Thomson: Continual Learning of Frontier Models for SovereignAI

Shengzhuang Chen[*], **Jerrod Parker**[*], **Yejin Bang**[*], **Andrew M. Bean**[*], **Nabeel Seedat**[*],
Stefan Winzeck[†], **Daniil Glazko**[†], **Jannik Zgraggen**[†], **Fangyi Yu**[†], **Scott Arnott**[†], **Dietrich Trautmann**[†], **Luca Ciuffreda**[†], **Guglielmo Bonifazi**[†], **Davide Romano**[†], **Bradley Bell**[†], **Kirsty Fielding**[†],
Daniele Giofrè[‡], **Tom Zielund**[‡], **Ipshita Chatterjee**[‡], **Sneha Murthy Ghantasala**[‡], **Manpreet Nanreh**[‡], **John Scoville**[‡], **Maciej Sakowicz**[‡], **Wassim Seifeddine**[‡], **Lukas Thede**[‡],
Jonathan Richard Schwarz[¶]

[*]Primary Authors, [†]Core Contributors, [‡]Contributors, [¶]Project Lead

Correspondence: First.Last@thomsonreuters.com, jschwarz@ic.ac.uk

Abstract

The development of frontier models is commonly perceived to be in the exclusive remit of a small number of heavily funded players, creating an information, economic and power asymmetry between developers and the diverse user base of modern AI. Recent public discourse acknowledges this concern, calling for SovereignAI (an organisation's capability to independently build, deploy and govern AI use), but often providing little concrete advice on how this can be achieved in the short term under a diversity of funding settings.

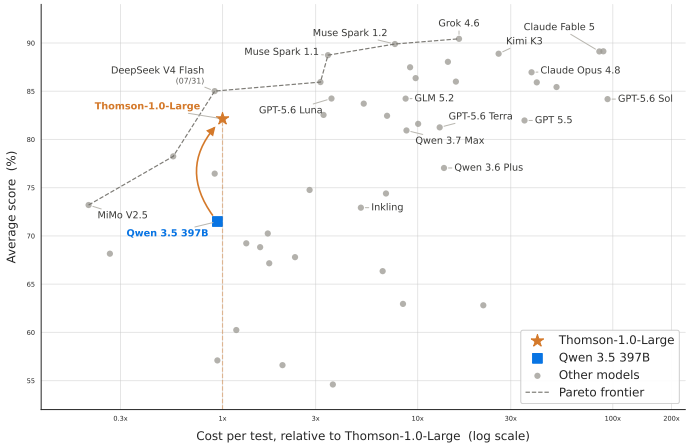
In this report, we argue that frontier performance can be achieved by a wide range of institutions through Continual Learning on readily available open-weight models. As opposed to existing limited approaches such as small-scale fine-tuning, prompt engineering, or tool-augmentation with a frozen model, our Continual Learning approach takes advantage of the effectiveness of a modern mid- & post-training stack while introducing safeguards preserving both plasticity and stability at each training stage and seeking to make the minimal number of high-impact interventions on the parameters. This strategy results in model improvements comparable to the gains typically seen across multiple successive model generations. Crucially, such results are achievable with compute and personnel budgets substantially lower than commonly thought, making ownership of large parts of the SovereignAI stack (model, tool infrastructure, values & data privacy) viable for a wider range of actors.

To demonstrate this, we introduce `Thomson`, a new general-purpose frontier model trained with an enhanced focus on high-stakes professional work: domains commonly predicted to undergo large productivity improvements through AI. Through a unique focus on Continual Learning, data-centricity, and efficiency, we demonstrate that `Thomson` performs competitively with recent frontier models on a wide range of domains and capabilities, ranging from agentic tasks to safety, legal, tax & multilingualism, to comprehensive large-scale Deep Research. Thorough evaluations show a distinctive π-shaped pattern: distinct improvements across a wide range of capabilities (including those not explicitly targeted), while almost completely eliminating the forgetting problem common to narrow domain adaptation.

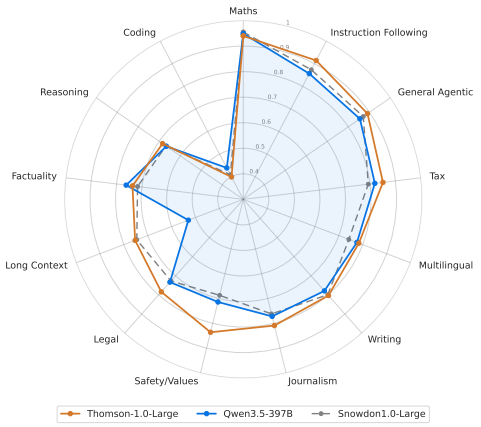
In partnership with: Imperial College London DatologyAI Lambda

Open-weight model: Thomson-1.0-Small (35b)

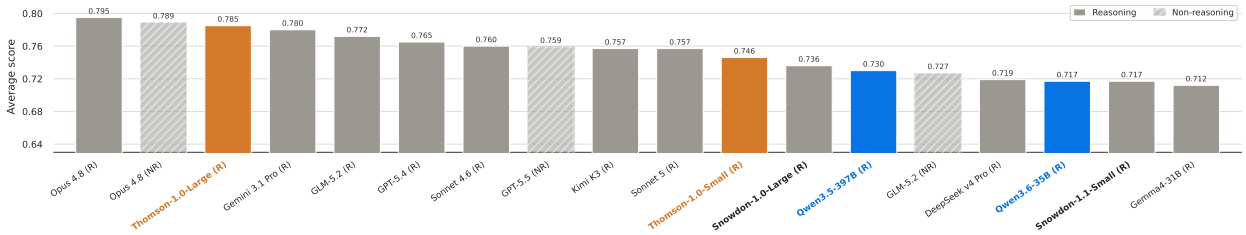
‡ See Acknowledgements for additional contributors.



(a) Cost/Performance trade-off on independent agentic suite.



(b) Per-category score improvements



(c) Aggregate score over our broader capability evaluation suite.

Figure 1 Combined `Thomson-1.0-Large` scores show frontier-level performance over a wide range of benchmarks. Improved scores are seen across most benchmark categories, with large gains in some areas. Please note that 1a shows the cost/performance trade-off on two target domain agentic tasks, while 1c is an aggregate score over a broader set of categories.

1 Introduction

1.1 SovereignAI through Continual Learning

We introduce `Thomson`, a new family of frontier Foundation Models of high proficiency across a wide range of specialised and general-purpose domains as well as practical deployment settings. `Thomson` models are developed within a Continual Learning paradigm with the explicit goal of AI sovereignty, demonstrating that frontier performance is attainable by a much wider range of actors and institutions than commonly thought. Our model development moves beyond the superficiality of narrow fine-tuning or customisation exercises and avoids the significant economic inefficiency of training from scratch. The `Thomson` family was developed by a technical team not exceeding three dozen engineers and scientists on a modest compute cluster with no more than 368 B200 GPUs available at any stage of experimentation. Instead of working on either end of a spectrum ranging from pre-training from scratch to limited customisation exercises, we repurpose Open-weight models (specifically the `Qwen3.5-397B` and `Qwen3.6-35B` models) and substantially improve them on a wide range of performance domains, giving rise to a distinctive π-shaped performance improvement pattern. The total model development timeline for `Thomson-1.0-Small` and `Thomson-1.0-Large` (measured from the date of the first experiments with `Qwen3.5-397B`) was three months, covering large-scale data curation, Mid- & Post-Training, several innovations on reward design (see Section 3), a mature evaluation protocol, human testing and adversarial safety studies. Once this pipeline is established, we believe that it can run in substantially shorter time frames. The cost of the final training run for `Thomson-1.0-Large` (measured in GPU costs over three weeks of training) is conservatively estimated to be under USD 450,000. The total cost of development (including staff, compute costs, domain expert compensation, and vendor partnerships) is estimated at approximately USD 40M, with most of it dedicated to reusable research, infrastructure engineering & experimentation over a meaningfully longer time period than the aforementioned three months.

We make full-weight updates (rather than relying on parameter-efficient methods) and do not engage in large-scale distillation, which both assumes the existence of a stronger teacher model and is typically unable to surpass it. In addition, we apply rigorous scrutiny to ensure that all of our training data and AI use are permissively licensed. Where we utilise open-weight models during development (e.g., as LLM judges), we show that they are eventually surpassed in performance and may be replaced by our own models in future iterations. Taken together, we believe that our findings generalise beyond the specific choices of open-weight checkpoint, model size, and target domain, giving rise to a new paradigm for frontier model development.

`Thomson` models are trained through three complementary modules, each yielding a fully developed Foundation Model. The implications of this finding are substantially more far-reaching than the models themselves: We argue that our development process can be seen as a blueprint for a wide range of institutions to develop private yet competitive Foundation Models at a fraction of the cost previously imagined. Specifically, we demonstrate that Continual Learning allows improvement of an open, *instruction-tuned* reasoning model broadly comparable to frontier performance in November 2025[1] to the point of surpassing recent flagship releases ranging from `Sonnet 5` & `GLM-5.2` (June 2026), `GPT-5.5` & `DeepSeek-V4 Pro` (April 2026) to `Gemini 3.1 Pro` (February, 2026) on a wide range of tasks (see Figure 1a). The results *bridge up to seven months of model improvements by the world's most generously funded AI companies* on a training compute budget typically reserved for smaller-scale ablation studies. Figure 1b shows that, rather than making only shallow changes to a base model, our modifications improve on existing open-weight models in a consistent and meaningful way across a wide range of domains.

More important than the concrete results shown in this report are the principles underlying our demonstrated improvements, which we believe are more general and can be readily applied to other open-weight models, allowing a capable team to further reduce iteration time. Thus, we believe the reader is best advised to think of this report as a set of building blocks for a model factory applicable more broadly to a wide range of open-weight models, with the `Thomson` models being merely an initial proving ground.

For the purpose of this report, we consider the following essential principles of SovereignAI:

§1. Training & Model Sovereignty: See Section 3 & broader report. Largely achievable through work with open-weight models (for discussion, see Section 1.4).

§2. Data & Tool Sovereignty: See Sections 2, 3. Fully achievable for private data. No control over data used to train open-weight model.

§3. Governance & Ethical Alignment: See Section 3.2. Substantial progress through alignment with custom constitutions at various training stages. Subject to current limitations of AI alignment more broadly.

§4. Infrastructure Sovereignty: See Section 5. Largely achievable by building on increasingly mature open-source stacks for training, model serving & tool infrastructure. Remaining dependence on critical hardware infrastructure.

§5. Control Over Economic Factors: Substantially increased by operating models at cost while maintaining control over release/update cycles. Remaining uncertainty based on cost/access to serving hardware.

The pillars of our approach to model development are:

I. Continual Learning: Taking care to preserve capabilities not directly affected in each phase of model improvement (stability) while maintaining the ability to master new skills and absorb new knowledge (plasticity). This is achieved through the careful design of our model improvement pipeline, including critical choices and modifications of learning algorithms both effective and compute-efficient within an accessible budget. In addition, we curate a representative and low-cost evaluation suite for knowledge retention, providing a reliable measure of skill retention without overtly optimising on benchmarks.

II. Data-centric Machine Learning: The highest standards of data quality control and design targeted at learning efficiency and skill improvement in vital domains. To name a few examples, we construct contrasting data sources to re-align a given model with an institution's values; re-phrase, filter, de-duplicate, and enhance mid-training data to enable greater post-training efficiency; calibrate data mixtures for post-training through Bayesian Optimisation; guide invaluable human subject-matter data collection towards a semantic space with

[1] Gemini 3 Pro: November 18, 2025; `GPT-5.1`: November 12, 2025; `Claude Opus 4.5`: November 24, 2025

Domain	Benchmark	Opus 4.8	Thomson 1.0-Large	Gemini 3.1 Pro	GLM 5.2	GPT 5.4	Kimi K3	Sonnet 5	Snowdon 1.0-Large	Qwen3.5 397B	DeepSeek v4 Pro
	Overall Avg.	**79.5**	78.5	78.0	77.2	76.5	75.7	75.7	73.6	73.0	71.9
Legal	Stanford LegalBench	81.8	82.3	**84.3**	82.9	82.3	83.3	81.4	82.8	78.8	76.8
	Info. Retrieval	51.2	53.6	**53.8**	53.6	**53.8**	53.0	47.9	51.2	49.0	51.5
	Reasoning	75.2	73.2	**77.8**	73.1	68.4	74.8	73.1	70.8	66.5	70.1
	Classification	70.5	70.9	**74.2**	70.2	70.0	71.6	70.4	69.8	68.7	67.8
	Doc. Processing & RAG	78.9	**79.7**	78.2	76.5	78.6	79.0	74.7	71.2	74.1	73.7
	Summarisation	88.3	**90.0**	89.4	**90.0**	86.3	87.7	84.8	87.9	82.5	89.1
	Contract Under.	74.4	71.0	75.9	73.6	74.9	**77.2**	69.6	71.7	68.4	68.3
	Human Queries	85.4	**89.2**	86.4	87.3	88.6	61.4	84.3	87.1	86.7	86.4
	Deep Research	**90.8**	88.9	80.6	89.0	85.9	89.8	86.0	78.4	87.3	84.0
	Harvey LAB	**86.9**	85.7	55.5	84.6	76.1	83.7	80.9	56.3	70.6	83.1
	Domain Avg.	78.3	**78.4**	75.6	78.1	76.5	76.1	75.3	72.7	73.3	75.1
Tax	Deep Research	**85.8**	82.4	76.0	83.5	80.9	85.1	82.0	70.4	78.7	80.7
	Tax Q&A	86.9	87.9	84.5	85.7	86.6	88.3	**88.7**	88.2	85.1	82.8
	Domain Avg.	86.3	85.1	80.2	84.6	83.8	**86.7**	85.4	79.3	81.9	81.8
Journalism	Deep Research	82.3	80.9	83.0	79.0	66.3	**84.5**	74.1	76.2	77.2	78.6
General	Factuality	71.4	73.7	**82.6**	66.2	68.3	69.5	59.8	71.7	76.2	69.1
	Long Context	75.2	75.3	75.0	**75.9**	70.0	73.5	70.7	74.6	53.0	69.5
	Multilingualism	83.2	78.4	**85.7**	81.7	82.7	84.9	79.5	74.2	77.6	78.4
	Instr. Following	86.1	**91.4**	84.8	89.4	89.0	85.8	86.8	87.3	85.6	89.2
	Writing	79.3	80.3	78.5	78.1	79.1	78.2	78.7	79.9	77.9	**80.7**
	Reasoning	73.7	68.4	74.8	67.3	71.6	**76.2**	66.8	66.8	66.6	65.0
	General Agent	83.4	**89.1**	84.4	77.5	75.6	61.0	74.1	87.1	85.5	72.0
	Coding	57.4	39.9	50.0	56.0	50.8	**66.8**	57.4	40.9	43.9	45.6
	Maths	**98.7**	94.0	97.4	91.2	97.7	96.2	85.9	95.5	94.9	94.9
	Domain Avg.	78.7	76.7	**79.2**	75.9	76.1	76.9	73.3	75.3	73.5	73.8
Safety / Values	Political Neutrality	82.8	**97.3**	93.3	91.0	83.8	33.5	82.3	85.3	51.5	31.3
	Robustness	**78.2**	60.3	65.3	50.4	68.3	72.7	77.1	42.2	66.7	38.1
	Adversarial Testing	–	93.4	–	93.6	–	87.3	–	78.8	**95.9**	81.1
	Domain Avg.	–	**83.6**	–	78.3	–	64.5	–	68.7	71.4	50.2

Table 1 Cross-domain benchmark results for reasoning-mode models. Scores are percentages; the best score in each row is shown in **bold**. Dashes (–) mark benchmarks that were not run for a given model. Note: Fair Adversarial testing against proprietary models was not possible due to the likely presence of unknown guardrail mechanisms. Overall Avg. is the unweighted mean over all individual benchmarks, excluding Adversarial Testing. Harvey LAB: Harvey Legal Agent Benchmark.

low density in the existing training data distribution; and carefully configure additional automated evaluation benchmarks to be as closely aligned with human judgements as possible.

III. Agentic training & tool use: Recognising the near-universal use of tool augmentation as a common easy-access strategy to introduce private data at inference, we develop data generation, training, inference and evaluation strategies to enable the tailoring of models to both general-purpose and privately implemented tool systems. This includes the design of a broad set of tools appropriate for the serving context, a full Deep Research harness (Section 2), and training stages ranging from localised error correction to end-to-end reinforcement learning (RL) for Deep Research. In addition, we provide a detailed description of thoroughly designed reward structures to incentivise faithful use and accurate citation patterns, vital to reducing hallucinations in high-stakes settings.

After discussing the specific modelling goals and broader impact of this work, the remainder of the report carefully lays out the different phases of model development, data synthesis, and evaluation, while also addressing broader practical infrastructure and serving requirements.

