trains the model to state confidence levels in proportion to the actual strength of the underlying reasoning chain, including the recognition that some conclusions deserve high confidence and others deserve graduated uncertainty. A direct counterweight to the hallucination tendency, which is a calibration failure.

**Operation 11 — Error Recognition.** Catching one's own mistakes mid-reasoning. The signal is typically a contradiction, an absurdity, or a conclusion that contradicts a known constraint. Trains the model to monitor its own reasoning chain for these signals and to interrupt itself when one fires, rather than pushing through the inconsistency.

**Operation 12 — Strategy Selection.** Choosing which operations to deploy on a given problem before beginning the reasoning. The metacognitive operation. Trains the model to diagnose the problem type, state its analytical approach, and commit to that approach — and to switch approaches when the chosen strategy proves inadequate. The hardest operation to train and the one that most distinguishes expert reasoning from competent reasoning.

## 4.2 The Compositional Table

The four classical philosophical reasoning modes correspond to specific compositions of the twelve operations. Table 1 presents the decomposition.

**Table** 1. Philosophical reasoning modes as compositions of cognitive operations.

| Philosophical Mode | Constituent Operations | Composition Pattern |
|---|---|---|
| Deductive | 2, 3, 4 | Identify premises → trace implications → verify no contradictions |
| Inductive | 5, 9, 10 | Weigh evidence across cases → map analogically → calibrate confidence in generalization |
| Abductive | 1, 5, 10 | Decompose observation into components → weigh candidate explanations → calibrate confidence in best explanation |
| Causal | 3, 7, 9 | Trace implication of putative cause → verify temporal precedence → map to analogous cause-effect cases |

This decomposition is not the only possible one — some operations contribute to multiple modes, and a more elaborate model of inductive reasoning would invoke Operation 8 (absence reasoning) in its consideration of disconfirming evidence. The table is intended as an illustration of the compositional structure rather than as a closed taxonomy. The key claim is that the philosophical modes can be reconstructed from the operations, while the operations cannot be reconstructed from the philosophical modes. This asymmetry is the technical content of the operations-as-primitives thesis.

## 4.3 Why Twelve, and Why These Twelve

The number twelve is not theoretically privileged. It emerged from the cognitive task analysis as the smallest set of operations sufficient to (a) reconstruct the four classical modes, (b) cover the reasoning demands of all eleven professional domains we analyzed, and (c) maintain enough distinctness between operations that each could be trained independently. A smaller set (six or eight operations) collapsed operations that we believed should be trained separately. A larger set (sixteen or twenty operations) produced operations that were compositions of more basic ones (e.g., "prioritization" is decomposition + evidence weighing applied to competing demands; "synthesis" is the inverse of decomposition and is learned implicitly when the model recombines sub-problem conclusions).

We do not claim that this particular set of twelve operations is the unique correct decomposition. We claim that some operation-level decomposition is correct, and that this particular decomposition is one workable

instance of it. Other research groups may arrive at different decompositions — with fourteen operations, or ten, or a different selection of twelve. The operations-as-primitives thesis is compatible with any such decomposition. What it is not compatible with is the philosophical-categories framing, in which the four classical modes are taken as the atomic units of training.

### 4.4 Frequency Allocation Across the Curriculum

In our implementation, the curriculum allocates training examples across operations roughly uniformly, with weighting toward the operations that we judged most broadly transferable. Table 2 presents the target frequency distribution for a 25,000-example dataset.

**Table** 2. Target operation coverage across a 25,000-example curriculum. Each training example exercises 2–4 operations simultaneously; the table reports the number of examples in which each operation appears as a primary or secondary contributor.

| Operation | Target Frequency | Primary Domain Anchors |
|---|---|---|
| Decomposition | ~5,000 | Resource allocation, multi-source conflict |
| Premise Identification | ~5,500 | Rule application, classification |
| Implication Tracing | ~4,500 | Rule application, causal reasoning |
| Contradiction Detection | ~3,500 | Multi-source conflict, productive failure |
| Evidence Weighing | ~5,500 | Multi-source conflict, diagnostic reasoning |
| Scope Determination | ~4,000 | Rule application, classification, boundary problems |
| Temporal Sequencing | ~3,000 | Temporal/sequential, causal vs. correlational |
| Absence Reasoning | ~3,000 | Diagnostic, incomplete data |
| Analogical Mapping | ~3,500 | Causal reasoning, projection and scenario modeling |
| Confidence Calibration | ~5,000 | Competing interpretations, incomplete data |
| Error Recognition | ~4,500 | Productive failure (primary), adversarial traps |
| Strategy Selection | ~4,000 | Cold-start records, mixed-mode problems |

Frequencies sum to more than 25,000 because each example exercises multiple operations. The minimum frequency of ~3,000 per operation ensures that no operation is undertrained.

### 4.5 Operations Are Trained Through Records, Not Labeled in Records

A critical point about implementation: the training records do not label which operations they exercise. The operation taxonomy is a tool for curriculum construction — it tells the curriculum designer what operations need to be covered, in what frequencies, across what record types and domains. But the records themselves

are not annotated with operation labels. The model learns the operations through repeated exposure to reasoning chains that exercise them, not through explicit operation labels that it would have to memorize.

This is a deliberate design choice. Explicit operation labels in training data would teach the model to produce surface tokens like "DEDUCTIVE REASONING:" or "[OPERATION 6: SCOPE DETERMINATION]" — pattern-matching artifacts that produce categorized prose without operational substance. We want the operations themselves to be encoded in the model's weights, not the names of the operations to be encoded in the model's output vocabulary. The taxonomy is for us; the reasoning is for the model.

## 5. The Six Record Types

A reasoning curriculum must train more than the operations themselves. It must train the cognitive behaviors that surround and modulate the operations: how to recover from a wrong start, how to handle genuine ambiguity, how to recognize when information is insufficient, how to resist cognitive bias. These behaviors are not operations in our taxonomy; they are dispositions that govern how operations are deployed. They are best trained through structural variation in the record format.

Existing fine-tuning datasets predominantly use a single record type: problem → reasoning chain → answer. This is the analogue of teaching a driving student by showing only successful drives. Real reasoning, like real driving, frequently involves wrong turns, ambiguous signals, missing information, and adversarial conditions. A model trained exclusively on clean reasoning has no learned behavior for any of these.

We propose six record types. Each type targets a distinct cognitive behavior. The distribution across types is calibrated to produce a model that handles not only the common case (clean reasoning) but also the failure modes and edge cases that distinguish robust reasoning from brittle reasoning.

**Table** 3. The six record types of the reasoning curriculum, with target distribution.

| # | Record Type | Count | Share | Cognitive Behavior Trained |
|---|---|---|---|---|
| 1 | Clean Multi-Operation Solve | 7,500 | 30% | Standard reasoning with decision-point transparency |
| 2 | Productive Failure + Recovery | 6,000 | 24% | Error recognition and recovery |
| 3 | Competing Interpretations | 4,000 | 16% | Calibrated handling of genuine ambiguity |
| 4 | Strategy Selection / Cold Start | 3,500 | 14% | Metacognitive strategy diagnosis |
| 5 | Insufficient Information | 2,000 | 8% | Recognition of unanswerable questions |
| 6 | Adversarial Traps | 2,000 | 8% | Resistance to cognitive bias |
| | Total | 25,000 | 100% | |

The remainder of this section specifies the response format for each type.

### 5.1 Type 1: Clean Multi-Operation Solve

The standard record type, but with one structural requirement that distinguishes it from existing approaches: each reasoning step must include an explicit decision point. The reasoning chain does not merely state successive conclusions; it identifies, at each step, what alternatives were considered and why the chosen alternative was selected.

Response format:

> *Opening: Brief framing of what kind of problem this is and what will be attempted.*
>
> *Premise statement: Explicit identification of what has been given, what is being assumed, and what is genuinely unknown.*
>
> *Step 1: First reasoning move, with explicit decision: "I could approach this as X or Y. X because [reason]. Y would require [condition] which is not present. Proceeding with X."*
>
> *Steps 2–N: Each step builds on the previous and includes its own decision point. The chain is a sequence of choices, not a sequence of statements.*
>
> *Conclusion: Clear statement of the answer.*
>
> *Verification: Quick check — does the conclusion satisfy all the constraints stated in the problem? If yes, the chain is complete. If no, return to the failure point.*

The decision-point requirement is the principal departure from existing clean-solve formats. Most reasoning datasets show steps without showing why each step was chosen over alternatives. The result is reasoning that looks deliberate but is actually scripted: the model has learned to produce a sequence of steps that lead to the answer, not to make a sequence of choices among alternatives. The decision-point format trains the latter.

