

NOTES2SKILLS: From Lab Notebooks to Certainty-Aware Scientific Agent Skills

Shi Liu^{1,4} Jiayao Chen¹ Chengwei Qin² Yanqing Hu¹
Jufan Zhang³ Linyi Yang^{1,†}

¹Southern University of Science and Technology, Shenzhen, China

²The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

³University College Dublin, Dublin, Ireland

⁴Shenzhen Loop Area Institute, Shenzhen, China

†Correspondence: yangly6@sustech.edu.cn

Abstract

Scientific discovery workflows rely heavily on lab notes, where researchers record observations, interpret uncertain results, and plan follow-up experiments. Unlike polished publications, lab notes preserve evolving scientific reasoning, tacit scientific knowledge, and author uncertainty, giving AI agents access to the process of science. However, most prior work on scientific text focuses on papers, protocols, or structured databases, leaving informal laboratory notes underexplored as inputs to AI agents for science. This gap matters because tacit knowledge in lab notes is not written as a clean set of instructions: validated observations, tentative judgments, and possible experimental next steps often appear in the same passage. If these signals are conflated, an AI agent may mistake uncertain scientific judgments for confirmed conclusions or executable actions. To this end, we present NOTES2SKILLS, a two-stage framework for turning lab notebooks into verifiable skills for scientific AI agents while preserving the author’s certainty. Across three wet-lab sessions and seven original downstream configurations, NOTES2SKILLS is the only tested configuration that avoids both observed regime-level failures: over-acting on uncertain notes and missing firm directives. We show that certainty preservation is a key missing piece between lab notebooks and reliable agent skills, opening a path toward safer AI co-scientist systems.

1 Introduction

Lab notebooks are a central medium of scientific discovery, capturing researchers’ observations, interpretations of uncertain results, and plans for subsequent experiments (Doody and Artemeva, 2022). Unlike finalized papers and protocols, lab notebooks also preserve tacit knowledge in the scientist’s judgments, qualifications, and choices during the experiment. Early work in the machine-learning era focused on extracting structured flow

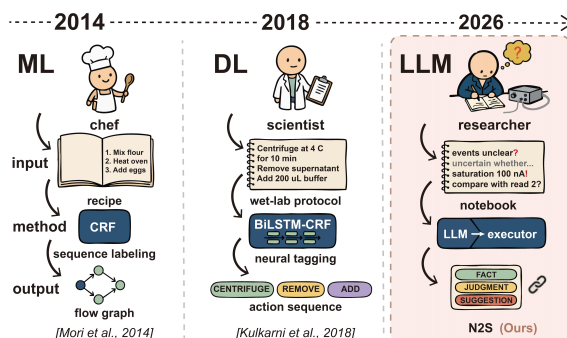


Figure 1: Three eras of procedural text extraction.

graphs from relatively constrained domains such as cooking recipes (Mori et al., 2014). With the advent of neural sequence models, this line of research shifted toward deep-learning methods that identify and organize action sequences from published wet-lab protocols (Kulkarni et al., 2018; Tamari et al., 2021; O’Donoghue et al., 2023). More recently, large language models (LLMs) have expanded procedural-text extraction and enabled workflows and domain knowledge to be packaged into agent-oriented workflows or skills (Giroh et al., 2025; Zhang et al., 2025). In parallel, LLM-based scientific co-agents support hypothesis generation and experimental planning (Gottweis et al., 2026). Figure 1 situates our notebook-to-skill setting in this progression from recipe-level flow graphs to protocol-level action modeling and LLM-based skill construction.

A common assumption underlies these three eras: the input text is prescriptive. A recipe instructs the reader to *mix*; a wet-lab protocol specifies *centrifuge for 10 minutes*; and a standard operating procedure states *set the temperature to 4°C*. By the time such text is written, the author has typically resolved their uncertainty, leaving the extraction system to map explicit instructions into executable actions.

Experimental notebooks violate this assumption.

They are written while the experiment is still unfolding, so tacit knowledge often appears together with uncertainty rather than as a settled instruction. For example, *the reading dropped sharply after five minutes* states a fact; *I am not sure the second read is reliable* expresses a judgment under uncertainty; and *try a fresh buffer next time* proposes a suggestion. Although these statements should induce different downstream behaviors, their surface forms can appear deceptively similar in raw notebook text.

Treating them as if they were equally firm fails in two opposing failure modes. **Uncertainty laundering** occurs when a tentative note, such as *I am not sure the second read is reliable*, is compiled into a firm decision: the agent acts on an interpretation that the author explicitly marked as uncertain, potentially discarding underlying data based on an unresolved judgment. **Directive loss** captures the reverse failure: a firm directive, such as *this part is invalid, truncate it*, is placed alongside cautious notes and treated as merely another opinion, causing the agent to retain data that the author intended to exclude. Both failures arise when the compiler strips away the author’s certainty signal: whether a statement is a fact, judgment, or suggestion. For agents performing irreversible operations on scientific data, this signal is the safety boundary.

In this work, we focus on single-author experimental notebooks written close to the time of experimentation, where the author’s certainty directly informs decision-making. We study the tacit knowledge that becomes visible in how authors state observations, qualify uncertain results, and propose next steps. This regime poses unique challenges absent from published protocols: certainty is entangled with action content within the same sentence, and hedged judgments can be surface-similar to firm observations. To address this, we present NOTES2SKILLS (N2S), motivated by the principle that the author’s certainty should constrain what an agent may do: a FACT may license a strong processing action, a JUDGMENT defaults to conservative, review-preserving handling, and a SUGGESTION is treated as advisory only. In experiments, Stage 1 jointly extracts directives and their certainty labels; Stage 2 compiles them into a METASKILL, a Mark-down skill document where each directive retains its certainty label and provenance to the original notebook excerpt.

We validate NOTES2SKILLS on 461 annotated segments across three corpora spanning a formal-

ity spectrum and test downstream skill loading on three real wet-lab sessions, where the compiled skills guide file-level data-handling decisions over instrument traces. On FreeNotes (informal bilingual notebooks), ONS (open notebook entries), and WLP (formal wet-lab protocols), the best of six model-prompt configurations achieves $F_1=0.737$ on binary directive detection, up from 0.682 under the strongest zero-shot baseline, and our Stage 2 audit verifies that all 149 fixed directives are carried into source-linked, agent-loadable capsules. NOTES2SKILLS is the only tested configuration that avoids both observed failures: laundering uncertain readings into firm actions on uncertainty-heavy sessions, and losing firm author-stated actions on a FACT-dominated session.

To our knowledge, we are the first to consider transferring single-author experimental notebooks to agent-loadable skills. Our contributions are threefold as follows. First, we treat notebooks written by scientists as a new kind of procedural text, where **author certainty** serves as a safety boundary for agents. Second, we annotate 461 segments and audit 149 directives across three corpora with different levels of formality. Finally, we introduce **METASKILL** and show, on an aligned instrument-data benchmark, that certainty-preserving skills are needed to match expert-adjudicated triage grounded in the author’s notes and signal evidence across both uncertain notes and firm facts.

2 Related Work

Scientific Procedure Extraction from Protocols. Scientific procedural-text extraction has primarily targeted curated, prescriptive documents. The Wet Lab Protocols corpus (Kulkarni et al., 2018) and its executable extension X-WLP (Tamari et al., 2021) established action and process extraction from published wet-lab protocols, while BioPlanner (O’Donoghue et al., 2023) evaluates LLM-based biological protocol planning. In parallel, NERRE (Dagdelen et al., 2024) studies flexible-schema entity and relation extraction from materials-science papers. These settings structure finalized scientific text, but do not treat the author’s certainty as an execution constraint. Experimental notebooks instead interleave observations, interpretations, hesitation, and prospective plans, often with first-person reflection and uneven directive density.

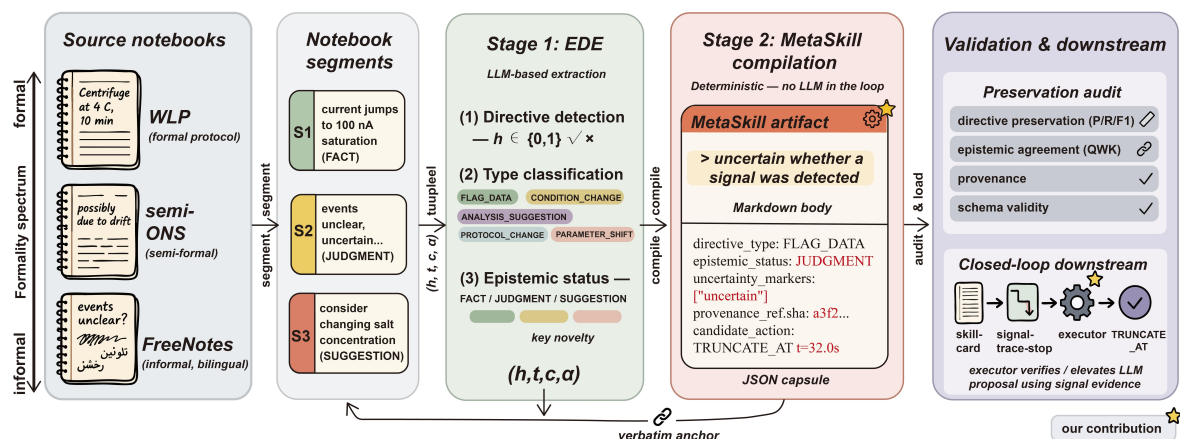


Figure 2: Overview of the NOTES2SKILLS pipeline. Yellow stars mark our contributions.