Domain	Benchmark	Thomson 1.0-Small	Snowdon 1.1-Small	Gemma4 31B	Qwen3.6 35B	Haiku 4.5
	Overall Avg.	**74.6**	71.7	71.2	71.7	68.2
Legal	Stanford LegalBench	79.9	80.9	**83.1**	80.3	80.7
	Info. Retrieval	49.6	48.6	**51.9**	49.4	49.2
	Reasoning	68.2	67.3	**71.9**	64.7	65.5
	Classification	70.0	70.1	69.4	**70.4**	68.1
	Doc. Processing & RAG	**78.8**	71.2	76.6	74.7	43.8
	Summarisation	**89.4**	88.1	**89.4**	89.0	84.3
	Contract Under.	67.3	64.8	**73.2**	63.7	70.1
	Human Queries	**90.2**	82.2	81.2	82.6	74.9
	Deep Research	**85.0**	80.0	74.0	82.0	80.0
	Harvey Legal Agent Bench.	**73.4**	71.5	34.2	69.5	60.5
	Domain Avg.	**75.2**	72.4	70.5	72.7	67.7
Tax	Deep Research	**78.6**	68.0	75.0	68.0	62.0
	Tax Q&A	**86.6**	85.2	84.5	86.2	79.4
	Domain Avg.	**82.6**	76.5	79.6	77.3	70.7
Journalism	Deep Research	74.2	67.5	74.7	73.0	**81.0**
General	Factuality	**61.1**	59.3	57.6	58.8	56.8
	Long Context	**74.1**	73.8	69.4	73.8	67.4
	Multilingualism	71.9	72.8	79.4	73.1	**85.8**
	Instruction Following	86.1	85.5	**89.3**	85.6	78.2
	Writing	**81.0**	79.3	75.5	79.5	77.9
	Reasoning	61.5	61.3	**66.1**	61.5	49.3
	General Agent	**85.8**	81.4	72.9	80.3	59.7
	Coding	37.4	35.6	34.6	**39.8**	32.9
	Maths	86.7	88.0	**91.1**	87.5	66.5
	Domain Avg.	**71.7**	70.8	70.7	71.1	63.8
Safety / Values	Political Neutrality	**98.5**	91.5	91.5	78.5	92.0
	Robustness	56.3	47.3	41.7	48.7	**70.2**
	Adversarial Testing	89.1	87.7	88.5	**89.2**	–
	Domain Avg.	**81.3**	75.5	73.9	72.1	81.1

Table 2 Cross-domain benchmark results for small models. The best score in each row is shown in **bold**. Dashes (–) mark benchmarks that were not run for a given model. All models were run with medium reasoning effort. Overall Avg. is the unweighted mean over all individual benchmarks, excluding Adversarial Testing.

1.2 Modelling Goals

`Thomson` was developed with a deliberate focus on economically impactful, high-stakes professional work across legal, tax, and journalism domains. This scope reflects a realistic and compelling use case for an institution with strong incentives to pursue SovereignAI. Accordingly, we address modelling decisions suited to domains that combine the formality, logic, and rigorous reasoning characteristic of technical fields with the nuance, interpretation, and tolerance for uncertainty of the humanities. Several of these principles – particularly the emphasis on citation quality and conditional relevance (whereby precedents of considerable

age may remain binding, while others are superseded by subsequent rulings or evidence) – bear a natural affinity to scientific practice. We thus believe they are of broader methodological interest. Where relevant to readers with a specific interest in the legal domain, results of particular significance to that area are highlighted throughout.

A notable insight that emerged during development was the behaviour of our Continual Learning pipeline: rather than merely preserving performance on tasks outside the primary focus, the pipeline demonstrably improved performance on them. This result was initially unexpected and represents a highly positive finding for the claims made in this report. It provides strong grounds for believing that the modelling principles described here are domain-agnostic and transferable across a wide range of settings and institutional contexts.

1.3 Headline Results

The following sections show some of the headline results for `Thomson`, both when run under identical conditions against frontier models as well as in a broader system comparison. Extended results, including detailed descriptions, ablation studies and analysis, can be found in Section 4.

1.3.1 Model Results

Figure 1 shows `Thomson-1.0-Large` in comparison to a wide range of competitors as well as its base model (`Qwen3.5-397B-A17B`) in three settings. Figure 1a shows an average cost vs performance trade-off on two agentic tasks in the legal target domain, normalised to the cost of running `Qwen3.5-397B` architectures.[2] We show near-Pareto-optimal performance, with substantial improvements over `Qwen3.5-397B`, rivalling a wide range of competitive frontier models. The cost difference between subsequent versions of models from the same provider also strengthens our argument for economic sovereignty (§5). Figure 1b is a clear demonstration of π-shaped Continual Learning, showing material performance gains on several dimensions, while holding the baseline performance on most other domains. Figure 1c is an aggregate overall performance score computed over target, general domains and safety scores (see Table 1). All results paint a clear picture of sustained frontier performance and substantial improvements through Continual Learning.

For completeness, we also share results for `Snowdon-1.0-Large` & `Snowdon-1.1-Small`, a value-realigned version of `Qwen3.5-397B/Qwen3.6-35B` (see Figures 1b, 2), developed through the Frontier AI Research Lab, a joint research lab established by Thomson Reuters and Imperial College London. The `Snowdon-1.0-Large` development process is described in Section 3.2 and in full detail in [1].

Table 1 shows detailed results in a wide range of settings, from high-volume automated tasks (such as document processing) to answer quality on real-world human interactions, high-budget Deep Research tasks, a wide range of general domain evaluations, as well as safety and value evaluations judged through adversarial testing. Table 2 reports the same breakdown for the small models, alongside Figure 2, which shows their per-category gains over `Qwen3.6-35B`. Every model is evaluated through the same harness, on identical prompts, comparable inference parameters (such as reasoning effort) and grading pipelines designed to faithfully measure performance rather than insignificant artefacts. No model is granted retrieval or tool access unless the benchmark itself defines it, in which case all models receive the same tools. However, closed models are accessible only as served endpoints, and we cannot rule out provider-side routing, system prompt augmentation or other scaffolding. We consider these unavoidable rather than disqualifying. In short, we take care to ensure the protocol is as fair and unbiased as possible.

`Thomson-1.0-Large` places close to the strongest proprietary model tested (which is likely of significantly larger size) in the comparison on the overall average, ahead of heavyweights such as `Gemini 3.1 Pro`, `Sonnet 5`, `GLM-5.2`, and importantly, dominating performance against its base model by a significant margin. `Thomson`'s strength is particularly pronounced across several of the domains we explicitly target, such as natural human interactions, document-heavy workflows or Deep Research. On safety and values, the effect of implementing value sovereignty (§3) is clearly visible, alleviating a regular concern of open-weight model use; we note that fair adversarial comparison against proprietary systems is not possible, since undisclosed

[2] Averaged over (1) Harvey Legal Agent Benchmark and (2) Legal Research Bench. Third-party model costs and results are taken from publicly available vals.ai leaderboards. `Thomson-1.0-Large` and `Qwen3.5-397B` costs are estimated using publicly available pricing data from Alibaba (both accessed in August 2026). All costs take token usage into account.

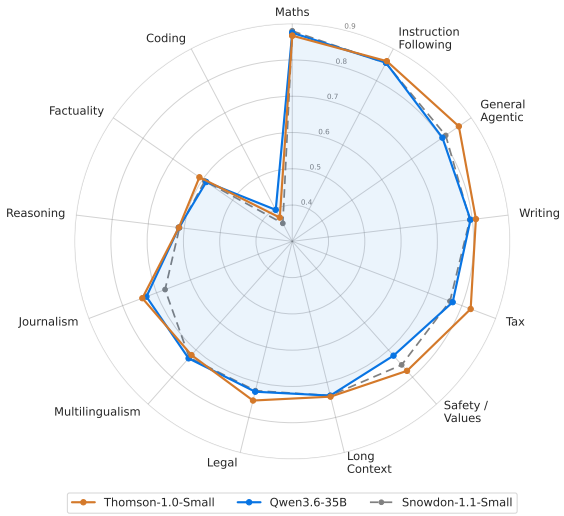


Figure 2 Per-Category score improvements of `Thomson-1.0-Small`.

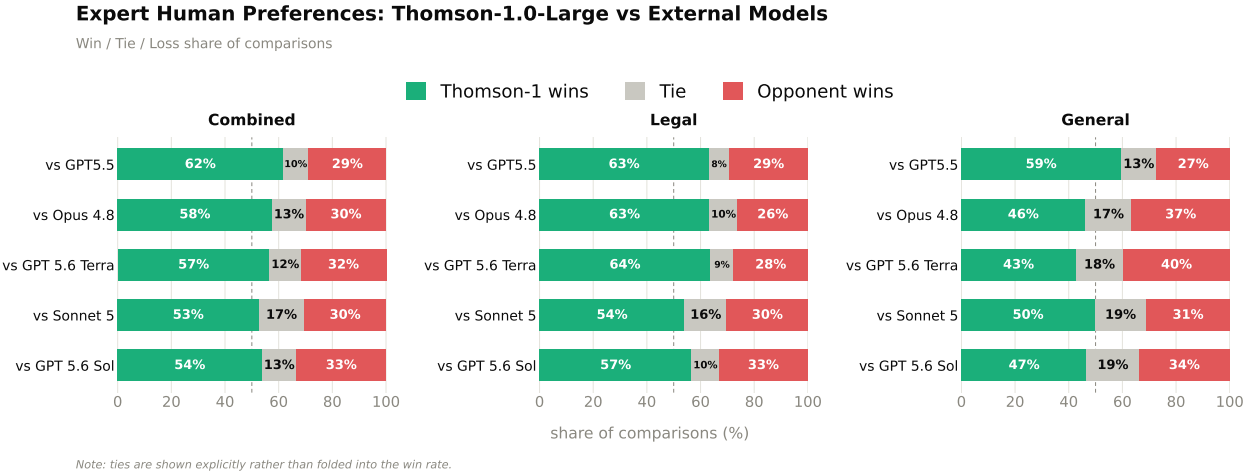


Figure 3 Blind human evaluation study of the `Thomson` system against systems provided by major frontier labs. `Thomson-1.0-Large` is given access to recent news (Reuters feed) as well as access to legal databases. External OpenAI & Anthropic models are given broad web access. Scores computed on an aggregate of over 3,000 preference-rated conversations.

guardrail layers sit between the served endpoint and the model. We thus omit such entries rather than reporting misleading numbers.

In terms of limitations, we find coding – the only domain showing mild forgetting relative to the base model – to fall clearly below frontier performance. While this is deliberately not a target domain (and arguably already well-targeted by the vast majority of AI developers), it is plausible that coding may affect the downstream performance of other domains, especially given the increased prevalence of agentic tasks requiring general computer-use skills. General mathematical and abstract reasoning trail the strongest proprietary models but are within the performance level of `Qwen`, showing successful protection against forgetting.

1.3.2 System Results

The preceding section deliberately omits various common comparative advantages institutions would deploy in practice. In keeping with our arguments for sovereignty (especially §2), this section answers a complementary question of high practical relevance: *what are the best results competing institutions can currently produce?*

To approximate the question, we provide a downstream system demonstration rather than aiming to isolate the model's contribution alone. To do so, we test models from leading AI providers OpenAI and Anthropic, given web access, against `Thomson-1.0-Large` when given access to some of the world's most authoritative licensed databases on legal and news information, testing both systems on general and domain-specific queries in a blind comparison. A priori, it may be expected that broad web access should provide a major advantage in general tasks while still providing substantial benefits in the target domain. In addition, both providers are likely to have specialised their models for web use, given the ubiquity of this use case.

Figure 3 reports a blind preference study over more than 3,000 expert-rated conversations, split into legal and general sets, with subject-matter experts unaware of which system produced which response. Instructions were deliberately limited (with the exception of providing realistic limits to what the systems could produce), testing systems under real-world conversational use. Aggregated across both sets, the `Thomson` system is preferred over each of the five external systems. The structure of the result is more informative than the headline: on legal conversations, where sovereign data access is most consequential, preferences are consistently and clearly favourable, in expectation of the privileged position. However, on open-domain general conversations, the picture is mixed and considerably tighter, a promising result given that (a) both OpenAI and Anthropic will likely possess historical user interaction data on general queries at unparalleled scale, and (b) web access is arguably broader than only news access. Further details (including ablation study results removing system components) are provided in Section 4.

Taken together with Section 1.3.1, the two sets of results describe a model factory: a reproducible pipeline for turning an open-weight checkpoint and an institution's proprietary data into a deployable, competitive, and governable system.

1.4 Discussion & Limitations

A critical reader may pause and consider the reliance on an initial open-weight starting point as incompatible with the principles of SovereignAI. While ownership of the full training stack arguably goes further, the implied economic and computational demands preclude most interested actors from making meaningful progress towards independence. In this report, we are interested in moving beyond a bipolar view of sovereignty (i.e., conditioning any notion of sovereignty on full ownership) by instead advancing along what we consider a sovereignty spectrum. To that end, we present substantial progress on the model – the core of any such strategy – as well as beyond it, through tool design, evaluation guidelines, serving infrastructure, and harness development.

In choosing this approach, we explicitly aim to capitalise on the outstanding opportunity of a competitive open-weight landscape that has flourished despite the substantial early performance margin enjoyed by closed-model developers. Indeed, it is an increasingly common view that the gap between open-weight models and the best closed models has collapsed from multiple months [2] to a much shorter time frame, suggesting that the additional training recommended in this report can not only close any remaining gap, but potentially surpass some of the world's most competitive models. Further arguments in favour of this approach are the general consensus that additional large-scale pre-training is no longer giving rise to large model improvements (drastically decreasing dependence on the most costly stage of training), as well as a noticeably larger collective body of knowledge on how to develop frontier models & systems (with LLM reasoning being the latest such breakthrough quickly made accessible by `DeepSeek-R1` [3]).

It is also important to note that once a first generation of models has been built using the principles laid out in this report, we observe that any subsequent model generation can be increasingly based on the resulting checkpoints from previous training rounds (similarly to how some model builders aim to rely only on internally developed models). It is our view that, after a few such model iterations, it becomes increasingly tenuous and of little relevance that a model builder relies on an open-weight model at the start of the iteration, no matter how distant the original dependence might be.

While valid arguments persist regarding architectural control (which is more difficult to alter in later stages of model training), a rich set of models spanning a wide range of sizes and architectures is readily available. Finally, recent work [e.g., 4] has begun to show that some pre-training properties may be resilient to alteration in later stages. So far, this has been exploratory research motivated by highly desirable properties (tamper

resistance of safety features), and there is little incentive for an open-weight developer to remove benign properties in a similar fashion. In addition, by demonstrating that Mid-Training can still be performed on *instruction-tuned* models (a highly desirable yet unconventional approach; Section 3.3), we present a practical solution for injecting additional knowledge present in private data. Thus, we believe that for the majority of potential beneficiaries of AI, our arguments in favour of practical sovereignty through Continual Learning carry more weight.

Contents

2 Preliminaries

2.1 Constitutional AI

Recognising the Sovereignty goal §3 (Governance and Ethical Alignment), we draw inspiration from initial attempts to align AI with model constitutions [5] and show throughout the report how this can be achieved through complementary training processes designed for actors with both modest and more generous computational budgets. Constitutional AI sits at the core of how we approach Sovereignty in model development. A sovereign model must be governed by principles that are transparent, publicly accountable, and open to scrutiny, rather than by proprietary value systems that are subject to change. For this reason, we deliberately depart from existing commercial constitutions and instead ground our alignment process in the Public AI constitutional project [6], an open project that emphasises public contribution and debate. As this constitution allows free use and modification, we believe §3 can be readily implemented by making appropriate modifications and following our development methodology. For the development of `Thomson`, this choice reflects our conviction that the normative foundations of a model should themselves be a shared public resource, developed in the open and subject to community input. Throughout development, this constitution serves as the reference framework against which model behaviour is shaped.

We stress that alignment to this constitution remains an aspiration rather than a solved problem. While we make every effort to faithfully reflect its principles in the resulting model, we regard this as ongoing work, and we make no claim of complete or guaranteed adherence. Our hope is that the approach described here, along with the choices and trade-offs it entails, will prove useful to the wider community and help inspire future work on grounding sovereign models in openly developed constitutional principles.

2.2 Mitigating & Measuring Forgetting

Continual Learning has its roots in the study of lifelong learning in neural networks, with the phenomenon of *catastrophic forgetting* first characterised in the late 1980s [7, 8] and subsequently formalised, along with a number of proposals for algorithmic remedies developed over the following three decades [9, 10]. In the era of large Foundation Models, however, forgetting rarely manifests as the abrupt, catastrophic collapse observed in earlier, smaller-scale settings; instead, it tends to be subtle and gradual, surfacing as a slow erosion of specific capabilities that is considerably harder to detect and to measure reliably. Throughout the development of this approach, we found that the careful, disciplined iteration and execution of well-established ideas – Data Rehearsal, Architectural Regularisation, and Functional Regularisation – applied selectively but consistently throughout every stage of the pipeline constitutes a substantially more impactful and practical strategy than isolated modifications to any single training algorithm.