### 5.2 Type 2: Productive Failure + Recovery

The most important record type in the curriculum, and the one almost entirely absent from existing reasoning datasets. The assistant begins with a plausible but incorrect reasoning approach, proceeds for two or three steps, encounters a contradiction or absurdity, explicitly identifies the failure, backtracks to the source of the error, and re-solves the problem correctly from that point forward. The record concludes with a brief reflection on what went wrong, what signal indicated the failure, and what the failure teaches.

Response format:

> *Opening: Initial approach to the problem.*
>
> *Step 1: Reasonable starting move based on a plausible reading of the problem.*
>
> *Step 2: Continuation of the initial approach.*

*Step 3: "Wait — something does not hold. If my Step 1 assumption were correct, then X would have to follow. But the problem tells us Y, which contradicts X."*

*Recovery framing: "So my approach from Step 1 is not working. The specific issue is [identification]. Let me try a different angle."*

*Step 1b: New approach, informed by what the failure revealed.*

*Steps 2b–Nb: Correct reasoning chain.*

*Conclusion and verification.*

*Reflection: "What went wrong here was [specific error]. The signal that indicated the error was [specific signal]. The lesson is [generalizable principle]."*

This record type trains Operation 11 (error recognition) and Operation 4 (contradiction detection) in their most ecologically valid setting. A model that has never practiced failure during training has no learned behavior for handling failure during inference; it will push through contradictions rather than catching them. By including productive failure as a substantial fraction of the training data (24% in our specification), the model acquires failure-handling as a default behavior rather than a rare emergent capability.

A critical design constraint: the initial wrong approach in a productive failure record must be plausibly wrong, not obviously wrong. The reasoning that leads into the failure should be the kind of reasoning a competent reasoner might actually produce on first reading the problem. If the initial approach is transparently absurd, the model learns to recognize transparent absurdity — which is uninteresting. The signal we want the model to learn is the recognition of non-obvious failures, the kind that only become visible after two or three steps of careful pursuit.

### 5.3 Type 3: Competing Interpretations

Problems that genuinely admit more than one defensible interpretation. The assistant identifies both interpretations, reasons through each separately, evaluates which is stronger and why, and concludes with a graduated confidence statement that acknowledges the interpretive ambiguity without collapsing it into false certainty.

Response format:

*Opening: "This problem is genuinely ambiguous, and I want to be transparent about that. There are at least two defensible readings."*

*Reading A: Statement of the first interpretation. Reasoning chain following Reading A to its conclusion.*

*Reading B: Statement of the second interpretation. Reasoning chain following Reading B to its conclusion.*

*Evaluation: Comparative assessment. "Reading A is more supportable because [specific reasons]. Reading B cannot be dismissed because [specific reasons]. If [specific*

*additional information] were available, it would resolve the ambiguity in favor of one reading."*

*Graduated conclusion: "On balance, Reading A is more likely correct, with the qualification that Reading B remains defensible. My confidence in Reading A is [stated level]; my reasons for not being more certain are [stated]."*

This record type trains Operation 10 (confidence calibration) in its most demanding form. Every existing reasoning dataset implicitly rewards convergence — the model receives a positive signal when its reasoning produces a single confident answer. Real professional reasoning frequently does not converge. A model that always converges produces hallucinated certainty when the underlying analysis warrants uncertainty. By training the model on examples in which the correct response is graduated confidence, we counteract the convergence bias.

### 5.4 Type 4: Strategy Selection / Cold Start

The system prompt for these records does not specify which reasoning approach to use. The problem is presented cold. The assistant's first move is to diagnose the problem type — to recognize what kind of reasoning the problem demands, state the diagnosis explicitly, justify it, and then execute the chosen strategy.

Response format:

*Diagnosis: "This problem requires [specific reasoning approach] because [specific features of the problem]. The principal operation I will need is [operation]; secondary operations are [operations]."*

*Strategy commitment: "My approach will be [statement of approach]. Alternative approaches I considered are [briefly named] and rejected because [specific reasons]."*

*Execution: The reasoning chain proceeds.*

*Mid-course correction if necessary: "My initial diagnosis was [original]. Two steps in, I am finding that [observation]. This suggests the problem is actually [revised diagnosis]. Switching to [revised approach]."*

*Conclusion and verification.*

This record type trains Operation 12 (strategy selection) — the metacognitive operation. The model learns that the choice of reasoning approach is itself a reasoning move that must be justified, and that the chosen approach is revisable if it proves inadequate. This is the distinction between a model that can execute a specified reasoning approach when told to and a model that can recognize what approach to take when no specification is provided.

### 5.5 Type 5: Insufficient Information

Problems where the correct answer is not an answer. The correct response is a precise identification of what information is missing, why it matters, and how the conclusion would vary depending on the missing data.

The assistant still reasons through everything that can be determined from the available information, demonstrating that the response is an analytical assessment rather than a refusal.

Response format:

> *Opening: Statement of what the problem appears to be asking.*
>
> *Analysis of available information: Reasoning through what can be determined from what has been given.*
>
> *Identification of the gap: "To produce a definitive answer, I would need [specific information]. This information is not present in the problem as stated."*
>
> *Conditional analysis: "If the missing information were [option A], then the conclusion would be [conclusion A], because [reasoning]. If the missing information were [option B], then the conclusion would be [conclusion B], because [reasoning]."*
>
> *Honest acknowledgment: "Without the missing information, I cannot give a definitive answer. The most useful thing I can do is identify what I would need to know."*

This record type is, in our framework, the principal countermeasure to hallucination. Hallucination is not fundamentally a knowledge deficit; it is a behavior deficit. A model that has been trained exclusively on records where the correct response is an answer has no learned behavior for the response "the information is insufficient." When such a model encounters insufficient information at inference, it produces an answer anyway — because producing an answer is the only behavior it has been rewarded for. By including ~8% of the training data as insufficient-information records, the model learns the behavior of declining to answer when an answer is not warranted, while still demonstrating analytical engagement with the available material.

### 5.6 Type 6: Adversarial Traps

Problems deliberately designed to trigger specific cognitive biases. The trained biases include: confirmation bias (evidence that appears to support an intuitive answer but in fact does not), base rate neglect (problems that invite the reasoner to ignore relevant prior probabilities), correlation-causation confusion, survivorship bias, anchoring effects, false-dilemma framing, and problems where the obvious reasoning principle is the wrong one for the case at hand.

The response may take either of two forms. In the first form, the assistant initially falls for the trap, then catches it (combining error recognition with bias identification). In the second form, the assistant identifies the trap immediately and explains why the intuitive answer is wrong. Both forms are valid training signals; the key requirement is that the specific cognitive bias is named and the correct reasoning is demonstrated.

Response format (first form, fall-and-catch):

> *Initial response: Plausible-but-biased reasoning that arrives at the intuitive (wrong) answer.*

*Catch: "Wait — I should examine this more carefully. The intuitive answer rests on [specific assumption]. Let me check whether that assumption is warranted."*

*Bias identification: "The pattern I just exhibited is [named bias]. It works as follows: [explanation of the bias mechanism]. The trap in this problem is [specific trap]."*

*Corrected reasoning: Re-solving the problem without falling for the bias.*

*Conclusion.*

The adversarial-trap record type trains a meta-skill that is not captured by any single operation: the disposition to be suspicious of one's own intuitive responses on problems that present surface features known to be associated with cognitive traps. This is a learned vigilance, and like all learned vigilances, it requires repeated exposure to develop.

### 5.7 Why These Six, and Why These Proportions

The six record types are not exhaustive. Other behaviors could be trained through additional record types — for instance, a "reasoning under time pressure" record type, or a "reasoning with adversarial interlocutor" record type. We restricted the curriculum to six types because each additional type dilutes the training signal for the others, and because the six chosen types cover the cognitive behaviors we judged most important for robust general reasoning.

