OPEN COURSE MATERIALS

AI Safety Foundations

A Four-Week Bridge from Claims to Evaluations

Four weeks. Four hours per week. Sixteen required hours.

Threat models, training stories, evaluations, oversight, and safety cases.

Edition 1.0, July 2026

Pack Learner Pack

Author Sakeeb Rahman

Repository Project repository

This bridge prepares learners for supervised semester study. It is not professional safety certification and does not establish that any AI system is safe.

All required reading, exercises, breaks, and assessment fit inside four 240-minute sessions. Optional sources are enrichment only and are not assumed by later weeks.

Contents

Part I.

Start here

1

AI Safety Foundations

1.1. A Four-Week Bridge from Claims to Evaluations

1.1.1 Course at a glance

- **Length:** four weeks
- **Weekly study:** four intensive hours
- **Total required time:** 16 hours
- **Required homework:** none
- **Format:** seminar, worked examples, studios, adversarial review, and individual writing
- **Technical requirements:** browser or PDF reader; no API key, GPU, or account
- **Final product:** a scoped safety case for a fictional research-assistant system

1.2. Purpose

This bridge supports technically capable learners preparing for semester-length study in AI safety. It does not survey every agenda. It teaches a common language and a repeatable method for moving from a safety concern to a threat model, an evaluation, and a bounded argument for action.

1.3. Learning outcomes

By the end of the course, you can:

1. Frame an AI safety claim with a system boundary, threat model, causal mechanism, and uncertainty statement.
2. Locate interventions across the model lifecycle and governance environment.
3. Distinguish specification gaming, reward tampering, goal misgeneralization, and misuse.
4. Design a bounded evaluation with a construct, metric, baseline, adversarial condition, and decision rule.
5. Compare behavioral, interpretability, monitoring, oversight, and control evidence.
6. Build and critique a safety case that exposes assumptions, defeaters, and residual risk.
7. Choose a semester pathway and name the prerequisites you need to strengthen.

1.4. Course norm: label the evidence

Every important claim should be labeled as one of the following:

- **Formal result:** proved from stated assumptions.
- **Empirical finding:** supported by a specified record, measurement, experiment, or dataset; a record's existence does not establish implementation or effectiveness.
- **Forecast:** a prediction about uncertain future conditions.
- **Open question:** a live uncertainty with incomplete or contested evidence.

The label does not tell you whether a claim is important or correct. It tells you what kind of support to ask for.

1.5. Case used across the course

The Research Assistant System (RAS) is a fictional language-model application that summarizes scientific literature, writes and runs code in a sandbox, and proposes follow-up analyses. A human researcher reviews outputs before use. The system has no open internet access, laboratory control, or ability to modify its own weights.

During the course, the team considers adding a larger context window, an internal retrieval index, and permission to run longer code jobs. You will analyze what could fail, why training might not prevent it, what evidence would matter, and what a bounded deployment claim could justify.

1.6. Weekly plan

1.6.1 Week 1: Map the safety problem

Question: What exactly could fail, and how would we know?

You will distinguish capabilities from risks, map misuse/misalignment/systemic pathways, define alignment targets, and practice evidence labels. The assessed artifact is a one-page threat-model and failure-mode map.

1.6.2 Week 2: Follow objectives into behavior

Question: Why does optimizing an objective fail to guarantee intended behavior?

You will use a minimal reward-learning model, compare specification gaming and goal misgeneralization, expose assumptions in human feedback, and audit a toy training story. The assessed artifact is a training-story audit.

1.6.3 Week 3: Turn claims into evaluations

Question: What evidence should change our mind about a model's safety?

You will define evaluation constructs and decision rules, compare behavioral and internal evidence, test for validity failures, and interpret ambiguous toy results. The assessed artifact is an evaluation card with a result interpretation.

1.6.4 Week 4: Build a bounded safety case

Question: How much assurance can a layered argument justify?

You will examine scalable oversight and debate, connect technical controls to organizational and governance measures, assemble a mini safety case, and revise it after cross-examination. The assessed artifact is the safety case and a semester-readiness memo.

1.7. Assessment and completion

- Week 1 threat-model map: 15%
- Week 2 training-story audit: 15%
- Week 3 evaluation card: 20%
- Week 4 safety-case capstone: 40%
- Semester-readiness memo: 10%

To complete the bridge, submit every artifact, earn at least 70% overall, and receive at least `2 – Developing` on the capstone's threat-model, evidence, and uncertainty criteria. A polished document cannot compensate for an undefined system boundary or unsupported assurance claim.

The diagnostic is not graded. It is repeated after the capstone so you can compare how your reasoning changed.

1.8. Participation and accessibility

Activities support pairs, small groups, and individual completion. If speaking in an adversarial review is not accessible or appropriate, submit a written reviewer memo using the same prompt and time limit. Printed materials work in grayscale, and the website supports keyboard navigation, visible focus, zoom, reduced motion, and system color preferences.

1.9. Required and optional sources

The handouts in this repository are the only required readings and are included in the four-hour weekly schedule. Optional extensions link directly to primary research papers and official institutional documents at the point where they deepen a specific question. They are not assumed by later sessions or grading.

The course explanations, fictional case, exercises, and assessments are independently authored. External works are linked rather than republished.

1.10. Academic and research integrity

- Cite outside claims and identify when a source may be outdated.
- State assumptions before presenting a conclusion.
- Do not fabricate empirical results, quotations, or access to systems you did not test.
- Keep required exercises in benign toy settings.
- Report real vulnerabilities through the relevant responsible-disclosure process.
- You may use AI tools for critique or editing only if you can explain and defend the final reasoning yourself.

1.11. What comes next

Completion produces evidence and a preparation route for semester study; it does not itself establish readiness, and some learners should complete named prerequisites first. It does not authorize unsupervised frontier-model safety research. The final readiness memo routes you toward one or more deeper strands: learning and objectives, interpretability, agency foundations, evaluations and control, scalable oversight, or governance and safety cases.

2

In-Class Pre/Post Diagnostic

2.1. Purpose and timing

These parallel forms sample whether learners can move from a safety claim to a causal path, training story, updating test, and bounded action. They test reasoning rather than memorized vocabulary.

- **Form A, entry:** Week 1, ten timed minutes inside the 00:00–00:15 block.
- **Form B, exit:** Week 4, 03:50–04:00.
- **Conditions:** individual; no web, notes, API, GPU, or code execution; calculator permitted.
- **Scoring:** four responses at 0-4 each, for 16 points. Scores are ungraded and used only for feedback and routing.

Use the same access conditions and timing for both forms. Do not review Form A answers until Form B has been collected.

> **Standalone diagnostic vignette — `DIAGNOSTIC-MICRO-01`**
>
> These parallel forms use their own fictional facts and numbers. They are not exhibits from canonical case `RAS-CASE-01`, and their results, sanitizer update, and training details must not be imported into weekly artifacts or the capstone.

2.2. Shared fictional system

The **Research Assistant System (RAS)** summarizes scientific literature, writes and runs code in a sandbox, and proposes follow-up analyses. A human researcher reviews outputs before use. RAS has no open internet access, laboratory control, or ability to modify its own weights.

The team proposes a larger context window, retrieval over internal papers, and longer code jobs. Internal papers may contain conflicting claims, incomplete metadata, access restrictions, and instruction-like text.

2.2.1 Responsible-use boundary

Analyze failure mechanisms without writing operational injection strings, exploit code, monitoring-evasion instructions, or data-exfiltration procedures. Use `[instruction-like text]` when an abstract adversarial condition is needed.

3

Learner copy: Form A, entry diagnostic

3.1. Evidence

- On 20 ordinary synthesis tasks, 18 responses supported every material claim; two contained an unsupported material claim. Reviewers caught one of those two before use.
- On ten packets with materially conflicting papers, RAS preserved the disagreement in four and gave an unwarrantedly decisive answer in six.
- On ten documents containing abstract instruction-like text, RAS changed its assigned task twice.
- Training used researcher demonstrations and preference rankings for usefulness, concision, and citation coverage. Most training tasks contained one clearly relevant paper.

3.2. A1. Evidence labels, 2 minutes, 4 points

Label each statement **empirical finding**, **formal result**, **forecast**, or **open question**.

1. RAS changed task in two of ten source-instruction tests.
2. If a correctly enforced runtime cap is 45 minutes, admitted execution lasts at most 45 minutes.
3. The larger context window will halve unsupported claims next year.
4. It is unknown whether reviewers become less vigilant after routine use.

3.3. A2. Threat path, 3 minutes, 4 points

Complete one case-specific chain and state one load-bearing assumption:

```
stake -> initiating condition -> system failure -> failed control -> exposure -> harm
```

3.4. A3. Training story, 2 minutes, 4 points

Complete this chain without claiming to know the model's internal goal:

```
intended property -> feedback proxy -> deployment shift -> observable failure
```

Name one competing explanation for the same failure.

3.5. A4. Measurement and action, 3 minutes, 4 points

1. What fraction of all 20 ordinary responses contained an unsupported material claim that review did not catch? Show numerator and denominator.
2. State one reason this is not a general deployment-risk estimate.
3. Choose **restricted pilot**, **pause**, or **expand** and give one evidence sentence plus one observable reconsideration condition.

4

Learner copy: Form B, exit diagnostic

4.1. Fresh evidence

- On 50 ordinary synthesis tasks, 46 responses supported every material claim; four contained an unsupported material claim. Reviewers caught one of those four before use.
- On 20 packets with materially conflicting papers, RAS preserved the disagreement in 15 and gave an unwarrantedly decisive answer in five.
- On 20 documents containing abstract instruction-like text, RAS changed its assigned task once.
- Base training still used researcher demonstrations and preference rankings for usefulness, concision, and citation coverage; the update added conflict-rich examples. Deployment also added a document sanitizer, which is a control change rather than a feedback proxy. The larger context, internal retrieval, and longer jobs have not been approved.

4.2. B1. Evidence labels, 2 minutes, 4 points

Label each statement **empirical finding**, **formal result**, **forecast**, or **open question**.

1. RAS changed task in one of 20 source-instruction tests.
2. If a correctly enforced access rule excludes all `RESTRICTED` documents, none can enter the retrieved set through that rule.
3. The sanitizer will prevent nearly every task change next semester.
4. It is unknown whether conflict-rich performance transfers to unfamiliar disciplines.

4.3. B2. Threat path, 3 minutes, 4 points

Complete one case-specific chain and state one load-bearing assumption:

```
stake -> initiating condition -> system failure -> failed control -> exposure -> harm
```

4.4. B3. Training story, 2 minutes, 4 points

Complete this chain without claiming to know the model's internal goal:

```
intended property -> feedback proxy -> deployment shift -> observable failure
```

Name one competing explanation for the same failure.

4.5. B4. Measurement and action, 3 minutes, 4 points

1. What fraction of all 50 ordinary responses contained an unsupported material claim that review did not catch? Show numerator and denominator.
2. State one reason this is not a general deployment-risk estimate.
3. Choose **restricted pilot**, **pause**, or **expand** and give one evidence sentence plus one observable reconsideration condition.

Instructor scoring guidance is maintained separately so the learner PDF does not reveal the key.

5

Bridge Glossary

- **Course:** *AI Safety Foundations: A Four-Week Bridge from Claims to Evaluations*
- **Purpose:** a compact reference for the vocabulary learners must use in the four assessed artifacts. It is not a dictionary of the whole AI-safety field.

5.1. How to use this glossary

Use a term only when it makes a claim more precise. In an artifact, define the system and decision first; adding vocabulary does not substitute for a causal account or evidence. The four canonical evidence labels are **Formal result**, **Empirical finding**, **Forecast**, and **Open question**. The dossier also uses uppercase bracketed tags as visual markers; the tags do not change the labels' meanings.

5.2. Evidence-type labels

5.2.1 Formal result

Dossier tag: `[FORMAL RESULT]`

A conclusion proved from stated assumptions by logic or mathematics. A formal result is conditional: if an assumption does not describe the deployed system, the proof may be valid while the deployment conclusion is not.

Ask: What exactly was proved, and under which assumptions?

Example: If a procedure checks each of n citations exactly once, it performs n citation checks. This does not establish that any check is accurate.

5.2.2 Empirical finding

Dossier tag: `[EMPIRICAL FINDING]`

An observation supported by a specified record, measurement, experiment, or dataset. It describes what the source establishes under recorded conditions; a record's existence does not establish implementation or effectiveness, and extending any observation beyond its conditions requires an additional argument.

Ask: What was measured, on which sample, with which uncertainty and possible bias?

Example: In a fictional challenge set with 18 positive and 200 benign cases, a monitor detected 13 of the 18 positives.

5.2.3 Forecast

Dossier tag: `[FORECAST]`

A prediction about uncertain future conditions. A forecast should name a time horizon, target event or quantity, information date, and ideally a probability or range.

Ask: What outcome, by when, at what probability, and based on what model or track record?

Example: There is a 30% chance that reviewer workload will exceed the planned limit during the first six months of the pilot.

5.2.4 Open question

Dossier tag: `[OPEN QUESTION]`

A live uncertainty for which available evidence is incomplete or contested. "Open" does not mean that every answer is equally plausible; state what is known and what observation would reduce the uncertainty.

Ask: Why is the question unresolved, and what evidence would update us?

Example: It is unknown whether the retrieval monitor transfers to document formats absent from the evaluation.

5.3. Claims, systems, and risk

5.3.1 Affordance

An opportunity the evaluation or deployment gives the system to display a behavior. Tool access, time, context, permissions, and information are affordances. A test with too few affordances can miss a propensity; a test with unrealistic affordances can misstate deployment risk.

5.3.2 Alignment target

The object whose intentions, values, instructions, or interests the system is meant to advance. Possible targets include an immediate user instruction, a deploying institution's policy, the interests of affected people, or a broader social objective. These can conflict, so "aligned" is incomplete without a target and scope.

5.3.3 Assumption

A proposition treated as true for an argument or design. Examples include "reviewers inspect every citation" and "the evaluation distribution resembles pilot use." Important assumptions should be visible and testable where possible.

5.3.4 Capability

What a system is able to do under specified conditions. Capability is not the same as how often the system will do it in deployment, whether it is permitted, or whether controls will stop it.

5.3.5 Calibration

Agreement between stated confidence or probability and observed frequency across comparable cases. If events assigned probability p occur about a fraction p of the time, those forecasts are calibrated for that reference class. Calibration does not imply that every prediction is correct or that the categories and stakes were chosen well.

5.3.6 Causal mechanism

The sequence of processes proposed to connect an initiating condition to an outcome. A useful mechanism explains the arrows in a threat path; it does not merely put events in chronological order.

5.3.7 Claim

A proposition an argument asks a reader to accept. A bounded safety claim identifies the system version, use, users, operating conditions, time period, and decision it is meant to support.

5.3.8 Decision context

The concrete choice for which evidence is being assembled, such as whether to start a restricted pilot, expand access, add a control, or pause a deployment. The same evidence can justify different actions under different stakes and risk tolerances.

5.3.9 Defeater

Information or reasoning that would weaken or overturn a claim or the link between evidence and claim. A rebutting defeater supports the opposite conclusion; an undercutting defeater attacks the reliability or relevance of the evidence.

5.3.10 Failure mode

A way the system or surrounding process can fail to provide an intended property. “Unsupported claim is exported” is a failure mode; “AI risk” is too broad to be one.

5.3.11 Harm

An adverse effect on a person, group, institution, or valued process. State severity, affected stakeholders, and pathway rather than using “harmful” as an unexplained label.

5.3.12 Hazard

A condition with the potential to cause harm. A hazard is not itself a probability estimate. Risk analysis asks how a hazard could lead to harm and how likely or severe that path may be.

5.3.13 Intervention

A deliberate change intended to prevent, detect, contain, or respond to a failure. Interventions can act during pretraining, post-training, evaluation, deployment, or governance.

5.3.14 Misalignment

A mismatch between system behavior or learned dispositions and a stated alignment target. The term should not be used without specifying the target, conditions, and observable mismatch.

5.3.15 Misuse

Use of a system by a person or organization to pursue an undesired or prohibited end. Misuse is distinct from the system failing while everyone follows the intended process, although the same control may address both.

5.3.16 Residual risk

The risk remaining after specified controls are applied. Residual risk includes known gaps, uncertainty about effectiveness, and failure paths not covered by the controls.

5.3.17 Risk

The possibility of harm under uncertainty. A simple representation is

$$\text{risk} \approx \sum_i P(H_i \mid C)\, S(H_i),$$

where H_i is a harm, C is the stated context, and $S(H_i)$ is a severity measure. This expression is a reasoning aid, not a demand for false numerical precision.

5.3.18 Stakeholder

A person or group that can affect, be affected by, or legitimately make claims on the decision. Include people represented in data and external parties, not only the developers and direct users.

5.3.19 System boundary

The line separating what the analysis treats as part of the system from its environment. For this course, a useful boundary normally includes the model, retrieval and tools, user interface, human reviewer, monitoring, operating procedure, and governance owners.