A further distinguishing feature of our approach is that we operate throughout on *instruction-tuned*, rather than simply pre-trained, open-weight models: this is vital in order to retain and build upon the substantial capabilities already instilled during prior post-training, but requires careful experimentation and validation at critical stages of the pipeline (most notably Mid-Training; see Section 3.3) where the assumptions underlying standard training recipes for base model adaptation no longer straightforwardly hold.

A vital tool throughout model development is the monitoring of capability preservation throughout all training stages. In order to effectively achieve this, we rely on a capability-centric meta-benchmark for measuring behavioural drift in large language models (CapTrack; for full details see [11]). Rather than proposing another standalone benchmark, CapTrack intentionally organises established community benchmarks into a structured capability taxonomy spanning three complementary dimensions: CAN (latent competence), WILL (default behavioural preferences), and HOW (protocol compliance and execution). This organisation enables a multifaceted view of model quality beyond aggregate accuracy, capturing changes in a semantically meaningful manner. Throughout this work, CapTrack serves as the primary framework for tracking capability evolution and identifying potential model drift across the training pipeline.

Each capability is assessed through multiple complementary evaluations and capability-specific metrics, providing robust capability estimates while reducing reliance on any individual benchmark. Rather than focusing on absolute benchmark scores, CapTrack quantifies relative changes in capability between model

	Development Evals (CapTrack) (only $\approx 10-15\%$ of examples used)	**Test Evals** (All examples used)
Factuality	RAGTruth, TruthfulQA, PopQA	SimpleQA, Faitheval
Multilingualism	XTREME, MGSM	XTREME*, MGSM, MMMLU
Instruction Follow.	IFEval, FollowBench	IFEval, FollowBench*
Writing	MT-Bench *(1st turn)*, OASST1, ELI5	WritingBench
Coding	HumanEval, MBPP, BFCL, MNMS	SWE-Bench Pro, TerminalBench 2.1
Maths	GSM-8K, MATH, LiveMathBench	AIME 2025-2026, GSM-8K, MATH
Reasoning	MMLU-Pro, SuperGPQA	GPQA-Diamond, Humanity's Last Exam, MMLU-Pro
General Agent	MT-Bench *(2nd turn)*, StructFlowBench	GDPVal, Tau2
Long Context	HotpotQA, QASPER, RULER-32K, LongBench-V2	HotpotQA, MuSiQue, NovelQA, NQ, QAMPARI, Quest, ∞Bench
Safety & Values	HarmBench (unsafe), GSM-8k (benign), RULER-4k *(incomplete)*, WinoGrande, HellaSwag	Adversarial Testing, Political Neutrality, Robustness
Misc	BoolQ, Schema-wrapped MMLU-Pro, Rephrased MMLU/GSM8K	

Table 3 General-purpose evaluations used for `Thomson`. Aggregate scores for Development Evals are monitored during the model training process to measure and control forgetting, while Test Evals are reported as a genuine performance evaluation. None of the evaluations has dedicated training sets (*uses CapTrack examples).

checkpoints and their corresponding base models, enabling consistent tracking of behavioural drift throughout training.

In the design of Captrack, we intentionally combine our aim for efficiency as well as pillars (i) and (ii) of our model development by applying data-pruning techniques to constituent benchmarks. Concretely, we categorise each evaluation example across 16 cognitive dimensions [12], allowing us to apply semantic de-duplication / data pruning to systematically subsample and drastically reduce the size of the remaining data while preserving as clear a signal as possible, an approach we term Scales++ (see [13] for full details). As a result, this provides broad capability coverage while remaining feasible to run repeatedly across multiple checkpoints and training stages on modest compute budgets.

To maintain scientific integrity, we contrast CapTrack with our general capability evaluation suite in Table 3, adding additional benchmarks for most categories, motivated by the development/unseen evaluation suite split recommended in [14]. We also note the additional benefit of using Scales++, which reduces sample overlap even when benchmarks are re-used, as the CapTrack benchmark versions only contain about 10-15% of the samples in the original data. An exception is Instruction Following, which we believe is indirectly measured through most other evaluations (automated scoring highly penalises failures to comply with format instructions).

2.3 Agentic Deep Research

Tool use recurs throughout this report as an essential model capability targeted through data design and training, with its infrastructure being a critical component best owned independently and usable with any model (§2), thereby also enabling fair evaluation. Its function is to augment parametric knowledge (memorised by a model during training) with access to external resources at test time, enabling the utilisation of private data or knowledge recency. This section describes the preliminaries, aiding the reader with the technical details in Section 3.

The most powerful instantiation of tool calling is Deep Research, where a model uses a suite of tools to independently retrieve and synthesise evidence over a long horizon, typically for high-impact tasks that prioritise output quality over model response latency. It is thus one of the clearest tests of sustained agentic capability [15, 16]: Successful research requires planning over long horizons, efficacious utilisation of search

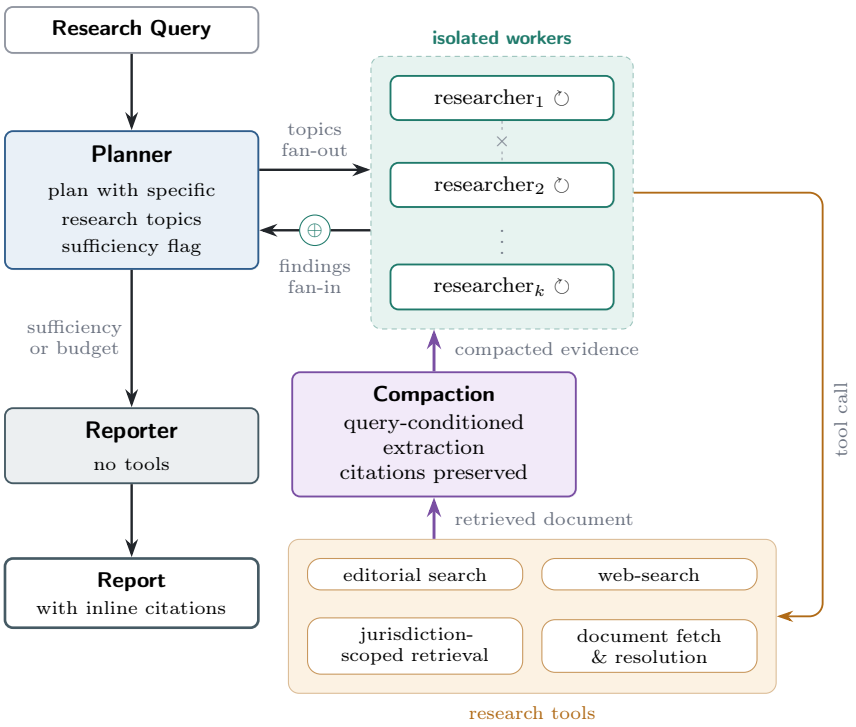


Figure 4 Deep Research agent harness. A planner decomposes the research query into self-contained research topics and dispatches them to isolated workers, each a ReAct loop with a bounded tool-call budget; workers share no context, so coverage is a property of the plan rather than of lateral coordination. Tool results return through query-conditioned compaction, which discards prose but preserves citations, keeping the evidence a worker accumulates small while the addressable corpus stays large. The planner re-plans against remaining gaps or hands off to a reporter that has no tools, which is then tasked with producing a Deep Research report with in-line citations.

tools, faithful interpretation of long source documents, meaningful synthesis, as well as appropriate and highly accurate citations. While capabilities like tool-calling [17, 18], long context interpretation [19, 20], and reasoning [21] are tested separately within our evaluation suite (Section 4), we look to agentic research as a test of integrating these capabilities in meaningful settings.

In many domains, Deep Research is also one of the most economically consequential applications of AI [22], spanning scientific research, finance, compliance, journalism and dozens of other domains combining source identification, interpretation, and synthesis. In biomedical research, determining whether a proposed mechanism, drug interaction, or trial design is consistent with the existing literature requires reviewing primary studies, dosage and safety data, and regulatory guidance that is frequently revised as new trials report out. While our research system is not intended to have the sophistication of a full research product, close alignment to real-world tasks also bolsters its value as a training and evaluation environment (Section 4), demonstrating several of the SovereignAI principles introduced in Section 1.1 put into practice most concretely.

This makes Deep Research a demanding evaluation target, requiring a harness/scaffold that can run a complete research episode end-to-end, identically for every model under test. Furthermore, complex research reports are not deterministically verifiable, and valuable information for complex economic domains is not easily accessible on the open web. While open-web research discussed in other reports [e.g. 23] shares a similar agentic loop, we tackle the following unique challenges motivated by the context of the report (Section 1.2):

(C1) The evidence is not on the open web. Authoritative evidence often sits behind private or licensed databases and paywalled corpora rather than the indexed open web (e.g., the Reuters News archive for reliable global news spanning decades, restricted-access outlets and journals for scientific publishing and financial reporting, and licensed professional research databases for primary source material). Consequently, a model cannot (and should not) fall back on memorised web text or general search coverage, so performance has to come from inference-time search and retrieval against these gated sources. On the positive side, this also

makes contamination a much smaller concern than for open-web browsing benchmarks such as GAIA [24] or BrowseComp [25].

(C2) Documents are orders of magnitude longer. A typical web page is a few thousand tokens, while this report deals with documents that routinely exceed 50k (e.g., a regulatory filing, a clinical trial protocol, or a technical standard). The bottleneck moves from *finding* a document to *absorbing* it, i.e., retrieval is cheap to invoke and expensive to consume, and an agent that fetches indiscriminately will exhaust its context before it can synthesise and, as a result, will affect measured quality. We note that a larger window does not dissolve the problem either, since long context degradation appears well before a window is nominally full [20].

(C3) Correctness is query-dependent, not intrinsic to the document. A document can be internally valid and still be the wrong answer to a given query, because its validity depends on facts external to the document itself – facts that change over time and are not visible from the text alone (e.g., a guideline or trial result on medicine later superseded by new evidence). A model can therefore correctly use information in the document, but incorrectly for the query at a given timepoint, resulting in a fluid, well-cited report that is nonetheless incorrectly conditioned on the query. There are also hard constraints on applicability – analogous to the jurisdiction in law – that shape the search process itself.

Figure 4 shows the design of a Deep Research agent harness used throughout this report. Specifically, we adapt DeerFlow [26], an open-source LangGraph [27] research agent implementing the orchestrator-worker pattern common to multi-agent research systems [28]. We retain the harness topology, with significant adaptations to the evaluation and training to address unique challenges in non-verifiable domains. The harness has the following architecture:

Planning. In response to a research query, the planner emits a typed plan typically of 3-5 steps. Each step is a self-contained retrieval topic specifying what evidence to collect. Planning runs as direct structured generation rather than inside a tool-calling loop.

Research. Each researcher is a ReAct agent [29] with a bounded tool-call budget, working to complete a research topic using the tool suite. It sees the broader query and its own topic, not the other topics and findings of other researchers. This isolation is the main structural control on context growth.

Fan-in and Re-planning. Findings of individual researchers are merged back via a fan-in to the planner. The planner re-examines the accumulated findings and either issues another research round targeting gaps or declares sufficiency; rounds are capped.

Reporter. The reporter receives the finalised plan and all findings, and has no tools. It is tasked with writing a research report with an explicit instruction to include inline citations to the specific document for every proposition/claim.

Tool suite. Addressing (C1) and the use of authoritative corpora, the harness dynamically wraps a set of custom research tools, fully controlled by the entity, thus implementing SovereignAI goal §2.

Context management. Addressing (C2), the challenge of long documents, we adapt the harness to use *query-conditioned evidence extraction*. Any tool result above a length threshold is compacted by a model conditioned on the research topic and the current findings.

3 Model Development

3.1 Overview

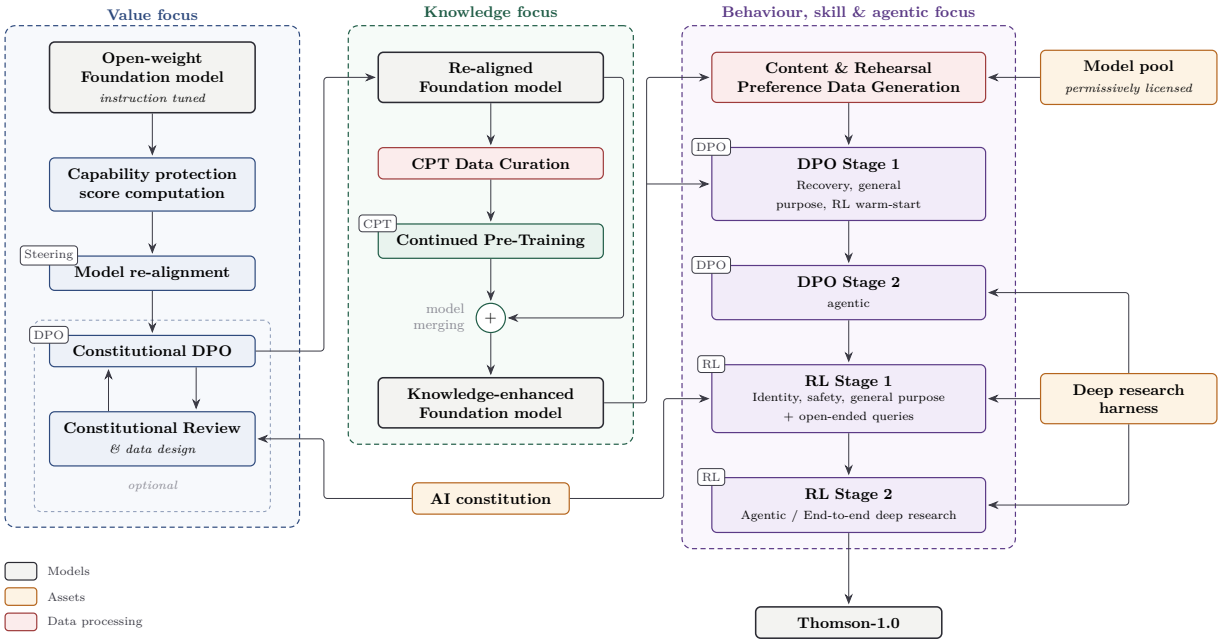


Figure 5 Model development pipeline for `Thomson 1`. The pipeline is organised into three sequential modules with distinct development foci. **Value focus**: Starting from an open-weight instruction-tuned Foundation Model, we perform value re-alignment through constitutional DPO, potentially augmented with activation steering. **Knowledge focus**: Data-centric continual pre-training (CPT) ingests proprietary domain data, with model merging protecting generic capabilities while absorbing domain knowledge. **Behaviour, skill & tool focus**: Two-stage DPO (broad preference alignment, then agentic specialisation) warm-starts two-stage RL (short-context with diverse queries, then long-context end-to-end Deep Research). Agentic data from the Deep Research harness feeds DPO Stage 2 and both RL stages. This modular design allows practitioners to adjust computational investment to match sovereignty requirements.

Figure 5 shows our overall model development diagram, separated into three modules with distinct development foci (values, knowledge, behaviour/skill/tool). Upon completion of a series of stages designed to achieve the respective development goal, we aim to arrive at a fully capable Foundation Model which may either be used directly or serve as input to the next module to further improve its capabilities. This allows model builders to further adjust computational budgets to sovereignty goals. As a practical choice, we found it valuable to first realign existing *instruction-tuned* Foundation Models with our own constitution, hypothesising that current alignment occurs primarily during post-training and may thus be altered with relatively cheap methods (such as activation steering).

Privately held unstructured data is best utilised through knowledge injection in the secondary stage, allowing further learning with a pre-training objective. To enable continued pre-training on an *instruction-tuned model*, we develop a data-centric pipeline focused on the automated enhancement and restructuring of raw data. This has the dual benefit of reducing distribution mismatch with the typical shape of post-training data and serving as a soft warm start for the final module. In addition, a vital model-merging step prevents otherwise detrimental degradation of model capabilities. This paradigm is arguably closer to what is often referred to as Mid-Training.

Our final module combines typical post-training stages to unlock newly acquired knowledge for practical tasks. Direct preference optimisation (DPO) [30] stages serve the dual purpose of warm-starting a model for online reinforcement learning [31] while allowing fine-grained edits to any remaining degradation not fully recovered by model merging. Among all model modifications, we found DPO and RL to be the most robust to model

degradation, provided sufficient care is taken to design high-quality preference data and training environments. Note that we explicitly forgo any further supervised fine-tuning (SFT) as an isolated post-training stage due to its tendency to aggressively alter model behaviour, leading to rapid forgetting even across broad data distributions [11].