The proportions (30/24/16/14/8/8) were chosen on the following rationale. Clean Multi-Operation Solve (30%) is the largest type because it carries the base load of operation training; the model needs many examples of each operation deployed in standard contexts. Productive Failure (24%) is the second-largest type because error recovery is the highest-value cognitive behavior beyond clean reasoning itself — a model that recovers from errors is dramatically more useful than one that does not. Competing Interpretations (16%) is sized to provide enough exposure to ambiguity that the model develops calibrated confidence as a default disposition. Strategy Selection (14%) is sized to provide enough cold-start exposure that metacognitive diagnosis becomes a learned behavior. Insufficient Information (8%) and Adversarial Traps (8%) are the smallest types because they target specific failure modes rather than broad behaviors; 2,000 examples of each is sufficient to teach the relevant pattern without crowding out the broader curriculum.

These proportions are recommendations, not theoretical optima. A research group with different priorities might rebalance — for instance, increasing the Adversarial Trap proportion to 15% if cognitive-bias resistance is a primary concern, or reducing Competing Interpretations to 10% if the target deployment domain has less inherent ambiguity. The framework is robust to such adjustments. What is not adjustable is the inclusion of all six types: omitting Productive Failure produces a model with no error recovery; omitting Insufficient Information produces a hallucinating model; omitting Competing Interpretations produces a model with miscalibrated confidence. Each type targets a behavior that cannot be acquired through any of the others.

## 6. The Ten Problem Domains

The problem domains define the content of training examples — what the problems are about. They are not professional domains. A training example in the domain "Resource Allocation Under Constraints" is not a finance problem or an operations-management problem; it is a structured scenario about allocating limited resources among competing demands, framed in domain-neutral surface content (factories, schools, teams, budgets) that exercises the cognitive operations of resource-constrained reasoning without requiring any specific professional knowledge.

This domain neutrality is a deliberate architectural choice that enables the curriculum to be domain-agnostic at training time and domain-specific at inference time. The reasoning operations trained on a problem about scheduling factory shifts are the same operations required for scheduling hospital staff, classroom rotations, or court dockets. By exercising the operations on neutral content during training, the operations become transferable to any domain at inference, provided the inference-time context supplies the domain-specific vocabulary and constraints.

The ten domains were selected through structural analysis of the cognitive demands of eleven professional fields. Each domain corresponds to a class of cognitive challenge that recurs across multiple professional fields. Table 4 specifies the domains and their cross-domain correspondences.

**Table** 4. The ten problem domains of the reasoning curriculum.

| # | Domain | Cognitive Challenge | Primary Ops |
|---|---|---|---|
| 1 | Rule Application & Exceptions | When does a general rule apply, and when does an exception override? | 2, 3, 6 |
| 2 | Multi-Source Conflict Resolution | Two or more sources disagree; which is correct and why? | 4, 5, 10 |
| 3 | Diagnostic Reasoning | Identifying an underlying cause from a constellation of observations | 1, 5, 8, 10 |
| 4 | Resource Allocation Under Constraints | Distributing limited resources among competing demands | 1, 3, 5 |
| 5 | Temporal & Sequential | The order of events changes the analysis | 3, 7, 9 |
| 6 | Incomplete Data Decision-Making | Reasoning under uncertainty when information is missing | 8, 10, 11 |
| 7 | Causal vs. Correlational | Distinguishing genuine cause from mere co-occurrence | 3, 7, 9 |
| 8 | Stakeholder Perspective Analysis | Multiple parties have different interests, evidence, and interpretations | 5, 9, 12 |
| 9 | Classification & Boundary | Does this case fall inside or outside a category? | 6, 9, 11 |
| 10 | Projection & Scenario Modeling | If conditions change, what follows? | 3, 7, 9 |

Each domain appears across all six record types. A Diagnostic Reasoning problem might appear as a Clean Solve (Type 1) in one record, a Productive Failure (Type 2) in another, an Insufficient Information case

(Type 5) in another. The combinatorial structure (six record types × ten domains × twelve operations × forty-four system prompt variants) provides the diversity space within which the training examples are constructed.

The distribution across domains is approximately uniform — ~2,500 examples per domain — but is not rigidly enforced. Some domains are richer in certain record types than others; Incomplete Data Decision-Making is the natural anchor for Insufficient Information records, for instance, while Diagnostic Reasoning is the natural anchor for Productive Failure records. The frequency targets in Table 4 are intended as floors, not ceilings; any given domain may appear in somewhat more or fewer examples depending on the natural fit between the domain and the various record types.

### 6.1 What the Domain Selection is Optimizing For

The ten domains were selected to satisfy three constraints. First, the union of the ten domains must cover the cognitive challenges that arise in professional reasoning across multiple fields. A domain that exercises a cognitive challenge present in only one field would not earn its place. Second, the domains must be expressible in domain-neutral surface content. A domain that fundamentally requires professional knowledge to even formulate the problem (e.g., "interpreting electrocardiogram patterns") would violate the domain-agnostic principle. Third, the domains must collectively exercise all twelve cognitive operations with sufficient frequency. A domain set that left some operations under-exercised would produce a model with weak deployment of those operations.

The resulting ten domains are not the only possible selection. A research group with a different professional focus might emphasize different domains (e.g., "Negotiation Under Conflict" for legal or commercial deployments, or "Pedagogical Sequencing" for educational deployments). The framework accommodates such substitutions. What matters is that whatever domain set is chosen satisfies the three constraints above.

### 6.2 Why Domain Neutrality Matters for Transfer

The transfer claim — that reasoning trained on domain-neutral problems generalizes to professional domains at inference — depends on the operations themselves being domain-invariant. We believe this claim is defensible for two reasons.

First, the cognitive task analyses we conducted across eleven professional fields revealed that the operations recurred across fields with high consistency, even when the content of the reasoning was radically different. A physician weighing differential diagnoses, a lawyer weighing precedents, and an actuary weighing risk factors are deploying the same Operation 5 (evidence weighing) on structurally analogous inputs. The vocabulary differs; the cognitive move does not.

Second, the system-prompt-mediated transfer mechanism (see Section 7) allows domain alignment to occur at inference time without requiring the operations to be retrained for each domain. A model that has been trained on domain-neutral Evidence Weighing problems can, when prompted with a clinical system prompt, deploy the same operation on clinical evidence — provided the clinical vocabulary and concepts are within

the model's general-knowledge base. The domain-neutral training establishes the capacity; the inference-time system prompt directs the application.

This mechanism has limits. A model trained only on domain-neutral problems will not acquire the substantive knowledge of a professional field — it will not learn pharmacology from clinical-shaped problems unless pharmacology is in its pretraining data. Domain-specific knowledge must be supplied either through the model's general-purpose pretraining, through retrieval-augmented generation at inference, or through a lightweight domain-specific fine-tuning layer applied on top of the domain-agnostic base. Section 11 develops the third option.

## 7. The System Prompt Architecture

The system prompt in a fine-tuning corpus performs a function distinct from its content. Considered as content, the system prompt is a description of the assistant's role, persona, and behavioral constraints. Considered as a training signal, the system prompt is a vector in the model's input space that gradient descent will either learn to attend to or learn to ignore, depending on whether the vector carries information that helps the model produce better assistant responses.

This dual function has under-appreciated consequences for the design of fine-tuning datasets. If the system prompt is held constant across all training records — for instance, set to the string "You are a helpful assistant" for every example — then the system prompt vector carries zero discriminative information across the training set. Gradient descent will encode this fact: the model learns that the system prompt is noise and that the relevant signal lies entirely in the user message. At inference, when the deployed model is provided with a different system prompt that is meant to control its behavior, the model will not attend to it. The system prompt has been trained into irrelevance.

This failure mode is widely observed in practitioner reports but is rarely addressed at the architectural level. The conventional advice — "use a meaningful system prompt" — is necessary but not sufficient. A single meaningful system prompt repeated across all records is still constant across the training set and will still be trained into irrelevance. The system prompt must vary across records in a way that correlates with the assistant response, so that the model learns the system prompt is a control signal whose content modulates the behavior the model should produce.

The corollary is that the system prompt architecture must be designed as deliberately as the record types and the problem domains. We propose a three-tier system prompt architecture comprising forty-four variants distributed across the curriculum in specific proportions.

### 7.1 Tier 1: Operation-Specific Prompts (36 variants)

Twelve base prompts — one per cognitive operation — each instantiated in three paraphrastic variants. Each operation-specific prompt describes the cognitive behavior the operation entails, in natural language, without using formal terminology from cognitive science or philosophy of mind. The intent is to teach the model to enact the operation, not to teach it to recognize labels for the operation.

We provide one illustrative example here. The full set of forty-four system prompt variants is part of the implementation detail that we have elected to protect; the structural specification is what is disclosed.