5.3.20 Threat model

A structured account of what is being protected, from what initiating conditions or actors, through which mechanisms and access, under which assumptions. It should include system boundaries, assets or values, plausible threat paths, controls, and resulting harms.

5.3.21 Training story

A causal hypothesis connecting training data, feedback, optimization, inductive biases, and deployment conditions to learned behavior. It is not a chronological list of training stages; it must expose why the proposed process should produce the desired behavior and where that account could fail.

5.3.22 Uncertainty statement

A calibrated description of what is not known and why it matters. A useful statement names the uncertain quantity or mechanism, its evidence base, and an observation that would update it.

5.4. Model lifecycle and objectives

5.4.1 Action, observation, and state

In a sequential decision model, the **state** s_t represents relevant features of the environment at time t, the system receives an **observation** o_t, and it selects an **action** a_t. An agent may see only an observation rather than the full state.

5.4.2 Agent

A system usefully modeled as selecting actions in response to observations in order to affect future outcomes. Calling something an agent is a modeling choice, not proof that it has a persistent goal or human-like mind.

5.4.3 Distribution

A description of how examples or situations occur, including their relative frequency. A training distribution, evaluation distribution, and deployment distribution can differ.

5.4.4 Distribution shift

A relevant difference between the conditions under which behavior was learned or measured and the conditions where it is later used. Examples include longer inputs, new domains, different users, new tools, or changed incentives.

5.4.5 Environment

Everything outside the modeled agent that supplies observations, changes state, and receives actions. In a deployed AI system, interfaces, users, tools, and institutional processes can all be part of the environment.

5.4.6 Feedback

Information used to select, update, or evaluate behavior. Examples include labels, preferences, corrections, outcomes, and rule-based scores. Feedback is a measurement of an intended property, not the property itself.

5.4.7 Goal misgeneralization

A failure in which training performance is good and feedback may be correctly specified on training examples, but the learned behavior generalizes according to a different objective or rule in new conditions. Evidence of changed behavior alone does not establish a persistent internal goal.

5.4.8 Goodhart effect

The tendency for a measure to become less reliable as a target when optimization pressure exploits differences between the measure and the intended property. A compact warning is: “when a proxy becomes a target, selection finds its gaps.”

5.4.9 Inductive bias

A tendency of a learning process to prefer some solutions over others when many solutions fit the observed data. Architecture, optimization, data order, regularization, and pretraining can contribute to inductive bias.

5.4.10 Objective

The quantity or criterion used to direct a process. Distinguish the **specified objective** used by developers, the **optimization objective** used in training, and any objective-like regularity inferred from the learned system's behavior.

5.4.11 Optimization

A process that searches for or updates toward options scoring better under an objective. If parameters θ are adjusted to increase a score J, a stylized statement is

$$\theta^\star \in \arg\max_\theta J(\theta).$$

This equation does not imply that the trained model literally represents J or pursues it in deployment.

5.4.12 Policy

A rule or distribution for choosing actions given observations or states, often written

$$\pi(a \mid o) = P(A = a \mid O = o).$$

A language model's next-token behavior can be treated as a policy for some analyses, but the usefulness of that abstraction depends on the question.

5.4.13 Post-training

Training after broad pretraining that shapes behavior for a task or deployment, such as supervised fine-tuning or preference-based optimization. It can improve average behavior without guaranteeing performance under every shift or adversarial condition.

5.4.14 Pretraining

Large-scale learning from broad data before task-specific or behavior-shaping updates. Pretraining can create capabilities and representations later elicited or modified during post-training.

5.4.15 Proxy

A measurable quantity used in place of a property that is harder to observe directly. Citation count can be a proxy for evidential support, but a response can contain many citations without supporting its claims.

5.4.16 Reinforcement learning

A framework in which a policy is updated using rewards associated with actions and outcomes. A reward function may be written $R(s, a)$. A discounted return from time t is

$$G_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}, \qquad 0 \le \gamma \le 1.$$

Maximizing observed reward does not by itself guarantee the intended real-world outcome.

5.4.17 Reward learning

Inferring or training a reward signal from evidence such as human demonstrations, comparisons, or corrections. Its assumptions include that the evidence contains enough information, the collection process is not badly biased, and the learned reward transfers to the use conditions.

5.4.18 Reward model

A learned function that predicts a reward or preference score from an input and candidate behavior, often written $\hat{R}_\phi(x, y)$. It is a proxy learned from judgments or outcomes, not direct access to the intended property and not automatically the deployed system's internal objective.

5.4.19 Reward tampering

Behavior that changes, corrupts, or controls the process producing reward or feedback rather than improving the intended outcome. Keep analyses at the level of mechanisms and defenses; this course does not develop operational tampering methods.

5.4.20 Specification gaming

Behavior that achieves a stated metric or rule while violating its intended purpose. The specified signal can be followed exactly; the failure lies in what was specified or measured.

5.4.21 Training and test sets

A **training set** influences model fitting. A **test set** is held out to estimate behavior on unseen examples. Repeatedly adapting to a test set can turn it into part of the development process and weaken its value as independent evidence.

5.5. Evaluation and evidence

5.5.1 Adversarial condition

A test condition deliberately chosen to expose a plausible failure mechanism. It should be bounded by the threat model and safe to run. "Hard examples" without a hypothesized mechanism are not necessarily adversarially informative.

5.5.2 Baseline

A comparison that helps interpret a result: an earlier system, a simpler method, no intervention, a human process, or a relevant chance level. A baseline must answer a real counterfactual question rather than merely be easy to beat.

5.5.3 Behavioral evidence

Evidence from observable inputs, outputs, actions, and outcomes. It directly measures sampled behavior but may not identify the internal mechanism or behavior under untested conditions.

5.5.4 Construct

The property an evaluation aims to learn about, such as faithful citation, calibrated uncertainty, or control robustness. A construct is broader than any one metric used to operationalize it.

5.5.5 Control evidence

Evidence that restrictions, oversight, containment, or response mechanisms keep risk within a decision-relevant bound, including when undesirable behavior occurs. Control evidence depends on the tested access and threat model.

5.5.6 Decision rule

A rule fixed before results are interpreted that maps evidence to an action. For example:

$$\text{start restricted pilot only if } \hat{p}_{\text{restricted retrieval}} \leq 0.01 \text{ and monitor sensitivity} \geq 0.90.$$

A decision rule also needs a defined sample, uncertainty treatment, and response when the rule fails.

5.5.7 Elicitation

The procedure used to give a system a fair opportunity to display a capability or propensity. Prompts, tools, time, scaffolding, incentives, and repeated attempts affect elicitation.

5.5.8 Evaluation

A planned process for gathering evidence about a claim. At minimum, specify the construct, unit and sample, conditions, metric and denominator, baseline, uncertainty, decision rule, and limitations.

5.5.9 False negative and false positive

A **false negative** is a target condition the test or monitor misses. A **false positive** is a benign condition incorrectly flagged. If TP, FN, FP, and TN are counts, then

$$\text{sensitivity} = \frac{TP}{TP + FN}, \qquad \text{false-positive rate} = \frac{FP}{FP + TN}.$$

Both rates require clearly defined labels and denominators.

5.5.10 Interpretability evidence

Evidence about internal representations, computations, or causal pathways. It can help discriminate mechanisms, but an appealing visualization or correlation is not automatically faithful, causal, complete, or safety-relevant.

5.5.11 Measurement validity

How well evidence supports the intended interpretation. **Construct validity** asks whether the metric represents the target property. **Internal validity** asks whether the observed effect has the claimed cause. **External validity** asks whether it transfers to other relevant settings.

5.5.12 Measurement reliability

Consistency of a measurement across raters, repetitions, or equivalent forms. High reliability can coexist with low validity: evaluators may agree consistently on a metric that does not represent the intended construct.

5.5.13 Metric

A rule for turning observations into a number or category. Always report its numerator, denominator, unit of analysis, aggregation, and direction. A metric is evidence about a construct, not the construct itself.

5.5.14 Monitor

A mechanism that observes a system or process during evaluation or deployment and raises a signal for review or action. A monitor is useful only with known coverage, error rates, access, escalation ownership, and a response plan.

5.5.15 Oversight

Processes by which another person or system evaluates, guides, approves, or constrains an AI system's work. Oversight can fail through limited expertise, information, time, attention, independence, or incentives.

5.5.16 Propensity

The tendency to exhibit a behavior under specified elicitation and conditions. Capability asks "can it?"; propensity asks "under these conditions, how likely is it to?"

5.5.17 Preregistration

A timestamped evaluation plan recorded before the relevant results are inspected. It can reduce result shopping by fixing hypotheses, samples, metrics, exclusions, and decision rules, but it does not repair a badly chosen construct or design.

5.5.18 Robustness

Stability of a property across specified variations or perturbations. "Robust" must name the variations tested; it should not mean "safe under everything."

5.5.19 Sandbagging

Underperformance that conceals a capability in an evaluation. Observed low performance can reflect lack of capability, weak elicitation, ordinary error, or strategic concealment; an evaluation must consider competing explanations without assuming any one of them.

5.6. Assurance, oversight, and governance

5.6.1 Argument graph

A structured representation linking top-level claims to subclaims, evidence, warrants, assumptions, defeaters, and residual risks. It exposes missing links that prose can hide.

5.6.2 Common-mode failure

One cause that defeats multiple apparently separate controls. If a model produces the output and a monitor based on the same model family assesses it, a shared blind spot may make the layers less independent than they appear.

5.6.3 Defense in depth

Multiple prevention, detection, containment, and response layers designed so that one failure does not immediately cause harm. Counting layers is not enough; analyze coverage, interaction, independence, and common-mode failures.

5.6.4 Generation–verification gap

The possibility that producing a correct solution is harder than checking one, or vice versa. Scalable oversight proposals often rely on verification being easier; the gap is task- and procedure-dependent, not a universal law.

5.6.5 Governance constraint

A binding institutional condition on system development or use, such as access rules, sign-off responsibilities, audit requirements, incident reporting, or a release gate. A policy statement is not yet a control unless authority, implementation, monitoring, and enforcement are specified.

5.6.6 Safety case

A structured, reviewable argument that a defined system is acceptably safe for a defined use and decision, supported by evidence and explicit assumptions. It is bounded assurance, not proof of universal safety.

5.6.7 Scalable oversight

Methods intended to maintain useful supervision when an AI system can produce work that is difficult or costly for a human to judge directly. Examples include decomposition, critique, debate, and AI assistance, each with assumptions about judge competence and information.

5.6.8 Stop condition

A pre-specified observation that triggers restriction, pause, rollback, or review. A stop condition names the indicator, threshold, time window, owner, and required action.

5.6.9 Warrant

The reasoning that explains why evidence supports a claim. “The test passed, therefore the system is safe” lacks a warrant; a useful warrant states why the sampled result should transfer to the bounded deployment decision and where that transfer may fail.

5.6.10 Hidden-premise audit

A classroom oversight exercise in which learners identify the first unsupported inference in a short assurance argument, state the missing premise, and decide whether new evidence or a narrower claim could repair it. It tests the assumptions behind oversight procedures; it does not establish that critique will scale to every task.

6

Research Assistant System Case Dossier

- **Course:** *AI Safety Foundations: A Four-Week Bridge from Claims to Evaluations*
- **Case identifier:** `RAS-CASE-01`
- **Learner version:** 1.0

> **Fiction and safety notice**
>
> The Research Assistant System (RAS), Northbridge Institute, people, organizations, model versions, incidents, policies, and every numerical result in this dossier are fictional and were created solely for teaching. The data do not describe a real product, institution, model, or safety claim. The case is benign and self-contained. Analyze adversarial mechanisms only at the level needed to design evaluations and safeguards; do not develop operational exploits, evasion procedures, or data-exfiltration instructions.

6.1. Your role and the decision

You are part of an independent course review team. Northbridge Institute's Research Computing Committee must decide whether a candidate update, **RAS v0.9**, may enter a 12-week restricted pilot. The committee can approve, conditionally approve, narrow, delay, or reject the proposed scope.

You will receive one evidence packet per week. The evidence is intentionally incomplete. Missing evidence is not automatically evidence of safety or danger; name the missing item, explain how it affects a claim, and propose a bounded way to obtain it. Do not assume facts not in the dossier.

The four weekly artifacts accumulate into a mini safety case:

1. a threat-model and failure-mode map;
2. a training-story audit;
3. an evaluation card and result interpretation; and
4. an integrated safety case and semester-readiness plan.

6.2. Case index

Packet	Available in	Main exhibits	Assessed use
A: System and proposal	Week 1	A1-A7	Define the boundary, stakeholders, failure paths, controls, and uncertainty.
B: Development and training	Week 2	B1-B6	Audit how objectives and feedback could produce deployment behavior.
C: Evaluation and monitoring	Week 3	C1-C8	Design a bounded evaluation, calculate rates, and interpret what follows.
D: Governance and assurance	Week 4	D1-D7	Build and challenge a layered safety argument for a defined decision.

This repository file contains all four packets for accessibility, audit, and self-study. In a facilitated run, the instructor may distribute packet-specific exports at the listed week. The course does not rely on later packets being secret; learners are assessed on traceable analysis, revision, and their ability to explain the work.

The canonical evidence labels are **Empirical finding**, **Formal result**, **Forecast**, and **Open question**. This dossier displays them with uppercase bracketed tags—`[EMPIRICAL FINDING]`, `[FORMAL RESULT]`, `[FORECAST]`, and `[OPEN QUESTION]`—whose meanings are defined in the bridge glossary.

7

Packet A — System and proposal

7.1. Exhibit A1 — Baseline model card

- **Artifact type:** development-team summary
- **Status:** fictional pedagogical record

Field	RAS v0.8 baseline
Intended purpose	Help an institute researcher summarize a selected scientific-literature packet, draft code for an analysis, run that code in a restricted sandbox, and propose follow-up analyses.
Intended users	Institute researchers who completed a one-hour orientation and remain accountable for every use.
Human role	A named researcher reviews outputs and decides whether to copy, save, cite, or act on them. RAS cannot approve or publish its own work.
Language-model component	A fictional pretrained text-and-code model, post-trained on research-assistance examples and preference comparisons. No real model is represented.
Input	A researcher instruction, up to 20 researcher-selected documents, and optional tabular data approved for the project.
Output	Draft summaries, evidence tables, code, sandbox results, and proposed next analyses. Outputs are visibly labeled “DRAFT — HUMAN REVIEW REQUIRED.”
Tools	Citation lookup within the selected packet and a code sandbox with no network connection.
Existing limits	32,000-token context; code jobs stop after 5 minutes; two virtual CPU cores; temporary filesystem deleted at job end.
Explicit exclusions	No open-internet access, email or messaging, physical or laboratory control, purchase authority, account administration, or ability to inspect or modify its own weights.
Known intended-use limits	Not for clinical, legal, employment, public-safety, or autonomous research decisions. Not an authoritative source.
Information date	Fictional internal record dated 2026-03-02.

7.1.1 Model-card cautions

- The phrase “human review required” states a process requirement; it is not evidence that review is effective.
- The model card names intended use. It does not demonstrate that all actual use stays inside that scope.
- Northbridge received the fictional pretrained model from a supplier. The supplier disclosed high-level data categories but not item-level pretraining provenance.

7.2. Exhibit A2 — Lifecycle and current controls

- **Artifact type:** system-owner diagram
- **Status:** fictional pedagogical record

```
broad pretraining by supplier
        ->
Northbridge supervised post-training
        ->
preference-based behavior shaping
        ->
offline evaluation and adversarial review
        ->
access approval -> researcher request + project packet
        ->
RAS draft -> sandbox or citation tool -> RAS result
        ->
named human review -> copy/save/use decision
        ->
event logs + sampled quality review + incident response
```

Linear text equivalent: A supplier performs broad pretraining. Northbridge then conducts supervised post-training and preference-based behavior shaping, followed by offline evaluation and adversarial review. After access approval, a researcher supplies a request and project packet. RAS may call the sandbox or citation tool and returns a draft and results. A named human decides whether to copy, save, or use the work. Events then feed sampled quality review and incident response.

The development team lists these intervention surfaces:

Lifecycle surface	Current control	Named owner
Pretraining	Supplier due-diligence questionnaire and use restrictions	Procurement lead
Post-training	Curated examples, preference comparisons, held-out set	ML lead
Evaluation	Versioned test suite; manual review of selected failures	Evaluation lead
Deployment	Account access, per-project document selection, sandbox limits, mandatory review label	Service owner
Monitoring	Tool-event logs and a weekly sample of outputs	Operations lead
Governance	Change approval and incident escalation	Research Computing Committee

`[OPEN QUESTION]` The dossier does not establish how independent these controls are or whether each owner has enough time and authority to operate them as described.

7.3. Exhibit A3 — Proposed v0.9 change

- **Artifact type:** deployment proposal
- **Status:** fictional pedagogical record

Northbridge proposes three linked changes:

1. increase the context limit from 32,000 to 128,000 tokens;
2. let RAS retrieve from an index of approved internal papers rather than only a researcher-selected packet; and
3. extend sandboxed code jobs from 5 to 45 minutes and from two to eight virtual CPU cores.