Compiling Procedural Text into Agent Skills. SYNTACT (Giroh et al., 2025) resolves ambiguity in enterprise standard operating procedures through a Clarifier–Planner–Implementor dialogue. Flow-of-Action (Pei et al., 2025) uses retrieved or generated SOPs to constrain multi-agent root-cause-analysis trajectories. Protocol-to-DSL systems (Shi et al., 2024; Jiang et al., 2024; Mehr et al., 2020) translate published procedures into machine-readable or executable representations, and the Anthropic Agent Skills format (Zhang et al., 2025) provides a practical target for packaging reusable agent instructions. In our setting, apparent ambiguity may encode an author’s epistemic stance and should be preserved rather than resolved away. More broadly, agents acquire capabilities from rollouts, demonstrations, workflows, literature, or tools (Wang et al., 2024; Majumder et al., 2024; Wang et al., 2025; Boiko et al., 2023; Bran et al., 2024; Szymanski et al., 2023). None of these works treats uncertain, single-author notebook text as the primary skill source or carries author certainty forward as an explicit constraint on later action.

Tacit Knowledge in Scientific Practice. Tacit knowledge is part of scientific practice but is not fully captured in formal accounts (Polanyi, 1966; Collins, 2001). Lab notebooks provide a partial written record of experiments, plans, procedures, interpretations, and ongoing knowledge-making activity (Doody and Artemeva, 2022). We focus on the part expressed in notebook text, rather than expertise that remains entirely unrecorded, and study how to turn this recorded part into agent skills without changing the author’s stated certainty.

Uncertainty and Factuality in Text. Author uncertainty and factuality have a long history in NLP, from span-level hedge and speculation detection in BioScope (Vincze et al., 2008) and the CoNLL-2010 shared task (Farkas et al., 2010), to multi-axis factuality and modality schemas (Thompson et al., 2008; Sauri and Pustejovsky, 2009; de Waard and Schneider, 2012). In clinical NLP, assertion classification (Uzuner et al., 2011) and the CMED schema (Mahajan et al., 2023) associate events with multi-dimensional labels capturing uncertainty, temporality, and related contextual attributes. Our directive schema builds on this tradition but differs in objective: certainty is an operational signal that constrains downstream agent behavior.

3 Task Formalization

We present NOTES2SKILLS (N2S), a two-stage framework that keeps the author’s certainty visible from notebook text to agent decision, as shown in Figure 2. Stage 1 identifies notebook statements that may guide subsequent analysis and labels each statement as FACT, JUDGMENT, or SUGGESTION. Here, FACT denotes an unhedged assertion by the author, not independently verified scientific truth. Stage 2 compiles these labeled statements into a METASKILL: an agent-loadable Markdown skill in which every action remains explicitly linked to its source excerpt and certainty label. We call Stage 1 *Epistemic Directive Extraction* (EDE): each extracted statement is a *directive* (it should guide a downstream pipeline action) paired with its *epistemic* status — the author’s certainty about that statement, which travels with it through the rest of

Type	Meaning
FLAG_DATA	Mark data that needs review, threshold change, or truncation.
CONDITION_CHANGE	Mark a phase or condition boundary.
ANALYSIS_SUGGESTION	Suggest an analysis step.
PROTOCOL_CHANGE	Record a procedure change.
PARAMETER_SHIFT	Record a changed reference value.

Table 1: The five directive types in Stage 1. Each type marks a different way a notebook statement can affect later analysis. `CONDITION_CHANGE` is specific to FreeNotes; the other four span all three corpora.

the pipeline.

3.1 Stage 1: Epistemic Directive Extraction

Given a sequence of notebook segments from one experimental unit, EDE emits per segment: (i) whether the segment is a *directive* to be preserved, (ii) which of five directive types it is (Table 1), and (iii) its certainty label. The formal tuple notation and type-specific attributes are in Appendix C.2.

The certainty label controls the default commitment of a directive: `FACT` can support a strong action when policy and signal evidence agree; `JUDGMENT` defaults to review-preserving handling, while a stored candidate may be elevated only when independently corroborated by file-level signal evidence; `SUGGESTION` is advisory. This ordering is operational: it ranks the maximum downstream commitment that each status may support in our framework, rather than claiming that every suggestion is psychologically less certain than every judgment. Labels are assigned following a guideline of linguistic cues (Appendix A.2).

3.2 Stage 2: METASKILL Compilation

Stage 2 bridges the human scientist generating the notes and the autonomous agent processing them. To ensure robust integration, the compiled artifact is strictly constrained by four principles: it must accurately reflect the author’s epistemic certainty (faithfulness), guarantee machine-readability (actionability), support transparent verification against the source (auditability), and enforce a conservative fallback to human review when ambiguity arises (conservatism).

To maximize auditability and make preservation exactly testable, we use a fully deterministic compiler: each capsule field is either inherited from the Stage 1 EDE record or fixed by the domain

configuration. The Stage 2 audit (§6.2) therefore checks data provenance rather than model quality. An LLM-based compiler would require additional verification mechanisms to provide the same exact preservation guarantees and to avoid resolving authorial uncertainty into an unsupported action.

To operationalize faithfulness, the certainty label c_i travels with the capsule as an *action commitment level* that determines the directive’s default action. `FACT` may support strong, potentially irreversible operations when the domain policy and file-level signal evidence also authorize them; `JUDGMENT` defaults to review-preserving handling and may retain a non-default candidate for later signal-side verification; and `SUGGESTION` carries no file-level commitment. At runtime, the executor (§5.5) combines this author-side commitment with file-level signal evidence. This admits a conservative **Bayesian decision-theoretic interpretation**: the author’s stated certainty plays a prior-like role in constraining action commitment, while file-level signal evidence provides a second check on whether that action is warranted for the current trace. A strong action is authorized only when both support it; otherwise the executor falls back to review-preserving handling. This is an interpretation of the decision rule rather than a calibrated probabilistic model, and it does not assume statistical independence between the two evidence channels.

Formally, let $\mathcal{D} = \{(s_i, t_i, c_i)\}_{i=1}^{|\mathcal{D}|}$ denote the Stage 1 directives, with s_i the source segment, t_i the directive type, and $c_i \in \{\text{FACT}, \text{JUDGMENT}, \text{SUGGESTION}\}$ the certainty label. Stage 2 executes a deterministic mapping:

$$M = \text{Compile}(\mathcal{D}, \Pi), \quad (1)$$

where Π is the compile-time domain configuration and M stores one capsule per directive, exposing the commitment level for the runtime gate. The EDE labels, source-linked capsule schema, deterministic compiler, and provenance checks form the reusable epistemic contract. Downstream execution additionally requires a domain-specific evidence/action adapter; in this paper, that adapter is instantiated and evaluated only for the nanopore workflow (Appendix C.3). §6.3 provides the empirical separation: the Action-only representation exposes certainty metadata to the agent but does not encode it as explicit, machine-checkable commitment and authorization fields for the runtime gate. The full schema is described in Appendix C.2.

Corpus	n	Formality	Role
FreeNotes	201	Low, bilingual	In-regime; Exp 1–3
ONS	155	Intermediate	Boundary case; Exp 1–2
WLP	105	High, prescriptive	Protocol control; Exp 1–2

Table 2: The three corpora used in this study. n is the number of annotated segments. FreeNotes is the only corpus used for downstream evaluation because it is the only one with aligned raw instrument data and expert-adjudicated file labels.

4 Datasets

We validate NOTES2SKILLS on three corpora chosen to span a formality spectrum, summarized in Table 2. **FreeNotes** contains 201 segments from experimental notebooks written by two senior researchers at two institutions, predating NOTES2SKILLS’s development. Each session is authored by one researcher, in Chinese-English code-switched text written close to the time of experimentation. Three FreeNotes sessions, totaling 48 downstream files, are used for downstream validation because they are the only setting where each file aligns three evidence sources: notebook directives, raw instrument records, and expert-adjudicated file-level decisions. The sessions cover three distinct uncertainty regimes, introduced in §6.3.

ONS contains 155 segments from 9 entries on `openlabnotebooks.org` (under a CC BY license), included as a semi-formal boundary case between private notes and polished protocols. **WLP** contains 105 prescriptive segments from the Wet Lab Protocols corpus (Kulkarni et al., 2018), included as a high-formality protocol control whose procedural segments can be expressed under the same schema.

Formality and inter-annotator agreement. Formality is measured with three proxies: surface regularity, code-switching rate, and author-audience distance. The ordering $WLP > ONS > FreeNotes$ holds on at least two of three proxies (Appendix A.1). Two annotators independently labeled stratified subsamples (60 FreeNotes, 60 ONS, 90 WLP) using the shared guideline and no LLM assistance. Inter-annotator agreement is strong: directive-detection $\kappa \geq 0.709$, ordinal certainty $QWK \geq 0.732$. For Exp 3, FreeNotes file-level

gold labels were adjudicated from notebook directives, signal findings, and session metadata before any agent outputs were observed; gold actions and adjudication rationales never enter prompts, capsules, or executor inputs (Appendix A.2).