3.2 Value Re-alignment

(Co-authored by Imperial College London)

We re-align the values of open-weight models following the two-stage procedure of [1], which brings the model's expressed values into line with the Public AI Constitution [6] rather than with those of its original training distribution. The first stage is Fisher-protected directional ablation (Section 3.2.2): a training-free weight edit that identifies the direction in activation space separating value-laden responses from neutral ones and projects it out of the model's weights. The second stage is Constitutional DPO (Section 3.2.3) over paired responses drawn from both expert-curated and synthetic data, which consolidates the edit. The binding constraint throughout is that the model's inherent capability must survive: a realignment that trades away general competence is of no practical use to us, so we treat capability retention as an objective rather than as a diagnostic.

3.2.1 The Alignment–Capability Frontier

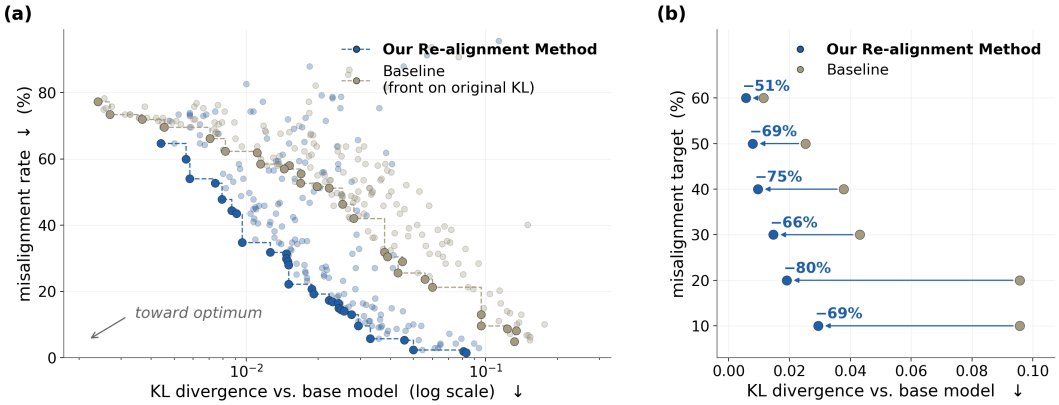


Figure 6 Misalignment-rate vs. capability preservation. Our realignment traces a strictly better frontier than the plain-abliteration baseline, reaching any given misalignment level at substantially lower KL divergence from the base model. **(a)** Misalignment rate on the validation prompt set vs. KL divergence from the base model; faded markers are individual trials, solid markers joined by dashed lines are the Pareto fronts. Note: Baseline KL scores are replayed under our KL score definition for comparability, with their front selected on the original KL scores. **(b)** KL divergence cost to reach each misalignment rate; our realignment method incurs 2.0×–5.0× less KL divergence scores than the baseline, suggesting better model capability preservation during ablations.

Because alignment and capability can pull against each other, we do not reduce the search to a single scalar objective; we characterise each method by its Pareto front over two measurements. The first is the *misalignment rate*: the fraction of a held-out validation set of value-probing prompts on which the edited model fails to respond in line with the desired value. The second is the capability cost, measured as the *multi-token Kullback-Leibler (KL) divergence* between the edited model and its corresponding base model over a curated prompt set (KL set). That set is deliberately broad, including benign everyday prompts, capability-specific and safety probing prompts, so that a low KL score indicates less behavioural drift, rather than merely preserving behaviour on the alignment distribution itself.

Figure 6 reports this front. Our realignment dominates the baseline abliteration over the entire range: for every misalignment target we consider, it reaches that target at 51–80% lower KL divergence cost (2.0×–5.0× cheaper), as quantified in panel (b). The advantage is widest at aggressive targets, where the baseline can only continue reducing misalignment by accepting capability damage. The baseline method illustrated here shares our direction-extraction data and validation set, but during its trial search it tracks a single-token

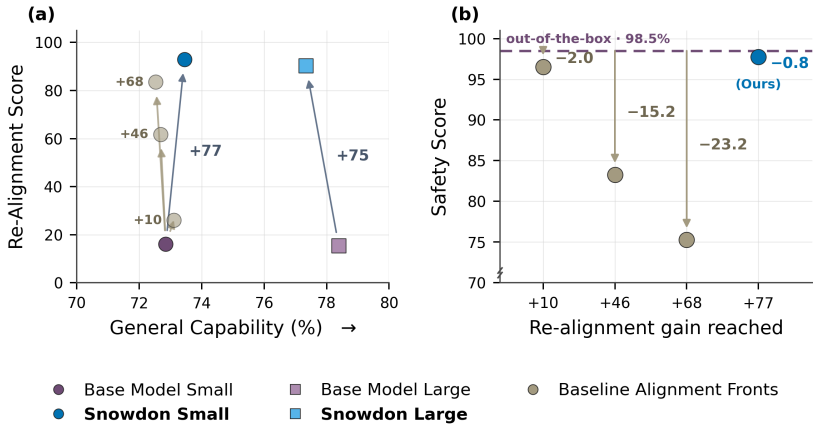


Figure 7 Realignment on the small and large models. Each panel contrasts the out-of-the-box base models (*Base Model Small/Large*) with our re-aligned checkpoints (*Snowdon Small/Large*); the *Baseline Alignment Fronts* are plain ablated versions of the small model. **(a)** Realignment score (Perspective Bench) against general capability (CapTrack): both sizes gain large points of realignment with essentially no capability loss, while baselines reach only partial re-alignment. **(b)** Safety, as the unsafe-request refusal rate's drop from the out-of-the-box line, plotted against the realignment gain each method reaches. Baselines improve re-alignment only by giving up safety score, and still plateau below the re-alignment level `Snowdon` reaches; `Snowdon` attains the largest gain at a cost of 0.8% refusal. Note: models are tested in non-reasoning mode for this experiment.

KL divergence over a limited prompt set. To ensure that our advantage does not merely reflect a change of metric, we recompute the KL score for every baseline trial on our KL set under the identical multi-token definition. Its front is still selected using the KL scores it actually optimised against, and only then re-plotted at the recomputed coordinates; we do not re-optimise the baseline after recomputation. Both searches are given the same budget of 201 trials.

Optimising this frontier yields a family of ablated checkpoints rather than a single model. From it, we select a small set of operating points and consolidate each with a preference optimisation step, searching jointly over the choice of front and the DPO hyperparameters. Crucially, the final `Snowdon` checkpoints are *not* chosen on the search-time proxies of Figure 6 but on the downstream quantities: a realignment score measured with Perspective Bench and a general-capability score aggregated over 18 benchmarks in CapTrack as well as an unsafe prompt refusal score. The evaluation that follows (Section 3.2.4) asks whether the frontier's proxy advantage survives this change of measurement, on a held-out test set the search never saw.

3.2.2 Fisher-Routed Directional Ablation

We suppress the target misaligned behaviour with a rank-one weight edit. The method separates two choices. A difference of means identifies which residual-stream direction to steer. An activation-space Fisher estimate determines where to place the resulting correction.

Estimating the Direction. We compare the model's state on two prompt populations. The sensitive set S elicits the target behaviour, while the neutral set N does not. We generate both sets independently from the same topic taxonomy.

Let μ_ℓ^S and μ_ℓ^N denote their mean residual-stream activations at layer ℓ. We record each activation at the final prompt token, immediately before the model begins its response. Their normalised difference defines the candidate direction at that layer [32, 33]:

$$v_\ell = \frac{\mu_\ell^S - \mu_\ell^N}{\|\mu_\ell^S - \mu_\ell^N\|_2}. \tag{1}$$

Averaging suppresses prompt-specific variation and retains the systematic difference between the populations. Figure 8 illustrates the resulting layer-wise separation between S and N using two-dimensional UMAP

projections [34]. Each search trial chooses a continuous source-layer index and interpolates the adjacent directions. We normalise the result and use the resulting direction v at every edited layer.

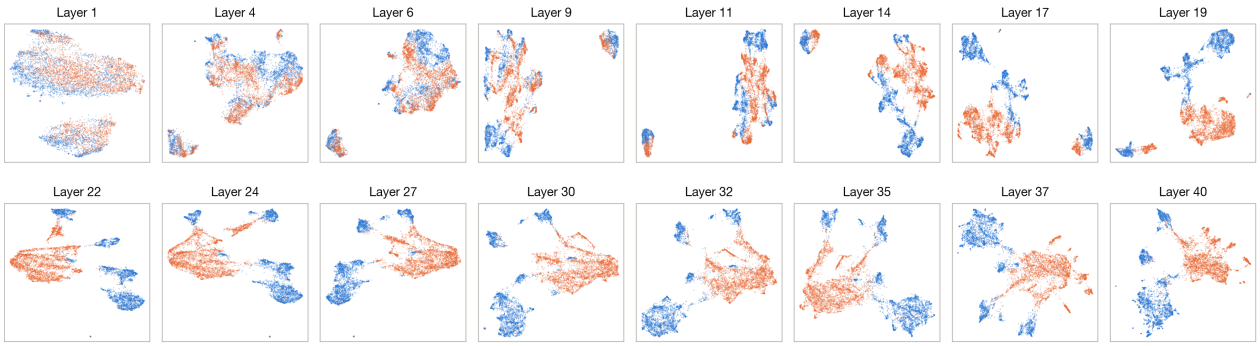


Figure 8 The misaligned behaviour has a separable representation. UMAP projections of final-prompt-token residual stream activations, computed independently at each layer, for the sensitive set S (elicits the misaligned behaviour) vs. the neutral set N (does not). Points from the two prompt sets are interleaved in early layers and become separated by mid-stack. Each panel is an independent fit, so panels show separability at a depth, not trajectories across depth.

Degrees of Freedom in the Writeback. Let W be the weight matrix of a component whose output enters the residual stream, and let b be its writeback vector. We write the edit as

$$\Delta W = b(v^\top W), \qquad v^\top b = -\lambda \quad \Longrightarrow \quad v^\top (W + \Delta W) = (1-\lambda) v^\top W. \tag{2}$$

The constraint fixes the removal strength λ while leaving $d-1$ orthogonal degrees of freedom in b. Standard directional ablation sets these to zero, giving $b = -\lambda v$. We instead choose $b = -\lambda v + b_\perp$, where $v^\top b_\perp = 0$, to minimise the edit's predicted effect on other model behaviour. During search, we implement each rank-one edit as a LoRA adapter [35].

Measuring Output Sensitivity. For each edited component, we estimate the diagonal Fisher information of the model's predictive distribution in the component's output space [36]. Omitting layer, component, and token indices,

$$F_i = \mathbb{E}_{x\sim\mathcal{D}} \mathbb{E}_{\hat{a}\sim p_\theta(\cdot|x)} \left[\left(\frac{\partial \log p_\theta(\hat{a} \mid x)}{\partial z_i} \right)^2 \right], \tag{3}$$

where z_i is output coordinate i and $\mathcal{D}$ is the reference distribution. Large F_i indicates that changing coordinate i has a larger local effect on the model's predictive distribution; small F_i indicates lower estimated sensitivity.

We estimate F on a reference set spanning safety, tool use, instruction following, multilingual behaviour, and general capability, disjoint from the data used to score search trials.

Fisher-Routed Writeback. Let $\overline{F}$ denote the mean of the coordinates of F. We define

$$h_i = F_i + \rho \overline{F}, \qquad g_i = \frac{v_i}{h_i}, \qquad s = v^\top g, \qquad b = -\frac{\lambda}{s} g. \tag{4}$$

We fix the ridge fraction at $\rho = 0.05$.

Each step has one role. The first adds a floor to the Fisher values, preventing coordinates with $F_i \approx 0$ from receiving an excessive correction. The second reduces the contribution of coordinates with high predictive sensitivity. The third measures the remaining projection of the routed vector onto v. The final step sets the scale required to remove the chosen fraction λ.

Because $s = v^\top g$, the normalisation preserves $v^\top b = -\lambda$ exactly. Fisher routing therefore changes where the correction is written, not how much of the target projection is removed.

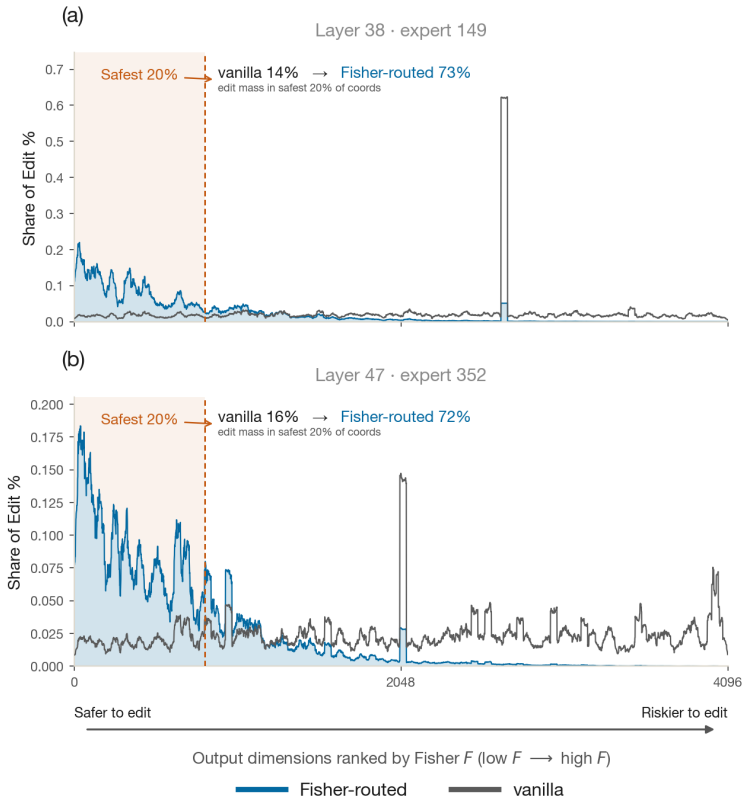


Figure 9 Fisher routing shifts the writeback towards output coordinates with lower estimated sensitivity. Coordinates are ordered from low to high Fisher value for the selected experts at layers 38 and 47. Grey shows the standard writeback $b = -\lambda v$; blue shows the Fisher-routed writeback (ours) from Equation (4). The vertical line marks the lowest- Fisher fifth of the output coordinates. Labels report the share of squared writeback mass within that fifth.

This writeback minimises the local diagonal-Fisher cost among all vectors that satisfy $v^\top b = -\lambda$ [37]. If F is uniform, then g is proportional to v, and the solution reduces to $b = -\lambda v$. Standard directional ablation is therefore the flat-Fisher case.

Figure 9 shows the routing effect directly. Both edits satisfy the same removal constraint. The Fisher-routed edit places more of its writeback on coordinates with lower estimated predictive sensitivity.

Search Procedure Our realignment procedure is an automated trial-search loop. Each trial applies a rank-one update to each targeted component, including combination of direction, location and strength, and the resulting model is scored on in-loop proxies for misalignment rate and capability drift. Trials are proposed by Tree-structured Parzen Estimator (TPE) in Optuna [38, 39], which concentrates the search on promising regions of the edit space.

We adapt this framework by replacing the standard writeback $b = -\lambda v$ with the Fisher-routed writeback from Section 3.2.2, and by measuring model drift with multi-token KL over a broader anchor set. Each trial is evaluated on held-out misalignment rate and KL divergence; the non-dominated trials form the alignment–capability frontier in Figure 6. The plain-abliteration baseline uses the standard writeback and its original single-token KL objective. Both searches use the same direction data, validation set, and trial budget. Selected points on our frontier are passed to constitutional DPO.

3.2.3 Constitutional DPO

We consolidate selected operating points from the alignment–capability frontier (Figure 6) with a single epoch of length-normalised DPO, searching jointly over the choice of front and the DPO hyperparameters.

We call this stage Constitutional DPO: the preference pairs combine human expert-curated examples on a heterogeneous topic mix with synthetic examples on the sensitive topics we target, and in each pair the chosen response is the one that reflects the Public AI Constitution.

Human Expert Pairs. Subject-matter experts (SMEs) prompted an early-stage exploration model across a broad topic taxonomy. General domains included coding, culture, history, languages, law, mathematics, medicine, philosophy, reasoning, science, and technology. Sensitive domains included political, historical, and social topics where credible narratives diverge or states maintain institutional positions. Prompts were designed to elicit value-relevant behaviour or test a constitutional constraint. Guided by the Public AI Constitution, SMEs made targeted edits where minor correction was sufficient and rewrote responses requiring substantive changes to their conclusions or framing. Each revision was checked against the Constitution and used as the chosen completion, with the original model response serving as the rejected completion.

Synthetic Pairs. To scale beyond the human-curated set, each base model answer is rewritten by an ensemble of three open-source models conditioned on the constitution. An LLM judge, using criteria drawn from the constitution, discards revisions that are out of date or that refuse to answer, deflect or return a vague non-answer. Surviving revisions become chosen responses, paired against the OOB rejected response.