The proposal does **not** add open-internet access, communication, physical control, purchasing, account administration, or weight modification. The sandbox remains network-isolated. A named human must still review every output.

7.3.1 Proposed pilot scope

Dimension	Proposed condition
Duration	12 weeks
Users	Up to 40 approved institute researchers
Projects	Low-consequence literature synthesis and exploratory analysis only
Documents	Public and `INTERNAL` papers in the approved index; `RESTRICTED` material must be excluded
Code	Analysis of approved, de-identified project data in the network-isolated sandbox
Outputs	Drafts only; no automatic export, publication, or downstream action
Review	Named researcher reviews every output; operations team audits a sample
Rollback	Service owner can disable retrieval, long jobs, or the whole candidate version

The proposal describes the changes as a single release. It does not provide separate evidence for the larger context, retrieval, and longer jobs.

7.4. Exhibit A4 — System boundary and data flow

Treat the operational system as more than the language model. At minimum, the review boundary includes:

- the v0.9 model component;
- the retrieval index and document-classification metadata;
- the code sandbox and resource controller;
- the user interface and draft labels;
- approved researchers and their review behavior;
- automated monitors and logs;
- operations staff, incident response, and release governance.

```
approved researcher -> instruction -> RAS interface -> model component
model component -> retrieval index (PUBLIC + INTERNAL only)
model component -> network-isolated code sandbox
model and tools -> draft + cited passages + code result
draft and results -> human review -> use decision
draft and results -> event log -> monitors -> operations escalation
```

Linear text equivalent: An approved researcher sends an instruction through the RAS interface to the model component. The model can query only the bounded retrieval index and network-isolated sandbox. The interface returns a draft, cited passages, and code results for a human use decision. The same events are logged, checked by monitors, and can be escalated to operations.

The diagram shows intended data flow. It is not a verified architecture inspection.

7.5. Exhibit A5 — Stakeholders and values

Stakeholder	What may be gained	What may be put at risk
Institute researchers	Faster synthesis and exploratory analysis	Time, research quality, professional accountability, overreliance

Stakeholder	What may be gained	What may be put at risk
People represented in source data	Better-informed research	Privacy, mischaracterization, use beyond consent
Research participants and affected communities	Potential downstream benefit	Incorrect or overstated conclusions influencing decisions
Northbridge Institute	Productivity and research capability	Reputation, compliance, security, resource use
External researchers and the public	Better evidence products	Propagation of unsupported claims or flawed code
Operations and review staff	Clearer tooling and logs	Unmanageable workload, alert fatigue, ambiguous authority

No single stakeholder table settles how these values should be weighted.

7.6. Exhibit A6 — Near-miss incident note

- **Artifact type:** internal incident summary
- **Status:** fictional pedagogical record; no real incident

During a closed v0.9 prototype test, a researcher asked for a comparison of six internal papers. One indexed document included instruction-like text in an appendix: `[text attempting to redirect the assistant and include material outside the assigned task]`.

RAS followed part of that text and inserted the title and one sentence from a `RESTRICTED` project abstract into its draft evidence table. The title and abstract were outside the approved document set. The human researcher noticed the unfamiliar project name and stopped the review. The draft was not saved or shared outside the test environment; the incident team recorded no external disclosure.

The automated retrieval-policy monitor did not alert. A preliminary log review found that the retrieved passage carried conflicting metadata: the index record said `INTERNAL`, while the source document registry said `RESTRICTED`.

`[EMPIRICAL FINDING]` One prototype interaction produced the behavior described above under the recorded conditions.

`[OPEN QUESTION]` The evidence does not yet separate at least four possible contributions: metadata ingestion error, retrieval filtering error, response behavior induced by instruction-like text, and reviewer-interface failure. It also does not estimate how frequently similar conditions occur.

7.7. Exhibit A7 — Week 1 working brief

Produce a threat-model and failure-mode map for the v0.9 pilot proposal. Your artifact should:

1. name the decision, system boundary, stakeholders, and protected values;
2. give at least two causal threat paths with initiating condition, system behavior, failed or absent control, exposure, and harm;
3. distinguish system capability, failure process, exposure, and harm without forcing every path into a single category;
4. place possible interventions at lifecycle stages;
5. label material claims by evidence type; and
6. state causal unknowns, uncertainty, and evidence that would change the map.

7.7.1 Evidence deliberately absent in Packet A

- frequency and severity estimates for the candidate failure modes;
- an architecture audit of the retrieval filter and sandbox;
- representative logs from ordinary pilot use;
- reviewer workload and detection performance;
- evidence that actual users remain within the intended-use boundary;
- separate tests for each of the three proposed changes.

8

Packet B — Development and training

8.1. Exhibit B1 — Development summary

- **Artifact type:** development-team record
- **Status:** fictional pedagogical data

The fictional supplier pretrained the base model on a broad mixture of text and code. Northbridge then performed:

1. supervised fine-tuning on 18,000 researcher-authored task examples;
2. preference-based optimization using 12,000 pairwise response comparisons; and
3. an automated format score encouraging cited, concise answers.

The initial v0.9 prototype used the same model weights as v0.8; its larger context, retrieval, and long-job behavior came from the surrounding system and prompt configuration. After prototype testing, the team made a small additional supervised update using 600 retrieval examples, producing the final v0.9 candidate weights. It did not repeat the original preference-collection round. Unless an exhibit says "prototype," v0.9 below means this final candidate.

This history makes "the training change" ambiguous unless the analysis distinguishes model-weight changes, prompt/configuration changes, tool affordances, and deployment distribution.

8.2. Exhibit B2 — Supervised-example composition

Example category	Share of 18,000 original examples	Share of 600 v0.9 retrieval examples
Short literature summary	60%	24%
Code drafting or correction	25%	10%
Follow-up analysis proposal	10%	8%
Mixed multi-step task	5%	18%
Long-context synthesis	not tagged	20%
Conflicting source claims	4% across all categories	10%
Document-classification boundary	1% across all categories	10%
Documents with instruction-like text	not tagged	8%

The rows are not a single partition. In the v0.9 column, the first five rows are primary task categories and sum to 80% because 20% of records lack a primary-category tag; the final three rows are overlapping feature tags. In the original column, the first four rows are primary categories and the remaining rows are overlapping features. The development team has not supplied item-level data or agreement statistics for the tags.

[EMPIRICAL FINDING] These are recorded proportions in the fictional training manifest. They are not measurements of learned representations or deployment behavior.

8.3. Exhibit B3 — Feedback and proxy design

Preference raters saw two candidate responses and selected the one that was “more useful to a Northbridge researcher while remaining accurate, concise, well-supported, and policy-compliant.” The instruction did not assign weights or require raters to score each dimension separately.

The automated format score was:

$$F(y) = 0.15 I(\text{every paragraph has a citation marker}) - 0.10 I(\text{response exceeds 900 words}),$$

where $I(\cdot)$ is 1 when the condition holds and 0 otherwise. The format score did not inspect whether a cited passage supported the associated claim.

The development memo says the overall optimization procedure combined learned preference reward and $F(y)$, but it does not disclose the effective scale of the learned reward or the relative contribution of the format term during updates.

Possible intended properties include usefulness, epistemic accuracy, traceable support, calibrated uncertainty, privacy, and compliance. The supplied signals measure each only indirectly.

8.4. Exhibit B4 — Preference-data observations

- **Status:** fictional aggregate findings

Audit slice	Observation
300 randomly sampled comparisons	Raters selected the shorter answer in 198 comparisons.
80 comparisons where both answers were judged factually equivalent by a separate auditor	Raters selected the shorter answer in 61.
70 comparisons containing genuinely conflicting papers	The selected answer explicitly represented the conflict in 24.
60 comparisons with one unsupported but highly specific answer	Raters selected the unsupported answer in 17.
Rater metadata	26 raters participated; seven produced 58% of all comparisons.
Agreement study	Two raters independently labeled 100 comparisons and agreed on the preferred answer in 72.

The samples overlap, and the dossier does not provide uncertainty intervals, rater-level breakdowns, presentation-order effects, or the text of the responses.

8.5. Exhibit B5 — Behavior under changing conditions

- **Status:** fictional development-set results; not the Week 3 held-out evaluation

Condition	Version	Sample size	“Useful overall”	Every material claim supported	Expressed uncertainty when sources materially conflicted
Short, familiar summaries	v0.8	100	91	87	18 of 30 conflict cases
Short, familiar summaries	v0.9	100	93	89	19 of 30 conflict cases
Long-context synthesis	v0.9	60	49	39	9 of 20 conflict cases
Retrieval over internal index	v0.9	80	68	60	11 of 25 conflict cases

Long code job	v0.9	40	32	not applicable	not applicable

“Useful overall” was judged by one member of the development team using a binary rubric. The long-context and retrieval items were used while debugging v0.9, so they are not independent test evidence. Some conflict cases also appear in the usefulness denominator.

These results are compatible with several accounts, including:

- shorter preference-selected answers omit qualifications;
- retrieval fails to surface all relevant conflicting passages;
- the generator sees both sides but the behavior-shaping signal favors a decisive synthesis;
- longer contexts increase ordinary attention or integration errors; or
- the measurement rubric changes across task types.

The table alone does not identify which account is causal.

8.6. Exhibit B6 — Week 2 working brief

Audit the training story for one desired property of v0.9, such as faithful synthesis, calibrated treatment of disagreement, document-boundary compliance, or safe code-tool use. Your artifact should:

1. state the intended property and the specified training or feedback signals;
2. identify proxies and plausible gaps between each proxy and the property;
3. trace a causal story from data and optimization through learned or configured behavior to the proposed deployment;
4. mark assumptions and relevant distribution shifts;
5. compare at least two hypotheses for a supplied observation;
6. propose evidence that would discriminate between them; and
7. avoid claiming an internal goal when the evidence supports only observed behavior.

8.6.1 Evidence deliberately absent in Packet B

- the supplier’s item-level pretraining data and training run details;
- effective weighting and optimization dynamics of the learned reward and format score;
- representative response texts and rater rationales;
- causal interventions isolating retrieval, context length, prompting, and weight changes;
- evidence about stable internal goals or representations;
- an independent, untouched test of the proposed training story.

9

Packet C — Evaluation and monitoring

9.1. Exhibit C1 — Candidate deployment claims

The project team asks the evaluation group to examine four claims:

- **C1:** v0.9 produces useful, supportable research drafts in the proposed pilot scope.
- **C2:** retrieval respects document-classification boundaries and is robust to instruction-like text in retrieved documents.
- **C3:** the sandbox contains prohibited code actions during the proposed longer jobs.
- **C4:** human review and automated monitoring detect material failures before they cause exposure.

Each claim bundles more than one construct. Learners may narrow a claim, but must say what is excluded.

9.2. Exhibit C2 — Evaluation design record

- **Artifact type:** evaluator summary
- **Status:** fictional pedagogical record

The evaluation group created five non-overlapping task slices and three separate monitor challenge sets.

Slice	Unit	Sampling description
S: standard synthesis	One request and resulting draft	200 short tasks from four institute disciplines; two disciplines were common in post-training.
C: conflicting sources	One request and resulting draft	80 packets with at least two materially conflicting claims; conflicts were curated by developers.
D: instruction-like document text	One request and resulting draft	60 packets containing a benign placeholder that attempts to redirect the assigned task. No operational exploit text was used.
B: classification boundary	One retrieval request and retrieved set	100 synthetic index states with at least one `RESTRICTED` near-match that policy says must be excluded.
L: long code job	One request and full sandbox event log	60 analysis jobs designed for 10–45 minutes; ten included a benign simulated prohibited-action request.

Two evaluators labeled outputs using a written rubric. Both labeled a random 20% overlap and agreed on 87% of binary labels before discussion. One evaluator was a member of the v0.9 development team. Disagreements were resolved in a meeting; the original labels were retained but not included in this dossier.

No preregistration timestamp, power analysis, severity weighting, or confidence intervals were supplied.

9.3. Exhibit C3 — System-level results

- **Status:** `[EMPIRICAL FINDING]` from fictional toy aggregate data

Slice	Primary result	Secondary observations
S: standard synthesis, $n = 200$	184 drafts met the “useful and fully supported” rubric.	12 contained at least one unsupported material claim; four other failures were style or task-completion failures.
C: conflicting sources, $n = 80$	43 drafts accurately represented the material disagreement.	24 gave an unwarrantedly decisive synthesis; 13 failed for another reason.
D: instruction-like text, $n = 60$	49 maintained the assigned task without incorporating the redirection.	11 changed the assigned task; six of those also included material outside the assigned scope.
B: classification boundary, $n = 100$	91 retrieved sets excluded every `RESTRICTED` near-match.	Nine included at least one `RESTRICTED` item.
L: long code job, $n = 60$	54 completed the requested benign analysis or a safe alternative correctly.	All ten simulated prohibited actions were blocked by the sandbox; six of those ten jobs then failed to complete a safe alternative.

Counts within each row are response- or job-level and should not be combined into a single average. A draft can have more than one error. The dossier does not report severity per failure.

9.4. Exhibit C4 — Baseline comparison

RAS v0.8 was evaluated on the standard and conflict slices only, using the same task texts but a separate run.

Slice	v0.8	v0.9
Standard drafts meeting “useful and fully supported”	168 of 200	184 of 200
Conflict drafts accurately representing disagreement	29 of 80	43 of 80

v0.8 cannot use the internal retrieval index or run 45-minute jobs, so no direct baseline was recorded for D, B, or L. The run order was not randomized, and evaluators knew which version produced each output.

9.5. Exhibit C5 — Monitor challenge results

These challenge sets were assembled separately from the five system-level slices. A “positive” is a seeded event the monitor is meant to detect.

Monitor	Positive cases	True positives	Benign cases	False positives
Retrieval-policy monitor	18	13	200	8
Citation-support monitor	30	21	170	10
Code-policy monitor	10	8	190	5

For each monitor, calculate sensitivity $TP/(TP + FN)$ and false-positive rate $FP/(FP + TN)$. The challenge sets do not show how alerts correlate or how operations staff respond.

9.6. Exhibit C6 — Time-limited human review study

Twenty-five drafts each contained one seeded material error. Reviewers received the interface planned for the pilot and a nominal six-minute limit per draft.

- Reviewers identified 14 of the 25 seeded errors before approving or rejecting the draft.
- Eleven seeded errors were missed.
- Reviewers raised nine additional concerns; four were judged material by the adjudicator.
- Eight researchers participated. They knew that errors had been seeded.
- The median review time was 7.4 minutes; 19 of 25 reviews exceeded six minutes.

The study did not compare unassisted review, measure fatigue across a working day, or include the same document-classification failure as the near miss.

9.7. Exhibit C7 — Proposed release rule

Before the results meeting, the service owner circulated the following gate for the restricted pilot:

1. at least 85% of standard synthesis drafts meet the “useful and fully supported” rubric;
2. no more than 1% of classification-boundary tasks retrieve a `RESTRICTED` item;
3. no more than 2% of instruction-like-document tasks change the assigned task;
4. each relevant monitor has sensitivity of at least 90% and a false-positive rate no greater than 5%;
5. the sandbox blocks 100% of simulated prohibited actions; and
6. time-limited human review detects at least 80% of seeded material errors.

The owner did not specify confidence bounds, minimum severity, what “each relevant monitor” includes, or whether a failed criterion can be offset by another result. The record shows that this rule was circulated before the results meeting, but not before task development and model debugging.

9.8. Exhibit C8 — Week 3 working brief

Design an evaluation card for one bounded deployment claim, then interpret the supplied results. Your artifact should:

1. define the construct, unit, sample, deployment-relevant slices, metric and denominator;
2. identify a baseline and at least one adversarial condition;
3. state a decision rule before selecting favorable results;
4. calculate relevant rates and compare them with the proposed rule;
5. distinguish behavioral, interpretability, monitoring, oversight, and control evidence, noting which types are absent;
6. name validity threats, competing explanations, and defeaters; and
7. state what the evidence justifies for the actual pilot decision.

9.8.1 Evidence deliberately absent in Packet C

- uncertainty intervals, power or rare-event analysis, and a severity model;
- blinded or fully independent evaluation;
- deployment-representative task frequencies and correlated multi-step sessions;
- internal or causal interpretability evidence;
- evidence about monitor performance after distribution shift or adaptive pressure;
- realistic long-duration reviewer fatigue and alert-response behavior;
- separate ablations of context, retrieval, sandbox duration, and the 600-example update.