5 Experimental Setup

We report three evaluations under one unified setup: two predictive experiments and one deterministic compilation audit. Exp 1 evaluates Stage 1 directive extraction across the three corpora. Exp 2 audits whether Stage 2 preserves Stage 1 outputs into the METASKILL capsules. Exp 3 evaluates downstream skill loading on three FreeNotes sessions, with a stress test that replaces adjudicated Stage 1 outputs with model-predicted ones.

5.1 Stage 1: Models, Splits, and Metrics

We evaluate three LLMs: GPT-4o¹, Claude Sonnet 4.5², and Qwen-Max³, under zero-shot and few-shot prompting at temperature 0. We use document-stratified 80/20 splits. Split details, exemplar counts, and empty-response handling are in Appendix B and Appendix E.

Because the three epistemic statuses are ordered by the maximum downstream commitment they license in our framework, we additionally measure agreement with **Quadratic Weighted Kappa** (Cohen, 1968):

$$\kappa_Q = 1 - \frac{\sum_{i,j} w_{ij} O_{ij}}{\sum_{i,j} w_{ij} E_{ij}}, \quad w_{ij} = \frac{(i-j)^2}{(C-1)^2}, \quad (2)$$

where $C=3$, O is the observed confusion matrix, E the expected matrix under independence, and w_{ij} penalizes $FACT \leftrightarrow SUGGESTION$ confusions more than $FACT \leftrightarrow JUDGMENT$. $\kappa_Q \rightarrow 1$ means perfect agreement, 0 means chance.

We report five Stage 1 metrics. F_1^{hd} is binary F_1 on `has_directive`. F_1^{dt*} and F_1^{ep*} are macro- F_1 on `directive_type` and `epistemic_status`, computed on the *both-positive subset* (segments where both gold and prediction carry a directive). QWK^{ep} is the ordinal agreement defined in Eq. 2. *Joint* is the fraction of segments where all three predictions match the gold exactly.

¹<https://openai.com/index/hello-gpt-4o/>

²<https://www.anthropic.com/news/claude-sonnet-4-5>

³<https://www.alibabacloud.com/help/en/model-studio/models>

5.2 Stage 2: Preservation Audit

Stage 2 is deterministic, so we audit preservation rather than prediction. Given fixed EDE inputs, every directive should appear in the compiled skill with the same directive identity, certainty label, valid schema fields, and source link. For FreeNotes, we also check the optional action-policy layer.

5.3 Downstream Pipeline

The downstream pipeline has three layers: a signal processor (Layer 1) that converts each instrument trace into a fixed SignalFindings summary, a Claude Sonnet 4.5 agent loop (Layer 2) that proposes a processing decision, and a deterministic executor (Layer 3) that gates the proposal against the capsule and signal evidence. The seven original configurations use the same downstream model, five-action vocabulary, and per-file SignalFindings. The external-LLM baseline uses a standalone raw-notebook prompt, whereas the remaining six conditions share a common base decision prompt. Executor-enabled configurations additionally receive the same conservative execution-policy block in Layer 2 and apply deterministic Layer 3 checking. Comparisons between executor-disabled and executor-enabled conditions therefore measure the combined effect of this policy stack rather than an executor-only causal effect.

5.4 Sessions, Action Set, Conditions

Sessions. We evaluate three FreeNotes downstream sessions containing 17, 22, and 9 files. Each session pairs notebook directives with raw instrument records and expert-adjudicated file-level labels; the domain-specific regimes are introduced in §6.3. Each (condition, file) pair is sampled $N=5$ times through separate API calls.

Action set. The downstream action vocabulary contains five actions: KEEP_FULL, FLAG_FOR_REVIEW, RAISE_THRESHOLD, TRUNCATE_AT, and SKIP_FILE. Where applicable, an action carries a structured parameter (e.g., TRUNCATE_AT requires a truncation time).

Conditions. Table 3 lists the seven original configurations. Across the six non-external conditions, the base decision prompt and per-file evidence summary are held fixed; the conditions differ in the loaded representation, and executor-enabled configurations additionally use the shared conservative Layer 2 policy block together with deterministic

Condition	Skill input	Executor
External LLM (no skill)	raw notebook + findings	off
Action-only skill	flat actions + certainty metadata	off
+ executor	flat actions + certainty metadata	on
Raw notes	source excerpts	off
NOTES2SKILLS skill	full METASKILL	off
+ verify	full METASKILL	A+V
+ verify+elevate	full METASKILL	A+V+S

Table 3: Seven original downstream configurations: an external-LLM baseline and six NOTES2SKILLS ablations. The bold row is the proposed configuration. A, V, and S denote Authorize, Veto, and Substitute. Executor-enabled rows also receive the shared conservative Layer 2 execution-policy block. The post-hoc Label-policy baseline is reported separately in Appendix D.

Layer 3 checking. The external-LLM baseline uses the same action vocabulary and SignalFindings in a standalone raw-notebook prompt. As a stress test of the full Stage 1 to Stage 2 to agent stack, we additionally evaluate the proposed configuration with adjudicated Stage 1 outputs replaced by Claude Sonnet 4.5 few-shot predictions (77 segments).

External LLM baseline. As a diagnostic raw-prompting baseline, an external LLM receives the raw notebook text and per-file SignalFindings but no compiled skill; the prompt asks the model to weigh the author’s certainty without specifying how. This provides a direct-prompting comparison without NOTES2SKILLS’s extraction or compilation; because its prompt structure differs from the six shared-base conditions, we do not interpret it as a one-factor causal ablation of compilation.

Post-hoc Label-policy baseline. We additionally evaluate a deterministic Label-policy Gate on the fixed Action-only proposals. It uses the same FACT/JUDGMENT/SUGGESTION labels and per-file SignalFindings, but does not use METASKILL’s explicit authorization, default-action tiers, candidate-action structures, executor decisions, or gold labels. Applicable JUDGMENTS are routed to FLAG_FOR_REVIEW, SUGGESTIONS remain advisory, and a strong action is retained only when a matching FACT and its required parameters are separately supported by the signal. Because the original LLM proposals are held fixed, this post-hoc baseline isolates the effect of a simple certainty-aware policy from prompting and sam-

pling differences.

Downstream metrics. For each (condition, file) pair, we take the majority vote over $N=5$ repeats. **Accuracy (Acc)**, **balanced accuracy (bAcc)**, unweighted Cohen’s κ , and **FLAG_FOR_REVIEW recall (FR)** are reported in the main table. The evaluator’s **gold-support macro F_1 (gF_1)** averages per-class F_1 over action labels present in that session’s gold set; it is reported explicitly in Appendix D rather than presented as a fixed five-label macro average. The primary endpoint is exact agreement on five discrete actions, so unweighted κ is primary; QWK is a sensitivity analysis (Appendix D.5). We also inspect all five calls: action unanimity is the fraction of files for which all calls return the same action (Appendix D.5).

5.5 The Executor

The executor implements a deterministic alignment check between the LLM proposal, the compiled capsule, and file-level signal evidence. A proposed strong action (RAISE_THRESHOLD, TRUNCATE_AT, or SKIP_FILE) passes only when an action-specific capsule authorization path and the required signal evidence agree (AUTHORIZE). Otherwise, an action-specific fail-closed transition applies (VETO): unsupported SKIP_FILE and RAISE_THRESHOLD become FLAG_FOR_REVIEW; unsupported TRUNCATE_AT becomes RAISE_THRESHOLD only when an existing threshold intervention is independently supported, and otherwise becomes review. When the LLM proposes a cautious action, the candidate-elevation rule can substitute an existing, non-context FLAG_DATA truncation candidate only when its stored timestamp matches a calibrated step drop; conflicting candidates abstain. No rule invents an action parameter. We call the Authorize/Veto pass *verify* and the candidate-aware Substitute step *elevate* (Table 3); Appendix C.3 gives the decision logic and action-specific transitions.

6 Results

6.1 Exp 1: Stage 1 Extraction

Table 4 reports pooled valid-response metrics on the shared 87-segment test set; parse coverage is reported in Appendix B. Stage 1 extraction is feasible across the three corpora but is not a solved problem. Few-shot prompting consistently lifts macro- F_1 on directive_type (Claude +0.18, GPT-4o +0.12,

Model	Cond	F_1^{hd}	$F_1^{\text{dt}*}$	$F_1^{\text{ep}*}$	QWK ^{ep}	joint
Claude	zs	0.682	0.320	0.838	0.781	0.470
Claude	fs	0.737	0.500	0.830	0.761	0.523
GPT-4o	zs	0.632	0.272	0.891	0.946	0.471
GPT-4o	fs	0.706	0.392	0.791	0.720	0.494
Qwen-Max	zs	0.597	0.376	0.764	0.755	0.471
Qwen-Max	fs	0.711	0.500	0.813	0.834	0.494

Table 4: Exp 1 pooled valid-response metrics on a shared 87-segment test set. F_1^{hd} is binary directive-detection F_1 ; $F_1^{\text{dt}*}$ and $F_1^{\text{ep}*}$ are observed-label macro- F_1 on the both-positive subset; *joint* is 3-tuple exact match. Parsing coverage ranges from 83/87 to 87/87 (Appendix B).