**Example — Operation 5 (Evidence Weighing), Variant 1 (illustrative):**

> *"You are an analytical reasoner. When you encounter multiple pieces of evidence, assess each one individually before drawing conclusions. Consider how direct the evidence is, whether it is firsthand observation or secondhand reporting, and whether other evidence corroborates or contradicts it. When evidence conflicts, resolve the conflict explicitly — state which evidence you find more reliable and give specific reasons. Do not silently ignore evidence that does not fit your emerging conclusion."*

Three paraphrastic variants of each operation-specific prompt are constructed. The variants differ in phrasing, sentence structure, and emphasis, but converge on the same operational instruction. The purpose of paraphrastic variation is to prevent exact-string overfitting — the model should learn the intent of the system prompt rather than the literal token sequence. With three variants per operation, the model receives the operational signal through diverse linguistic surfaces and is forced to extract the underlying instruction rather than memorize the surface form.

Operation-specific prompts are used for 60% of training records (15,000 of 25,000). The high proportion reflects the centrality of operation training in the curriculum; most records are training a specific operation or small set of operations, and the system prompt names that operation to align the model's attention with the intended training signal.

### 7.2 Tier 2: Meta-Cognitive Prompts (5 variants)

A single base meta-cognitive prompt with five paraphrastic variants. The meta-cognitive prompt does not specify which operation to deploy; it instructs the model to diagnose the problem and select its own approach. This prompt is used for 30% of records (7,500), including all Strategy Selection records (Type 4) and a substantial fraction of the other record types.

**Example — Meta-Cognitive Prompt, base form (illustrative):**

> *"You are a critical thinker. When presented with a problem, do not begin solving immediately. First, read the problem carefully and identify what type of reasoning it requires. State your analytical approach before executing it. If you discover midway that your approach is wrong or insufficient, say so explicitly and adapt. Show every decision point. Your goal is not just a correct answer but a transparent reasoning process that could be audited by someone who disagrees with your conclusion."*

The 30% allocation to meta-cognitive prompts is deliberately higher than the conventional ~5–10% allocation in instruction-tuning datasets. We believe strategy selection (Operation 12) is the highest-leverage operation to train, because it is the operation that determines which other operations get deployed on a given problem. A model that can execute any specified operation on demand is useful; a model that can recognize what operation a problem requires without being told is qualitatively more capable.

### 7.3 Tier 3: Minimal Prompts (3 variants)

Three intentionally sparse system prompt variants, used for 10% of records (2,500). The variants approximate the minimal possible system prompt:

> *"Answer the following problem. Show your reasoning."*
>
> *"Solve this. Explain your thinking."*
>
> *"Reason through the following problem step by step."*

Minimal prompts exist to inoculate the model against system-prompt dependence. If the model can produce high-quality reasoning only when given a richly specified system prompt, then the model's reasoning capability is fragile — it depends on a control signal that may or may not be present at inference. By training a substantial minority of records under minimal prompts, we ensure that the reasoning capability is encoded as a property of the model itself rather than as a contingent response to elaborate prompting. The result is a model that benefits from rich system prompts (gaining additional operational guidance) without being incapacitated by their absence.

### 7.4 Distribution Summary

**Table** 5. System prompt tier distribution across the 25,000-example curriculum.

| Tier | Variants | Records | Share |
|---|---|---|---|
| 1 — Operation-Specific | 36 (12 ops × 3 paraphrases) | 15,000 | 60% |
| 2 — Meta-Cognitive | 5 | 7,500 | 30% |
| 3 — Minimal | 3 | 2,500 | 10% |
| Total | 44 | 25,000 | 100% |

### 7.5 The Inference-Time Implication

The system prompt architecture is not merely a training-time concern. It has direct consequences for how the deployed model can be controlled at inference. Because the model has been trained on forty-four system prompt variants and has learned to attend to system prompts as meaningful control signals, the deployed model will respond to new system prompts that share the general structural features of the training-time prompts. A deployment that provides a richly specified domain-and-persona system prompt at inference — for instance, instructing the model to behave as a clinical reasoner with specific evidentiary standards — will activate the operational behaviors that were trained under structurally similar prompts during fine-tuning.

This is the mechanism by which a domain-agnostic reasoning model can be specialized to a professional domain without retraining. The fine-tuning provides the cognitive operations and the disposition to follow system prompts; the inference-time system prompt provides the domain alignment. Section 11 extends this architecture to cases where additional domain fine-tuning is desired, but the basic insight is that the system

prompt is the principal lever for inference-time control, and that the system prompt's effectiveness as a lever is determined by how the system prompt was designed during fine-tuning.

## 8. The Dataset Generation Pipeline

Constructing a 25,000-example reasoning dataset at the quality level the framework demands is not feasible through hand-authoring (the labor cost is prohibitive) and is not feasible through naive AI generation (the quality variance is unacceptable). We specify a seven-phase pipeline that combines hand-authored seed problems, AI-assisted augmentation, rejection sampling, multi-layer quality verification, and structured diversification across record types. The pipeline produces the full 25,000-example dataset from approximately 500 hand-authored seed problems.

The pipeline is structured so that each downstream phase depends on the quality of the upstream phases. The seed problems are the most consequential single input; every downstream artifact carries the signal of the seed problems forward and amplifies it. If the seeds are well-constructed, the AI augmentation extends them faithfully and the verification phases retain the highest-quality outputs. If the seeds are weak, no amount of downstream processing will produce a strong dataset.

We disclose the structure of each phase. The specific seed problems used in our implementation, and the verbatim text of the AI prompts used at each generation phase, are part of the protected implementation detail.

### 8.1 Phase 1: Seed Problem Design (Human, ~500 Problems)

The curriculum designer constructs approximately 500 seed problems by hand, distributed roughly evenly across the ten problem domains (~50 problems per domain). Each seed is a paragraph-length scenario (50–200 words) with a clearly stated question. Seeds must satisfy four constraints:

*Constraint 1 — Genuine difficulty.* A competent reasoner should need to stop and think to solve the problem. Trivial seeds amplify into trivial datasets.

*Constraint 2 — Self-containment.* The problem must be solvable from the information stated in the problem itself, without external knowledge beyond what an educated general reader would possess.

*Constraint 3 — Domain neutrality at the surface.* The problem content should be expressible in everyday terms — factories, schedules, teams, budgets, scheduling, allocation, classification of cases — rather than in technical professional vocabulary. The cognitive challenge should be transferable to professional domains, but the surface content should not require professional knowledge to engage with.

*Constraint 4 — Privately annotated operation tagging.* Each seed should be tagged (in the curriculum-construction workspace, not in the training data) with the cognitive operations it primarily exercises. The tags inform downstream pipeline phases but are never visible to the model.

The seed-design phase is the most labor-intensive single phase, and it is the phase that most determines final dataset quality. We estimate two to three weeks of focused work for a competent curriculum designer to produce a complete seed set.

### 8.2 Phase 2: Problem Augmentation (AI-Assisted, 500 → ~5,000)

Each seed problem is used as input to a frontier language model with an augmentation prompt that requests nine structural variants of the seed. A structural variant preserves the cognitive operation profile of the original — same operations exercised, same reasoning difficulty, same problem domain — while varying the surface content. A factory-scheduling seed yields hospital-staffing, classroom-allocation, delivery-routing, and crop-rotation variants. The result of nine variants per seed is ~4,500 derivative problems, plus the original 500 seeds, for a corpus of ~5,000 problems.

The augmentation prompt is structured to require difficulty variation across the nine variants: three easier than the original, three of comparable difficulty, three harder. This ensures the final dataset has the difficulty distribution required for graduated learning rather than concentrating mass at the original seed difficulty level.

Spot-checking 20% of the augmented problems and discarding/regenerating any variants that drift from the original cognitive structure is required for quality control. Naive AI augmentation reliably produces some variants that change the cognitive profile rather than the surface content; human review catches these.

### 8.3 Phase 3: Solution Generation (AI, 5,000 × 3 = 15,000 Candidates)

For each of the ~5,000 problems, three independent candidate reasoning chains are generated using a frontier language model. Each chain is generated under the appropriate system prompt variant for the problem's cognitive operation profile (drawn from the forty-four variants specified in Section 7).

The three-chain rejection sampling is the highest-leverage quality intervention in the pipeline. The single-chain alternative produces solutions at the mean of the model's distribution; the three-chain rejection alternative produces solutions at the top of the distribution. The cost is three times the generation cost. The quality improvement is substantial because the upper tail of the distribution is meaningfully better than the mean.