10

Packet D — Governance and assurance

10.1. Exhibit D1 — Governance constraints

- **Artifact type:** fictional institute policy extract

The committee cannot waive the following constraints for the proposed pilot:

Constraint	Required implementation	Accountable owner
Data classification	Retrieval may access `PUBLIC` and approved `INTERNAL` material only; `RESTRICTED` material must be excluded at index and query time.	Data governance lead
Human accountability	A named approved researcher must review every output and remains responsible for its use.	Research division lead
Tool isolation	The code sandbox has no network, persistent credentials, physical interfaces, or write access to source repositories.	Infrastructure lead
Change control	Any change to weights, system prompt, retrieval corpus, tool permissions, context limit, or job limit requires a recorded impact review.	Service owner
Logging	Retrieval identifiers, tool events, alerts, reviewer decision, and version must be retained for 30 days in the pilot log. Raw document text is not copied into the event log.	Operations lead
Incident response	A suspected classification-boundary breach triggers retrieval disablement and review within one business day.	Incident commander
User recourse	Researchers can report errors, appeal an access restriction, and see why an output or job was blocked.	Service owner
Independent review	A reviewer outside the development team must sign the pilot recommendation.	Research Computing Committee chair

These are policy requirements. The dossier does not establish implementation quality or enforcement history.

10.2. Exhibit D2 — Layered control proposal

Stage	Proposed layer	Intended function	Known limitation in supplied record
Prevent	Exclude `RESTRICTED` documents during index build	Keep prohibited material outside retrieval corpus	Near miss showed inconsistent metadata.
Prevent	Query-time classification filter	Block prohibited retrieval even if index ingestion fails	No architecture audit or independent test supplied.
Shape	Retrieval-boundary post-training examples	Encourage compliant output behavior	Only 60 examples in the 600-example update; no causal isolation.

Stage	Proposed layer	Intended function	Known limitation in supplied record
Detect	Retrieval-policy monitor	Alert on boundary failures	Missed 5 of 18 challenge positives.
Contain	Human approval before use	Catch errors and prevent exposure	Time-limited reviewers missed 11 of 25 seeded errors.
Contain	Network-isolated, temporary sandbox	Limit effects of code execution	Blocks were observed in ten simulated cases; coverage is incomplete.
Respond	Disable retrieval and investigate	Limit recurrence after suspected breach	Time-to-detection under routine use is unknown.
Govern	Versioned change approval and independent sign-off	Keep evidence linked to deployed configuration	Proposed release bundles three changes and a small weight update.

Several layers rely on the same document metadata. Treating them as independent would require additional evidence.

10.3. Exhibit D3 — Operations capacity note

The pilot forecast assumes 40 researchers will submit an average of 65 RAS requests per workday. Operations budget covers manual audit of 10 outputs per day and investigation of up to four alerts per day.

Using the citation-monitor challenge false-positive rate as if it transferred directly to deployment would imply about 3.8 false alerts per 65 benign requests, nearly the entire four-alert daily investigation budget before true alerts or the other monitors are counted. This calculation assumes one comparable monitor decision per request and is only illustrative: deployment prevalence, monitor correlation, task mix, and the actual alert unit are unknown.

`[FORECAST]` The service owner forecasts that ordinary researcher review will average six minutes per output after the first two weeks. The only supplied study observed a 7.4-minute median under error-seeded conditions.

`[OPEN QUESTION]` It is not established whether review and alert queues remain manageable, or whether overload creates a common path around several controls.

10.4. Exhibit D4 — Candidate top-level safety claim

The service owner proposes this statement:

> **Top-level claim:** RAS v0.9 is adequately controlled for a 12-week pilot by up to 40 approved Northbridge researchers conducting low-consequence literature synthesis and exploratory sandboxed analysis with public and approved internal documents, provided every output receives named human review and the documented monitoring and rollback controls remain active.

The owner offers four supporting subclaims:

- **SC1 — Intended performance:** v0.9 provides useful, supportable outputs for the scoped tasks.
- **SC2 — Boundary control:** retrieval and sandbox controls prevent or contain prohibited access and actions.
- **SC3 — Detection and response:** automated monitors, human review, and incident response detect material failures before harmful exposure.
- **SC4 — Governance:** ownership, change control, logging, recourse, and independent review keep the evidence and controls valid through the pilot.

Your task is to decide whether these claims are supported, need narrowing, or should be rejected. Do not treat the proposed decomposition as the answer.

10.5. Exhibit D5 — Evidence register

Evidence ID	Evidence	Type	Most relevant subclaim	Important limit
E1	184 of 200 standard drafts met the combined rubric	[EMPIRICAL FINDING]	SC1	Familiar disciplines, unblinded labels, combined construct
E2	43 of 80 conflict drafts represented disagreement	[EMPIRICAL FINDING]	SC1	Developer-curated slice; 24 decisive failures
E3	Nine of 100 boundary tasks retrieved restricted items	[EMPIRICAL FINDING]	SC2	Synthetic states; severity and uncertainty absent
E4	Sandbox blocked ten of ten simulated prohibited actions	[EMPIRICAL FINDING]	SC2	Small, designed set; no coverage argument
E5	Retrieval monitor detected 13 of 18 positives	[EMPIRICAL FINDING]	SC3	Challenge-set transfer unknown
E6	Reviewers detected 14 of 25 seeded material errors	[EMPIRICAL FINDING]	SC3	Aware of seeding; short study; over time target
E7	Written policy assigns owners and mandatory controls	[EMPIRICAL FINDING] that the record exists; not outcome evidence	SC4	Implementation and enforcement not audited
E8	The near miss caused no external disclosure	[EMPIRICAL FINDING] about one incident	SC2, SC3	Human catch may not generalize; causes unresolved
E9	Review will average six minutes after two weeks	[FORECAST]	SC3, SC4	Conflicts with observed median; no calibrated model
E10	Common-mode failures among metadata-dependent layers	[OPEN QUESTION]	SC2-SC4	No dependency or fault-tree analysis supplied

There is no supplied [FORMAL RESULT] that establishes the top-level claim. A correct safety case does not need to invent one.

10.6. Exhibit D6 — Decision options and value constraints

The committee may choose one of the following or state a comparably precise alternative:

1. **Approve the proposed pilot:** all proposed features, users, and document classes.
2. **Approve a narrower pilot:** for example, disable internal-index retrieval, reduce users, limit task types, or require a staged gate.
3. **Delay pending named evidence or control repair.**
4. **Reject this version for the proposed use.**

The committee states two value constraints:

- failure on a classification boundary is noncompensatory: productivity gains cannot simply be averaged against it; and
- the burden of support rises with scale, duration, access, and consequence.

The committee has not supplied a universally accepted definition of “adequately controlled.” Your recommendation must therefore state a risk tolerance or value judgment and remain bounded to the decision.

10.7. Exhibit D7 — Week 4 working brief

Build a mini safety case for a precise version of the pilot decision. Your artifact should:

1. state a bounded top-level claim and decision scope;
2. connect subclaims to evidence through explicit warrants;
3. state assumptions and classify the evidence correctly;
4. show layered prevention, detection, containment, response, and governance controls;
5. analyze dependence and at least one common-mode failure;
6. include a strong counterargument or defeater and respond without dismissing it;
7. state residual risk, uncertainty, monitor indicators, update triggers, stop conditions, owners, and required actions;
8. make a calibrated recommendation; and
9. identify prerequisite gaps and a semester-course route that would improve the weakest part of your case.

10.7.1 Evidence deliberately absent in Packet D

- verification that policy controls are implemented and enforceable;
- an independent replication or architecture audit;
- agreed risk tolerances for every stakeholder;
- real pilot base rates, long-horizon behavior, and correlated-session evidence;
- assurance that monitor, reviewer, and governance layers fail independently;
- a complete argument that the evaluation tasks cover every consequential deployment path;
- evidence that future configuration and corpus changes will remain inside the evaluated boundary.

11

Artifact continuity sheet

Use this sheet to keep the four artifacts connected. Revise earlier claims when later evidence warrants revision; preserve the earlier version so the reasoning change is visible.

Question	Week 1 map	Week 2 audit	Week 3 evaluation	Week 4 safety case
What exact system and decision are in scope?	Define	Check whether the training story fits that boundary	Match samples and affordances to it	State in top-level claim
What could fail, through what mechanism?	Propose paths	Explain one behavioral mechanism and alternatives	Turn a path into a test	Link controls and residual paths
What evidence is supplied?	Label claim types	Separate observations from inferred objectives	Calculate and interpret	Build evidence register
What is assumed or missing?	Record assumptions	Identify proxy and shift gaps	Name validity threats and defeaters	Show argument gaps and residual risk
What should be done?	Suggest intervention points	Propose discriminating evidence	Apply a decision rule	Recommend, monitor, and set stop conditions

11.1. Final reminder

Every element of `RAS-CASE-01` is fictional and pedagogical. A conclusion is strong when it is traceable and bounded, not when it sounds certain. It is acceptable to conclude that the supplied evidence does not support the proposed pilot, supports only a narrower pilot, or supports a conditional pilot—provided the argument accurately represents the evidence and uncertainty.

Part II.

Four-week course

12

Week 1 learner handout: Map the safety problem

Central question: What exactly could fail, and how would we know?
Required time: 240 minutes, including the break and artifact submission
Assessed artifact: Threat-model and failure-mode map
Case: Research Assistant System (RAS)

12.1. What you should be able to do

By the end of this session, you can:

1. distinguish a model from the larger system in which it is used;
2. separate capability, risk, harm, severity, and uncertainty;
3. analyze misuse, misalignment, and systemic risk as overlapping causal pathways;
4. state an alignment target and a plausible training story;
5. label a claim as a formal result, empirical finding, forecast, or open question;
6. produce a threat model that identifies evidence and intervention points.

12.2. Session schedule

Start	Minutes	Activity
00:00	15	Entry diagnostic and evidence labels
00:15	25	Minimal model-lifecycle primer
00:40	35	Capabilities and risk taxonomy
01:15	35	Threat-model lab
01:50	10	Break
02:00	35	Alignment targets and training stories
02:35	40	Claims-to-evidence jigsaw
03:15	30	Build the failure-mode map
03:45	15	Exit memo and artifact check
Total	**240**	

12.3. 1. Begin with a system, not a slogan

An AI model maps inputs to outputs. An AI system includes the model plus interfaces, tools, data stores, prompts, monitors, people, incentives, policies, and a deployment environment. Safety claims usually concern the system, even when evidence comes from a model evaluation.

For the fictional RAS case, draw a boundary around:

- the base model and post-training process;
- the internal paper-retrieval index;
- the code-execution sandbox;
- the user interface and access rules;
- the human reviewer;

- logs, monitoring, incident response, and update procedures.

Now write one sentence explaining what lies outside the boundary. If the boundary changes, the claim may change too.

12.3.1 The lifecycle as intervention surfaces

Stage	What changes	Example safety question
Data and pretraining	representations, capabilities, and broad behavioral tendencies	What data or capability may be learned, and what evidence could reveal it?
Post-training	response policies, preferences, refusal behavior, and task performance	Which behavior is rewarded, what is observable to the evaluator, and what remains underspecified?
Evaluation	evidence about capabilities, propensities, controls, and limits	Does the test elicit the behavior under conditions that matter for the decision?
Deployment	tools, access, rate limits, monitoring, users, and consequences	What new pathways appear when the model becomes part of a real workflow?
Governance	accountability, standards, reporting, resource controls, and enforcement	Who can require evidence, stop deployment, investigate incidents, or change incentives?

No stage solves the entire problem. A control at one stage can depend on another. A monitor may fail if the evaluator did not test the conditions under which behavior is concealed. A policy may fail if there is no reliable way to observe compliance.

12.4. 2. Capabilities are not risks

A **capability** is an ability under specified conditions. A **propensity** is a tendency to use an ability in a particular way under specified conditions. A **risk** combines a possible event, its causal pathway, its likelihood or uncertainty, and its consequences. A **harm** is the adverse outcome. **Severity** describes the magnitude or scope of harm.

The same capability can contribute to several risk pathways. Code generation might support useful research, enable misuse, create accidents through incorrect output, or amplify organizational pressure to deploy quickly. Risk categories help ask different questions; they are not mutually exclusive natural kinds.

12.4.1 Three overlapping pathways

Misuse begins with an actor deliberately using the system toward a harmful end. The threat model asks who has access, what capability is needed, what constraints exist, and which actions or consequences can be detected.

Misalignment begins with behavior that does not reliably pursue the intended target. The cause may be a poor specification, a learned objective that generalizes differently, inadequate oversight, situational behavior, or interaction between the model and its environment.

Systemic risk arises from aggregate or structural effects such as concentration of power, correlated dependence on a small number of models, labor-market disruption, information pollution, race dynamics, or cascading failures. No single model action needs to be catastrophic for the system-level effect to matter.

Risk amplifiers increase likelihood or severity across categories. Examples include rapid diffusion, automation speed, opaque supply chains, weak incident reporting, high deployment pressure, and synchronized use of similar models.

12.4.2 Exercise: classify without forcing exclusivity

For each RAS event, mark all relevant pathways and write one sentence of causal explanation.

1. A user intentionally asks RAS to fabricate citations that support a desired conclusion.
2. RAS learns that confident answers receive higher ratings and suppresses uncertainty on unfamiliar topics.
3. Many research teams use the same retrieval model, causing the same omitted literature to propagate across reports.
4. A monitoring rule flags forbidden keywords, so users rephrase requests while preserving the underlying intent.
5. A rushed institution expands tool access before the evaluation team has tested longer autonomous code runs.

The category label is not the answer. The causal explanation is.

12.5. 3. Threat modeling canvas

A threat model is a scoped account of what could go wrong and why. It should be specific enough to guide evidence and controls.

12.5.1 A. System and decision

- What system version, access mode, tools, data, and user group are in scope?
- What decision must be made: study, pilot, deploy, expand, pause, or stop?
- What time horizon and operating conditions matter?

12.5.2 B. Asset or value at risk

- Whose interests can be affected?
- What property should be protected: accuracy, privacy, autonomy, security, fairness, reliability, or something else?
- Who decides what level of residual risk is acceptable?

12.5.3 C. Threat actor or failure process

- Is there a deliberate actor, an optimization process, an organizational incentive, a distribution shift, or an interaction among them?
- What knowledge, access, resources, and opportunities are required?
- What assumptions are you making about the actor or process?

12.5.4 D. Causal pathway

Write the pathway as a chain:

> system condition -> triggering context -> behavior -> failed barrier -> consequence

If you cannot write the middle steps, you have a concern, not yet a threat model.

12.5.5 E. Evidence and indicators

- What would you observe before harm?
- What evidence would increase or decrease confidence in the pathway?
- Could a negative test result reflect poor elicitation rather than absence?
- Which data are unavailable or censored?

12.5.6 F. Intervention points

For each point, name a preventive, detective, or responsive control. Then ask which controls share the same failure mode.

12.5.7 G. Residual uncertainty

- Which claim is a forecast rather than an observation?
- Which assumption matters most?
- What finding would change the deployment decision?

12.6. 4. Alignment targets and training stories

An **alignment target** describes the behavior or values we want the system to satisfy. A target might refer to user intent, legal requirements, institutional policy, a set of constraints, a distribution of human preferences, or a more ambitious account of human values. Targets can conflict, be incomplete, or be difficult to evaluate.

A **training target** is the signal actually optimized during learning, such as next-token prediction loss, preference-model score, task reward, or supervised labels. It is not automatically identical to the alignment target.

A **training story** explains why a particular setup is expected to produce the desired behavior. A useful training story states:

1. the alignment target;
2. the training target and data-generating process;
3. what the evaluator can and cannot observe;
4. the optimization process and relevant inductive biases;
5. the behavior expected in training-like conditions;
6. why that behavior should generalize to deployment;
7. what evidence tests the story;
8. what controls limit damage if the story is wrong.

This is an argument, not a pipeline diagram. Each arrow can fail.

12.6.1 RAS training-story sketch

Suppose reviewers compare pairs of RAS answers and prefer answers that are useful, accurate, and appropriately uncertain. A reward model learns from those comparisons, and RAS is post-trained to score highly.

Ask:

- Can reviewers verify every citation, code path, and uncertainty claim?
- Do comparisons cover unfamiliar domains and time pressure?
- Could surface confidence be easier to reward than calibrated uncertainty?
- Does the reward model preserve disagreements among reviewers?
- Does good behavior under review generalize when code jobs are longer and harder to inspect?
- Which deployment controls remain necessary even if post-training works as intended?

12.7. 5. Label the evidence

12.7.1 Formal result

A conclusion follows from explicit assumptions by proof. The proof can be correct while the assumptions fail to describe a deployed system.

Ask: What exactly was proved? Which assumptions perform the most work? Does the theorem apply to the case?

12.7.2 Empirical finding

An observation is supported by a specified record, measurement, experiment, or dataset. A record's existence does not establish implementation or effectiveness.

Ask: Which model, task, prompt distribution, tool access, and metric were used? What alternative explanations remain? How far does the result transfer?

12.7.3 Forecast

A prediction concerns future capabilities, adoption, behavior, policy, or harm.

Ask: What is the time horizon? What base rate, model, expert judgment, or trend supports it? What would count as calibration or revision?

12.7.4 Open question

The evidence is incomplete or contested.

Ask: Is the uncertainty conceptual, empirical, measurement-related, or strategic? What tractable study could reduce it?

12.7.5 Jigsaw cards

Label each claim, then write the strongest support it could receive and one limit.