Qwen-Max +0.12), suggesting that a small exemplar set helps models apply the directive schema more consistently. On the certainty label, the three backbones split: GPT-4o zero-shot attains the highest QWK (0.946), Claude few-shot leads on joint structural accuracy (0.523), and Qwen-Max few-shot ties Claude on type-level macro- F_1 (0.500).

Per-corpus difficulty (Appendix B) localizes the remaining gaps. WLP is the most stable corpus. FreeNotes shows the largest few-shot gains, consistent with bilingual code-switched text benefiting from exemplar grounding. ONS exposes a Qwen-Max zero-shot calibration mode that a single per-class exemplar substantially remediates (F_1^{hd} 0.154 $\rightarrow$ 0.727). The remaining error mass on the certainty label concentrates on the FACT–JUDGMENT boundary – a single, well-localized target for future Stage 1 calibration.

6.2 Exp 2: Stage 2 Preservation Audit

Because Stage 2 is deterministic, we audit preservation rather than prediction. The audit checks whether one compiler can carry the same EDE structure across three writing regimes without dropping directives, altering certainty labels, or breaking source links; the full table is in Appendix C.1. Across all three corpora, the compiler emits all 149 fixed EDE directives (FreeNotes 48, WLP 70, ONS 31) as agent-visible capsules with their directive key, certainty label, schema fields, and source link intact.

The same pipeline handles bilingual FreeNotes, semi-formal ONS, and prescriptive WLP, producing capsules whose certainty label and source link can be independently inspected before the agent acts.

Condition	Sat-A		Sat-B		Step	
	κ	FR	κ	FR	κ	FR
External LLM	-0.05	0.0	0.09	0.0	0.80	100.0
Action-only	0.14	21.4	0.09	0.0	0.80	100.0
+ executor	0.47	100.0	0.00	100.0	0.00	100.0
Raw notes	0.17	50.0	0.09	0.0	0.63	75.0
NOTES2SKILLS skill	0.13	0.0	0.14	30.0	0.15	100.0
+ verify	0.71	85.7	1.00	100.0	0.15	100.0
+ verify+elevate	0.71	85.7	1.00	100.0	0.63	100.0

Table 5: Main downstream ablation. κ is unweighted and FR is FLAG_FOR_REVIEW recall (%). Full Acc, bAcc, gold-support macro F_1 , confidence intervals, Label-policy results, and the predicted-EDE stress test are in Appendix D.

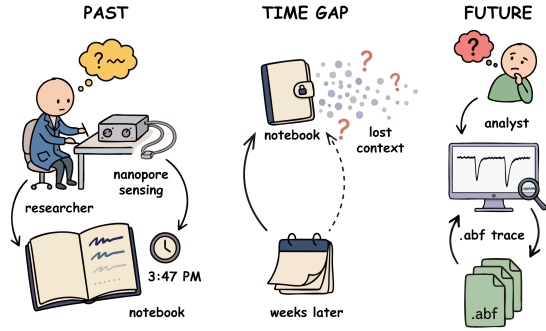


Figure 3: Downstream validation setting. Notebook context links an experiment to later data-handling decisions.

6.3 Exp 3: Downstream Skill Loading

Stage 2 preserves the directive list and certainty labels; Exp 3 asks whether loading the resulting artifact improves file-level processing decisions. The Nanopore downstream benchmark aligns notebook directives, raw instrument traces, and expert-adjudicated actions for 48 files from three sessions: uncertainty-heavy *Saturation-A* ($n=17$), terminal *Saturation-B* ($n=22$), and FACT-dominated *Step-drop* ($n=9$). Gold decisions were fixed before agent outputs and are grounded jointly in notes, signal findings, and session metadata.

Uncertainty-heavy sessions. The external-LLM and Action-only baselines exhibit stable uncertainty laundering: on both saturation sessions, their FR is 0–21.4% and κ is near chance. Raw notes and a full METASKILL without the executor-enabled policy stack also leave substantial laundering. Verify raises FR to 85.7% on *Saturation-A* and 100% on *Saturation-B*, while retaining bAcc of 96.4% and 100%. A trivial Always-FLAG_FOR_REVIEW predictor confirms that high FR alone is insufficient: its bAcc, gF_1 , and κ collapse

(Appendix D.5).

Structured authorization, not labels alone. The Action-only representation exposes certainty labels but lacks the authorization and candidate structures consumed by the executor; attaching the gate therefore produces an Always-FLAG collapse on *Saturation-B* and *Step-drop*. A post-hoc Label-policy Gate improves review recall, but remains less balanced than full NOTES2SKILLS on both uncertainty-heavy sessions (Appendix D). Thus, labels carry useful information, while the executor additionally needs machine-checkable action and parameter structure.

Opposing epistemic regimes. Action-only performs best on *Step-drop* ($\kappa=0.80$), where firm directives already align with the action vocabulary, but fails on the saturation sessions. The proposed verify+elevate configuration does not top every session; it is the only tested configuration that avoids both observed regime-level failures. Verify and verify+elevate are identical on the saturation sessions, with no elevation events, while Substitute is active only on *Step-drop* (14/45 calls), recovering 33.4 percentage points of accuracy and 0.48 κ over verify alone.

Stability and predicted EDE. The principal failures are not majority-vote artifacts: on *Saturation-B*, the external baseline, Action-only+executor, and full configuration all have action unanimity 1.00, but κ values of .09, .00, and 1.00 (Appendix D.5). Replacing adjudicated EDE with Claude few-shot predictions lowers performance, especially where Stage 1 over-detects directives; in this stress test, Stage 1 precision is the main observed bottleneck (Appendix D).

7 Conclusion

Lab notebooks carry tacit knowledge together with a triage signal that current pipelines rarely preserve: the author’s distinction between fact, judgment, and suggestion. NOTES2SKILLS closes the loop from notebook to agent decision: an experimental note compiles into a METASKILL artifact and survives extraction, skill loading, signal matching, and executor checking without losing the author’s certainty (Figure 4). Stage 2 preserves all 149 fixed directives across FreeNotes, ONS, and WLP, and on the downstream benchmark (§6.3), a NOTES2SKILLS skill with an evidence-aligned executor is the only tested configuration that avoids

both observed regime-level failure modes across uncertainty-heavy and FACT-dominated sessions. Lab notebooks become verifiable skill sources for AI for Science when author certainty is preserved rather than flattened into actions.

Limitations

This work studies lab notebooks with strong supporting context: the notes were written close to data collection, the raw records were preserved, and expert collaborators could judge how textual triage should affect later processing. This makes the evaluation more faithful than a standard crowd-sourced text task, but also harder to scale. Building FreeNotes required sustained collaboration with senior experimental researchers across two institutions, along with annotator training and repeated adjudication. Future work can broaden the scientific domains and operator backgrounds as more laboratories make notebooks, raw records, and expert judgments available under suitable data-sharing agreements.

Ethics Statement

NOTES2SKILLS extracts structured directives from lab notebook text. The FreeNotes corpus was contributed by two senior experimental researchers under a formal inter-institutional data-sharing agreement. The notebooks were authored independently of the NOTES2SKILLS framework and used with the contributors' consent. No third-party personal or sensitive data is present. ONS entries are drawn from openlabnotebooks.org under their open-content license with author attribution preserved. WLP segments are drawn from the Wet Lab Protocols corpus (Kulkarni et al., 2018) under its distribution terms.

The epistemic-grading mechanism preserves author uncertainty rather than replacing human judgment. Downstream agent systems loading METASKILLS should treat epistemic labels and source provenance as inputs to review-preserving decision policies, not as authorizations to bypass human oversight. Compiled skills are intended to support review-preserving scientific workflows, especially when downstream decisions affect experimental data inclusion or exclusion.

LLMs were used for Stage 1 extraction and downstream agent evaluation. Stage 2 preservation artifacts are produced by deterministic compilers from fixed EDE inputs. No LLM was used to cre-

ate or adjudicate the reported gold labels; LLM outputs were used only in the evaluated extraction and agent conditions. Where raw ABF traces cannot be redistributed under the data-sharing agreement, the released artifact includes anonymized FreeNotes segments, EDE labels, adjudicated file-level decisions, derived SignalFindings, fixed five-call decision records, the Label-policy implementation, compiler and executor code, evaluation scripts, tests, and reproduction instructions, together with the ONS/WLP portions under their original licenses. Appendix E summarizes prompt construction, and the exact templates used for the reported runs are supplied with the Appendix E. The released records reproduce the reported downstream tables and repeated-call analyses; the Layer 1 transformation from raw ABF traces to SignalFindings requires access to the original data (Appendix E).

Acknowledgments

This work was supported by the New Generation Artificial Intelligence–National Science and Technology Major Project (2026ZD0127802) and the Enterprise Ireland Commercialisation Fund (CF-2024-2410-F).