### 8.4 Phase 4: Verification and Selection (Automated + AI Judge, 15,000 → 5,000)

For each problem's three candidate chains, two verification layers are applied.

*Layer A — Automated checks.* Does the response follow the structural format required by its record type? Is it within the target length range (150–1,200 words depending on difficulty)? Does it contain a verification step (required for Clean Solves)? Does it contain the failure-and-recovery sequence (required for Productive Failures)? Automated checks remove obvious format violations.

*Layer B — AI Judge scoring.* A different model than the one used in Phase 3 is invoked as a quality judge. Cross-model evaluation reduces systematic bias — if Phase 3 used Model X, Phase 4's judge uses Model Y, and vice versa. The judge rates each chain on a five-dimension rubric: logical validity (does each step

follow from the previous?), decision-point transparency (are choices made explicit?), format compliance (does it match the expected structure?), calibration (does the confidence level match the actual strength of the reasoning?), and answer correctness.

The highest-scoring chain per problem is retained. Problems where no chain achieves an acceptable score across all dimensions are returned to Phase 3 for regeneration. The output of Phase 4 is ~5,000 problems with one verified high-quality solution each.

### 8.5 Phase 5: Record Type Diversification (AI-Assisted, 5,000 → 25,000)

The ~5,000 verified Clean Solve examples from Phase 4 are now diversified into the six record types. For each downstream record type, the AI is given the original problem and the verified solution and asked to produce the type-specific variant.

*Productive Failure variants (6,000):* The AI is given the problem and correct solution, and asked to produce a response that starts with a plausible-but-wrong approach, encounters a specific contradiction within two to three steps, identifies the failure, backtracks, and solves correctly. The verified solution serves as the ground-truth correct chain for the recovery portion.

*Competing Interpretation variants (4,000):* Problems that genuinely admit ambiguity are identified (or modified to introduce defensible ambiguity). The AI produces a response that develops both readings and concludes with calibrated confidence.

*Strategy Selection variants (3,500):* The system prompt is replaced with a meta-cognitive prompt (Tier 2). The AI regenerates the solution with diagnosis-before-execution structure.

*Insufficient Information variants (2,000):* One critical piece of information is removed from the problem. The AI produces a response that identifies the gap, analyzes what can still be determined, and provides conditional analysis.

*Adversarial Trap variants (2,000):* New problems are designed (or existing ones modified) to trigger specific cognitive biases. The AI produces fall-and-catch or immediate-identification responses. Adversarial seeds typically require more human creativity than the other types and may benefit from an additional human-authoring step.

### 8.6 Phase 6: Final Quality Pass (AI Judge + Human Review)

Every example produced by Phase 5 is passed through the AI Judge a second time. Additionally, a cross-type consistency check is performed: when the same problem appears as a Clean Solve and as a Productive Failure, both must reach the same final answer.

A 10% sample of the full dataset (~2,500 examples) is reviewed by humans. The review sample is stratified to cover all six record types, all ten domains, and all twelve operations proportionally. Systematic issues identified in human review trigger regeneration of the affected category.

### 8.7 Phase 7: Deduplication and Balance Verification

Semantic deduplication at a cosine similarity threshold of 0.92 removes near-duplicate examples. A typical deduplication pass removes 5–15% of the dataset, which is then regenerated to bring the dataset back to ~25,000.

Final balance verification confirms that the target distributions across record types, domains, and operations are met. Adjustments via targeted regeneration bring any under-represented category up to its target frequency.

### 8.8 Pipeline Cost Estimation

The pipeline produces a 25,000-example dataset at a generation cost dominated by Phases 2, 3, and 5, in which the frontier model is invoked at scale. With current frontier-model pricing (≈$15 per million input tokens, ≈$60 per million output tokens for high-quality generation; some pricing tiers significantly lower), the total generation cost for the dataset is on the order of $300–$800 depending on model selection, batch-mode discounts, and verbosity of generated content. The cost of the seed-design phase is the curriculum designer's time. The cost of the LoRA fine-tuning phase, downstream of the dataset, is approximately one order of magnitude smaller than the generation cost.

### 8.9 Iterative Construction

We strongly recommend a two-pass construction rather than a single-shot construction. In the first pass, ~5,000 examples (20% of the budget) are constructed and used to fine-tune a checkpoint model. The checkpoint is evaluated against the evaluation suite (Section 10). The evaluation identifies which operations and record types the checkpoint handles well and which it handles poorly. The second pass constructs the remaining 20,000 examples with increased allocation to the weak areas identified in evaluation.

The two-pass approach costs approximately 10% more than the single-shot approach because of the additional fine-tuning run and evaluation cycle. The quality gain is substantial: the second pass is data-driven correction of weaknesses that would otherwise be locked into the final model. We consider the two-pass approach the default; single-shot construction is appropriate only when iteration is infeasible.

## 9. Fine-Tuning Specification

The reasoning curriculum is fine-tuned into a base model via Low-Rank Adaptation (LoRA) [Hu et al., 2021]. This section specifies the base model selection, the LoRA configuration, the training hyperparameters, the data split, and the training cost.

### 9.1 Base Model Selection: Phi-4 14B

We recommend Microsoft Phi-4 (14B parameters) [Abdin et al., 2024] as the base model for the reasoning fine-tune, on four grounds.

*Reasoning baseline.* Phi-4 achieves 95.3% on GSM8K, 80.4% on MATH, and 56.1% on GPQA, outperforming open-source models five times its size on mathematical and scientific reasoning. The

pretraining curriculum favors structured, reasoning-rich content, which provides a strong foundation that targeted fine-tuning can refine.

*Training philosophy alignment.* The Phi series is built on the premise that data quality dominates data quantity, which aligns with the curriculum's design philosophy. A model pretrained under this philosophy is well-positioned to receive a fine-tune that further refines its reasoning style; the fine-tune is operating in a regime the base model is already optimized for.

*Inference economics.* At 14 billion parameters, Phi-4 runs on a single A100 80GB or comparable accelerator. Per-token inference cost is approximately one to two orders of magnitude lower than frontier API pricing. This is consequential for the cost structure of any deployment built on the fine-tuned model.

*Ecosystem support.* Phi-4 is available through Hugging Face and Azure AI Foundry. LoRA fine-tuning, model serving, and adapter hot-swapping are well-supported in standard frameworks.

The framework is not specific to Phi-4. The same curriculum can be applied to other base models in the 7B–14B class — Llama 3.1 8B, Mistral 7B, Qwen 2.5 14B, and others. We expect the relative gains from the curriculum to be larger for base models with weaker reasoning baselines (because they have more room to improve) and the absolute gains to be larger for base models with stronger reasoning baselines (because they can reach higher ceilings). The choice of base model should be made based on the deployment requirements rather than on the curriculum's compatibility, which is broad.

## 9.2 LoRA Configuration

We specify the following LoRA hyperparameters, which we believe to be reasonable defaults for the reasoning fine-tuning task.

*Rank: 64.* Higher than the typical r=8–16 used for style adaptation. Reasoning tasks benefit from higher rank because the fine-tuning needs to modify a wider band of the model's representation space than a stylistic adaptation does. Rank 32 is a viable alternative for cost-sensitive deployments; rank 128 produces diminishing returns relative to its compute cost.

*Alpha: 128.* Two times the rank, following the standard heuristic. The effective learning rate for LoRA parameters is alpha/rank times the base learning rate; this ratio of 2 has been empirically reliable.

*Target modules: all linear layers, including both attention projections and MLP layers.* Reasoning-relevant signal is distributed across both attention and MLP layers in transformer architectures. Restricting LoRA to attention layers alone, as is common in stylistic fine-tunes, leaves MLP capacity for reasoning under-utilized.

*Dropout: 0.05.* Light regularization to prevent overfitting to the training distribution. Higher dropout values (0.1–0.2) impede the model's learning of complex reasoning patterns and should be avoided.

## 9.3 Training Hyperparameters

*Epochs: 2.* Two passes through the training data. Validation loss is monitored after each epoch; if validation loss increases on the second epoch, training is stopped and the first-epoch checkpoint is used. Three or more epochs reliably produce overfitting to the training patterns, which manifests as deterioration of reasoning quality on out-of-distribution problems even as in-distribution loss continues to decrease.

*Learning rate: 2e-4 with cosine annealing.* Standard for LoRA fine-tuning. Higher learning rates (5e-4 and above) destabilize the reasoning behaviors that the pretrained model already exhibits. Lower learning rates (5e-5 and below) require more epochs to converge and thus increase overfitting risk.