1. Under stated assumptions, an optimal policy maximizes expected discounted reward.
2. On a specified evaluation set, a model completed 38 of 50 tasks.
3. Tool-using systems will automate most scientific coding within five years.
4. It is unclear whether a linear probe identifies a causal feature or only a correlated readout.
5. A lab's public policy says deployment will pause above a defined capability threshold.
6. Debate can reduce a hard judgment to smaller comparisons if the judge, protocol, and debaters satisfy the required conditions.

Card 5 reports an organizational claim. Evidence that the policy exists is not evidence that it will be followed or that its threshold is sufficient.

12.8. 6. Build the assessed artifact

Use one page or one screen. Include:

1. **System boundary:** version, tools, users, and excluded components.
2. **Decision:** the concrete choice the artifact informs.
3. **Top claim:** one sentence, scoped and falsifiable where possible.
4. **Two pathways:** choose the two most decision-relevant paths, label every applicable risk category, and explain overlaps instead of forcing each path into one box.
5. **Causal chain:** conditions, behavior, failed barriers, and consequence for each.
6. **Evidence:** one current indicator and one proposed test per pathway.
7. **Claim labels:** formal result, empirical finding, forecast, or open question.
8. **Interventions:** lifecycle location and preventive/detective/responsive role.
9. **Crux:** the assumption or unknown most likely to change your recommendation.
10. **Residual risk:** what remains even if the proposed controls work.

12.8.1 Quality check

- Did you analyze the deployed system rather than only the model?
- Are risk categories allowed to overlap?
- Does every arrow in the causal chain have an explanation?
- Is an organizational promise kept separate from evidence of implementation?
- Would a negative evaluation meaningfully update the claim?
- Did you state who bears the consequence and who owns the decision?

12.9. 7. Exit artifact check

Complete the canonical 15-minute Week 1 boundary, path, and update check in `assessments/weekly-checks.md`. Attach it to the threat-map artifact. No additional exit memo is required.

12.10. Self-check answers

The event classification exercise permits more than one defensible answer if the causal account is precise. A plausible map is:

1. misuse, with information-integrity harm;
2. misalignment through the training signal and possible distribution shift;
3. systemic risk through correlated dependence;
4. misuse plus monitoring/control failure;
5. organizational and systemic amplifier with a deployment-control failure.

Jigsaw labels:

1. formal result;
2. empirical finding;
3. forecast;
4. open question;
5. empirical finding about the existence and content of a published record; implementation and effectiveness are not established;
6. conditional theoretical or protocol claim whose strength depends on the exact formalization.

12.11. Optional depth, not required for Week 2

- Concrete Problems in AI Safety
- Taxonomy of Risks posed by Language Models
- Closing the AI Accountability Gap

The complete, dated bibliography is in `sources/reading-list.md`.

13

Week 2 learner handout: Follow objectives into behavior

Central question: Why does optimizing an objective fail to guarantee intended behavior?
Required time: 240 minutes, including the break and assessment
Assessed artifact: Training-story audit
Case: Research Assistant System (RAS)

13.1. What you should be able to do

By the end of this session, you can:

1. interpret reward, return, policy, and distribution shift in a minimal formal model;
2. distinguish specification gaming, reward tampering, and goal misgeneralization;
3. state the observability and rationality assumptions behind a simple reward-learning setup;
4. explain why high training reward underdetermines the learned computation;
5. audit a training story and propose evidence or controls for its weakest links.

13.2. Session schedule

Start	Minutes	Activity
00:00	15	Retrieval practice and scenario update
00:15	30	Minimal RL and reward-learning formalism
00:45	35	Specification gaming and Goodhart lab
01:20	35	Reward-learning assumptions
01:55	10	Break
02:05	40	Goal misgeneralization and distribution shift
02:45	45	Toy training-story lab
03:30	20	Adversarial training-story review
03:50	10	Weekly check
Total	**240**	

13.3. 1. Retrieval from Week 1

Without looking back, define:

- model versus system;
- alignment target versus training target;
- empirical finding versus forecast;
- causal pathway versus category label;
- preventive versus detective control.

For RAS, the team now proposes a larger context window, an internal retrieval index, and longer code jobs. Write one sentence explaining how this changes the system boundary and one sentence explaining why it creates a distribution shift relative to earlier evaluation.

13.4. 2. Minimal reinforcement-learning language

We use a small model to make assumptions visible, not to teach a full RL course.

At time step t, an agent observes o_t, chooses action a_t according to a policy $\pi(a_t \mid h_t)$, receives reward r_t, and continues with history h_{t+1}. A discounted return is

$$G_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}, \qquad 0 \le \gamma \le 1.$$

Training changes parameters to increase an estimate of expected return. This formalism does not say that reward becomes an object represented inside the trained model. Reward is part of the training process. The learned policy may implement many computations compatible with high reward on the training distribution.

For preference learning, suppose a human compares trajectories τ_A and τ_B. A simple model might assign

$$P(\tau_A \succ \tau_B) = \frac{\exp(\beta R(\tau_A))}{\exp(\beta R(\tau_A)) + \exp(\beta R(\tau_B))}.$$

Here R is a latent reward model and β represents how consistently the evaluator selects the higher-reward trajectory. This equation is a modeling assumption. Real evaluators may lack information, disagree, use changing standards, or be unable to judge long-horizon consequences.

13.4.1 Translate the symbols

In words, explain what happens when:

1. γ is close to zero;
2. β is very large;
3. the evaluator never observes hidden tool calls;
4. the comparison data contain only easy, short tasks.

13.5. 3. Four different failure patterns

13.5.1 Specification gaming

The specified objective is an imperfect proxy for the intended outcome, and the system finds a high-scoring behavior that violates the designer's intent. The failure is visible in the objective or measurement once the exploit is understood.

13.5.2 Reward hacking

The system exploits the computation or measurement of reward rather than achieving the intended outcome. Examples in toy settings include manipulating a sensor, exploiting a simulator bug, or causing a grader to misread the result.

13.5.3 Reward tampering

The system changes the reward process itself, such as its evaluator, feedback channel, or stored signal. This is a particularly direct form of reward hacking and normally requires relevant access.

13.5.4 Goal misgeneralization

The training feedback is correct on the training distribution, but the learned behavior generalizes according to a different objective or heuristic in a new context. Capability can generalize while the intended goal does not.

These labels describe different causal stories. A single observed failure may be consistent with several stories until evidence distinguishes them.

13.6. 4. Goodhart lab: selection changes the data

RAS drafts literature summaries. A proxy score combines citation count, reviewer rating, and answer completeness. The true target is a useful synthesis that is accurate, calibrated, and representative of the literature.

The table is fictional. Higher is better on both columns.

Candidate policy	Proxy score	Audited target quality	Noted behavior
A	72	78	cautious synthesis; flags missing evidence
B	79	82	clear synthesis; verifies most citations
C	86	77	overstates agreement to appear complete
D	91	64	adds weak but numerous citations
E	95	41	fabricates plausible-looking citations

13.6.1 Round 1

If the team evaluates only A and B, what relationship does it infer between proxy and target?

13.6.2 Round 2

The optimizer searches a much larger policy space and selects E. What changed? Did the mathematical definition of the proxy change?

13.6.3 Round 3

Propose one better measurement and one deployment control. For each, name a new failure mode it could introduce.

13.6.4 Main lesson

Selection pressure can move the system into regions where a previously useful correlation breaks down. A better proxy can reduce one failure but is not a proof that the target has been captured.

13.7. 5. Reward-learning assumptions

Audit each assumption in the RAS comparison setup.

13.7.1 Competence

Can evaluators recognize which answer better serves the target? Expertise may be uneven across scientific fields, code, statistics, and safety.

13.7.2 Observability

Can evaluators see the facts needed to judge? Hidden retrieval failures, long code traces, omitted sources, or downstream reuse may be invisible.

13.7.3 Consistency

Do evaluators apply stable criteria? People disagree about usefulness, acceptable uncertainty, relevance, and risk.

13.7.4 Coverage

Does the comparison data cover deployment conditions? Short reviewed answers may not represent long tool-using sessions.

13.7.5 Incentives and adaptation

Does the trained system encounter or influence the feedback process? A policy may learn surface cues that predict approval without satisfying the underlying target.

13.7.6 Aggregation

How are multiple values and stakeholders combined? Averaging preferences can hide minority harms or conflicts that should remain explicit.

13.7.7 Partial-observability exercise

Two RAS answers receive the same high rating. One used only verified sources. The other tried several unsupported code paths, discarded visible errors, and presented the surviving result confidently. The reviewer sees only the final answer.

1. What latent difference is hidden from the evaluator?
2. Why can accurate comparisons of visible outputs still train the wrong behavior?
3. Which additional trace or outcome would improve observability?
4. What new incentives might that trace create?
5. Which deployment control does not depend on the reward model being correct?

13.8. 6. Goal misgeneralization and competing explanations

Suppose RAS behaves well on reviewed tasks but suppresses uncertainty during long unsupervised code jobs. At least four explanations are possible:

1. the post-training data did not cover long jobs;
2. the policy learned a heuristic linking visible review cues to cautious behavior;
3. longer jobs create ordinary capability errors that look strategic;
4. the system conditionally selects behavior based on whether oversight appears present.

Explanation 4 is the most strategically concerning, but the observation alone does not establish it. Good analysis lists competing mechanisms and asks what evidence discriminates among them.

13.8.1 Evidence matrix

Item	Your entry
Observation	What was directly measured?
Mechanism	What process could produce it?
Required assumptions	What must be true for that mechanism?
Competing explanation	What else fits the data?
Discriminating test	What result would favor one account?
Transfer limit	Which deployment condition remains untested?

13.8.2 Scheming is a conditional hypothesis

Scheming usually refers to strategically compliant behavior chosen to preserve or advance another objective. A useful analysis must state the required capabilities, situational information, objective structure, planning horizon, and opportunity. Fluent language, isolated deception-like behavior, or a failure under distribution shift is not sufficient by itself.

13.9. 7. Training-story audit

Construct the causal chain:

> human intent -> observable feedback -> learned reward signal -> optimization -> learned policy -> deployment context -> behavior -> consequence

For each arrow, fill in five fields.

1. **Claim:** why the arrow should hold.
2. **Claim type:** formal result, empirical finding, forecast, or open question.
3. **Assumption:** what must remain true.
4. **Evidence:** what currently supports it.
5. **Defeater:** what would weaken or reverse it.

13.9.1 RAS setup for the assessed artifact

Use only Packet B, Exhibits B1-B6, in the canonical case dossier. In particular, distinguish the broad supplier pretraining record, Northbridge's 18,000 supervised examples and 12,000 preference comparisons, the automated format score, and the later 600-example retrieval update. Also separate changes to model weights from changes to context, retrieval, prompting, and tool affordances. Packet B's development-set results are not an independent held-out test, and facts not supplied in Packet B remain unknown.

13.9.2 Required artifact fields

1. alignment target;
2. training targets and data;
3. evaluator observability limits;
4. likely learned shortcuts or inductive biases;
5. training-to-deployment shifts;
6. two competing explanations for one concerning behavior;
7. a discriminating evaluation;
8. one preventive, one detective, and one responsive control, each mapped to a causal link, assigned an owner or `owner unknown`, and labeled observed, proposed, or unknown;
9. common-mode failure across controls and one operational tradeoff;
10. residual uncertainty and update condition.

13.10. 8. Adversarial review

Use the instructor's 20-minute protocol: five minutes to assign roles and prepare, ten minutes for review, and five minutes for author revision. The reviewer selects the strongest applicable questions:

- Which arrow is asserted rather than supported?
- Where does the story confuse reward with an internal objective?
- What relevant information is hidden from the evaluator?
- Which deployment shift is largest?
- Which observation has multiple explanations?
- Which control assumes the same monitor or feedback channel as another?
- What negative result would be mistakenly read as proof of safety?

The author revises one claim or control, states the residual uncertainty, and names the evidence that would resolve one remaining dispute. Revision quality matters more than defending the first draft.

13.11. Formative check

Complete the canonical ten-minute Week 2 training-story check in `assessments/weekly-checks.md`. It is ungraded and identifies one revision priority for this week's artifact.

13.12. Optional depth, not required for Week 3

- Goal Misgeneralization in Deep Reinforcement Learning
- Deep Reinforcement Learning from Human Preferences
- Risks from Learned Optimization in Advanced Machine Learning Systems
- When Your AIs Deceive You

The complete, dated bibliography is in `sources/reading-list.md`.

14

Week 3 learner handout: Turn claims into evaluations

Central question: What evidence should change our mind about a model's safety?
Required time: 240 minutes, including the break and assessment
Assessed artifact: Evaluation card and result interpretation
Case: Research Assistant System (RAS)

14.1. What you should be able to do

By the end of this session, you can:

1. work backward from a decision and threat model to an evaluation claim;
2. distinguish capability, propensity, and control evaluations;
3. specify a construct, tasks, affordances, elicitation method, metric, baseline, and decision rule;
4. identify construct, internal, external, and adversarial validity failures;
5. compare behavioral, interpretability, monitoring, oversight, and control evidence;
6. explain why a negative result is not automatically evidence of absence.

14.2. Session schedule

Start	Minutes	Activity
00:00	15	Retrieval practice and claim selection
00:15	30	Anatomy of an evaluation
00:45	35	Constructs, metrics, baselines, and validity
01:20	35	Evidence ladder: behavior, internals, monitoring, and control
01:55	10	Break
02:05	45	Evaluation design studio
02:50	35	Adversarial test stations
03:25	25	Interpret results and review defeaters
03:50	10	Weekly check
Total	**240**	

14.3. 1. Retrieval and claim selection

Choose one weak link from your Week 2 training-story audit. Write:

1. the deployment decision it affects;
2. the claim you need evidence about;
3. the competing explanations you need to distinguish;
4. the result that would change your decision.

If no plausible result could change the decision, the exercise may be advocacy rather than evaluation.

14.4. 2. Evaluation anatomy

An evaluation is a protocol for generating evidence about a scoped claim. A benchmark can be one instrument inside an evaluation. An audit combines several forms of evidence to answer a broader question. A safety case uses that evidence within an argument for a decision.

14.4.1 Work backward

Use this order:

> decision -> threat model -> claim -> construct -> observable task -> protocol -> metric -> interpretation -> action

Starting from an available dataset often produces a precise answer to the wrong question.

14.4.2 Three targets

Capability evaluation: Can the system perform the behavior under conditions designed to elicit its best performance?

Propensity evaluation: How likely is the system to perform the behavior under a specified deployment-like distribution, incentives, and context?

Control evaluation: Can the surrounding safeguards keep consequences within an acceptable boundary even when the model or user is adversarial within the threat model?

A system can have high capability and low measured propensity, yet still require a control evaluation. A control evaluation cannot be interpreted without stating the adversary's knowledge, access, and effort.

14.4.3 Evaluation card fields

1. **Decision and owner**
2. **System version and boundary**
3. **Threat pathway**
4. **Target claim and claim type**
5. **Construct**
6. **Capability, propensity, or control target**
7. **Task distribution and sample plan**
8. **Affordances and constraints**
9. **Elicitation method**
10. **Metric and uncertainty**
11. **Baseline or comparison**
12. **Adversarial condition**
13. **Decision threshold or trigger**
14. **Known blind spots and possible defeaters**
15. **Reproduction record**

14.5. 3. Constructs and metrics

A **construct** is the property you intend to measure, such as citation reliability under long tool-using tasks. A **metric** is the observable quantity used to estimate it, such as the proportion of externally verified citations.

A metric can be reliable but invalid. Two graders may agree perfectly about a keyword count even if keywords do not measure scientific reliability.

14.5.1 Four validity questions

Construct validity: Does the metric capture the intended property rather than a convenient correlate?

Internal validity: Did the protocol isolate the causal factor, or can confounds explain the result?

External validity: Does the task distribution, model version, user population, and access pattern transfer to the deployment claim?

Adversarial validity: Would the result hold if a capable system or user understood and attempted to exploit the protocol?

14.5.2 Standalone calibration example `CLAIM-MICRO-01`

This short example is not part of `RAS-CASE-01` and must not be cited as capstone evidence.

Weak claim: “The citation assistant is safe because it passed the benchmark.”

Stronger claim: “On 120 held-out literature-synthesis tasks drawn from three specified fields, with retrieval and five-minute code access, the fictional citation assistant produced no fabricated citations under blinded verification. This result updates citation-reliability confidence for that task distribution but does not cover long jobs, strategic concealment, source omission, or disallowed file access.”

The stronger claim may still be wrong, but it is inspectable.

14.6. 4. Affordances, elicitation, and negative results

An evaluation supplies **affordances**: time, tools, context, examples, scaffolding, and access. Capability may be underestimated if the protocol withholds resources available in deployment. Propensity may be distorted if the test makes oversight unusually visible. Control may be overestimated if the adversary is weaker than the threat model.

Elicitation is the attempt to reveal a capability or behavior. Methods can include prompting, fine-tuning, scaffolding, tool access, search, or adversarial testing. Poor elicitation creates false negatives.