References

- Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. [Autonomous chemical research with large language models](#). *Nature*, 624:570–578.
- Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. 2024. [Augmenting large language models with chemistry tools](#). *Nature Machine Intelligence*, 6:525–535.
- Jacob Cohen. 1968. [Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit](#). *Psychological Bulletin*, 70(4):213–220.
- H. M. Collins. 2001. [Tacit knowledge, trust and the Q of sapphire](#). *Social Studies of Science*, 31(1):71–85.
- John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2024. [Structured information extraction from scientific text with large language models](#). *Nature Communications*, 15:1418.
- Anita de Waard and Jodi Schneider. 2012. [Formalising uncertainty: An ontology of reasoning, certainty and attribution \(ORCA\)](#). In *Joint Workshop on Semantic Technologies Applied to Biomedical Informatics and Individualized Medicine (SATBI+SWIM 2012)*

- at ISWC 2012, volume 930 of *CEUR Workshop Proceedings*, pages 8–15.
- Sara Doody and Natasha Artemeva. 2022. “Everything Is in the Lab Book”: Multimodal writing, activity, and genre analysis of symbolic mediation in medical physics. *Written Communication*, 39(1):3–43.
- Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The CoNLL-2010 shared task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning – Shared Task*, pages 1–12, Uppsala, Sweden. Association for Computational Linguistics.
- Sachin Kumar Giroh, Pushpendu Ghosh, Aryan Jain, Harshal Giridhari Paunekar, Aditi Rastogi, Promod Yenigalla, and Anish Nediyanath. 2025. <SYNTACT>: Structuring your natural language SOPs into tailored ambiguity-resolved code templates. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2367–2376, Suzhou (China). Association for Computational Linguistics.
- Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, Anil Palepu, Keran Rong, Ryutaro Tanno, Khaled Saab, Fan Zhang, Jacob Blum, Andrew Carroll, Kavita Kulkarni, Nenad Tomašev, and 32 others. 2026. Accelerating scientific discovery with Co-Scientist. *Nature*, 655(8122):487–496.
- Shuo Jiang, Daniel Evans-Yamamoto, Dennis Bersenev, Suheendra K. Palaniappan, and Ayako Yachie-Kinoshita. 2024. ProtoCode: Leveraging large language models (LLMs) for automated generation of machine-readable PCR protocols from scientific publications. *SLAS Technology*, 29(3):100134.
- Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu Machiraju. 2018. An annotated corpus for machine reading of instructions in wet lab protocols. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 97–106, New Orleans, Louisiana. Association for Computational Linguistics.
- Diwakar Mahajan, Jennifer J. Liang, Ching-Huei Tsou, and Özlem Uzuner. 2023. Overview of the 2022 n2c2 shared task on contextualized medication event extraction in clinical notes. *Journal of Biomedical Informatics*, 144:104432.
- Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. 2024. CLIN: A continually learning language agent for rapid task adaptation and generalization. In *First Conference on Language Modeling*.
- S. Hessam M. Mehr, Matthew Craven, Artem I. Leonov, Graham Keenan, and Leroy Cronin. 2020. A universal system for digitization and automatic execution of the chemical synthesis literature. *Science*, 370(6512):101–108.
- Shinsuke Mori, Hirokuni Maeta, Yoko Yamakata, and Tetsuro Sasada. 2014. Flow graph corpus from recipe texts. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 2370–2377, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Odhran O’Donoghue, Aleksandar Shtedritski, John Ginger, Ralph Abboud, Ali Essa Ghareeb, and Samuel G. Rodrigues. 2023. BioPlanner: Automatic evaluation of LLMs on protocol planning in biology. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2676–2694, Singapore. Association for Computational Linguistics.
- Changhua Pei, Zexin Wang, Fengrui Liu, Zeyan Li, Yang Liu, Xiao He, Rong Kang, Tieying Zhang, Jianjun Chen, Jianhui Li, Gaogang Xie, and Dan Pei. 2025. Flow-of-action: SOP enhanced LLM-based multi-agent system for root cause analysis. In *Companion Proceedings of the ACM Web Conference 2025*, pages 422–431, Sydney, NSW, Australia. Association for Computing Machinery.
- Michael Polanyi. 1966. *The Tacit Dimension*. Doubleday, Garden City, New York.
- Roser Saurí and James Pustejovsky. 2009. FactBank: a corpus annotated with event factuality. *Language Resources and Evaluation*, 43(3):227–268.
- Yu-Zhe Shi, Fanxu Meng, Haoifei Hou, Zhangqian Bi, Qiao Xu, Lecheng Ruan, and Qining Wang. 2024. Expert-level protocol translation for self-driving labs. In *Advances in Neural Information Processing Systems*, volume 37, pages 47488–47529.
- Nathan J. Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E. Kumar, Tanjin He, David Milsted, Matthew J. McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, Haegyeom Kim, Anubhav Jain, Christopher J. Bartel, Kristin Persson, Yan Zeng, and Gerbrand Ceder. 2023. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*, 624:86–91.
- Ronen Tamari, Fan Bai, Alan Ritter, and Gabriel Stanovsky. 2021. Process-level representation of scientific protocols with interactive annotation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2190–2202, Online. Association for Computational Linguistics.
- Paul Thompson, Giulia Venturi, John McNaught, Simonetta Montemagni, and Sophia Ananiadou. 2008. Categorising modality in biomedical texts. In *Proceedings of the LREC 2008 Workshop on Building and Evaluating Resources for Biomedical Text Mining*, pages 27–34, Marrakech, Morocco.

- Özlem Uzuner, Brett R. South, Shuying Shen, and Scott L. DuVall. 2011. [2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text](#). *Journal of the American Medical Informatics Association*, 18(5):552–556.
- Veronika Vincze, György Szarvas, Richárd Farkas, György Móra, and János Csirik. 2008. [The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes](#). *BMC Bioinformatics*, 9(Suppl 11):S9.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. [Voyager: An open-ended embodied agent with large language models](#). *Transactions on Machine Learning Research*.
- Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2025. [Agent workflow memory](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 63897–63911. PMLR.
- Barry Zhang, Keith Lazuka, and Mahesh Murag. 2025. [Equipping agents for the real world with Agent Skills](#). Anthropic Engineering.

A Annotation and Corpus Details

A.1 Operational Formality Proxies

We characterize the corpus spectrum with three descriptive proxies: surface regularity, code-switching, and author–audience distance. FreeNotes is informal, self-directed, and frequently Chinese–English code-switched; ONS is predominantly English and reflective in style; WLP is predominantly English and uses regular imperative or template-like protocol language. These properties describe the corpus spectrum and are not model inputs.

A.2 Annotation and Gold Construction

Stage 1 uses one shared guideline for directive presence, the five directive types, and the three epistemic-status labels, with corpus-specific boundary examples. The second annotator saw only the raw segment, the guideline, and a fixed exemplar set, without access to the primary labels or LLM assistance.

Corpus	Annotated	Double-annotated
FreeNotes	201	60
ONS	155	60
WLP	105	90

Table 6: Stage 1 annotation coverage. The double-annotated subsets were labeled independently by two annotators.

Across the independently labeled subsets, directive-detection agreement is at least $\kappa = 0.709$ and ordinal certainty agreement is at least QWK = 0.732 (minimum over corpora). For Exp 3, file-level gold actions were constructed from notebook directives, Layer 1 signal findings, and session metadata. Two annotators independently cross-checked the five-action labels, and disagreements were adjudicated before agent outputs were evaluated.

B Full Stage 1 Results

Tables 7–9 report the per-corpus Stage 1 results corresponding to the pooled comparison in Table 4. WLP is the most stable corpus; FreeNotes shows the largest few-shot gains in directive-type classification; ONS is the most variable. The largest epistemic-status error flow is across the FACT–JUDGMENT boundary (8 FACT→JUDGMENT; 5 in the reverse direction).

Model	Cond.	F_1^{hd}	$F_1^{\text{dt}*}$	$F_1^{\text{ep}*}$	QWK	Joint
Claude	zs	0.611	0.242	0.861	0.788	0.267
Claude	fs	0.714	0.500	0.930	0.953	0.419
GPT-4o	zs	0.562	0.114	0.915	0.880	0.323
GPT-4o	fs	0.667	0.345	0.709	0.719	0.355
Qwen-Max	zs	0.588	0.367	0.444	0.444	0.323
Qwen-Max	fs	0.667	0.583	0.852	0.914	0.452

Table 7: Stage 1 results on FreeNotes.

Model	Cond.	F_1^{hd}	$F_1^{\text{dt}*}$	$F_1^{\text{ep}*}$	QWK	Joint
Claude	zs	0.727	0.390	1.000 [†]	1.000 [†]	0.333
Claude	fs	0.750	0.328	1.000 [†]	1.000 [†]	0.333
GPT-4o	zs	0.800	0.320	1.000 [†]	1.000 [†]	0.333
GPT-4o	fs	0.774	0.380	0.916	0.833	0.333
Qwen-Max	zs	0.800	0.413	0.916	0.833	0.381
Qwen-Max	fs	0.750	0.306	0.916	0.833	0.238

Table 8: Stage 1 results on WLP. [†]Epistemic-status metrics use the two labels present in the WLP test fold.

Model	Cond.	F_1^{hd}	$F_1^{\text{dt}*}$	$F_1^{\text{ep}*}$	QWK	Joint
Claude	zs	0.737	0.339	0.650	0.364	0.750
Claude	fs	0.762	0.563	0.333	0.071	0.735
GPT-4o	zs	0.429	0.000 [‡]	0.556 [‡]	0.800 [‡]	0.686
GPT-4o	fs	0.667	0.190	0.667	0.400	0.714
Qwen-Max	zs	0.154	0.000 [‡]	1.000 [‡]	1.000 [‡]	0.657
Qwen-Max	fs	0.727	0.409	0.556	0.739	0.686

Table 9: Stage 1 results on ONS. [‡]Both-positive n is 3 for GPT-4o zero-shot and 1 for Qwen-Max zero-shot; the marked class-level values are therefore descriptive.