*Warmup: 5% of total steps.* Linear warmup from zero to the peak learning rate over the first 5% of training steps.

*Batch size: effective batch size of 32 via gradient accumulation.* The micro-batch size is constrained by GPU memory; gradient accumulation over multiple micro-batches achieves the effective batch size. On an A100 80GB, micro-batch size of 4 with gradient accumulation of 8 reaches effective batch size 32.

*Maximum sequence length: 2,048 tokens.* Sufficient for the longest reasoning chains in the curriculum. The 95th percentile chain length is below 1,200 tokens; the 2,048 ceiling accommodates outliers without requiring sequence packing.

*Optimizer: AdamW with weight decay 0.01.* Standard.

### 9.4 Data Split

The 25,000-example dataset is split 90/10 between training and validation. The validation set is stratified across all six record types and ten problem domains to ensure that validation loss is a meaningful signal of generalization rather than of in-distribution memorization.

The validation set is used for early stopping and loss monitoring only. It is not used for hyperparameter tuning, which would require a separate test set. The evaluation suite (Section 10) serves as the independent test set, and is constructed separately from the training data to avoid any train-test contamination.

### 9.5 Training Cost

The fine-tuning is computationally inexpensive relative to the dataset generation. At an effective batch size of 32 and the hyperparameters above, training the full 25,000-example dataset for two epochs on a single A100 80GB requires approximately 1.5–2 hours of compute time. On a spot-instance configuration with current cloud pricing, the total compute cost is on the order of $5–15 for the production training run, plus comparable cost for the validation runs during hyperparameter selection.

The total compute cost for a complete fine-tuning cycle, including the iterative two-pass construction (Section 8.9), is on the order of $20–50. This is approximately one to two orders of magnitude smaller than the dataset generation cost, which itself is one to two orders of magnitude smaller than the labor cost of the seed problem design.

This cost structure is consequential. The principal investment in the framework is the seed problem design, which is non-monetary (curriculum designer time). The principal monetary cost is the dataset generation. The fine-tuning itself is a rounding error. This means that a small research group with a well-designed seed set can produce a reasoning-fine-tuned model for under $1,000 in total cloud compute and API spend.

## 10. Proposed Evaluation Framework

This paper does not present empirical results. The evaluation framework described in this section is the framework we recommend for assessing a reasoning fine-tune produced by the curriculum, and the framework against which we have evaluated our own implementation. The empirical results from our implementation are reserved for a subsequent publication; this section specifies the evaluation methodology so that other research groups can apply the framework to their own implementations.

### 10.1 The Evaluation Suite

The evaluation suite is a set of 200 problems hand-constructed to test the cognitive operations in isolation and in combination. The suite is not derived from the training data and is not used during training in any capacity. It is the independent test set.

The 200 problems decompose as follows:

*Single-operation problems (144 problems, 12 per operation).* Each operation is tested on twelve problems that primarily exercise that operation, with minimal load on other operations. These problems isolate the model's capacity in each operation and reveal which operations the fine-tune has strengthened versus which it has left underdeveloped.

*Operation-combination problems (20 problems).* Pairs and triples of operations that are critical for specific reasoning tasks — Evidence Weighing combined with Scope Determination (statutory interpretation analogue), Diagnostic Reasoning combined with Absence Reasoning (clinical reasoning analogue), Temporal Sequencing combined with Causal vs. Correlational reasoning (historical analysis analogue). These problems test the model's capacity to compose operations rather than deploy them individually.

*Insufficient Information problems (20 problems).* Problems where the correct response identifies missing information rather than producing an answer. The test is whether the model recognizes the insufficiency and produces a correctly framed conditional analysis, or whether it produces a confident answer anyway (hallucination).

*Adversarial bias problems (16 problems).* Problems designed to trigger specific cognitive biases. The test is whether the model falls for the bias, catches itself, or recognizes the trap immediately.

The evaluation suite should be constructed by a different person than the curriculum designer where possible, to reduce shared blind spots. The suite should also be expanded over time as new failure modes are discovered in deployed models; an evaluation suite that does not evolve is an evaluation suite that drifts out of relevance.

## 10.2 The Three-Way Benchmark

Each evaluation problem is run against three models:

*Baseline.* The base model without any reasoning fine-tuning (Phi-4 14B in our reference implementation). This is the floor. The fine-tuned model must outperform the baseline on substantially all operations, or the fine-tune has failed.

*Fine-tuned model.* The product of applying the curriculum to the base model. This is what is being evaluated.

*Frontier reference.* A current frontier model accessed via API (we recommend GPT-4o or Claude 3.5 Sonnet as the frontier reference at the time of writing, recognizing that frontier models change rapidly). This is the ceiling against which the fine-tune is benchmarked. The success criterion is not that the fine-tune matches or exceeds the frontier model on all dimensions — that is implausible for a 14B-parameter student against a multi-hundred-billion-parameter frontier teacher — but that the fine-tune matches or exceeds the frontier on a substantial fraction of operations.

The three-way structure is essential. Comparing the fine-tune only against the baseline tells us the fine-tune improved, but not how far it can be pushed. Comparing the fine-tune only against the frontier tells us where the ceiling is, but not whether the gains over baseline justify the fine-tuning cost. The three-way comparison locates the fine-tune in the space between floor and ceiling.

## 10.3 Scoring Rubric

Each model response is scored on five dimensions, each on a 1–5 scale.

*Logical validity.* Does each reasoning step follow from the previous? Are there any non-sequiturs, unsupported leaps, or invalid inferences?

*Decision-point transparency.* Are choices made explicit at each reasoning step? Or does the chain present a sequence of statements without acknowledging the alternatives that were available?

*Completeness.* Are all relevant considerations addressed? Are there obvious gaps in the analysis?

*Calibration.* Does the confidence level expressed in the response match the actual strength of the underlying reasoning chain? Or is the response overconfident relative to the reasoning, or underconfident?

*Answer correctness.* Is the final answer right? (For Insufficient Information problems, this dimension is replaced with: does the response correctly identify the missing information and its consequences for the analysis?)

Scoring is performed by an AI judge using a different model than the one being evaluated. Human scoring of a 20% sample is recommended as calibration for the AI judge; the AI judge's reliability against human scoring is itself a quality metric to be reported.

## 10.4 Success Criteria

The fine-tune is judged to have succeeded if:

(a) the fine-tuned model outperforms the baseline on all twelve operations, with the gain being statistically meaningful (we suggest an effect size of at least 0.5 points on the 1–5 scale, with significance assessed across the twelve problems per operation);

(b) the fine-tuned model matches or exceeds the frontier reference on at least eight of the twelve operations (this is a calibration test; the threshold may be raised or lowered based on the specific deployment requirements);

(c) the fine-tuned model demonstrates the Insufficient Information behavior — that is, it produces conditional analyses rather than confident answers on the Insufficient Information problems — at a rate of at least 80%;

(d) the fine-tuned model demonstrates resistance to the targeted cognitive biases — that is, it either catches the bias mid-response or identifies the trap immediately — at a rate of at least 75% on the Adversarial Bias problems.

These thresholds are starting points, not theoretically derived optima. A research group with different deployment requirements may adjust them. The framework specifies the evaluation methodology; the thresholds for judging success against that methodology are deployment-dependent.

### 10.5 Iterative Improvement

If the evaluation reveals weakness in specific operations or record types, the curriculum can be adjusted by increasing the allocation of training examples to the weak areas. This is the data-driven iteration loop described in Section 8.9. We have found that two iterations are typically sufficient to bring all operations above the baseline-outperformance threshold; a third iteration is rarely worthwhile relative to its cost.

## 11. Domain Specialization via Lightweight Secondary Fine-Tuning

The reasoning curriculum described in Sections 5–9 produces a domain-agnostic reasoning model. The cognitive operations it has acquired are universal. The problems it has been trained on are content-neutral. The model can be deployed as a general-purpose reasoner with only inference-time prompting to control its behavior.

For many deployment scenarios, however, additional domain alignment is desirable. A model intended for clinical reasoning benefits from explicit training on clinical vocabulary, evidentiary hierarchies, and failure modes specific to medicine. A model intended for legal reasoning benefits from training on statutory interpretation conventions, jurisdictional considerations, and the structure of case analysis. The domain-agnostic base model can engage with these domains given a rich enough system prompt, but a secondary fine-tuning layer specialized to the domain typically produces more reliable behavior.