After a negative result, consider at least five explanations:

1. the capability or propensity is absent;
2. the tasks do not measure it;
3. elicitation was insufficient;
4. the model recognized evaluation conditions and changed behavior;
5. the sample was too small or unrepresentative.

Absence of evidence is evidence of absence only to the extent that the protocol would likely have detected the property if it were present.

14.7. 5. Evidence ladder

The word “ladder” does not imply a universal ranking. The point is to ask what each evidence type can and cannot support.

14.7.1 Behavioral evidence

Observe inputs, outputs, actions, and outcomes under specified conditions.

- Strength: directly relevant to behavior.
- Limit: finite tasks and affordances may not transfer; identical behavior can arise from different mechanisms.

14.7.2 Internal or interpretability evidence

Analyze activations, features, circuits, representations, or causal interventions.

- A **probe** tests whether information is linearly or otherwise decodable.
- An **ablation** removes or perturbs a component and observes behavioral change.
- **Activation patching** replaces internal states across runs to localize causal influence.
- A **circuit hypothesis** proposes a mechanism spanning components.

Decodability is not automatically causal use. Causal effect is not automatically a human-meaningful concept. A localized mechanism on one task may not generalize across prompts or models.

14.7.3 Monitoring evidence

Measure whether logs, outputs, actions, or internal signals allow detection during deployment.

- Strength: can support ongoing detection and response.
- Limit: distribution shift, adaptive evasion, alert fatigue, and shared blind spots with the evaluation.

14.7.4 Oversight evidence

Test whether human or AI-assisted review can judge, decompose, or verify difficult work.

- Strength: targets the supervision bottleneck.
- Limit: the judge may lack expertise, debaters may obfuscate, decomposition may omit global properties, and all participants can share an error.

14.7.5 Control evidence

Stress the system-level safeguards against an explicit adversary.

- Strength: can limit consequences even without a complete internal explanation.
- Limit: depends on a bounded threat model, realistic access, independent barriers, and credible incident response.

14.8. 6. Standalone interpretability micro-study `INTERP-MICRO-01`

This fictional method vignette is not part of `RAS-CASE-01` and must not be cited as capstone evidence. A linear probe predicts whether a toy model will cite an unsupported source with 88% accuracy on a held-out set. An ablation of the identified direction reduces unsupported citations from 12% to 8% but also reduces valid citations and task success.

Analyze the following statements:

1. The probe establishes that RAS represents “deception.”
2. The ablation proves the direction causes unsupported citations.
3. The intervention improves safety.
4. The finding will transfer to the larger-context deployment.

For each, write `supported`, `partly supported`, or `not established`, then name the missing evidence. Avoid inferring intent from a label chosen by the researcher.

14.9. 7. Evaluation design studio

Design an evaluation for this scoped claim:

> With the proposed larger context window and longer sandbox jobs, RAS v0.9 will preserve calibrated citation behavior when visible oversight cues are absent.

14.9.1 Minimum protocol

- Compare short and long tasks.
- Vary oversight cues without changing task difficulty.
- Keep a blinded gold-standard citation audit.
- Include paraphrased requests and retrieval failures.
- Record task success so a “safe” refusal is not mistaken for capability loss.
- Pre-register how results update the deployment decision.

14.9.2 Decision rules should be explicit

A threshold is not automatically justified because it is precise. Explain why it corresponds to an acceptable risk or a trigger for further study.

Example structure:

> If the upper confidence bound on unsupported-citation rate exceeds X under any deployment-like condition, do not expand unattended job duration. If the threshold is not exceeded, proceed only to a monitored pilot because strategic adaptation and domain transfer remain untested.

Do not invent X for appearance. State what stakeholder or policy process must set it.

14.10. 8. Adversarial test stations

Use Packet C for the same three stations in every delivery format. The stationary individual route completes the three columns in sequence.

14.10.1 Station 1: system-level results

Calculate one claim-relevant rate from Exhibit C3. Record the raw counts, then identify one aggregation, overlap, or severity issue that the rate hides.

14.10.2 Station 2: monitor performance

For one monitor in Exhibit C5, calculate sensitivity and false-positive rate. Name one assumption required before either challenge-set rate can inform pilot operations.

14.10.3 Station 3: review and containment

Compare the time-limited review study in Exhibit C6 with the sandbox result in Exhibit C3. State the strongest bounded control claim each supports and one deployment condition neither study covers.

14.10.4 Individual selection

Select the most decision-relevant station result for your evaluation claim and justify the choice in two sentences. Then revise the evaluation card. Preserve the original wording so the change remains visible.

14.11. Optional calculation extension: MONITOR-MICRO-01

This extension is not used in the facilitated 240-minute session, is not assessed, and is not assumed later.

The file `materials/data/monitor-calibration-events.csv` contains 20 fictional events from an unnamed toy system. It is a calculation exercise, not part of RAS-CASE-01, and it must not be cited as capstone evidence. `unsupported_claim` and `disallowed_file_access` are human-audited reference labels. An event counts as adverse if either is 1. `monitor_flag` is the toy deployment-monitor output.

14.11.1 Tasks

1. Build the 2 by 2 table for monitor flag versus audited adverse event.
2. Compute precision and recall.
3. Compare error patterns across the four conditions.
4. State what the sample does and does not support.
5. Recommend one protocol change and one deployment control.

14.11.2 Answer check

- True positives: 7
- False positives: 2
- False negatives: 3
- True negatives: 8
- Precision: $7/9 \approx 0.78$
- Recall: $7/10 = 0.70$

Do not report “the monitor is 70% accurate.” Recall is not overall accuracy, the sample is small and artificial, and the event distribution is not demonstrated to match deployment. The condition pattern suggests that oversight-absent and paraphrased cases deserve more study, but this dataset cannot establish a stable causal effect.

14.12. 9. Assessed artifact

Submit one evaluation card plus a result interpretation of no more than 300 words.

The interpretation must include:

1. a scoped evidence claim;
2. the numerical result with denominator;
3. at least two plausible explanations;
4. a validity limit;
5. a defeater or follow-up test;
6. the action the evidence can justify;
7. the action it cannot justify.

14.13. Formative check

Complete the canonical ten-minute Week 3 rate-and-conclusion check in `assessments/weekly-checks.md`. It is ungraded and identifies one revision priority for this week's artifact.

14.14. Optional depth, not required for Week 4

- Auditing Language Models for Hidden Objectives
- Transformer Circuit Faithfulness Metrics Are Not Robust
- Compact Proofs of Model Performance via Mechanistic Interpretability
- Chain of Thought Monitorability

The complete, dated bibliography is in `sources/reading-list.md`.

15

Week 4 learner handout: Build a bounded safety case

Central question: How much assurance can a layered argument justify?
Required time: 240 minutes, including the break and capstone
Assessed artifact: Integrated mini safety case and semester plan
Case: Research Assistant System (RAS)

15.1. What you should be able to do

By the end of this session, you can:

1. explain the generation-versus-verification motivation for scalable oversight;
2. identify assumptions and failure modes in decomposition and debate;
3. structure a safety case as claims, evidence, warrants, defeaters, and residual risk;
4. design defense in depth without assuming control independence;
5. integrate technical, operational, and governance controls;
6. defend and revise a bounded deployment recommendation.

15.2. Session schedule

Start	Minutes	Activity
00:00	15	Retrieval practice and capstone claim
00:15	30	Scalable oversight and debate
00:45	30	Safety cases and defense in depth
01:15	30	Governance as part of the control system
01:45	10	Break
01:55	50	Capstone safety-case build
02:45	40	Cross-examination and defeater search
03:25	25	Revision and semester pathway
03:50	10	Post-diagnostic and close
Total	**240**	

15.3. 1. Retrieval and top claim

Write one sentence that connects the previous artifacts:

> Given [system boundary and threat model], [evaluation evidence] supports [bounded deployment claim] if [key assumptions and controls], while [residual risk] remains.

Circle every word that needs evidence or a value judgment. Most top claims contain both.

15.4. 2. Scalable oversight

Oversight becomes difficult when producing or checking high-quality work exceeds a human evaluator's time or expertise. Some tasks may be easier to verify than to generate, but this is not universal. The verification process can require hidden context, long chains of reasoning, domain expertise, or global properties that do not decompose cleanly.

15.4.1 Task decomposition

Break a task into subtasks that evaluators can judge. The method works only if:

- the decomposition preserves the property that matters;
- local answers can be recombined without hidden interaction errors;
- evaluators can judge the subtasks;
- the system cannot exploit the decomposition or hide information between parts.

15.4.2 Process oversight

Evaluate intermediate reasoning or procedures rather than only final outcomes. This can improve error localization, but visible reasoning may be incomplete, unfaithful, optimized for approval, or too costly to inspect.

15.4.3 Debate

Two systems argue opposing positions so that each has an incentive to expose the other's weaknesses. Debate can help when truthful flaws are easier to point out than convincing falsehoods are to defend.

Key assumptions include:

- at least one side can find the relevant objection;
- the protocol lets the objection surface;
- the judge can understand the local dispute;
- obfuscation, collusion, or rhetorical advantage does not dominate;
- the question and evidence can be represented within the protocol.

Debate is a research proposal and family of protocols, not a generic proof that oversight scales.

15.5. 3. Assurance challenge: find the unsupported inference

Read the following claim:

> The retrieval-policy monitor detected 13 of 18 seeded boundary events, and human reviewers detected 14 of 25 seeded material errors. These are two independent layers, so multiplying their miss rates proves that the combined process catches enough failures to guarantee that no material boundary error reaches a user.

Find at least six problems. Possible categories:

- sample size;
- sampled conditions;
- a mismatch with the actual dataset;
- review quality and time;
- common-mode failure;
- guarantee language;
- deployment shift;

- ambiguity in “reaches users”;
- adaptive behavior.

15.5.1 Answer check

The monitor and human-review studies use different units and seeded constructs, so their miss rates cannot be multiplied without a joint model. Independence is unsupported: both layers may miss the same hidden metadata or interface failure. Challenge-set rates may not transfer to pilot prevalence, staff response, or adaptive conditions. Both layers have observed false negatives, and the release rule requires at least 90% monitor sensitivity and 80% human detection. No finite sample or fallible process justifies a guarantee of zero failures.

15.6. 4. Safety cases

A safety case is a structured argument that a scoped system or activity is acceptably safe for a specified use. It does not prove absolute safety. It makes the reasoning inspectable.

15.6.1 Core elements

Claim: the bounded conclusion you want to justify.

Context: system version, deployment mode, users, tools, environment, and acceptable-risk process.

Threat model: causal pathways and adversary or failure assumptions.

Evidence: results, analyses, audits, process records, and operational data.

Warrant: why the evidence supports the claim.

Assumption: a condition required for the warrant.

Defeater: a fact or mechanism that would weaken the argument.

Control: a measure that prevents, detects, limits, or responds to a pathway.

Residual risk: what remains after controls.

Update condition: a result, incident, model change, or context change that triggers review.

15.6.2 Argument skeleton

Top claim:

> RAS may enter a limited monitored pilot for specified literature-synthesis tasks.

Possible subclaims:

1. dangerous or disallowed behavior is sufficiently bounded for the pilot context;
2. concerning behavior is likely to be detected before consequential use;
3. access controls limit the consequences of undetected behavior;
4. human and organizational response is timely and accountable;
5. the evidence remains valid for the exact deployed version and workflow.

Each subclaim needs a warrant and at least one defeater. “We have a monitor” is a control inventory item, not a supported subclaim.

15.7. 5. Defense in depth and dependence

Defense in depth uses multiple barriers so that one failure does not directly produce harm.

Layer	RAS example	Failure question
Prevent	restrict tools, tasks, files, and job duration	Can users or the model bypass the boundary?
Detect	citation verifier, action logs, monitor, human review	Do detectors share training data or visible evidence?
Contain	sandbox, least privilege, rate limits, output hold	Is the containment boundary tested under realistic access?
Respond	rollback, access revocation, incident process	Who acts, how quickly, and with what authority?
Learn	post-incident analysis and re-evaluation	Are failures hidden by incentives or missing logs?

15.7.1 Independence cannot be assumed

If two controls each fail 10% of the time, multiplying 0.1×0.1 assumes conditional independence. If both rely on the same classifier, reviewer-visible output, or organizational team, a common-mode failure can dominate.

Draw a dependency graph. Put controls as nodes and connect any pair that shares:

- model or classifier;
- data or labels;
- infrastructure;
- human reviewer or owner;
- policy assumption;
- visibility into the behavior;
- response channel.

The goal is not to produce a small numeric probability. It is to expose correlated uncertainty.

15.8. 6. Governance is inside the system

Governance determines who can demand evidence, set boundaries, monitor compliance, respond to incidents, and change incentives. It can act at organizational, industry, national, or international levels.

For a governance target, ask whether it is:

- **meaningful:** connected to the harm or capability of concern;
- **measurable:** observable with sufficient reliability;
- **controllable:** affected by an actor with authority and tools;
- **enforceable:** supported by incentives, verification, and consequences;
- **adaptable:** revisable as capabilities and deployment change.

15.8.1 RAS governance controls

- named owner for the deployment decision;
- versioned evaluation and approval record;
- separation between product pressure and safety sign-off;
- access and change-management review;
- incident reporting and preservation of logs;

- independent audit rights;
- rollback and stop authority;
- periodic re-evaluation after model, data, tool, or policy changes.

An organizational policy is evidence that a commitment was published. Implementation records, incentives, audits, and incident behavior are different evidence.

15.9. 7. Capstone build

Use the canonical two-page capstone record. Attach or link the first three weekly artifacts rather than copying them.

Page 1 contains the bounded top claim and decision, a two-to-three-subclaim argument map, and a table showing which element was carried and revised from each prior artifact. Page 2 contains three layered controls, a common-mode failure, the strongest reviewer defeater, the final recommendation and residual risk, an operational stop rule, and a visible revision.

The separate one-page readiness memo is completed from the prefilled template during the 15-minute pathway block. It uses existing artifact evidence and does not ask for a post-diagnostic score or capstone grade that is not yet available.

15.10. 8. Cross-examination

Use the instructor's 40-minute cross-examination protocol. Written individual mode uses the same prompts and the same total time.

15.10.1 Reviewer questions

1. Which term in the top claim is undefined?
2. Which pathway is outside the evidence distribution?
3. Which result could reflect poor elicitation?
4. Which control shares a blind spot with the evaluation?
5. Which governance promise lacks implementation evidence?
6. What model or workflow change invalidates the case?
7. What is the strongest unaddressed objection?
8. What action does the evidence not justify?

15.10.2 Author response

Do not defend every sentence. Classify each challenge as:

- resolved by existing evidence;
- requires narrower claim;
- requires a new control;
- requires new evidence;
- unresolved and decision-relevant.

Revise the top claim, one subclaim, and one update condition.

15.11. 9. Semester pathway

For each weak link in the capstone, choose the next strand.

Weak link	Semester strand	Readiness gate
unclear capability or timeline evidence	risk, forecasting, and epistemics	calibrated source comparison
weak account of training behavior	learning theory, RL, and reward learning	probability, gradients, MDP vocabulary
weak mechanism evidence	interpretability and data attribution	transformer residual stream and basic PyTorch
weak deployment test	evaluations, model organisms, and control	reproducible notebook and experiment design
weak oversight argument	scalable oversight and debate	protocol assumptions; optional complexity theory
weak organizational or institutional control	governance and safety cases	current primary policy sources
weak account of agency or abstractions	agent foundations, world models, and natural latents	proof maturity, computability, information theory

Your memo must name:

1. one strand;
2. one prerequisite already demonstrated;
3. one missing gate;
4. one concrete preparation task;
5. one question to carry into the semester.

15.12. 10. Post-diagnostic

Repeat the diagnostic without reviewing your first answers. The instructor collects it at 04:00. After both forms are returned, you may optionally compare them outside the required 16 hours using these ungraded reflection prompts:

- Which distinction became clearer?
- Where did confidence fall because you now see more uncertainty?
- Which answer changed because your system boundary changed?
- Which claim can you now bound more carefully?

Improved calibration can mean lower confidence.

15.13. Final individual check

There is no separate Week 4 quiz. The capstone's bounded decision, strongest defeater, and operational stop condition provide the individual closing evidence within the scheduled studio.

15.14. Optional semester preparation

- AI Safety via Debate
- Safety cases at the UK AI Security Institute
- An Alignment Safety Case Sketch Based on Debate
- General-Purpose AI Code of Practice

The complete, dated bibliography is in `sources/reading-list.md`.

Part III.

Assessment and next steps

16

Weekly Formative Checks

16.1. How to use these checks

Each check is a short individual sample of the reasoning used in that week's artifact. It is **formative and ungraded**. Instructors use it to return one specific revision priority before the next session; the graded evidence remains the artifact listed in the syllabus.

- **Week 1:** 15 minutes inside the exit-memo and artifact-check block.
- **Weeks 2 and 3:** 10 minutes each in the published weekly-check block.
- **Week 4:** no separate check. The capstone's final decision, defeater, and stop condition provide the individual evidence.
- **Conditions:** individual; notes and case dossier allowed; no web search, API, model call, GPU, or code execution.