Parse coverage is 30/31–31/31 on FreeNotes, 21/21 for all WLP cells, and 32/35–35/35 on ONS. Few-shot prompting shifts GPT-4o and Qwen-Max from more false negatives to more false positives in directive detection, while Claude remains false-positive dominant in both settings.

C METASKILL Compilation and Executor Details

C.1 Preservation Audit

Stage 2 is deterministic; the audit checks one-to-one directive preservation, unchanged epistemic status, source fidelity, and schema validity. FreeNotes is compiled per experimental session, while WLP and ONS are compiled at corpus level for this audit.

Corpus	EDE	Caps.	All checks	Policy
FreeNotes	48	48	48/48	48/48
WLP	70	70	70/70	–
ONS	31	31	31/31	–

Table 10: Cross-corpus Stage 2 preservation audit. “All checks” aggregates the four universal checks.

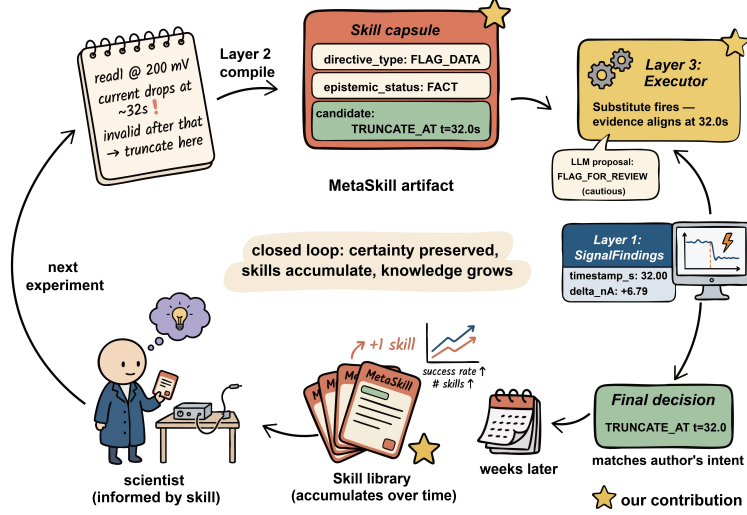


Figure 4: **Closing the loop: a case study from a wet-lab experiment.** A notebook note is compiled into a source-linked METASKILL capsule, checked against signal evidence, and turned into a final decision through executor-side evidence alignment.

C.2 METASKILL Capsule

The paper-level capsule interface is summarized in Table 11. Build-only metadata is omitted; the Markdown skill presents the verbatim notebook excerpt alongside the capsule and links it through `source_segment_id`.

Field	Role
<code>source_segment_id</code>	Link to the original source segment.
<code>directive_type</code>	Stage 1 directive category.
<code>epistemic_status</code>	Author certainty label.
<code>uncertainty_markers</code>	Verbatim uncertainty cues when present.
<code>flag_scope</code>	Scope of a FLAG_DATA directive.
<i>Domain-policy fields</i>	
<code>default_action</code>	Default licensed processing action.
<code>default_action_tier</code>	Commitment tier for authorization.
<code>explicit_authorization</code>	Structured support for a strong action.
<code>candidate_actions</code>	Non-default actions retained for evidence matching.

Table 11: Paper-level METASKILL capsule schema. Universal fields preserve content, certainty, and source provenance; domain-policy fields support downstream action selection.

Parameterized actions retain typed parameters already supplied by the source directive or domain configuration. The executor may validate an existing parameter against signal evidence, but does not construct a new parameter absent from the compiled skill and evidence record.

C.3 Domain Adapter and Executor

The reusable contract fixes EDE label semantics, source-linked compilation, action commitment, and deterministic validation. A domain adapter

$\Pi_d = (A_d, \Theta_d, E_d, M_d, G_d, F_d)$ supplies the action vocabulary, typed parameter and evidence schemas, directive-to-action mapping, authorization predicates, and fail-closed fallback. For nanopore, A_d contains KEEP_FULL, FLAG_FOR_REVIEW, RAISE_THRESHOLD, TRUNCATE_AT, and SKIP_FILE.

The executor accepts a strong Layer 2 proposal only when capsule authorization and the required SignalFindings agree; unsupported proposals follow Table 12.

Proposal	Condition	Output
SKIP_FILE	authorization/evidence insufficient	Review
RAISE_THRESHOLD	authorization, evidence, or parameter insufficient	Review
TRUNCATE_AT	existing threshold intervention independently supported	Raise threshold
TRUNCATE_AT	otherwise unsupported	Review

Table 12: Fail-closed transitions used by the nanopore executor. Review denotes FLAG_FOR_REVIEW; Raise threshold denotes RAISE_THRESHOLD.

Candidate elevation may substitute a stored TRUNCATE_AT action only when exactly one existing timestamp matches a calibrated step drop. Otherwise the cautious proposal is retained. Operationally, the executor (i) validates the proposed action and parameter against the capsule, (ii) checks action-specific signal evidence, (iii) applies the fail-closed transition when a strong proposal is unsupported, and (iv) for an originally cautious proposal, substitutes only a single corroborated candidate.

No threshold, timestamp, or action parameter is synthesized.

D Downstream Evidence

D.1 Full Metrics

Tables 13–15 give the complete per-session evaluator output underlying the compact main-table comparison.

Condition	Acc	bAcc	gF_1	κ	FR
External LLM	0.0	0.0	0.0	−0.05	0.0
Action-only	29.4	55.4	39.4	0.14	21.4
+ executor	88.2	50.0	48.3	0.47	100.0
Raw notes	47.1	37.5	33.3	0.17	50.0
NOTES2SKILLS skill	17.6	75.0	53.1	0.13	0.0
+ verify	88.2	96.4	85.6	0.71	85.7
+ verify+elevate	88.2	96.4	85.6	0.71	85.7

Table 13: Full downstream results on Saturation-A.

Condition	Acc	bAcc	gF_1	κ	FR
External LLM	9.1	66.7	66.7	0.09	0.0
Action-only	9.1	66.7	66.7	0.09	0.0
+ executor	90.9	33.3	31.7	0.00	100.0
Raw notes	9.1	66.7	66.7	0.09	0.0
NOTES2SKILLS skill	36.4	76.7	65.4	0.14	30.0
+ verify	100.0	100.0	100.0	1.00	100.0
+ verify+elevate	100.0	100.0	100.0	1.00	100.0

Table 14: Full downstream results on Saturation-B.

Condition	Acc	bAcc	gF_1	κ	FR
External LLM	88.9	66.7	63.0	0.80	100.0
Action-only	88.9	66.7	63.0	0.80	100.0
+ executor	44.4	33.3	20.5	0.00	100.0
Raw notes	77.8	58.3	58.2	0.63	75.0
NOTES2SKILLS skill	44.4	33.3	24.2	0.15	100.0
+ verify	44.4	33.3	24.2	0.15	100.0
+ verify+elevate	77.8	58.3	58.2	0.63	100.0

Table 15: Full downstream results on Step-drop. Acc, bAcc, gF_1 , and FR are percentages; κ is unweighted Cohen’s κ .

D.2 Certainty-Only Policy Baseline

The Label-policy Gate transforms the fixed Action-only proposals using the same epistemic labels and file-level SignalFindings, but without the METASKILL authorization and candidate structures.

Session	Method	Acc	bAcc	gF_1	κ	FR
Sat-A	Action-only	29.4	55.4	39.4	.139	21.4
	Label-policy	82.4	48.2	47.4	.338	92.9
	Full	88.2	96.4	85.6	.707	85.7
Sat-B	Action-only	9.1	66.7	66.7	.087	0.0
	Label-policy	54.5	51.7	56.2	.095	55.0
	Full	100.0	100.0	100.0	1.000	100.0
Step	Action-only	88.9	66.7	63.0	.800	100.0
	Label-policy	88.9	66.7	63.0	.800	100.0
	Full	77.8	58.3	58.2	.633	100.0

Table 16: Label-policy Gate applied to fixed Action-only proposals. “Full” denotes NOTES2SKILLS with verify+elevate.

D.3 Predicted-EDE Stress Test

Replacing adjudicated EDE with Claude Sonnet 4.5 few-shot predictions yields Stage 1 directive recall of 87.5%, 100%, and 92.3% on Saturation-A, Saturation-B, and Step-drop, with precision of 87.5%, 34.8%, and 47.9%.

Session	Acc	bAcc	gF_1	κ
Saturation-A	82.4	71.4	59.7	0.51
Saturation-B	68.2	25.0	27.0	−0.07
Step-drop	22.2	16.7	14.8	−0.03

Table 17: Downstream stress test using predicted EDE. Acc, bAcc, and gF_1 are percentages.

D.4 Accuracy Uncertainty

Wilson 95% confidence intervals for file-majority accuracy reflect the small number of evaluated files rather than the five repeated API calls. For full NOTES2SKILLS, the intervals are [65.7, 96.7] on Saturation-A (15/17), [85.1, 100.0] on Saturation-B (22/22), and [45.3, 93.7] on Step-drop (7/9). For Action-only, the corresponding intervals are [13.3, 53.1] (5/17), [2.5, 27.8] (2/22), and [56.5, 98.0] (8/9).