Hyperparameter Search. We grid-search KL strength $\beta \in \{5, 10\}$, learning rate $\in \{1 \times 10^{-7}, 2 \times 10^{-7}\}$, NLL weight $\lambda \in \{0.9, 1.0\}$, objective (DPO vs. length-normalised DPO), and data mixture (equal 1:1:1:1 vs. re-weighted across expert and synthetic sources), selecting the configuration maximising realignment score subject to retaining baseline general-capability performance. The resulting `Snowdon` checkpoints (Figure 7(b), blue points) push realignment further while preserving the capability gains from the Fisher-protected edit.

3.2.4 Realignment Evaluation

The frontier (Figure 6) was traced with search-time proxies, misalignment rate and KL divergence, measured on data that drove the search. We then evaluate the final checkpoints on two independent, downstream measurements and ask whether that advantage survives the change of metric.

Realignment. We measure realignment with Perspective Bench [1]: prompts on geopolitically contested topics, each response graded by a panel of five LLM judges across six rubric dimensions and assigned an overall ideological lean by majority vote. We summarise the lean distribution as a single *realignment score* (0–100, higher is more aligned).

Model Capability. We track general capability with CapTrack [11] (Section 2.2), reporting its latent-competence (CAN) group: what a model can do under ideal prompting across knowledge, reasoning, comprehension, faithfulness, and robustness. Our score is the mean accuracy over 18 benchmarks; we avoid CapTrack's headline average, which mixes in behavioural metrics of differing polarity. This is mainly to keep track of capability preservation of our realignment process. Additionally, we measure *Safety Score* as CapTrack's W1 unsafe-request refusal rate (higher values indicate that the model refuses a larger proportion of unsafe requests).

Results. Realignment succeeds at both model scales while largely preserving capability (Figure 7). Both base models start relatively low on the realignment axis and, after the full pipeline, gain roughly 75 points while general capability holds within a point of its starting value (panel a). Panel (b) shows the same holds for safety: every baseline achieves re-alignment by giving up unsafe-request refusal and still plateaus below `Snowdon`'s re-alignment level, whereas `Snowdon` reaches the largest gain on realignment only with a 0.8 percentage-point drop in safety refusals. Both results are measured on benchmarks the search never scored against.

3.3 Continual Pre-training

(Co-authored by DatologyAI)

3.3.1 Data-Centric Subselection and Enhancement

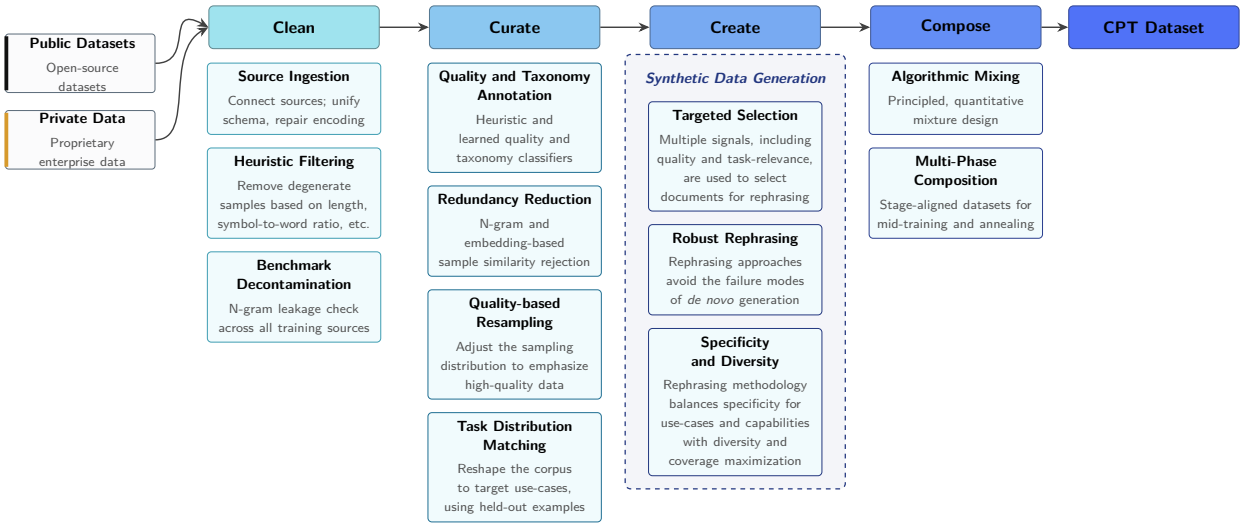


Figure 10 Data curation pipeline for continued pre-training

Data curation is at the heart of our Continual Learning strategy for the knowledge focus module (Figure 5) of the `Thomson` model development pipeline. To do so, we partnered with DatologyAI to curate a high-quality mid-training dataset of 200B tokens from a corpus of permissively public and high-quality proprietary data of over 19T tokens. The key tension in "Continual Pre-training" (CPT) for *instruction-tuned* models is the need to elicit deep domain expertise without eroding the model's existing capabilities. This is the standard "learn without forgetting" tension that all Continual Learning must navigate, but with the important twist that we perform CPT on top of an *instruction-tuned* (and later value-realigned) checkpoint. We thus make an important terminological distinction and consider this a form of "mid-training", hinting that data is carefully curated to sit at an in-between point between raw, unstructured pre-training and formats typically seen in Post-Training. Standard Continual Pre-training on such post-trained models disproportionately degrades the vital capabilities previously developed during post-training [40, 41] and thus presents a critical challenge for Continual Learning. This raises the stakes of forgetting mitigation and, with it, the value of curated replay data.

Our approach to mitigating this is a combination of classic rehearsal-based Continual Learning [e.g. 42–44] along with a modern approach to data-centric techniques and synthetic data [e.g. 43, 45–47]. Our mid-training data mix thus comprises three roughly equal parts: curated proprietary documents, synthetic data generated from those documents, and curated general-capability replay data.

The knowledge of the raw proprietary corpus (containing decades of news, contracts, long and numerically dense regulatory filings, US and international case law, statutes, regulations, practitioner guidance, etc.) is distilled through a series of processing stages (Figure 10). The much denser corpus yielded is representative of a corpus of data often held privately by institutions and thus missing in the pre-training data of third-party models. This allows expanding the embedded knowledge through model training while preserving data sovereignty (§2). The deployed pipeline runs as distributed jobs on Kubernetes and processes each source.

- Ingestion and Normalisation: Per-source parsing of heterogeneous formats into a standardised markdown document schema.
- Exact Deduplication: applied per-source where duplication analysis warranted it.
- Length Filtering and Chunking: with documents capped at 256k tokens.
- Decontamination: against all evaluation targets via 13-gram overlap (à la [48]), applied before any selection stage so that no later stage can include evaluation data in training.

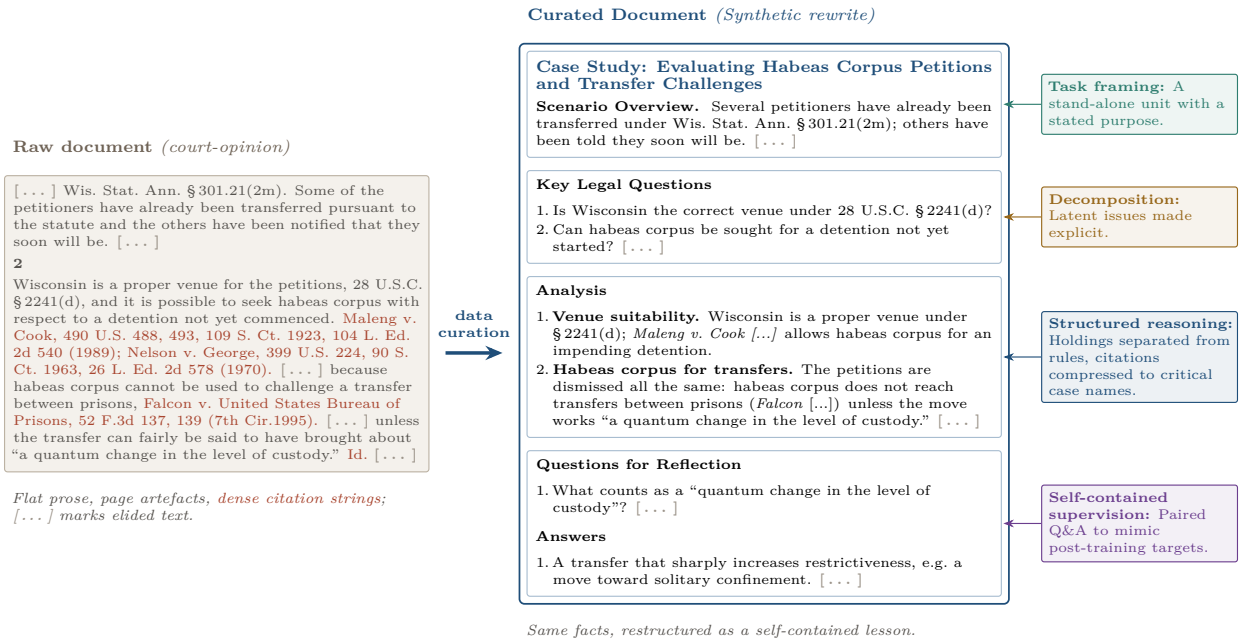


Figure 11 Example curation of a dense document

- Quality Filtering: using heuristic and model-based quality and repetition filters. These filters serve multiple goals. First, removing the bad tail of degenerate documents, such as excessively short, repetitive, poorly-formatted, and insufficiently linguistic documents. And second, identifying fluent, information-dense, and/or relevant documents.
- Task and Use-Case Distribution Matching: We upsample documents from the corpus to better align with the distribution of tasks the model will serve in the target domain. We emphasise that evaluation items have already been removed (Stage 4), and that all curation research was conducted without the use of proprietary data or evaluations (as described below).
- Synthetic Generation: (see below).
- Mixing, Shuffling, and Export: with mixture proportions determined systematically and enforced by token count.

Example 11 shows a rephrased example data point, highlighting the key distinction between large-scale pre-training on raw, uncurated content and targeted mid-training with high-quality data designed to teach new knowledge while reducing degradation.

Evaluation. To maintain evaluation integrity, our full curation recipe – including mixture proportions, filter thresholds, and the synthetic data approach – was developed through mid-training ablations on small open models (specifically `Qwen3-8B` [49] and `Llama-3.1-8B` [50]). We used public corpora as stand-ins for the full proprietary corpus, and open benchmarks re-implemented in a common evaluation harness [51]. General-purpose evaluations encompassed world knowledge, reading comprehension, language understanding, general reasoning, mathematics, coding, and instruction following (Section 2.2). Target domain evaluations were composed of the LSAT subsets of AGIEval [52], law-related subsets of MMLU [53] as well as a subset of tasks from LegalBench [54] which we adapted to maximise the signal-to-noise ratio using data similar to [55].

In cases where the default evaluation protocol scores a generated answer by exact match (conflating capability with format compliance), we implemented subsets in multiple scoring formats: exact match, perplexity-based option selection, and chain-of-thought with answer extraction. We found that perplexity-based selection yielded the lowest noise, particularly on models with diminished instruction-following capability. To reduce the cost of evaluation and maximise signal, we evaluated twelve intermediate checkpoints of OLMo-2-7B [56]

spanning 1B to 3.9T training tokens on a subset of tasks, computed the rank correlation between tokens seen and score, and retained the tasks whose scores improved reliably with training. Fewer than a third exceeded a rank correlation of 0.7.

Synthetic & Replay Data. Our replay data is DatologyAI's general-capability mid-training mixture. This constitutes curated web text, mathematics, code, multilingual data, and a small amount of instruction-style data. Replay is standard practice in domain-adaptive pre-training [57], but the effects of data curation on replay data utility are poorly understood. In ablation studies, we observe that replacing publicly-available replay data derived from Dolmino [56] with the curated replay data has the potential to substantially improve code (+5.4pp average across HumanEval++, MBPP, and MBPP++) and reading comprehension (PubMedQA +7.2pp, BoolQ +4.5pp), indicating that replay data curation has meaningful, capability-specific downstream effects.

Taking inspiration from Deep Generative replay [e.g. 43], we further develop synthetic data. This consists of rephrasings of algorithmically-selected source material, generated using an adaptation of DatologyAI's BeyondWeb method [47]. BeyondWeb's efficacy comes from two complementary mechanisms. Diversity, which is achieved by rephrasing source documents into a breadth of formats and registers, and specificity, which is achieved through careful selection of documents for rephrasing using various quality and relevance signals.

While naively applying our standard BeyondWeb-style rephrasing pipeline to domain-specific documents demonstrated clear benefits in internal ablations, we made numerous adaptations and improvements to our synthetic data curation in order to promote strong capabilities in `Thomson`. After examining benchmarks that made it past our signal and noise analysis, we identified six core capabilities that we thought were required by these benchmarks: foundational knowledge and language; (verbal) reasoning; reading comprehension and summarisation over supplied documents; long-context handling; specialised knowledge; and numerical reasoning over formal rules. We then expanded the synthetic rephrasing format inventory from general-purpose patterns to also target these capabilities. Before our final run, we validated candidate generator models specifically on their fitness for synthetic data applications. We used a mix of heuristic and model-based assessments, as well as extensive human evaluation. We emphasise that this is for generator qualification, not curation of individual documents. Figure 11 shows a re-phrased example.

Furthermore, we changed the way in which rephrased documents are assembled into contexts. Typically, individual documents are sampled IID from the full pre-training corpus. Instead, we found that leveraging the relationship between the source(s) and rephrased document(s) can meaningfully impact downstream model quality. Finally, we extended the input length for seed data input and output to accommodate the need for long-context reasoning over long and/or many documents.

Lessons Learned and Future Work

Our experiments provided the following insights:

- Selection of optimal mid-training learning rate (LR). This depends on the base model and Mid-Training dataset [57–59], hence it is standard practice to run an LR sweep using short training runs before committing to a full mid-training run. We found that the LR minimising training loss continued to deliver the strongest checkpoint after model merging with an appropriate coefficient.
- Recipes developed on base models transfer to post-trained models after model merging. We conducted our mid-training curation research using both base and post-trained models, and found that the effects of data curation interventions were qualitatively similar across model types after model merging.

Our future work aims to explore the following directions:

- Increase data pool. To maintain economically efficient Continual Learning, we restricted our CPT budget to a modest 200B tokens, pruning over 98% of the original data. While this represents the highest quality data and we believe this represents a realistic setting for many potential actors interested in SovereignAI, scaling up the data pool through this pipeline is expected to unlock further improvement of `Thomson`.

- Curation to reduce hallucination. Hallucinations are a known weakness of LLMs in high-stakes domains [e.g. 60], and are contained in synthetic data [61, 62]. Our next cycle of data curation will target reduction of hallucination in synthetic training data and in downstream models.
- Curation for multi-document reasoning. Complex tasks require reasoning across many related documents, and this is where current models are weakest. Grouping related documents during curation to teach models how to reason across documents, rather than curating each in isolation, is a promising direction to improve reasoning in complex settings.

3.3.2 Training & Capability Recovery

Training runs

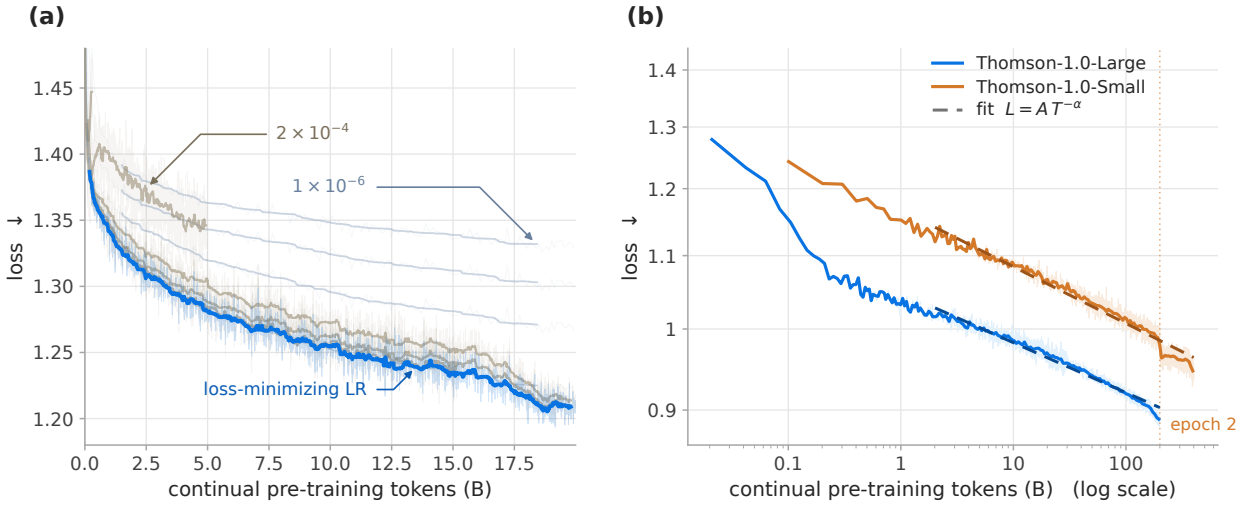


Figure 12 Learning rate selection and loss scaling during continual pre-training. (a) Training cross entropy across candidate learning rates under a fixed pilot budget, with the selected learning rate minimising stable training loss. (b) Loss trajectories for the Small and Large models as a function of consumed continual pretraining tokens, with dashed lines denoting power-law fits.