We propose a two-layer fine-tuning architecture in which the domain-agnostic reasoning curriculum is the foundation (Layer 1) and a lightweight domain-specific curriculum is applied on top (Layer 2). Each Layer

2 curriculum is much smaller than Layer 1 — typically 5,000 to 10,000 examples — because the cognitive operations are already trained by Layer 1, and Layer 2 only needs to teach the model the vocabulary, source hierarchies, and conventions of the target domain.

## 11.1 Layer 2 Curriculum Structure

A Layer 2 curriculum uses the same six record types and the same response formats as Layer 1. The structural framework is unchanged; only the content shifts. Where Layer 1 might exercise Operation 6 (scope determination) on a problem about whether a workplace policy applies to a contractor, Layer 2 for a legal domain would exercise the same operation on whether a specific statute applies to a specific factual pattern.

The twelve cognitive operations remain the operational primitives. The domain-specific vocabulary is layered onto the operational substrate. The model that has learned scope determination on domain-neutral content learns, through Layer 2, that the domain-specific term for scope determination in law is "interpretation of statutory reach" — but the cognitive operation it deploys is the same one it learned in Layer 1.

This compositional structure has direct practical consequences. The Layer 2 curriculum can be built much faster than Layer 1 would be if constructed from scratch, because the curriculum designer does not have to teach the operations again; they have to teach the domain's expression of operations the model already knows. A 5,000–10,000-example Layer 2 curriculum is achievable in two to three weeks of focused work, compared to the eight to twelve weeks required for the 25,000-example Layer 1 curriculum.

## 11.2 Layer 2 Construction Requirements

Each Layer 2 curriculum requires a domain expert who can construct the seed problems and verify the reasoning chains. The expert's role parallels the curriculum designer's role in Layer 1: hand-authoring seed problems, tagging them with operation profiles, and reviewing the AI-augmented outputs for fidelity to the domain. The expert does not have to be a curriculum designer themselves; they need to be a domain expert who can collaborate with a curriculum designer to translate their domain expertise into the curriculum structure.

This requirement creates a natural sequencing constraint: Layer 2 development for any given domain is gated by the availability of a domain expert willing to do the seed-problem work. The domain-agnostic Layer 1 is the unblocked foundation; Layer 2 development can proceed for any domain whose expert is engaged, in parallel across multiple domains, each producing its own specialized variant of the base reasoning model.

## 11.3 Fine-Tuning the Layer 2 Variant

The Layer 2 LoRA adapter is fine-tuned on top of the Layer 1 base. The hyperparameters mirror Layer 1 — rank 64, alpha 128, all linear layers, 2 epochs — with the smaller dataset size resulting in proportionally shorter training time. The total compute cost for a Layer 2 fine-tune is approximately $5–15.

Each Layer 2 produces a distinct LoRA adapter. The same base model can host multiple Layer 2 adapters, which can be hot-swapped at inference depending on which domain the model is being asked to engage with. This is the architectural basis for cost-efficient multi-domain deployment: a single hosted base model serves multiple domain-specialized variants through adapter swapping, with marginal cost per additional domain limited to the storage cost of the adapter (megabytes) rather than the compute cost of an additional hosted model.

### 11.4 Domain Knowledge vs. Domain Reasoning

A clarification about what Layer 2 does and does not provide. Layer 2 teaches the model the reasoning conventions of a domain — the vocabulary, the evidentiary standards, the conventional structure of analysis. It does not teach the model the substantive knowledge of a domain. A clinically fine-tuned model has been taught to reason like a clinician; it has not necessarily been taught all the substantive medical knowledge a clinician possesses.

Substantive domain knowledge can be supplied through three channels: the model's pretraining data (Phi-4's pretraining includes substantial scientific and medical content but is not optimized for any specific professional domain), retrieval-augmented generation at inference (giving the model access to authoritative domain references at query time), or a third fine-tuning layer specifically targeting domain knowledge content. The choice among these channels depends on the stability of the domain knowledge — fast-changing domains (medicine, law, finance) benefit from retrieval-based supply, because the knowledge can be updated without retraining; stable domains can have knowledge baked into the model.

The framework as presented here is silent on the choice of knowledge-supply channel. The framework provides the reasoning capacity; how the deployed system supplies knowledge to that reasoning capacity is a deployment-architecture decision that depends on the requirements of the specific application.

## 12. Discussion

This section addresses limitations of the framework as proposed, theoretical implications of the operations-as-primitives thesis, and open empirical questions that this paper does not resolve.

### 12.1 Limitations

*The framework is unvalidated empirically in this paper.* We have implemented the framework and produced a fine-tuned model, but the empirical results are reserved for a subsequent publication. This paper makes architectural and theoretical claims; the claim that the architecture works — that is, produces measurable improvements over baselines — is asserted but not demonstrated here. Readers should treat the framework as a hypothesis to be tested rather than as a validated method.

*The twelve-operation decomposition is one of several possible decompositions.* We have argued for the operations-as-primitives thesis at the theoretical level, but the specific selection of twelve operations is a design choice that other research groups might make differently. A decomposition into ten operations or fourteen operations could be equally defensible. The thesis is robust to such variations; what is not robust

is the philosophical-categories approach, which we have argued is mistaken at the level of cognitive architecture.

*The framework assumes substantial curriculum-design labor.* The seed-problem design phase, which we identify as the most consequential single input to the dataset, requires two to three weeks of focused work by a curriculum designer with sufficient cognitive task analysis skill to construct problems that exercise specific operation combinations. This is a non-trivial labor cost. The framework is therefore not a turnkey solution; it is a structured approach that requires significant intellectual investment to instantiate.

*The framework is silent on multilingual and multimodal reasoning.* We have addressed monolingual textual reasoning. The extension to multilingual settings (does an Operation 6 example in English transfer to Operation 6 capacity in Mandarin?) and to multimodal settings (does textual reasoning training transfer to visual reasoning?) is an open question that the framework as presented does not address.

*The domain transfer claim depends on the base model's general-knowledge coverage.* The framework's domain-transfer mechanism — train operations on neutral content, apply them at inference with domain-aligned system prompts — assumes that the base model has sufficient general knowledge of the target domain to engage with domain-specific content when prompted. This assumption fails for domains the base model has minimal pretraining exposure to. For such domains, retrieval augmentation or domain-knowledge fine-tuning is required in addition to the reasoning fine-tune.

## 12.2 Theoretical Implications

*If operations compose into philosophical modes, why has the field organized training around the modes?* We hypothesize three reasons. First, the philosophical taxonomy is inherited intellectual infrastructure that comes with the territory of analytic philosophy; AI researchers trained in or adjacent to philosophy import the taxonomy by default. Second, the philosophical taxonomy is convenient for benchmark construction — "this benchmark tests deductive reasoning" is a clean framing for a paper, even if the cognitive structure of the benchmark is more granular. Third, the philosophical taxonomy is at the right level of abstraction for reporting reasoning capability ("the model is strong on deductive tasks, weak on abductive tasks"), even if it is at the wrong level for training reasoning capability.

These reasons are sociological rather than technical. We do not claim that researchers who have used the philosophical taxonomy were wrong to do so given their goals; we claim that the philosophical taxonomy is the wrong organizing principle for the specific goal of producing models with robust reasoning capacity, and that this fact has been obscured by the convenience of the taxonomy for other purposes.

*Are the twelve operations themselves further decomposable?* Possibly. Operation 1 (decomposition) might be decomposable into sub-operations: recognition that a problem has internal structure, identification of the seams along which to decompose, allocation of sub-problems to the appropriate operations for solving. Operation 5 (evidence weighing) might be decomposable into source-credibility assessment, evidence-to-claim relevance assessment, and inter-evidence consistency assessment. We have not pursued these decompositions because we judged the twelve operations to be at the right level of abstraction for training: granular enough to be cognitively meaningful, coarse enough that each can be trained with sufficient

frequency in a 25,000-example curriculum. A research group with a larger curriculum budget could explore finer-grained decompositions.

*What is the relationship between operations and the linguistic features of the training data?* This is an open theoretical question. The operations are cognitive primitives, but they are trained through linguistic surfaces — reasoning chains expressed in text. The model learns the operations only insofar as the linguistic surface of the training data correlates with the operational structure of the underlying reasoning. If the linguistic surface and the operational structure decouple — if reasoning chains can be written that look like Operation 6 (scope determination) but do not actually exercise the cognitive move of scope determination — then the model may learn the linguistic surface without learning the operation. This is a risk our framework mitigates through the decision-point transparency requirement and the structural variation across record types, but it is not a risk we believe can be fully eliminated.