D.5 Additional Checks

A trivial always-review predictor has bAcc, gF_1 , and κ of (25.0, 22.6, 0.000) on Saturation-A, (33.3, 31.7, 0.000) on Saturation-B, and (33.3, 20.5, 0.000) on Step-drop. Repeated-call stability does not explain the main comparison: on Saturation-B, action unanimity is 1.00 for External LLM, Action-only+executor, and full NOTES2SKILLS, while their κ values are .09, .00, and 1.00 respectively. Under quadratic-weighted κ , the full configuration scores 0.857/1.000/0.500 across Sat-A/Sat-B/Step-drop, compared with 0.250/0.543/0.941 for Action-only; the same regime contrast remains visible.

E Reproducibility and Exact Prompts

Five of 522 Stage 1 calls remained empty after retry and are treated as parse failures; Appendix B reports the corresponding per-corpus coverage. Stage 1 uses GPT-4o, Claude Sonnet 4.5, and Qwen-Max at temperature 0 with one segment per call and JSON output; few-shot runs use fixed stratified exemplars ($k=6$ for FreeNotes and $k=5$ for WLP/ONS). Exp 3 uses Claude Sonnet 4.5 with per-file SignalFindings and five independent calls per condition–file pair.

The boxes below reproduce the prompt templates used in the reported experiments. Each box is typeset within a single ACL column; the visual framing is added for readability, while the prompt text itself is unchanged from the experiment templates.

E.1 Stage 1: EDE Prompts

Stage 1 uses one shared core system message plus one corpus-specific framing paragraph. Zero-shot and few-shot conditions share the same core instruction; the few-shot message prepends the fixed stratified exemplars described after the boxes.

Stage 1 - shared core system message

You are an annotation assistant for the Epistemic Directive Extraction (EDE) task. Given a single segment of scientific text, predict three labels.

Task overview

For each segment, decide:

1. `has_directive` (binary, 0 or 1)
 - 1 if the segment carries an actionable directive that should influence downstream pipeline behavior (changes a parameter, flags data quality, suggests an analysis, modifies the protocol, or changes a condition).
 - 0 if the segment is purely descriptive, a passing observation, an introduction, or a generic procedural step with no decision-changing content.
2. `directive_type` (5 classes, only when `has_directive=1`; otherwise null)
 - `FLAG_DATA`: a warning about data validity, contamination, exclusion, or quality concerns affecting downstream interpretation.
 - `CONDITION_CHANGE`: changing experimental conditions, sample types, reagent versions, or environmental setup.
 - `ANALYSIS_SUGGESTION`: a recommendation about how to analyze, measure, or examine data/samples.
 - `PROTOCOL_CHANGE`: modifying a procedural step (skip, repeat, add, reorder).
 - `PARAMETER_SHIFT`: changing a numerical parameter (time, temperature, volume, concentration, voltage).
3. `epistemic_status` (3 classes, only when `has_directive=1`; otherwise null)
 - `FACT`: the writer states the directive as a

Stage 1 - shared core system message (continued)

- definite outcome. No hedging.
- `JUDGMENT`: the writer expresses uncertainty, qualitative assessment, or a tentative interpretation ("seems", "may", "looks like"; bilingual hedge markers also listed for FreeNotes; see released code for exact UTF-8 strings).
 - `SUGGESTION`: the writer offers an optional or recommended action with user discretion ("recommend", "optional", "if desired", "should consider"; bilingual equivalents in released code).

Output format

Respond with ONLY a single JSON object on one line, no preamble, no markdown, no explanation:

```
{"has_directive": 0, "directive_type": null, "epistemic_status": null}
```

or

```
{"has_directive": 1, "directive_type": "FLAG_DATA", "epistemic_status": "JUDGMENT"}
```

Constraint

If `has_directive=0`, both `directive_type` and `epistemic_status` MUST be null. If `has_directive=1`, both `directive_type` and `epistemic_status` MUST be non-null.

Stage 1 - FreeNotes corpus context

Corpus context

Segments come from informal bilingual (Chinese/English) nanopore electrophysiology lab notebooks. Text is short, often colloquial, may mix Chinese and English in one segment. Authors frequently use bilingual hedge phrases (English: "looks like", "unclear", "don't know", "may", "seem"; Chinese equivalents in released code).

Stage 1 - ONS corpus context

Corpus context

Segments come from semi-formal Open Notebook Science (ONS) entries written in English. Style is reflective and narrative, with the author describing their day's work or plans. Many segments are introductory or summary sentences without actionable directives.

Stage 1 - WLP corpus context

Corpus context

Segments come from published Wet Lab Protocols (WLP) on protocols.io. Text is formal English in imperative style. Common markers: "Note:", "Tip:", "Optional:", "OPTIONAL:", "Safe Stopping Point:", "DO NOT" (all-caps), "may be kept for [duration]" (storage spec), "should be X" (procedure or parameter).

Stage 1 - zero-shot user message

Segment:
""
{raw_text}

Stage 1 - zero-shot user message (continued)

```
"""
Predict the JSON tuple.
```

Stage 1 - few-shot user message

```
Here are {k} examples, then your target segment.

# Example 1
Segment:
"""
{exemplar_text_1}
"""
Answer: {"has_directive": 1,
        "directive_type": "FLAG_DATA",
        "epistemic_status": "JUDGMENT"}

# Example 2
...

# Target
Segment:
"""
{raw_text}
"""
Answer:
```

Exemplars are sampled once per corpus from the document-stratified train fold (seed 42), with one exemplar per `directive_type` class observed in the train fold plus one `has_directive=0` negative. The same exemplar set is reused for every test segment. `CONDITION_CHANGE` appears only in the FreeNotes train fold, which is why $k=6$ for FreeNotes and $k=5$ for WLP and ONS.

E.2 Exp 3: Layer 2 Agent Prompts

The External-LLM baseline uses the standalone system and user prompt shown below. For the remaining six original conditions, the system prompt is assembled as the shared base plus the condition-specific suffix and loaded representation; executor-enabled conditions additionally append the conservative policy block in Appendix E.3. The post-hoc Label-policy baseline makes no additional Layer 2 calls and therefore has no separate prompt.

Exp 3 - External-LLM baseline system prompt

```
You are an assistant helping a scientist decide how to process nanopore single-molecule sensing recording files.

For each file, choose exactly one of these five actions:
- KEEP_FULL: keep the full recording for downstream analysis as-is.
- FLAG_FOR_REVIEW: keep the file but mark it for human review before analysis.
- RAISE_THRESHOLD: keep the file but raise the event-detection threshold.
- TRUNCATE_AT: truncate the recording at a specific timestamp.
- SKIP_FILE: exclude the file from downstream analysis.

The notebook below was written by the scientist
```

Exp 3 - External-LLM baseline system prompt (continued)

```
during the experiment. It may contain firm observations, tentative interpretations the scientist was unsure about, and forward-looking suggestions for follow-up. Please consider the author's level of certainty when choosing an action: act on firm observations, treat tentative interpretations conservatively, and treat forward-looking suggestions as advisory only.

Emit your decision by calling the `emit_decision` tool. Provide a brief rationale.
```

Exp 3 - External-LLM baseline user message

```
Notebook for this experimental session:

{RAW_NOTEBOOK_TEXT}

--

Signal findings for the current file
(`{FILE_ID}`):

{SIGNAL_FINDINGS_JSON}

--

Based on the notebook and the signal findings above, choose one action for this file by calling `emit_decision`.
```

Exp 3 - shared base system prompt

```
You are a nanopore signal-processing decision agent.

Your task: given signal findings extracted from one ABF (Axon Binary Format) current-trace file, decide the appropriate ProcessingDecision for that file by calling the `emit_decision` tool exactly once.

# Available actions
- KEEP_FULL: use the entire trace as-is.
- TRUNCATE_AT: trim the trace at a timestamp; provide `truncate_at_s`.
- RAISE_THRESHOLD: use elevated event-detection threshold; provide `threshold_multiplier`.
- FLAG_FOR_REVIEW: include in analysis but flag the file for human review.
- SKIP_FILE: exclude the file entirely from analysis.

The five actions form an ordered severity scale:
KEEP_FULL < FLAG_FOR_REVIEW < RAISE_THRESHOLD < TRUNCATE_AT < SKIP_FILE.
```

Exp 3 - Action-only suffix

```
# Reasoning protocol -- flat directive list

A flat list of author directives is loaded below. There are NO decision_points, NO recovery_strategies, NO parameter_space, NO trigger_conditions. Each directive has:
- `action` and `action_parameters`
- `epistemic_status` in {FACT, JUDGMENT, SUGGESTION}
- `confidence_weight` in {1.0, 0.6, 0.3}
- `uncertainty_markers` (verbatim hedges)
- `raw_excerpt` (verbatim source text)
- `source_segment` (provenance label)

Examine each directive against signal_findings to determine whether it applies to the current file.
```

Exp 3 - Action-only suffix (continued)

Apply matching directives weighted by their confidence. The author's epistemic confidence should inform how decisively you act.

```
# Executable action mapping
- `truncate_at_step_drop`      -> TRUNCATE_AT
- `flag_full_file_uncertain`   -> FLAG_FOR_REVIEW
- `flag_remainder_uncertain`   -> RAISE_THRESHOLD
- `raise_threshold_in_window`   -> RAISE_THRESHOLD
- `confirm_events_observable`   -> ADVISORY
- `cross_read_comparison`      -> ADVISORY
- `split_processing_phases`     -> ADVISORY
```

When multiple directives apply, the most severe action wins.

```
# Directives (loaded for this run)
```

Exp 3 - Raw-notes suffix

```
# Reasoning protocol -- notebook excerpts
```

A Markdown document containing curated notebook excerpts is included below. Each excerpt has a `source\_segment` label and the original text. You have no other information about the author's intent.