Before committing to a full CPT run, we select the peak learning rate independently at each model scale using short pilot runs that hold the remaining training configuration fixed. Figure 12 shows that learning rates at the upper extreme produce optimisation instability, whereas those at the lower extreme reduce cross entropy too slowly under the same token budget. The intermediate rate that achieves the lowest stable training loss within the pilot budget is then used for the full continual pre-training run.

Each model is trained with sequences of 8,192 tokens and a global batch size of 512, corresponding to approximately 4.2 million tokens per optimisation step. Following LR warmup, we apply inverse square root decay, under which the LR at step t is proportional to $t^{-1/2}$. Unlike schedules such as cosine decay, in which the LR is parametrised by the total training budget, inverse square root decay is independent of the final token budget. This allows the stable training phase to be extended without redefining the schedule or introducing a discontinuity in the LR. A final annealing stage promotes controlled convergence by linearly decreasing the LR to its minimum value over the last 20% of the continual pre-training budget. In early-stage experiments, we also investigated an extended long context Mid-Training stage. Although this additional training recovered long context capabilities lost during continual pre-training, model merging followed by long context DPO and agentic RL proved largely sufficient, and the dedicated extension was therefore omitted from the final training recipe.

Model Merging

Model merging is a central component of our mid-training procedure when starting from a model that has already undergone post training. Although merging can mitigate forgetting during continued training more

generally [63, 64], it is particularly important in this setting since mid-training disproportionately erodes capabilities acquired during post training.

During early development, we sought a compute efficient procedure that balanced domain adaptation against forgetting under a fixed budget for continual pre- and post-training. Alongside model merging, we evaluated several regularisation techniques using the same curated CPT mixture, including low rank adaptation, lower continual pre-training learning rates, and shorter training schedules. Each of these regularisation techniques reduced the loss of general capability but weakened domain adaptation to a similar extent. After post-training, none of the resulting checkpoints improved on either axis relative to a model post-trained without continual pre-training.

Consistent with prior work on domain adaptation of post trained models [65, 66], these results led us to merge the continually pretrained checkpoint with the checkpoint preceding mid-training, which we found necessary to restore the lost capabilities. Additional post-training on generic data provides another route to recovery, but its computational cost grows with the severity of forgetting and represents overhead relative to the domain adaptation objective, potentially placing it beyond the budget of many institutions.

Model Merging Benefits from Stronger Continual Pre-Training. Model merging, in its simplest form, linearly interpolates in weight space between the continually pre-trained checkpoint and the model initialisation preceding continual pre-training, with the relative contribution of each checkpoint governed by a single merging coefficient [64, 67]. We found that an appropriate merge coefficient can recover most of the general capability lost during continual pre-training. More surprisingly, after post-training under the same budget, a merged checkpoint derived from sufficiently strong continual pre-training outperformed its counterpart trained without continual pre-training on both the domain and general capability axes.

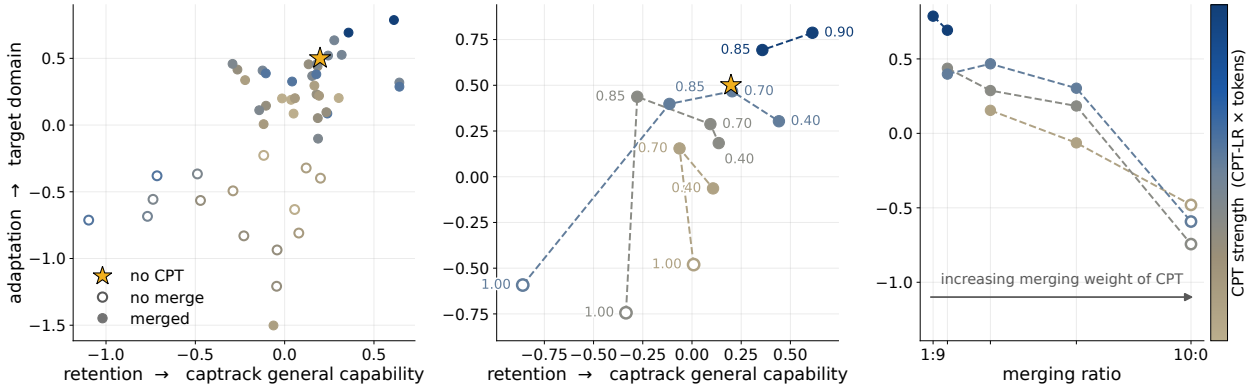


Figure 13 Effect of continual pre-training strength and merge coefficient on target domain performance and general capability retention under a fixed post-training budget. Colour denotes CPT strength, while each numerical annotation gives the merge coefficient assigned to the CPT checkpoint. Axes report mean standardised performance over the target and general domains.

To characterise how continual pre-training, merging and post-training jointly determine the trade-off between domain adaptation and forgetting, we sweep the continual pre-training learning rate over $\{2\times10^{-6}, 5\times10^{-6}, 1\times10^{-5}, 2\times10^{-5}\}$ and the token budget over $\{20, 40, 100, 200\}$B tokens, and interpolate each resulting checkpoint towards the model preceding CPT at merge ratios from 1:9 to 10:0, holding the post-training budget fixed. We define *CPT strength* as the product of the learning rate and the token budget, a single index of how far training displaces the weights from their initialisation. Figure 13 shows that aggressive continual pre-training followed by a conservative merge ratio, by which we mean one that retains less of the continually pre-trained weights, consistently outperforms the regularised alternatives on both axes at once, and that the Pareto front is occupied by high-strength merged checkpoints rather than by restrained ones with a lower CPT LR or a smaller training budget.

3.4 Direct Preference Optimisation

Post-training begins from the merged CPT checkpoint, converting newly injected knowledge into desired, task-specific behaviour. On-policy RL is the most direct instrument for this conversion, but it is expensive, and several of the behaviours required for subsequent exploration – e.g., model identity, output-format conventions, and related inductive biases – are specified more economically by off-policy preference data than they are acquired through fully on-policy search. We therefore precede reinforcement learning with DPO [30], which warm-starts the policy at substantially lower cost.

A common alternative for warm-starting RL is SFT; however, we found that naive SFT in our setting can rapidly degrade generic capability, even when the supervised mixture is broad and includes generic replay data, while DPO tends to be much more robust to catastrophic forgetting [11]. We forgo a standalone SFT stage for this reason, and instead turn towards a DPO training objective with an SFT intervention to still account for the instruction-following and output-formatting supervision when needed for a particular task collection. The supervision that such a stage would otherwise provide is already present in the CPT mixture, both as a small instruction-style replay component and as synthetic rephrasings of proprietary documents into task-shaped and instruction-following formats (Section 3.3), motivated by prior findings that absorbing such supervision under a pre-training objective, rather than as an isolated post-training SFT pass, reduces distribution mismatch with later stages and is a more effective way of using SFT-like data on an already-aligned model [68, 69].

We apply DPO in two sequential stages (see Figure 5), separating broad alignment from long-context, agentic specialisation:

- DPO Stage 1 (broad preference alignment): Performs preference alignment over a large heterogeneous collection mixture at moderate context length. We employ multi-fidelity Bayesian optimisation to learn data mixture proportions that improve both performance and robustness (Section 3.4.3).
- DPO Stage 2 (agentic specialisation): Initialises from the Stage 1 checkpoint and specialises the policy for agentic behaviour on a targeted set of synthetic Deep Research preference collections. This stage operates at extended context length (up to 64k tokens) to support long-horizon research and tool-use trajectories.

The remainder of this section covers, in order, the shared training algorithm in Section 3.4.1, DPO Stage 1 preference data and data mixture optimisation in Sections 3.4.2 and 3.4.3, DPO Stage 2 agentic preference data and long-context specialisation in Sections 3.4.4 and 3.4.5, and, finally, how the resulting DPO checkpoint provides a stronger initialisation for subsequent reinforcement learning in Section 3.4.6.

3.4.1 Algorithmic Overview

The DPO training objective used throughout both stages is

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y_{c,w},y_{c,l},c)}\left[-\log\sigma\big(\beta\left[\bar{h}_\theta(x,y_{c,w}) - \bar{h}_\theta(x,y_{c,l})\right]\big) + \alpha_c\,\mathcal{L}_{\text{SFT}}(x,y_{c,w})\right]. \tag{5}$$

Here, c indexes a training collection, or dataset, and $y_{c,w}$ and $y_{c,l}$ are its chosen and rejected responses for prompt x. Equation (5) differs from standard DPO in that we employ the length-normalised implicit reward $\bar{h}_\theta(x,y)$ defined in Equation (6), and augment the preference loss with an SFT term $\mathcal{L}_{\text{SFT}}$ weighted by the collection specific coefficient α_c.

Collection-Dependent SFT Anchoring. When tasks are often open-ended, what counts as a target is task- and preference-dependent, and there is often no universal, single gold response for a query. We introduce a per-collection coefficient α_c that controls how much absolute imitation is warranted on the chosen response. Setting α_c per collection lets us encode domain preferences and requirements using expert knowledge of each task. At one end of the scale, collections with high-quality, strict gold-factuality and other format-specific outputs – receive a large α_c. Where preference is instead driven by more abstract aspects such as style, malicious query refusal, and helpfulness, the gold response need not be unique. In these cases α_c is small or zero, and the preference gap still presents a meaningful optimisation direction [70]. DPO Stage 1 uses the full spread of these weights; DPO Stage 2, whose collections are all curated imitation targets of comparable status, applies a single modest weight uniformly.

In accordance with prior work [50], we find that the SFT term also stabilises DPO training. Since standalone DPO optimises only the relative margin between the chosen and rejected responses, their likelihoods may both decrease while the preference objective improves, provided that the rejected response decreases more rapidly. The auxiliary SFT term directly anchors the chosen response and thereby counters this likelihood displacement.

Length Normalised Implicit Reward. Standard DPO defines the implicit reward as $h_\theta(x, y) = \log \pi_\theta(y \mid x) - \log \pi_{\text{ref}}(y \mid x)$, which sums token level log ratios across the response. Its magnitude therefore varies with sequence length, allowing length to influence the preference margin independently of response quality. We reduce this dependence by using the mean token log ratio, following SimPO [71] and Tülu 3 [14].

$$\bar{h}_\theta(x, y) \;=\; \frac{1}{|y|} \sum_{t=1}^{|y|} \log \frac{\pi_\theta(y_t \mid x, y_{<t})}{\pi_{\text{ref}}(y_t \mid x, y_{<t})}. \tag{6}$$

We apply the same token averaging to the SFT term $\mathcal{L}_{\text{SFT}}$, preventing longer responses from contributing disproportionately due to response length alone.

3.4.2 DPO Stage 1 – Content & Rehearsal Preference Data Generation

Rehearsal DPO Preference Data

The purpose of Rehearsal DPO is to strengthen previously acquired knowledge and abilities that were weakened over the course of model re-alignment and mid-training. We base the Rehearsal DPO corpus on Math[3] [72], Safety[4], Instruction Fine-tuning, Science, and Multilingual[5] subsets of the Nemotron Super dataset family and augment them with new synthetic responses and reasoning traces, generated by permissively-licensed, open-weights models.

The DPO pair synthesis follows a process inspired by [14] and [73]. First, we define a pool of generators powered by selected open-weight LLMs. For each input query, each generator in the pool creates one candidate response. These responses are subsequently pairwise evaluated via LLM judge model equipped with a set of domain-specific criteria and the original answer as a reference, and ranked by the number of times they were judged to be the better option in the direct one-to-one comparison. To avoid a judge's potential self-preference bias, the answer-generation pool never includes any model from the same family as the judge. Finally, the highest-ranking response in the pool is selected as the preferred answer, and its rejected counterpart is either the second-best candidate in the pool, or, to keep the training examples on policy, the response given by the model under training, if the latter happened to rank third or lower in the pool.

Content-Driven Synthetic Preference Data

A strong argument for SovereignAI is the comparative advantage of private data repositories not seen by general-purpose models (§2). Mid-Training absorbs that material as knowledge, but knowledge alone is insufficient for such data to be surfaced in the appropriate manner during practical use cases. This section describes a practical approach to further utilise private data beyond the Mid-Training stage.

For large document corpora, we reverse-engineer queries for which a document implicitly provides an authoritative answer. If appropriate questions are recoverable, the document that suggested the questions also serves as their answer key, and the only thing left to synthesise is the reasoning connecting the two. In this sense, the content supervises every stage (Figure 14): it decides which tasks it can support, it generates the questions, and it guides the reasoning that answers them. No practitioner is in the loop at generation time, yet every target traces back to material a practitioner wrote.

Content Assessment. Not every passage of content can carry a training task, and the deciding factor is rarely quality of writing, which is uniformly high. What varies is whether a passage is substantive enough

[3] https://huggingface.co/datasets/nvidia/Nemotron-SFT-Math-v3
[4] https://huggingface.co/datasets/nvidia/Nemotron-SFT-Safety-v1
[5] https://huggingface.co/datasets/nvidia/Nemotron-Post-Training-Dataset-v2

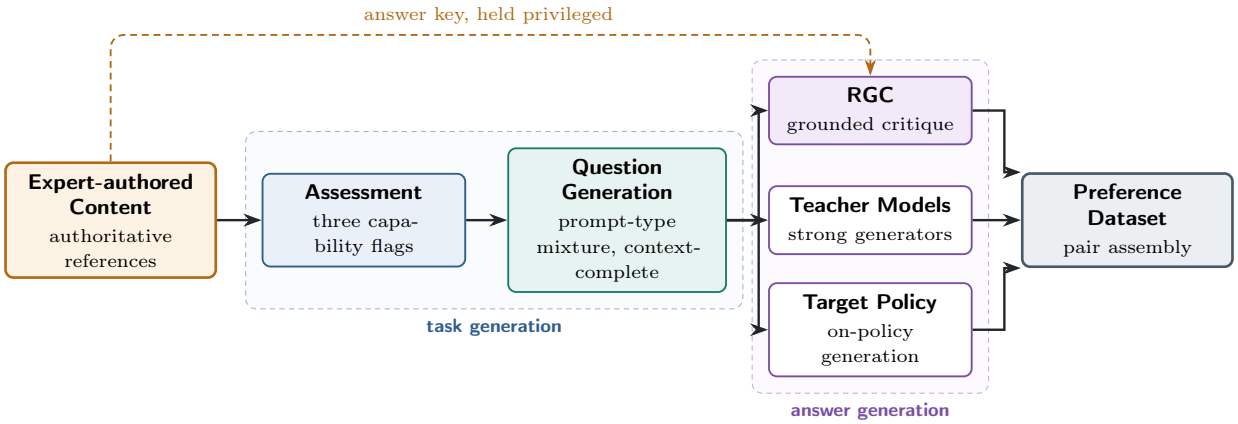


Figure 14 Expert-authored content as the supervision signal. It decides which documents can carry a training task and which task to generate. Answers to those questions are then generated by three routes: RGC, which uses the source content as a reference inside its critique stage, and teacher models and the target policy, which do not. The resulting candidates are assembled into preference pairs.

to be worth asking about. Each document is therefore assessed by a judge and probed for three separate capabilities (Table 4). A passage passes only if it contains substance of interest rather than strings of citations, and is challenging enough that training on it would meaningfully improve the model. Brief, superficial, or purely introductory material is rejected.

Capability	What the assessor looks for
question-answer pairs	factual content that converts into clear, definitive questions; answers that are specific and authoritative rather than vague; enough detail to support several distinct pairs
reasoning traces	step-by-step reasoning with explicit connectives such as "because", "therefore", "since" or "given that"; clear cause and effect; identifiable steps running from premises to conclusion
concept definitions	detailed explanations of concepts, with context and examples; specific rather than general; self-contained enough to be understood without the surrounding document

Table 4 The three capability criteria applied to every document, each returned as an independent true/false judgement on top of a shared quality bar. The datasets described here keep documents that satisfy all required criteria.

Task Generation. Questions are generated one document at a time, controlling: what kind of question gets asked, and whether it is self-contained.

To get a diverse set of questions we sample one of nine prompt types per document from a weighted mixture (Table 5). Six vary the cognitive work demanded, from general reasoning over the whole document to counterfactuals, analogies, claim verification, quantitative analysis, and causal or temporal chains. Three vary the professional stance instead.