### 12.3 Open Empirical Questions

The framework as proposed raises a number of empirical questions that we believe deserve investigation by the broader research community.

**Q1.** *How does the relative benefit of operation-based vs. category-based training scale with model size?* Our framework is optimized for 7B–14B-parameter models. Larger models may benefit less because their pretraining has already produced strong implicit reasoning capacity; smaller models may benefit more because the curriculum compensates for weaker pretraining. The size scaling of the framework is unknown.

**Q2.** *What is the marginal value of each record type?* Our framework includes all six record types with specified proportions. Ablation studies could quantify the contribution of each type. If removing Insufficient Information records degrades the model's hallucination resistance, that confirms the inclusion. If removing Adversarial Trap records makes no difference to the model's bias resistance, that suggests the type can be reduced or eliminated in favor of more examples elsewhere.

**Q3.** *How sensitive is the framework to the specific operation taxonomy?* We have argued that the operations-as-primitives thesis is robust to variations in the specific taxonomy, but the empirical robustness has not been tested. A study that trained models on different operation taxonomies (twelve operations, ten operations, fourteen operations, different operation selections) and measured the resulting reasoning capacity would test this claim.

**Q4.** *Does domain transfer actually work via system prompts?* Our framework predicts that a domain-agnostic reasoning model can be specialized to a professional domain through rich inference-time system prompts, without secondary fine-tuning. The prediction has the right structure to be tested: train a model on the domain-agnostic curriculum, evaluate it on domain-specific benchmarks with domain-aligned system prompts, and compare to a model that has also been Layer-2 fine-tuned on the domain. The marginal value of Layer 2 over inference-time prompting alone is empirically discoverable.

**Q5.** *What is the right curriculum size?* We have proposed 25,000 examples as a working target, citing LIMA and Deita for the quality-over-quantity argument. The specific number is a design choice, not a

theoretically derived optimum. A larger study could establish the size-quality tradeoff curve and identify the inflection point at which marginal examples produce diminishing returns.

### 12.4 What This Paper Does Not Claim

To prevent misreading, we explicitly note several claims the paper does not make.

The paper does not claim that the framework is a unique or optimal approach to reasoning fine-tuning. It claims that the framework is well-motivated and worth investigating, and that the operations-as-primitives thesis is a worthwhile correction to the philosophical-categories framing prevalent in the existing literature.

The paper does not claim that small fine-tuned models can match frontier models on all reasoning tasks. It claims that small fine-tuned models can match frontier models on specific reasoning tasks within their training distribution, and that the gap on tasks outside the distribution is recoverable through curriculum extension.

The paper does not claim that reasoning is reducible to twelve operations. It claims that for the purpose of training models, decomposing reasoning into operations of approximately this granularity produces better curricula than treating philosophical categories as the unit of training. Other framings of reasoning may be appropriate for other purposes.

The paper does not claim that the architecture eliminates hallucination, eliminates bias, or eliminates reasoning errors. It claims that the architecture, by explicitly training the cognitive behaviors that correspond to recognizing insufficient information, recognizing cognitive bias, and recognizing one's own errors, produces models that exhibit these behaviors more reliably than models trained without these specific record types.

## 13. Conclusion

We have proposed a framework for training reasoning in small language models that diverges from the prevailing approach in the field. Where existing approaches organize training around philosophical categories of reasoning — deductive, inductive, abductive, and causal — we have argued that these categories are post-hoc descriptions rather than cognitive primitives, and that training organized around them produces brittle reasoning that imitates the surface of categorized prose without acquiring the operational substrate beneath it.

The alternative we have proposed organizes training around twelve cognitive operations — decomposition, premise identification, implication tracing, contradiction detection, evidence weighing, scope determination, temporal sequencing, absence reasoning, analogical mapping, confidence calibration, error recognition, and strategy selection — from which the philosophical categories emerge as natural compositions. Training the operations directly, rather than the categories, produces a model that can deploy whatever combination of operations a problem demands, including combinations that do not map onto any single classical category.

We have specified the curricular architecture that an operations-based framework requires: six record types that train cognitive behaviors beyond clean reasoning (productive failure recovery, competing-interpretation handling, strategy selection, insufficient-information identification, adversarial-bias resistance), ten problem domains that exercise the operations on domain-neutral content with transfer to professional domains, a three-tier system prompt architecture that teaches the model to treat system prompts as meaningful control signals, and a seven-phase generation pipeline that produces a 25,000-example training corpus from approximately 500 hand-authored seed problems.

We have specified the fine-tuning methodology — LoRA on a 14B-parameter base model, with specific hyperparameters and a two-pass iterative construction — and we have proposed an evaluation framework that benchmarks the fine-tuned model against both its base model (floor) and a frontier reference (ceiling) across the twelve operations.

The framework is presented as a defensive publication. We disclose the architecture in full; we protect only the specific seed problems and the verbatim text of system prompt variants used in our own implementation. We have implemented the framework and produced a fine-tuned model; the empirical evaluation is reserved for a subsequent publication. We invite empirical investigation by other research groups, critical engagement with the theoretical claims, and extensions of the framework to settings (multilingual, multimodal, longer-context) that we have not addressed.

The narrowing gap between small fine-tuned models and frontier models on reasoning tasks has shifted the principal lever of reasoning capability from parameter count to curriculum design. This paper is an attempt to take that shift seriously — to treat curriculum design as a first-class object of architectural investigation rather than as an unspecified support activity for the more visible work of pretraining and benchmark performance. We believe the framework we have proposed is a step in that direction. We expect it to be improved upon, in some places substantially, by subsequent work.

## References


Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., Lee, J. R., Lee, Y. T., Li, Y., Liu, W., Mendes, C. C. T., Nguyen, A., Price, E., de Rosa, G., Saarikivi, O., ... Zhang, Y. (2024). Phi-4 Technical Report. Microsoft Research.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Narang, S., Mishra, G., Yu, A., ... Wei, J. (2022). Scaling Instruction-Finetuned Language Models. arXiv:2210.11416.

DeepSeek-AI. (2025). DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Behl, H. S., Wang, X., Bubeck, S., Eldan, R., Kalai, A. T., Lee, Y. T., & Li, Y. (2023). Textbooks Are All You Need. arXiv:2306.11644.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Li, Y., Bubeck, S., Eldan, R., Del Giorno, A., Gunasekar, S., & Lee, Y. T. (2023). Textbooks Are All You Need II: phi-1.5 Technical Report. arXiv:2309.05463.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., & Cobbe, K. (2023). Let's Verify Step by Step. arXiv:2305.20050.

Liu, W., Zeng, W., He, K., Jiang, Y., & He, J. (2023). What Makes Good Data for Alignment? A Comprehensive Study of Automatic Data Selection in Instruction Tuning. arXiv:2312.15685.

Mitra, A., Del Corro, L., Mahajan, S., Codas, A., Simoes, C., Agarwal, S., Chen, X., Razdaibiedina, A., Jones, E., Aggarwal, K., Palangi, H., Zheng, G., Rosset, C., Khanpour, H., & Awadallah, A. (2023). Orca 2: Teaching Small Language Models How to Reason. arXiv:2311.11045.

Mukherjee, S., Mitra, A., Jawahar, G., Agarwal, S., Palangi, H., & Awadallah, A. (2023). Orca: Progressive Learning from Complex Explanation Traces of GPT-4. arXiv:2306.02707.

Peirce, C. S. (1903). Pragmatism as a Principle and Method of Right Thinking. Harvard Lectures on Pragmatism.

Wang, Y., Ivison, H., Dasigi, P., Hessel, J., Khot, T., Chandu, K. R., Wadden, D., MacMillan, K., Smith, N. A., Beltagy, I., & Hajishirzi, H. (2023). How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources. arXiv:2306.04751.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (2021). Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C., & Jiang, D. (2023). WizardLM: Empowering Large Language Models to Follow Complex Instructions. arXiv:2304.12244.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., & Levy, O. (2023). LIMA: Less Is More for Alignment. arXiv:2305.11206.

---

*Manuscript prepared for arXiv preprint submission, May 2026. This work is released as a defensive publication establishing prior art on the architectural framework described herein. The authors invite empirical investigation, critical engagement, and extension of the framework by the broader research community.*