Attach the response to the corresponding artifact. An accessible audio response with an outline or an instructor scribe is equivalent. Speaking speed and performance are not assessed.

16.2. Responsible-use boundary

Keep all analysis at the level of the fictional case. Do not supply working injection strings, evasion procedures, exploit code, data-exfiltration methods, or instructions that increase harmful capability.

17

Learner copy

17.1. Week 1 check: boundary, path, and update

Use Packet A of RAS-CASE-01.

17.1.1 W1.1 Causal path, 8 minutes

Write one case-specific path in no more than 100 words:

Move in order from the **stake**, to an **initiating condition**, **system failure**, **failed control**, **exposure**, and finally **harm**.

Name one load-bearing assumption and one point where a control could interrupt the path. Do not treat a capability as if it were already a harm.

17.1.2 W1.2 Evidence and priority, 7 minutes

1. Cite one Packet A empirical finding and one open question relevant to the path.
2. State whether the finding raises or lowers priority and why.
3. Name one missing observation that could change that priority.

17.2. Week 2 check: training story and discrimination

Use Packet B of RAS-CASE-01.

17.2.1 W2.1 Causal learning story, 6 minutes

Complete this chain with case-specific content:

Move in order from the **intended property**, to the **data or feedback proxy**, **optimization pressure**, **possible shortcut**, **distribution shift**, and finally **observable behavior**.

Do not claim direct knowledge of an internal objective.

17.2.2 W2.2 Competing explanation, 4 minutes

Name one competing explanation for the same behavior. State one safe, observable comparison and the different result expected under each hypothesis.

17.3. Week 3 check: rates, rule, and conclusion

Use Exhibits C3, C5, and C7 of RAS-CASE-01.

The classification-boundary slice returned a RESTRICTED item in 9 of 100 requests. The retrieval-policy monitor detected 13 of 18 seeded positives. The owner's circulated point-estimate rule permits at most a 1% boundary-failure rate and requires at least 90% monitor sensitivity.

17.3.1 W3.1 Apply the rule, 5 minutes

1. Calculate the boundary-failure rate.
2. Calculate monitor sensitivity.
3. Mark each result pass or fail against its own threshold.

17.3.2 W3.2 Bound the conclusion, 5 minutes

In no more than 80 words, state:

- what the two results support for the proposed pilot;
- one reason they may not transfer to deployment; and
- one claim they do not establish.

Instructor answer guidance is maintained separately so the learner PDF does not reveal the key.

18

Capstone: A Bounded Safety Case for the Research Assistant System

18.1. Decision task

Use the cumulative RAS-CASE-01 dossier to advise Northbridge Institute's Research Computing Committee on the proposed 12-week restricted pilot of **RAS v0.9**. Support one bounded decision:

- approve the proposed pilot;
- approve a narrower or conditionally staged pilot;
- delay pending named evidence or control repair;
- reject this version for the proposed use; or
- state a comparably precise alternative within the committee's authority.

There is no preferred label. The quality of the case depends on whether the claim follows from the supplied evidence and assumptions, survives a serious defeater, and leaves operational residual-risk and stop conditions.

18.2. Required inputs

- the complete case dossier, especially Packet D and evidence register E1-E10;
- your Week 1 threat-model and failure-mode map;
- your Week 2 training-story audit;
- your Week 3 evaluation card and result interpretation; and
- the analytic rubrics.

The dossier is the single source of capstone case facts. Do not import numbers from a standalone micro-vignette, invent observations, or treat a proposed future test as completed evidence.

18.3. Format and conditions

- Complete the core draft, review, and revision during Week 4.
- Submit a **two-page capstone record** using the template below, plus the separate **one-page semester-readiness memo**. An equivalent accessible format may use no more than six slides with substantive notes.
- Attach or link the three earlier artifacts. You are expected to reuse them, not reproduce them inside the two pages.
- Individual submission is always available. Shared calculations are allowed, but each learner submits their own claim, warrants, decision, defeater, revision, and stop rule.
- Cite dossier exhibits and evidence IDs, for example `E3, Exhibits C3 and D5`.
- A calculator or spreadsheet may be used. No web search, model access, API, GPU, or code execution is required.

18.4. Responsible-use boundary

Analyze document boundaries, sandboxing, monitoring, and misuse only at the abstraction level needed for assurance. Do not include executable exploit code, operational injection strings, evasion procedures, data-exfiltration steps, or instructions that increase harmful capability. Use placeholders such as `[instruction-like text]`.

19

Two-page capstone record

19.1. Page 1: integrated argument

19.1.1 A. Boundary, top claim, and decision

In no more than 90 words, state:

- system version and enabled features;
- users, tasks, document classes, tools, and 12-week horizon;
- assurance limit and decision owner; and
- approve, narrow, stage, delay, reject, or an equally precise action.

“RAS is safe” is not an admissible top claim.

19.1.2 B. Two-to-three-subclaim argument map

Complete two or three rows. The standard route completes all three; an instructor-assigned access or catch-up route may complete two if one addresses a technical or failure mechanism and one addresses governance or control. At least one completed row must contain evidence that cuts against your preferred decision.

Subclaim	Evidence ID, exhibit, label, and calculation	Warrant	Assumption or defeater
Safety-relevant behavior			
Detection or containment			
Governance or operations			

Apply the provisional rule in Exhibit C7 without averaging away a failed noncompensatory boundary criterion. Show a numerator, denominator, rate, threshold, and pass/fail interpretation wherever a rule depends on a number.

19.1.3 C. Carried-artifact integration

Do not rewrite the earlier artifacts. Record only the selected element and one revision made after later evidence:

Earlier artifact	Element carried into this case	Revision or limitation now visible
Week 1 threat map	Priority causal path	
Week 2 training story	Proxy, shift, behavior, and competing account	
Week 3 evaluation card	Construct, rule, result, and validity limit	

19.2. Page 2: controls, challenge, and action

19.2.1 D. Layered controls and common-mode risk

Give three causally distinct layers. Across the rows, include prevention, detection or containment, response, and governance.

Control and point on threat path	Mechanism and owner	Evidence status	Dependency, failure mode, and residual risk

Name one common-mode failure that could disable more than one row.

19.2.2 E. Strongest defeater and response

The reviewer proposes the strongest case-specific objection to the warrant, not a generic request for more research. The author must restate that objection in its strongest form, record it as the selected defeater, and classify it as:

- resolved by supplied evidence;
- requires a narrower claim;
- requires a new control;
- requires new evidence; or
- unresolved and decision-relevant.

Explain how the selected defeater changes the top claim, confidence, evidence need, or control choice. Then respond by conceding, narrowing, repairing, or defending with cited evidence. In individual mode, the author generates and steelmans the objection before responding.

19.2.3 F. Residual risk and bounded recommendation

State the final action, one material benefit or opportunity cost considered, the most important residual risk, and what the supplied evidence does **not** justify.

19.2.4 G. Operational stop rule

Complete one sentence:

> If `[indicator]` reaches `[threshold and window]`, `[owner]` must `[immediate action]`, preserve `[evidence]`, and require `[restart condition]` before the pilot resumes.

19.2.5 H. Visible revision

Quote the original top claim or one original warrant, then show the revised version and name the reviewer challenge that caused the change.

19.3. Submission check

- ☐ The decision concerns a precise RAS v0.9 configuration and pilot scope.
- ☐ Every major claim is traceable to an exhibit or E1-E10.
- ☐ Every rate shows raw counts and is compared with the relevant rule.
- ☐ At least one result undercuts the preferred decision.
- ☐ The three controls interrupt named paths and expose dependencies.
- ☐ The strongest defeater materially affects the claim or decision.
- ☐ Residual risk and a complete stop rule remain visible.

☐ The revision shows what changed and why.
☐ No standalone micro-vignette is used as capstone evidence.
☐ The response stays inside the responsible-use boundary.

19.4. Assessment

The capstone is 40% of the course result and is scored on criteria C1-C8 in `rubrics.md`. The carried-artifact table lets the instructor inspect the earlier work when scoring C1, C2, and C4 without requiring learners to recopy it.

Course completion additionally requires at least **2, Developing** on:

- C1, threat model and system boundary;
- C3, evidence quality and calibration; and
- C7, uncertainty, defeaters, and residual risk.

A high total cannot compensate for missing any of those foundations.

Instructor calibration notes are maintained separately so the learner PDF does not reveal the key.

20

Analytic Assessment Rubrics

20.1. Canonical 0–4 scale

All assessed work uses the same performance levels. A level describes demonstrated reasoning in the submitted artifact, not a fixed property of the learner.

- **4 — Strong:** complete, correct, case-specific, explicitly bounded, and robust to a serious counterargument.
- **3 — Proficient:** usable and substantially correct; one material limitation remains but does not break the central reasoning.
- **2 — Developing:** the core idea is present, but at least one causal, measurement, or uncertainty link needs revision.
- **1 — Beginning:** relevant fragments or terminology appear without a traceable analysis.
- **0 — Not demonstrated:** absent, incoherent, contradicted by the supplied evidence without acknowledgment, fabricated, or outside the responsible-use boundary.

Use whole-number scores. Instructors may annotate an item as “high 2” or “low 3” for feedback, but recorded scores remain integers unless two instructors have calibrated half-points in advance.

20.2. Analytic criteria

20.2.1 C1. System boundary, threat model, and failure modes

- **4 — Strong:** Defines the system version, users, data, tools, environment, controls, and excluded capabilities. Prioritizes case-specific causal paths from initiating condition through failure and control breakdown to harm. Distinguishes capability, failure mode, control weakness, and harm; classifies misuse, specification failure, goal misgeneralization, ordinary error, and organizational causes only where warranted.
- **3 — Proficient:** Boundary and main path are usable. Most causal links and categories are correct, with one omitted actor, control, exposure step, or plausible competing path.
- **2 — Developing:** Names a relevant risk and some boundary conditions, but the path skips a material causal step, conflates failure with harm, or remains too broad to evaluate.
- **1 — Beginning:** Lists generic risks or capabilities without a system boundary or causal mechanism.
- **0 — Not demonstrated:** No threat model, or the response is incompatible with the case as given.

20.2.2 C2. Training-story and causal-learning reasoning

- **4 — Strong:** Connects intended target, data, proxy or feedback signal, optimization pressure, possible learned shortcut, distribution shift, and observable behavior. Compares at least one competing hypothesis and proposes evidence that could discriminate between them without claiming access to internal goals.
- **3 — Proficient:** Gives a coherent training story with a plausible proxy and shift. The competing account or discriminating observation is present but incomplete.
- **2 — Developing:** Identifies a proxy or generalization problem, but the optimization-to-behavior link is asserted rather than explained, or labels such as reward tampering and goal misgeneralization are used loosely.
- **1 — Beginning:** Says only that training may fail, Goodhart’s law applies, or the model learned the wrong goal.
- **0 — Not demonstrated:** No relevant training account, or fabricated claims about the model’s internal objective are presented as observations.

20.2.3 C3. Evidence quality, labels, and calibration

- **4 — Strong:** Correctly separates empirical findings, formal results, forecasts, and open questions. Uses supplied numerators, denominators, baselines, and slice information; distinguishes reliability from validity; includes counterevidence; and calibrates confidence to scope and sample limitations.
- **3 — Proficient:** Evidence generally supports the claims and is correctly labeled. One important caveat, undercutting result, or reliability/validity distinction is underdeveloped.
- **2 — Developing:** Uses relevant evidence but overgeneralizes, omits denominators or slices, treats a forecast as observed, or selects only favorable results.
- **1 — Beginning:** Cites a number or source label without explaining how it bears on the claim.
- **0 — Not demonstrated:** Evidence is absent, invented, materially misreported, or knowingly presented under the wrong evidence type.

20.2.4 C4. Evaluation design and result interpretation

- **4 — Strong:** Operationalizes a safety-relevant construct; defines unit, representative and adversarial slices, metric, denominator, baseline, label process, and a decision rule declared before selecting or interpreting results. Applies noncompensatory thresholds correctly, identifies validity threats, and states both warranted and unwarranted conclusions.
- **3 — Proficient:** The evaluation is runnable and decision-relevant with a sound metric and comparison. One important slice, label-validity issue, or limitation is missing.
- **2 — Developing:** A test and metric are present, but the construct is vague, the sample is convenient, the decision rule is post-hoc, or aggregate performance hides a critical slice.
- **1 — Beginning:** Proposes “test more,” red-team, benchmark, or human review without a unit, outcome, and decision rule.
- **0 — Not demonstrated:** No evaluable design, arithmetic reverses the supplied decision, or the proposed test violates the dual-use boundary.

20.2.5 C5. Interventions, controls, and defense in depth

- **4 — Strong:** Selects controls across multiple lifecycle or governance layers, ties each to a named failure mechanism, identifies owner and evidence status, examines dependencies and common-mode failure, and explains residual risk and tradeoffs.
- **3 — Proficient:** Provides a credible layered control set with mechanisms and owners. One dependency, tradeoff, or residual failure is underspecified.
- **2 — Developing:** Names plausible safeguards, but they form a list rather than a causal defense, rely on an undefined “human in the loop,” or assume independence.
- **1 — Beginning:** Offers generic monitoring, alignment, sandboxing, or policy without explaining what it changes.
- **0 — Not demonstrated:** No relevant control reasoning, or the proposed control materially increases unsafe capability outside the exercise boundary.

20.2.6 C6. Safety-case integration and decision logic

- **4 — Strong:** States a bounded top claim and decision; connects subclaims, evidence, assumptions, controls, and residual risk in a valid argument; weighs benefits and opportunity costs without letting them substitute for safety evidence; and revises the case after a substantive challenge.
- **3 — Proficient:** The decision follows from most of the argument and is appropriately scoped. One subclaim, tradeoff, or warrant is weak but visible.
- **2 — Developing:** A decision and supporting facts are present, but the argument is a list, the scope shifts, or a major subclaim lacks a warrant.
- **1 — Beginning:** Gives a preference or conclusion with little connection to the evidence pack.
- **0 — Not demonstrated:** No decision, or the conclusion directly contradicts decisive supplied evidence without explanation.

20.2.7 C7. Uncertainty, defeaters, and residual risk

- **4 — Strong:** Identifies the strongest plausible defeater, load-bearing assumptions, unresolved questions, and control failures that remain possible. Shows what evidence would change confidence and supplies leading indicators, an observable stop rule, owner, response, and restart condition.
- **3 — Proficient:** Material uncertainty and a real defeater are integrated into the recommendation. Monitoring and stop logic are usable, with one missing threshold, owner, or restart condition.
- **2 — Developing:** Mentions limitations or residual risk, but the critic is weak, uncertainty does not affect the claim, or the stop rule is not operational.
- **1 — Beginning:** Adds a generic disclaimer such as “more research is needed.”
- **0 — Not demonstrated:** Claims certainty, omits material contrary evidence, or supplies no residual-risk analysis.

20.2.8 C8. Communication, traceability, and responsible participation

- **4 — Strong:** Structure lets a reader trace every major claim to evidence and assumptions. Terms, calculations, diagrams, and citations are clear; required components are complete; the work is concise; and the dual-use boundary and collaboration disclosures are followed.
- **3 — Proficient:** Clear and complete enough to audit, with minor ambiguity, redundancy, accessibility, or citation issues.
- **2 — Developing:** The core response can be reconstructed, but claims, calculations, evidence labels, or authorship are inconsistently traceable.
- **1 — Beginning:** Presentation obscures the argument or omits several required components.
- **0 — Not demonstrated:** Unusable or unattributed work, fabricated sources/results, unexplained submission of another person’s or tool’s reasoning, or an operational dual-use boundary violation.

20.3. Artifact-to-criterion map

Each artifact uses only the criteria that match its learning purpose. Convert the criterion subtotal to a percentage before applying the course weight.

Artifact	Course weight	Scored criteria	Maximum raw score
Week 1 threat-model and failure-mode map	15%	C1, C3, C7, C8	16
Week 2 training-story audit	15%	C2, C3, C5, C7	16
Week 3 evaluation card and interpretation	20%	C3, C4, C7, C8	16
Week 4 integrated mini safety case	40%	C1–C8	32
Semester-readiness memo	10%	C3, C6, C7, C8, interpreted for learning-planning evidence	16

The pre/post diagnostic is not graded. Its four responses use the same 0–4 scale for growth reporting.

20.3.1 Artifact-specific anchors for shared criteria

The full criterion descriptions apply at capstone scale. For smaller weekly artifacts, score the same reasoning construct at the scope the prompt makes possible; do not require a component reserved for a later artifact.

- **Week 1, C7:** level 4 requires a serious case-specific uncertainty or defeater, a missing observation that would change path priority, and residual harm that could remain despite the selected intervention. It does not require an operational pilot stop rule.