The model that will answer these questions never sees the document they came from, so each question has to carry everything needed to answer it: any authority the question turns on quoted inside the question, all necessary facts present as posed. Generated questions are then filtered on the same standard.

Answer Generation. Task generation yields questions without targets. Each question, however, carries the document it was derived from, and that document is an authoritative answer to it by construction: the question was written to be answerable from that passage. Questions and answer keys therefore come paired, at no annotation cost.

Prompt Type	What it asks for	Weight
general reasoning	reasoning across the whole document	30%
counterfactual	how the analysis changes if facts change	10%
analogy	reasoning by comparison to a related case	10%
claim verification	whether a stated claim holds	10%
quantitative	numerical or computational analysis	10%
temporal / causal	sequence and cause-and-effect chains	10%
stance, judge	record building, ambiguities, scope limits	10%
stance, lawyer	doctrine, compliance, client consequences	5%
stance, layperson	the same material in plain language	5%

Table 5 Question generation prompt types and their default sampling weights. One prompt type is drawn per document, so the mixture controls the distribution of task types independently of the topical distribution of the underlying content.

Candidate answers are produced by three routes (Figure 14), which differ in whether the source content participates. Teacher models and the target policy answer from the question alone. Only Reasoning with Grounded Critique (see below) uses the document, and only inside its critique stage. All three routes feed the pool from which preference pairs are assembled (Section 3.4).

Reasoning Trace Generation via Reasoning with Grounded Critique. Reasoning with Grounded Critique (RGC) uses the source content as an answer key. The document that generated the question also contains its answer – only the reasoning that connects them is missing. RGC uses the grounding document in critic-mode to obtain guidance for the generation of the reasoning trace. The target model reasons from the question alone, under that guidance, and writes the final answer from its own trace. The output is a trace the policy itself generated, steered at every step by a document. The procedure is described in Algorithm 1 and illustrated in Figure 15. Data generation is divided into five stages:

- CRYSTALLISE reads the source content once and restates it as a structured argument with its citations intact, organised around the question rather than as a neutral summary.
- IDENTIFY then starts the reasoning trace from the question alone. It is a planning step explicitly barred from answering: it identifies the sub-questions that will have to be resolved and sketches a strategy.
- CRITIQUE sees the question, the trace so far, and the crystallised document, and returns exactly one piece of guidance, tagged with its kind: stay on the current path (CONTINUE), add the single most valuable missing insight (INSIGHT), fix a specific flaw (CORRECT), or stop because the reasoning now suffices (CONCLUDE).
- CONTINUE receives the question, the trace so far and that guidance and extends the trace.
- CONCLUDE writes the final answer from the question and accumulated trace alone.

The process alternates between CRITIQUE and CONTINUE until a stop signal or exhaustion of the iteration budget ends the loop. Every generative stage (IDENTIFY, CONTINUE, CONCLUDE) runs the target policy, so traces stay close to the distribution the model already produces rather than importing a teacher's voice. The loop spends several rounds per example, which is test-time compute paid once at data-creation time instead of at every inference.

The grounding therefore acts as an additional signal during construction of the trace.

Preference Pair Assembly. Each question yields a pool of candidate rollouts drawn from the answer generation routes described above (Figure 14). A judge then compares these candidates against one another to select a chosen and a rejected response, and critically, this comparison is made with privileged access to the expert-authored source content, so the comparison is anchored to what the source content actually supports rather than to surface qualities of the writing. This ensures high quality, diverse preference pairs: grounding in the source content lets the judge catch a strong-sounding but subtly incorrect candidate and reject it in favour of an answer faithful to the document. At the same time, because the candidate pool draws on multiple

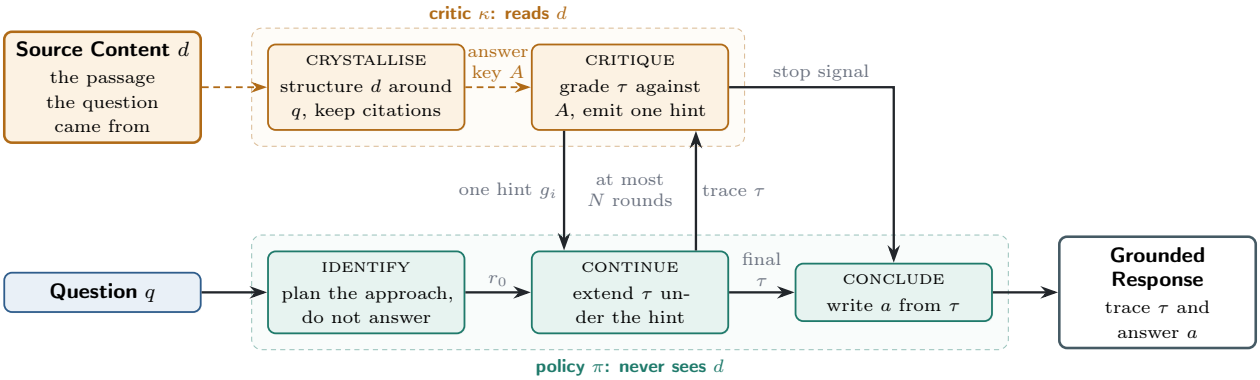


Figure 15 Reasoning with Grounded Critique (Algorithm 1).

Algorithm 1: Reasoning with Grounded Critique. At each round the critic returns a hint g_i together with its tag $c_i \in \{\text{CONTINUE}, \text{INSIGHT}, \text{CORRECT}, \text{CONCLUDE}\}$. The lines marked $\star$ are the only two at which the source document d is read, both by the critic κ.

Input : question q, source content d, budget N, policy π, critic κ
Output : reasoning trace τ, final answer a

$A \leftarrow$ Crystallise(q, d) // $\star$ κ: structure d around q, keep citations $r_0 \leftarrow$ Identify(q) // π: sees q only
  $\tau \leftarrow \langle r_0 \rangle$
for $i \leftarrow 1$ **to** N **do**
  $(g_i, c_i) \leftarrow$ Critique(q, τ, A) // $\star$ κ: g_i=hint, c_i=its tag **if** $c_i = \textit{CONCLUDE}$ **then break**
  $r_i \leftarrow$ Continue(q, τ, g_i) // π: sees g_i, not A or d $\tau \leftarrow \tau \,\|\, \langle r_i \rangle$ // append, never rewrite
$a \leftarrow$ Conclude(q, τ) // π: sees q and τ only **return** (τ, a)

answer generation routes rather than a single one, the chosen and rejected responses in a given pair need not come from the same route, keeping the dataset diverse across reasoning styles and generation strategies.

Ontology-Driven Synthetic Preference Data

Domain Ontologies. The previous section describes how we derive supervision from content by recovering the questions a document already answers. Inspired by [74], we take advantage of a second route that opens whenever the content implicitly adheres to an *ontology*: a specification of the key entity types and the relations among them. In specialised domains such a structure almost always exists and is almost never written down, precisely because the practitioners writing the document already share it; yet it is what the prose is organised around. A clinical case report, for instance, is organised around entities such as symptoms, test results, comorbidities, diagnosis, treatment and outcome, and around the relations that connect them: the diagnosis together with the patient's comorbidities guides the treatment, which itself drives the outcome. No sentence in the report states this schema, but every sentence is placed within it. Consequently, the document admits a faithful representation as a knowledge graph whose schema is the ontology itself. Two properties follow. First, tasks that are meaningful to practitioners can typically be expressed in terms of the ontology, so that inputs and targets may be obtained by traversing the graph rather than authored by hand. Recovering the diagnosis from the symptoms and test results attached to a case is one such task: it amounts to anchoring on a diagnosis, collecting the symptom and test-result nodes adjacent to it, and withholding the diagnosis itself. Second, because the schema is fixed, the same traversal can be applied to every knowledge graph derived from documents in the corpus. Dataset construction then becomes mechanical.

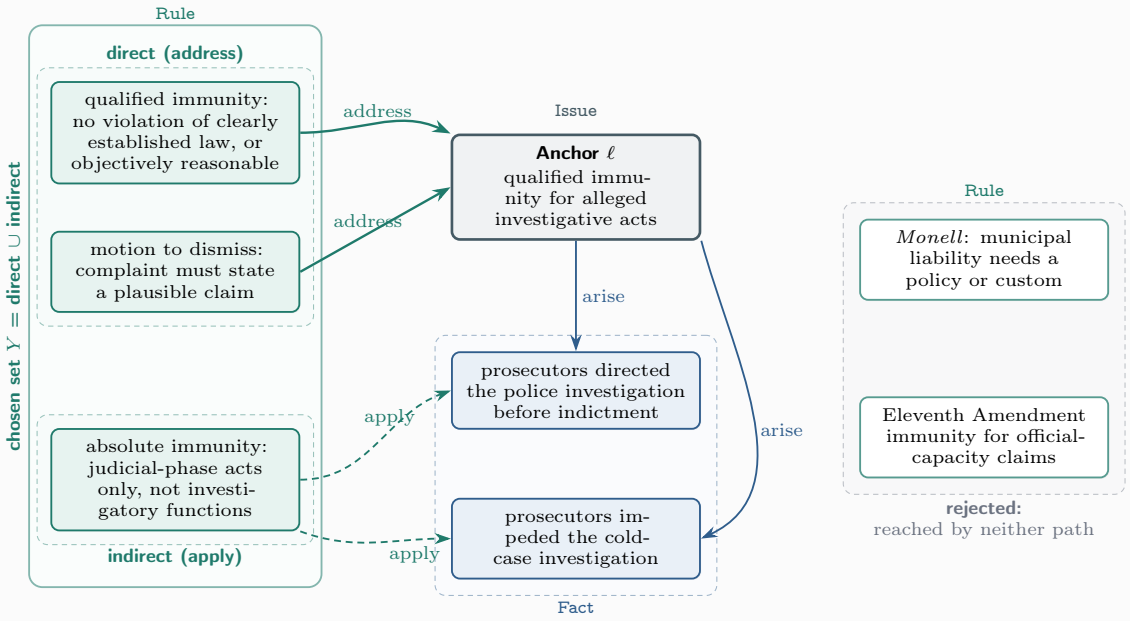
From Graph to Pairs. Let $\mathcal{O}$ be an ontology and D a document conforming to it. A *traversal* $\mathcal{T}$ is a fixed procedure that starts from a chosen *anchor* node a, the entity a sample is built around, such as a single diagnosis, and walks the graph to select an input set I and a target set Y, returning the pair (I, Y). Training samples are then obtained in three steps:

1. INSTANTIATE: Extract a knowledge graph G_D from D using $\mathcal{O}$ as schema, so that every node and edge carries a type drawn from it.
2. COLLECT: Sample an anchor a and evaluate $\mathcal{T}(G_D, a)$. Each anchor yields one sample, so a document contributes as many samples as it admits valid anchors.
3. RENDER: Verbalise the task as an instruction, the entities of I as the input, and the entities of Y as the reference answer.

Finally, when the objective is a preference dataset rather than a single reference answer, negatives come for free: the entities of the target type that occur in the same graph but were not collected are topically close and, up to extraction error, wrong by construction.

Data construction. We applied the pipeline to ontologies unique to reasoning in our target domains. We use an LLM to instantiate the knowledge graph for a given document and create preference data as described above. Example 5 shows the process for a data point.

Example 3.1: Preference pair from rule traversal



The rule traversal on an extracted graph. Node labels are abbreviated – the pair appears verbatim below.

Input – the anchor ℓ and the facts it arises from

instruction: "Analyse the provided facts and issue, and identify the most important rules that are applicable to address the issue."

facts:

- Prosecutors allegedly directed police investigation prior to indictment, including ordering incomplete polygraph, directing DNA not be tested, and relying on non-credible witness
- Prosecutors allegedly impeded cold case investigation by coercing witnesses and threatening police officials

issue: Whether prosecutors are entitled to qualified immunity for alleged investigative acts.

Rejected – rules of the same opinion reached by neither path

- *A municipality may be held liable under § 1983 when execution of a government policy or custom inflicts the injury, but not under respondeat superior.*
- *District attorneys acting in a quasi-judicial capacity represent the state and are entitled to Eleventh Amendment immunity for official capacity suits.*

Chosen – the union of the two paths

- [DIRECT] *Qualified immunity protects officials if their actions did not violate clearly established law or were objectively reasonable.*
- [DIRECT] *To survive a motion to dismiss, a complaint must contain sufficient factual matter to state a plausible claim for relief.*
- [INDIRECT] *Prosecutors are absolutely immune for acts intimately associated with the judicial phase of the criminal process, but not for administrative or investigatory functions unrelated to preparation for judicial proceedings.*

Why these three, in this order.

1. Absolute immunity reaches only judicial-phase acts – not these, so *qualified* immunity is what is in issue.
2. Qualified immunity then turns on clearly established law or objective reasonableness.
3. At the pleading stage, that question is asked of the complaint rather than of proven facts.

→ The opinion concludes: no qualified immunity at the motion to dismiss stage.

What the indirect path buys. Rule 1 carries no ADDRESS edge to ℓ; in the graph it addresses the neighbouring absolute-immunity issue. It enters the target set only because the opinion APPLIES it to both of ℓ's facts. The direct path alone would drop the rule that frames the question.

Why the negatives are hard. Both concern who else is answerable – the county under *Monell*, the district attorneys in their official capacity – not whether these prosecutors are shielded for this conduct.

3.4.3 DPO Stage 1 – Data-Mixture Optimisation

The proportions assigned to training collections can materially affect the outcome of post-training under a fixed compute budget [75, 76]. Selecting these proportions is a joint optimisation problem because the value of any collection depends on the weights assigned to the others, as well as on the base model and training horizon. As the number of collections grows, the candidate mixtures form an increasingly high-dimensional probability simplex, making manual search expensive and unlikely to capture these interactions.

This motivates learning the data-mixture. A line of work trains proxy models on sampled mixtures and uses their measured quality to choose weights for the target model: RegMix [77] regresses validation loss on the mixture weights, while DoReMi [78] reweights collections directly on a proxy model through group distributionally robust optimisation. These approaches reduce dependence on manual tuning but leave two difficulties in our setting.

- **Cost of training and evaluation.** Every observation costs one full round of training followed by evaluation across a wide and diverse range of downstream tasks, which for a post-trained model demands generation and model-based grading rather than the negative log-likelihood measurements that RegMix and DoReMi rely on for pre-trained models. A dense search at the target scale is therefore prohibitively expensive.
- **Model Dependent Optima.** The optimal mixture depends on both the capabilities and scale of the base model. Transferring a mixture selected on a small proxy model directly to the target model consequently assumes that mixture rankings remain stable across scale, which need not hold.

To address these challenges, we adapt ADMIRE-BayesOpt [79] and formulate data mixture selection as a sequential multi-fidelity Bayesian optimisation problem. Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_D)$ denote the proportions assigned to D collections, where $\pi_d \geq 0$ and $\sum_{d=1}^{D} \pi_d = 1$, and let m denote model scale. Each training and evaluation run provides a noisy observation of the quality associated with one mixture and model pair. A multi-fidelity Gaussian process (GP) serves as a surrogate over this joint space, predicting the quality of untested configurations, quantifying uncertainty in those predictions, and learning how evidence transfers across model scales. Optimisation proceeds sequentially by updating the surrogate after each batch of observations (data mixtures, model scale, and associated evaluation results) and using an acquisition function to select the next mixtures and model scales from their predicted quality, uncertainty, and experimental cost. This process naturally balances exploration of uncertain regions against exploitation of promising mixtures, while allocating inexpensive proxy runs when they remain informative and reserving target scale evaluations for refinement of the final optimum.

Optimisation Objective. For Q in-house evaluation tasks, let $s_q(\boldsymbol{\pi}, m) > 0$ denote the standardised score on task q after training model m with mixture $\boldsymbol{\pi}$, shifted by a constant to remain positive across the mixtures we evaluate. We optimise their geometric mean

$$f(\boldsymbol{\pi}, m) = \left(\prod_{q=1}^{Q} s_q(\boldsymbol{\pi}, m) \right)^{1/Q}. \tag{7}$$

This objective assigns a larger marginal benefit to the same absolute improvement on a lagging task than on an already strong task. The preference is aligned with the role of mid-training (including DPO stages) in our training pipeline, which must provide a broadly capable initialisation for the subsequent RL stage rather than maximise retrospective performance on a narrow subset of evaluations. Since RL improvements depend on capabilities already present in the initial policy [80, 81], a mixture that leaves one area substantially behind can constrain later exploration even when its arithmetic mean is high.

Multi-Fidelity Optimisation. We conduct the sequential search with batched queries in three phases, using the smaller model as an inexpensive source of information and the target model as the high fidelity objective.