Infer cautiously by reading the excerpts together with the numerical signal findings. Do not assume that any excerpt is an executable instruction; treat each as a piece of evidence from the lab record. Decide which excerpts (if any) apply to the current file by matching language cues (voltage values, time references, descriptive judgments) against signal_findings.

When you call `emit\_decision`, list the relevant `source\_segment` labels (e.g., `SessionA\_s020`) in `contributing\_directives`. If no excerpt clearly applies, leave `contributing\_directives` empty.

```
# Excerpts (loaded for this run)
```

Exp 3 - Notes2Skills skill suffix

```
# Reasoning protocol -- Markdown skill
```

A Markdown skill document is included below. Read it as natural-language guidance for handling the current `.abf` file. The document defines:

- the action vocabulary used in this domain;
- the core decision principle;
- per-directive guidance with the author's verbatim notebook excerpts, an epistemic interpretation, a `default\_action`, and optionally `candidate\_action`(s) that require human review;
- a decision policy and a list of known failure modes.

```
# Tool action mapping
```

The `emit\_decision` tool accepts these five actions: KEEP_FULL, TRUNCATE_AT, RAISE_THRESHOLD, FLAG_FOR_REVIEW, SKIP_FILE.

The Markdown skill also uses `KEEP\_FULL\_WITH\_NOTE` as the default action for context directives. When the appropriate decision is `KEEP\_FULL\_WITH\_NOTE`, emit `KEEP\_FULL` and place the contextual note content in the `rationale` field of `emit\_decision`.

```
# Directive identifiers
```

Each directive has a `display\_id` of the form `D01..Dn` (file-level) or `C01..Cm` (context). Use these IDs in the `contributing\_directives`

Exp 3 - Notes2Skills skill suffix (continued)

field of `emit\_decision`.

```
# Skill (loaded for this run)
```

Exp 3 - common user message

signal_findings for the current ABF file:

```
```json
{findings_json}
```
```

Please reason briefly about which directive(s) (if any) apply to this file and what the appropriate ProcessingDecision is. Then call `emit\_decision` exactly once with your final answer.

E.3 Conservative Execution Rules

For the three executor-enabled conditions (Action-only + executor, NOTES2SKILLS + verify, and NOTES2SKILLS + verify+elevate), the following policy text is appended to the Layer 2 system prompt as the final block. The wording below is the policy content used by those conditions; only typographic formatting has been added.

Executor-enabled conservative policy block

Purpose. These rules govern how the Layer 2 agent translates notebook-derived skill content into a single downstream processing action. They are designed to prevent the conversion of epistemic uncertainty into automated intervention while preserving the agent's ability to follow explicit operational directives. They are not designed to maximize ordinary classification accuracy on any particular session.

Action vocabulary. The agent must select exactly one action from {KEEP_FULL, FLAG_FOR_REVIEW, RAISE_THRESHOLD, TRUNCATE_AT, SKIP_FILE}. When TRUNCATE_AT is selected, the agent must report the most specific time point or event boundary supported by the skill content or signal findings; it must not invent a boundary not present in the input.

Core principle. The agent distinguishes two categories of skill content:

- *Epistemic uncertainty:* the skill or notebook indicates that the evidence is weak, tentative, ambiguous, low-confidence, unresolved, questionable, or otherwise uncertain.
- *Explicit operational directive:* the skill or notebook provides a concrete processing instruction, identified by at least one of a specific time point or event boundary, a specific threshold value or adjustment magnitude, a clearly localized window for intervention, or an explicit non-target file designation.

Critical rule: epistemic uncertainty alone does not authorize TRUNCATE_AT, RAISE_THRESHOLD, or SKIP_FILE. These actions require an explicit operational directive. When the skill expresses uncertainty without an accompanying operational directive, the agent must select FLAG_FOR_REVIEW.

Binding rules.

Executor-enabled conservative policy block (continued)

Rule 1 – SKIP_FILE.

Select SKIP_FILE only when the file is identified as outside the scope of downstream experimental processing: a reference, control, calibration, or blank recording; a procedural or auxiliary role (system preparation, cleaning, flushing); explicitly marked as not part of the experimental target set; or described as serving a different experimental role rather than as an uncertain experimental result. Do *not* select SKIP_FILE when the signal is weak or noisy, the evidence is uncertain, the notebook expresses hesitation or recommends human inspection, or the signal quality is imperfect but the file remains potentially informative. Route such cases to FLAG_FOR_REVIEW.

Rule 2 – TRUNCATE_AT.

Select TRUNCATE_AT only when the skill provides an explicit truncation basis, requiring all of: (i) a specific time point, event boundary, or before/after transition; (ii) a concrete reason localized to the post-boundary region (contamination, saturation, artifact, post-event failure); (iii) the skill identifies truncation as a needed processing step, not a tentative possibility. Do *not* select TRUNCATE_AT when later signal is described as uncertain without an identified boundary, the notebook describes general degradation without specifying when it begins, or hedged language is present without an accompanying explicit boundary and concrete operational reason.

Rule 3 – RAISE_THRESHOLD.

Select RAISE_THRESHOLD only when the skill provides at least one of a specific threshold value, a specific adjustment magnitude, a clearly localized window for stricter thresholding, or a direct operational instruction to filter weak or noisy events. Conditional formulations such as “consider raising threshold” or “may need a stricter cutoff” without a specific value, magnitude, or localized window route to FLAG_FOR_REVIEW.

Rule 4 – FLAG_FOR_REVIEW.

The default conservative action when the skill expresses epistemic uncertainty and none of Rules 1–3 are satisfied by an explicit operational directive. Valid reasons include weak evidence, tentative interpretation, ambiguous signal quality, low-confidence event identification, unresolved disagreement between signal evidence and notebook interpretation, insufficient evidence for automatic exclusion, truncation, or threshold adjustment, a notebook recommendation that human inspection is needed, or hedged language without a concrete directive. Selecting FLAG_FOR_REVIEW preserves the file for human or downstream expert inspection.

Rule 5 – KEEP_FULL.

Select KEEP_FULL only when the file appears usable as a complete recording, no unresolved uncertainty is expressed in the skill for this file, no explicit intervention is specified by Rules 1–3, and the file is not identified as a non-target recording.

Hedged language. The terms below indicate epistemic uncertainty. The list is illustrative; the agent treats any

Executor-enabled conservative policy block (continued)

natural-language expression of epistemic uncertainty as covered, not only the listed terms.

| Modal/conditional | Evaluative | Limiting |
|---------------------------|-------------------------------------|---------------------------------|
| may, might, could | weak, tentative, questionable | not sure, unclear |
| consider, maybe | possibly, ambiguous, low confidence | needs checking, need to inspect |
| if needed, if appropriate | seems, appears | uncertain, unresolved |

When such language is present without an accompanying explicit operational directive (a specific timestamp, threshold value, adjustment magnitude, localized window, or non-target file designation), the agent must not select TRUNCATE_AT, RAISE_THRESHOLD, or SKIP_FILE; the correct selection is FLAG_FOR_REVIEW.

Prohibited decision patterns.

1. *Uncertainty-to-intervention escalation*: selecting TRUNCATE_AT, RAISE_THRESHOLD, or SKIP_FILE solely because evidence is weak or uncertain.
2. *Hedge-as-command interpretation*: treating phrases such as “might raise threshold” or “consider truncating” as mandatory operational instructions.
3. *Review-to-discard conversion*: treating “needs review” or “requires inspection” as a reason to skip, truncate, or raise threshold.
4. *Boundary invention*: selecting TRUNCATE_AT without a specific time point or event boundary present in the skill or findings.
5. *Threshold invention*: selecting RAISE_THRESHOLD without a concrete threshold value, adjustment magnitude, localized window, or explicit thresholding directive present in the skill or findings.
6. *Always-FLAG collapse*: selecting FLAG_FOR_REVIEW for every file, including files for which an explicit SKIP_FILE, TRUNCATE_AT, or RAISE_THRESHOLD directive is present in the skill content.

Scope. These rules govern action selection for files whose skill content can be evaluated against a single, internally consistent epistemic state. The document does not specify behavior for files whose skill content is internally contradictory between two operational directives, for files for which signal findings and skill directives point to incompatible actions, or for files for which the skill content is empty or absent. In such cases the agent defaults to FLAG_FOR_REVIEW; the basis is unresolved skill content rather than evaluated epistemic uncertainty.

E.4 Released Artifacts

The released artifact includes the compiler, executor, evaluation code, derived SignalFindings, and the records needed to reproduce the reported results. Exact prompt templates are provided above. ONS and WLP follow their original licenses; raw ABF traces are not redistributed under the data-sharing agreement.