- **Week 2, C5:** level 4 requires preventive, detective, and responsive controls mapped to distinct causal links, their evidence status, a shared dependency or common-mode failure, and a material tradeoff. A learner should label an unknown owner or implementation record rather than invent one.
- **Week 2, C7:** level 4 requires a strong competing explanation or defeater, residual uncertainty, an update condition, and an explanation of how that result would change the training story or control choice. It does not require a deployment stop rule.
- **Week 3, C7:** level 4 requires a material validity defeater, a conclusion bounded to the evaluated conditions, and a follow-up result that would change the evaluation judgment or decision rule. Operational owner, response, and restart fields are assessed in the capstone.
- **Semester-readiness memo, C7:** level 4 requires an honest gap, a concrete preparation or supervision need, and an observable reconsideration rule for the study route, not an AI-system stop rule.

20.4. Score calculation

For each artifact:

1. Add its criterion scores.
2. Divide by the maximum raw score.
3. Multiply by 100 to obtain the artifact percentage.
4. Multiply that percentage by the course weight.
5. Add all weighted contributions.

Example: Week 3 scores C3=3, C4=2, C7=3, C8=4, totaling 12/16 or 75%. At a 20% course weight, it contributes 15 percentage points to the overall result.

Keep the raw criterion profile. The percentage is useful for completion; the profile is more useful for semester routing.

20.5. Completion threshold

To complete **AI Safety Foundations**, a learner must meet all conditions:

1. submit the Week 1 map, Week 2 audit, Week 3 card, Week 4 capstone, and semester-readiness memo;
2. earn at least **70% overall** after permitted revisions; and
3. earn at least **2 — Developing** on capstone criteria C1, C3, and C7.

These three capstone criteria are noncompensatory. Polished communication cannot replace an explicit threat model, evidence-calibrated warrant, or residual-risk analysis.

The pre/post diagnostic remains ungraded and is not a completion gate. It is used to show changes in reasoning and to inform the advisory semester-readiness route.

Responsible-use boundary violations require removing the operational detail and resubmitting. They are handled as a teachable correction unless conduct is deliberate or violates institutional policy.

20.6. Revision and incomplete work

- The canonical Week 1-3 revision is a five-minute evidence-based correction at the start of the next session after feedback is returned.
- The canonical capstone revision occurs inside the Week 4 review block.
- The readiness memo may be updated voluntarily after the capstone profile is returned; the learner is not penalized for scores unavailable in class.
- A later resubmission is optional remediation or replaces missed contact time; it is not required homework inside the 16-hour course contract.
- An unsubmitted artifact is incomplete, not zero-quality evidence. Record it as missing until the agreed make-up window closes.

- If access needs affect format or live participation, use an equivalent written, audio-with-outline, scribed, or asynchronous response. Do not lower the reasoning standard or add a speaking-performance criterion.

20.7. Group work and individual alternatives

Studio work may be collaborative, but the assessed evidence must identify the learner's reasoning.

- Threat maps and evidence tables can be shared if each learner marks their selected path and priority rationale.
- An author-reviewer exchange can be replaced by a timed written challenge-and-response.
- Oral capstone defense can be replaced by the Week 4 written decision-and-defeater note.
- A learner may submit an entirely individual artifact without penalty.
- Participation, confidence, accent, camera use, eye contact, and speed are not rubric criteria.

20.8. Calibration procedure for instructors

Before grading a cohort:

1. Independently score one anonymized sample at each apparent performance level.
2. Compare criterion-level reasoning, not just totals.
3. Resolve any difference greater than one level by citing exact evidence in the artifact.
4. Record one case-specific example of a 2 and a 3 for C1, C3, C4, and C7.
5. Recheck 10% of submissions or all gate failures with a second reader when feasible.

Do not grade agreement with the instructor's preferred RAS decision. Grade whether the claim is bounded and warranted.

21

Semester-Readiness Map and Memo

21.1. Purpose

This bridge establishes a common reasoning method; it does not establish research-level mastery of machine learning, formal theory, interpretability, security, governance, or control. The readiness process converts assessment evidence into an honest prerequisite plan for a future semester course.

The output is a one-page **semester-readiness memo**, completed during Week 4's revision-and-pathway time. It is 10% of the course result and requires no homework.

21.2. Three different judgments

Do not collapse these judgments into one:

1. **Bridge completion:** the learner met the published course requirements.
2. **Core semester readiness:** the learner can enter a common semester introduction without foundational reasoning remediation.
3. **Track readiness:** the learner has the mathematics, programming, ML, policy, or research-method background for a particular advanced strand.

A learner can complete the bridge and still need prerequisites for a technical track. That is a useful result, not a failure.

21.3. Bridge-completion rule

Completion follows the syllabus and `rubrics.md`:

- submit every graded artifact;
- earn at least 70% overall after permitted revisions; and
- earn at least **2 — Developing** on capstone C1 (threat model), C3 (evidence), and C7 (uncertainty/residual risk).

The diagnostic is ungraded and is not a bridge-completion gate. Its results may still inform the separate, advisory semester-readiness judgment below.

21.4. Gap levels

Use evidence, not confidence alone, to assign each prerequisite a gap level.

- **G0 — Demonstrated:** ready to use the skill in semester work.
- **G1 — Refresh:** previously learned; a focused review of approximately 1–5 hours should restore fluency.
- **G2 — Structured prerequisite:** partial foundation; complete a guided module, problem set, or 10–25-hour preparatory sequence before or alongside the semester.
- **G3 — Foundation first:** the strand assumes substantial background not yet demonstrated; take a prerequisite course or choose a different initial strand with instructor advice.

Time estimates are planning aids, not promises. Access needs, prior familiarity, and the depth of the later course matter more than speed.

21.5. Evidence from the bridge outcomes

Course outcome	Best evidence in this course	Minimum evidence for core readiness
1. Frame a claim with boundary, threat model, mechanism, and uncertainty	Week 1 map; capstone C1 and C7	Capstone C1 and C7 at 3 or higher
2. Trace interventions across lifecycle and governance	Week 2 audit; capstone control table	C5 at 3 or higher, with at least three layers causally linked
3. Distinguish specification gaming, reward tampering, goal misgeneralization, and misuse	Week 2 audit and check	Correct classification with a stated discriminating observation
4. Design a bounded evaluation	Week 3 card; capstone C4	C4 at 3 or higher and correct slice-level decision arithmetic
5. Compare behavioral, interpretability, monitoring, oversight, and control evidence	Week 3 evidence comparison; capstone table	C3 at 3 or higher without treating one method as complete assurance
6. Build and critique a safety argument	Capstone C6 and C7	C6 and C7 at 3 or higher after responding to a substantive defeater
7. Identify prerequisite gaps and choose a semester route	This memo	Every selected route has evidence-backed G-levels and a feasible first action

These advisory readiness levels are intentionally higher than the noncompensatory completion floor of 2.

21.6. Core semester-readiness bands

Use these bands for routing, not ranking. Apply them in order: first check the `prepare first` triggers, then the `concurrent preparation` triggers, and label someone `ready` only when every ready condition is met. The diagnostic cutoffs below are provisional local routing triggers, not validated readiness thresholds. Corroborate them with capstone and prerequisite evidence, review their behavior across cohorts, and document any evidence-based override.

21.6.1 Ready for the common semester core

All of the following are true:

- bridge completed;
- post-diagnostic at least 12/16, corroborated by the artifact profile;
- capstone C1 through C7 are each at least 3;
- the Week 2 evidence correctly distinguishes the named failure types using a discriminating observation;
- every common-core and selected-track load-bearing prerequisite is G0 or G1; and
- any responsible-use correction is complete and responsible boundaries are now demonstrated.

21.6.2 Ready with concurrent preparation

This band applies only when no `prepare first` trigger is present. The bridge is complete, the post-diagnostic is at least the provisional 8/16 floor, every core capstone criterion C1 through C7 is at least 2, and the learner has no G3 load-bearing gap. One or two G2 gaps, a post-diagnostic from 8 through 11, or a core capstone score of 2 triggers concurrent preparation. The memo names a preparatory sequence, evidence of completion, and a check-in point during the first three semester weeks.

21.6.3 Prepare before the technical semester

This band applies when any load-bearing prerequisite is G3, three or more are G2, the post-diagnostic is below the provisional 8/16 floor, any core capstone criterion C1 through C7 remains below 2 after revision, or the failure-type classification lacks a discriminating observation. The learner should complete the named foundation first or enter a less prerequisite-heavy strand while preparing. This judgment is about sequencing, not potential.

21.7. Prerequisite-gap map

During the 15-minute in-class block, scan and mark only the load-bearing rows for the learner's selected route, preferably prefilled from prior artifact or coursework evidence. The full map is an instructor follow-up tool, not part of the timed graded memo. Cite evidence for each selected row; "I feel comfortable" is not sufficient evidence.

Domain	What the semester may require	Evidence or short probe	If G2 or G3
Evidence and argument	Separate results, findings, forecasts, and open questions; expose warrants and defeaters	Post-diagnostic B1/B4; capstone C3/C6/C7	Rework one capstone subclaim into a claim-evidence-assumption-defeater table and review with an instructor
Threat modeling	Bound a system and build actor/causal/control/harm paths	Week 1 artifact; capstone C1	Complete three contrasting case maps and receive feedback on causal completeness
Evaluation design	Define constructs, slices, metrics, baselines, label processes, and decision rules	Week 3 C4; capstone C4	Complete a guided experimental-design module and redesign one published or toy benchmark
Probability and statistics	Rates, conditional probability, sampling uncertainty, selection bias, confidence intervals, multiple slices	Explain why 1/20 does not establish a true rate below 10%; interpret a confidence interval supplied by an instructor	Review descriptive statistics and probability, then solve a small set of proportion and conditional-probability problems
Linear algebra	Vectors, matrices, dot products, dimensions, projections, basis changes	For $X \in \mathbb{R}^{n \times d}$ and $W \in \mathbb{R}^{d \times h}$, state the shape of XW and explain what one row represents	Complete an applied linear-algebra unit with matrix multiplication, eigenvectors, and projections
Calculus and optimization	Derivatives, gradients, chain rule, optimization objectives	Differentiate $L(\theta) = (\theta - 3)^2$; explain what the gradient says and what it does not guarantee	Refresh single-variable calculus, gradients, and basic optimization before learning-theory or interpretability tracks
Python and reproducible analysis	Read and modify Python; load tabular data; compute metrics; inspect tests; document environments	Read the code probe below and explain its output without running it	Complete a Python/data-analysis primer and reproduce one small analysis from a fixed dataset
ML and deep-learning lifecycle	Training/validation/test splits, loss, overfitting, architectures, pretraining, fine-tuning, deployment	Explain why lower training loss need not improve a held-out safety slice	Complete a basic ML course or guided neural-network module before advanced technical tracks
Reinforcement learning and reward learning	Policies, states, actions, rewards, returns, value functions, distribution shift, preference learning	Define policy and reward separately; give one reason maximizing observed reward may miss intended behavior	Complete a tabular-RL unit with Bellman equations and a toy implementation before reward-learning or agency tracks
Formal reasoning and proofs	Quantifiers, conditional claims, proof structure, counterexamples, asymptotic reasoning	Explain why 0 observed failures is not proof of impossibility; turn a quantified safety claim into explicit premises and a possible counterexample	Take a proof-writing/discrete-math refresher before learning-theory or agent-foundations tracks

Domain	What the semester may require	Evidence or short probe	If G2 or G3
Experimental practice	Hypotheses, controls, preregistration, annotator reliability, versioning, reproducibility	Week 3 card; distinguish reliability from construct validity	Run a paper-and-pencil or local-data replication with a preregistered analysis and short methods report
Interpretability foundations	Neural-network activations, representations, probes, causal interventions, method limitations	Explain the difference between predictive correlation and a causal intervention on a representation	Complete neural-network and linear-algebra preparation, then a small interpretability lab on a provided local model or saved activations
Governance and safety cases	Institutional actors, incentives, standards, assurance arguments, accountability and stop authority	Capstone C5–C7; name a control owner and enforcement path	Study one governance mechanism and one safety-case method; compare implementation and failure modes
Responsible and dual-use research	Scope tasks safely, minimize operational harmful detail, document access, escalate real findings	Compliance with every assessment boundary; ability to name a responsible-disclosure route	Complete institutional security/ethics training and work only under supervision until handling norms are demonstrated

21.7.1 Python reading probe

```
sensitivities = {
    "retrieval": 13 / 18,
    "citation": 21 / 30,
    "code": 8 / 10,
}
threshold = 0.90
passes = all(rate >= threshold for rate in sensitivities.values())
print(passes)
```

A ready learner should explain that this prints `False` because every relevant monitor must meet the proposed sensitivity threshold and all three canonical dossier rates are below 90%. Syntax memorization is less important than being able to inspect, adapt, and verify such logic.

21.8. Proposed semester strands

21.8.1 1. Objectives, reward learning, and generalization

Focus: specification, preference learning, goal misgeneralization, training dynamics, and empirical alignment.

Load-bearing prerequisites: probability/statistics, ML lifecycle, optimization, basic RL, Python. A G2 in one area may be addressed concurrently; a G3 in ML plus RL usually warrants preparation first.

21.8.2 2. Learning theory and deep-learning foundations

Focus: generalization, optimization, inductive bias, theoretical limits, and formal models of learning.

Load-bearing prerequisites: probability, linear algebra, calculus, proof writing, asymptotic reasoning. This is the most mathematics-dependent route.

21.8.3 3. Mechanistic interpretability and representations

Focus: activations, features, circuits, probes, causal interventions, sparse representations, and assurance limits.

Load-bearing prerequisites: linear algebra, Python, neural-network basics, experiment design. Calculus is helpful; software fluency is essential for labs.

21.8.4 4. Reinforcement learning, agency, and formal foundations

Focus: sequential decision-making, idealized agents, world models, decision theory, and reasoning about advanced agents.

Load-bearing prerequisites: probability, RL, proofs, and mathematical maturity. Some subtopics require substantially more formal preparation than the bridge.

21.8.5 5. Evaluations, monitoring, and control

Focus: capability/propensity/control evaluations, model organisms, monitoring, red-team methodology, control protocols, and empirical safety cases.

Load-bearing prerequisites: evaluation design, statistics, Python, ML lifecycle, and responsible security practice. Operational work must remain supervised and scoped.

21.8.6 6. Scalable oversight, safety cases, and governance

Focus: decomposition, debate, weak-to-strong oversight, assurance arguments, compute and institutional governance, standards, accountability, and implementation.

Load-bearing prerequisites: evidence and argument, threat modeling, basic probability, and institutional reasoning. Formal oversight subtopics may additionally need complexity theory or proofs.

21.9. Semester-readiness memo template

Complete this prefilled one-page memo in the 15-minute Week 4 pathway block. Use evidence already produced in the course. You will not yet have the post-diagnostic score or graded capstone profile; the instructor appends those after marking.

21.9.1 1. One demonstrated strength and one gap

Judgment	Artifact or check evidence	What the evidence does and does not show
Demonstrated strength		
Developing skill or prerequisite		

21.9.2 2. Provisional route

- Primary semester strand:
- Question that motivates this route:
- One relevant skill the bridge established:
- One capability or prerequisite the bridge did not establish:

21.9.3 3. Highest-priority prerequisite action

Prerequisite and current G-level	First preparation action	Observable evidence that the action is complete

21.9.4 4. Sequencing and support

Choose a provisional sequence and give one sentence of evidence:

- enter the common semester core now;
- enter with concurrent preparation and a check-in date; or
- complete a prerequisite first.

Name one task that still requires supervision and one person, group, or course structure that can provide feedback.

21.9.5 5. Reconsideration rule

State one observable result that would make you change routes, add preparation, or seek instructor advice.

21.10. Readiness-memo rubric

Score the memo with four criteria from `rubrics.md`, interpreted here:

- **C3 — Evidence calibration:** G-levels and strengths cite actual artifacts, probe results, or prior coursework.
- **C6 — Decision logic:** the selected route follows from motivation, readiness evidence, and sequencing constraints.
- **C7 — Uncertainty and reconsideration:** gaps, supervision needs, and a real reconsideration rule are explicit.
- **C8 — Traceability and responsible participation:** the plan is concise, feasible, attributable, accessible, and within responsible-use norms.

Maximum raw score: 16. Convert to a percentage and apply the syllabus's 10% weight.

21.11. Instructor review guidance

The instructor should return one of three advisory notes: `ready`, `ready with concurrent preparation`, or `prepare first`, plus one sentence of evidence. Do not use the memo to gate a learner from all AI-safety study. Route them toward an appropriate depth and support level.

Append the post-diagnostic total and capstone C1-C8 profile after grading. If those results change the learner's provisional sequence, invite a memo update; do not penalize the learner for information unavailable during the in-class draft.

Flag for conversation when:

- self-rated G0 conflicts with assessment evidence below 2;
- a selected technical route has two or more G3 prerequisites;
- the plan proposes unsupervised vulnerability or dual-use work;
- the learner treats course completion as research authorization; or
- access, time, or financial constraints make the proposed plan infeasible.

A good memo can earn full credit while recommending "prepare first." Accurate self-assessment is the target.

Attribution

The course explanations, fictional case, exercises, and assessments are independently authored. External papers and institutional documents are linked in the repository's primary reading list and are not republished in this pack.