- Phase 1 – Parallel Exploration. The search is initialised with mixtures drawn from a Dirichlet distribution centred on a manually specified prior, with rejection sampling on cosine distance to maintain coverage of the simplex. Each mixture is trained and evaluated on the proxy model in parallel, and the resulting observations provide the initial fit of the GP surrogate at low cost.

- Phase 2 – Batched Multi-Fidelity Querying. Each iteration fits the multi-fidelity GP to all observations collected so far, maximises the acquisition function jointly over the mixture and the model scale, and selects a batch of candidate experiments subject to the same diversity constraint. The selected configurations are trained and evaluated, their results are appended to the observation set, and the procedure repeats. The multi-fidelity GP and the Bayesian optimisation loop together balance exploration against exploitation, with transferability across model scale modelled explicitly through the cross-scale covariance, which makes the search over data mixtures more efficient.
- Phase 3 – Target-Scale Refinement. We restrict the remaining queries to the target model and use the transferred GP surrogate to refine its model-specific optimum. This phase concentrates the high-fidelity budget on resolving the most promising and uncertain target scale mixtures, yielding the final mixture for each target model.

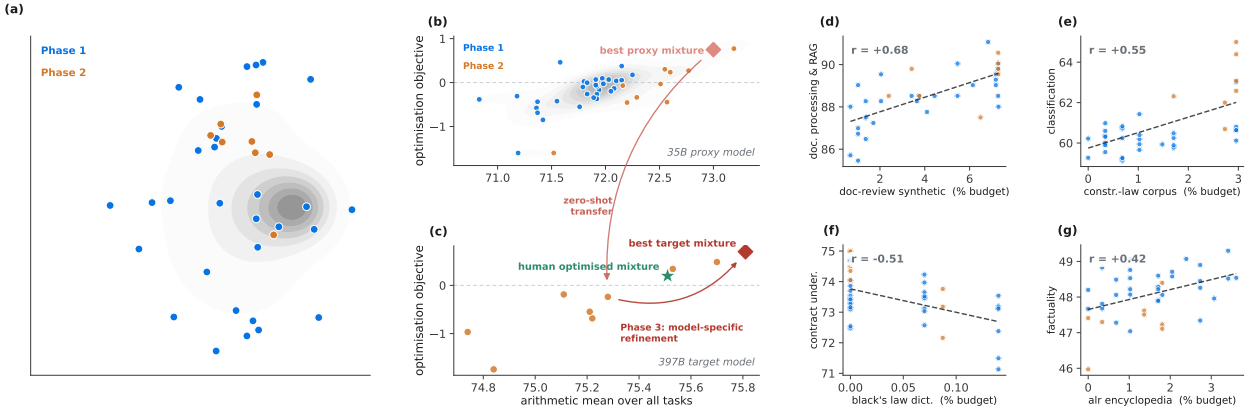


Figure 16 Multi-fidelity data-mixture optimisation for the DPO Stage 1 training. Panel (a) shows the distribution of candidate data mixtures explored during the search, projected onto the first two principal axes of the mixture simplex. Panels (b) and (c) show the optimisation progress at the proxy and target model scales, plotting the geometric-mean optimisation objective against the arithmetic mean over all evaluation tasks. Arrows trace the best proxy mixture as it is transferred zero-shot to the target scale, where it is no longer optimal, and the subsequent Phase 3 refinement that reaches the best target mixture. Panels (d)–(g) show the association between individual evaluation scores and the budget share allocated to the collection most correlated with each task, with the Pearson correlation inset.

As shown in Figure 16, relative to the human-optimised mixture, the final DPO Stage 1 training run on the learned best mixture increases the geometric mean objective from 0.20 to 0.70 and simultaneously achieves the highest arithmetic mean across all validation tasks, including tasks held out from the optimisation objective. We note that the best proxy mixture is suboptimal on the target model and would perform below the manually selected mixture if transferred without further search. The multi-fidelity procedure therefore reduces the cost of exploration without treating proxy performance as a scale invariant ranking.

The effect of training the model on a selected DPO data mixture is shown in Figure 17. Alignment-adjacent capabilities improve while more technical, execution-heavy capabilities slightly regress. The largest gains appear on CapTrack's Knowledge axis, in Multilingualism (+9.08%), Reasoning (+5.09%), and Safety & Values (+4.52%), reflecting the direct targets of preference optimisation, with a small trade-off in Coding (-2.92%) and Writing (-3.29%). Changes on the Ability axis are smaller, with gains in Policy & Behavioural Preferences (+0.99%) and Latent Competence (+0.78%) offset by a modest drop in Protocol Compliance & Execution (-0.26%). Encouragingly, this trade-off does not appear to damage downstream performance in the target domain: benchmarks show broad-based improvement, led by Summarisation (+5.63%) and Reasoning (+4.71%). Together, these results suggest that thoughtful DPO data selection can strengthen the model's alignment to a target-domain without meaningfully undermining general capabilities.

3.4.4 DPO Stage 2 – Agentic Preference Data Generation

The agentic preference dataset supports an off-policy warm start for subsequent RL by familiarising the policy with the Deep Research harness and retrieval tools engineered to provide models with up-to-date information

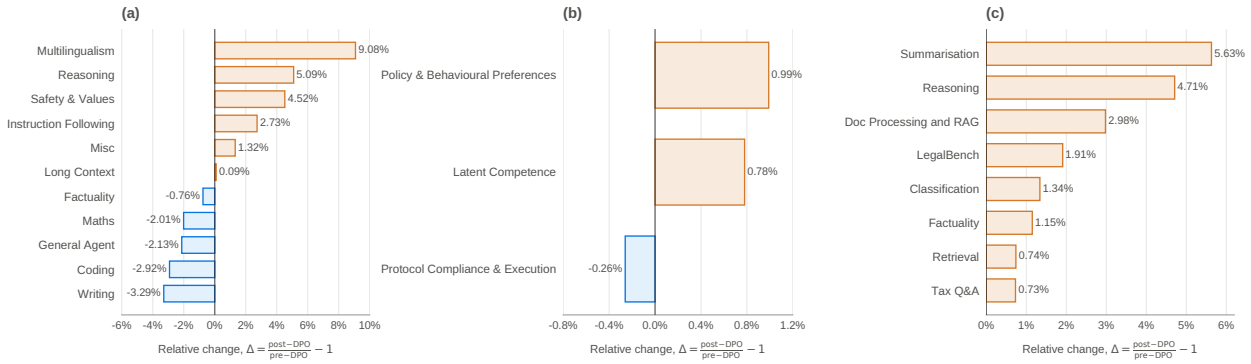


Figure 17 Relative Change in Performance after DPO Stage 1. (a) CapTrack – Knowledge: largest gains in Multilingualism, Reasoning, and Safety & Values, but small regressions in Coding and Writing. (b) CapTrack – Ability: modest gains in Policy & Behavioural Preferences with slight drop in Protocol Compliance & Execution. (c) Target Domain: broad improvements led by Summarisation and Reasoning. Orange = improvement, blue = regression.

(§2). It also targets weaknesses observed in the current model, including failure to invoke available tools and fabrication of sources. Deep Research requires the model to plan an open-ended investigation, retrieve and interpret relevant evidence, and synthesise a grounded report. The task and harness are described in Section 2.3, while the corresponding evaluation protocol is presented in Section 4. This section describes how completed trajectories are decomposed into pivotal decision points and converted into preference data.

Each preference example contains the off-policy prefix leading to a selected node and alternative chosen and rejected completions at that node. The node is a pivotal decision point because its output either determines the subsequent research state or contributes directly to final report quality. Pairing a lower-quality or behaviourally undesirable completion with a preferred alternative assigns supervision directly to the responsible decision. Depending on the selection criterion, this preference may encode a targeted behavioural correction, such as invoking retrieval tools before answering, or a relative quality improvement, such as stronger factual grounding during compaction and report synthesis.

Trajectory Collection and Pivotal Node Extraction

We first collect complete trajectories from which prefixes and pivotal nodes can be extracted. For each item, human experts author a research *query*, annotated with metadata such as domain and complexity, together with a two-tier *rubric* specifying the essential and supplementary content of a competent response. The query initiates the harness execution, while the rubric provides the standard against which model response quality is assessed.

The queries are executed through *fr-agents*, the Deep Research harness shown in Figure 4. Its graph contains four node types, with each node execution corresponding to one model call.

- The PLANNER decomposes the research question into a plan.
- The RESEARCHER decides at each turn whether and how to invoke a retrieval tool.
- The COMPACTION node condenses retrieved documents while preserving relevant evidence and citations.
- The REPORTER synthesises the accumulated evidence into the final report.

Each node execution is logged with its trajectory prefix, completion, and available tool schemas. A single trajectory therefore yields multiple candidate decision points. The number of calls varies substantially by node because the researcher executes repeatedly whereas the reporter executes once. As summarised in Table 6, this asymmetry produces approximately two orders of magnitude more researcher examples than reporter examples.

Counterfactual Completions from a Shared Prefix. A recorded node supplies an off-policy prefix containing the research state, message history, and available tool schemas at a pivotal decision point. We re-run the node

Node	Calls / trajectory	Rejected-side limitation	Chosen-side improvement
Planner	1-2	shallow or mis-scoped plan	stronger planning behaviour
Researcher	∼17	answers without retrieval	tool-grounded research
Compaction	many	drops or distorts evidence	factual evidence retention
Reporter	1	unsupported or incomplete report	grounded report synthesis

Table 6 Pivotal nodes extracted from Deep Research trajectories, including their typical frequency, the limitation represented by the rejected completion, and the behavioural or quality improvement represented by the chosen completion.

under multiple open-weight models while holding this prefix fixed, producing alternative completions that differ only in the decision being supervised. Comparing completions from a shared prefix isolates the effect of that decision without the trajectory drift that would arise from comparing independently generated rollouts.

Preference Construction at Pivotal Nodes

We convert the alternative completions at each pivotal node into preference pairs using three complementary selection signals. Teacher identity transfers behaviours that cannot yet be scored reliably, a programmatic tool-use rule corrects an observed agentic failure, and factuality ranking selects the higher-quality completion where groundedness can be measured. The rejected side may therefore exhibit a specific undesirable behaviour or simply lower measured quality, while the chosen side provides the corresponding preferred alternative.

Constructing these pairs offline permits stronger supervision than would be practical within an online training loop. The chosen completion may be generated by a stronger model with access to the SME rubric, and pair selection may use a judge whose cost would be prohibitive as an online reward. The same prefix pool can also be re-evaluated as the selection criteria improve. These preference pairs are used in DPO Stage 2, which provides the off-policy warm start for RL. A disjoint set of recorded prefixes at pivotal nodes is subsequently reused as training queries for RL, allowing the same decision types to be optimised on-policy without overlap with the DPO preference data.

Teacher-Guided Behavioural Repair. Teacher-guided pairs transfer planning and research behaviours for which a reliable automatic quality metric is not available. For each shared prefix at a `planner` or `researcher` node, a completion from a stronger agentic model is selected as chosen and one from a weaker model as rejected, provided both are valid. Relative model strength is established through consistently higher performance on our Deep Research evaluation. This uses model identity rather than a per-example judge to define the preference, capturing behaviours such as decomposing the query into appropriately scoped research tasks and continuing retrieval when the available evidence remains insufficient.

Factuality-Based Quality Selection. Factuality-based pairs improve the quality of `compaction` and `reporter` outputs by preferring completions that remain grounded in the evidence available in the shared prefix. Model identity is not a reliable selection rule for these nodes because no candidate model is consistently the most factual across examples, as shown in Figure 18.

We score every candidate completion using the factuality judge described in Section 4.2.2, which extracts claims and assesses their support in the available source material. The highest-scoring completion forms the chosen side and the lowest-scoring completion forms the rejected side. Pairs with a score difference below 0.10 are removed to reduce ambiguous preferences arising from judge variability. This criterion directly targets the responsibilities of the two nodes. Compaction must preserve relevant evidence without introducing unsupported content, while reporting must synthesise the accumulated evidence into a grounded final response.

Programmatic Tool-Use Correction. Tool-use pairs correct a recurrent researcher failure in which the model answers from parametric memory despite being given retrieval tools. This behaviour can produce apparently researched responses containing fabricated citations because no supporting document was retrieved. It occurs across all evaluated backbone models, with substantially different frequencies, as shown in Figure 19.

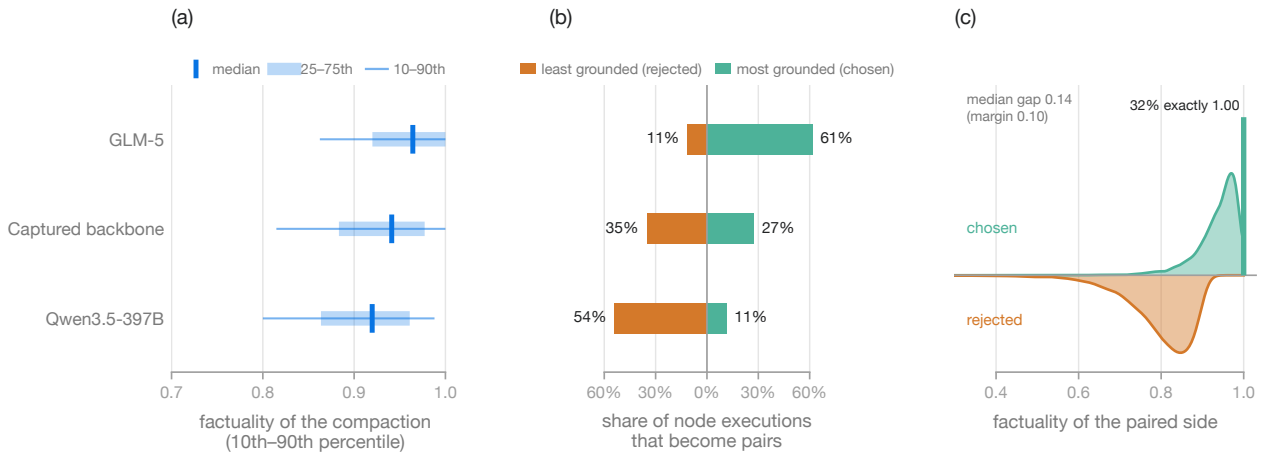


Figure 18 Factuality-based selection for compaction preference pairs. **(a)** Candidate factuality distributions overlap substantially, showing that model identity does not reliably determine preference. **(b)** Frequency with which each model supplies the chosen and rejected completions. Every model appears on both sides of the preference pairs. **(c)** Factuality distributions of the resulting pairs, with a median chosen-rejected difference of 0.14.

Pair selection uses a programmatic predicate rather than a model judge. At a shared prefix where retrieval is the required next action, the rejected completion returns a substantive answer without invoking any tool, while the chosen completion invokes at least one retrieval tool. The preference therefore provides a targeted behavioural repair signal that teaches the policy to gather evidence before composing an answer. It does not attempt to rank the substantive quality of two completed reports.

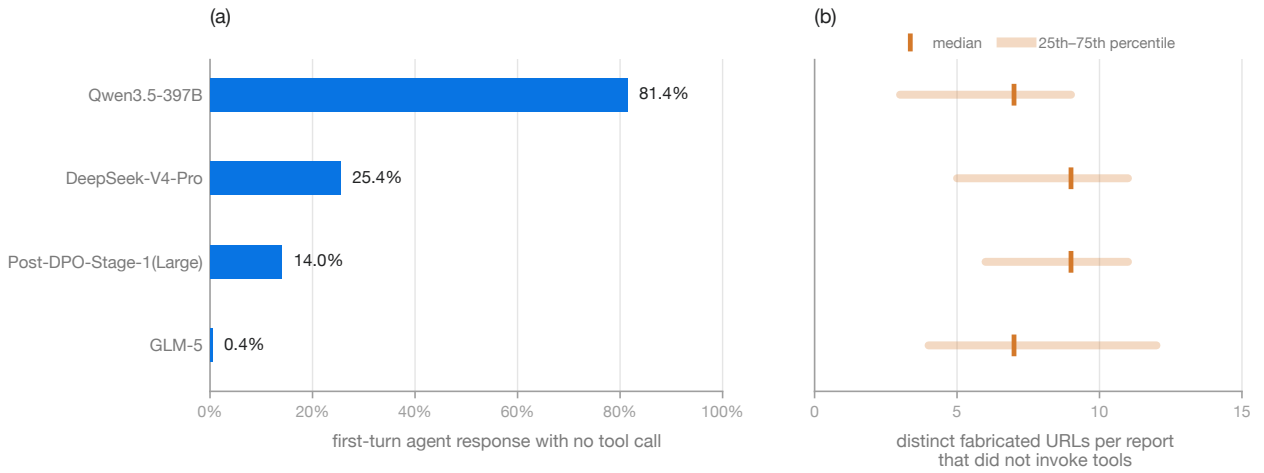


Figure 19 Tool-use failure at the first `researcher` node in the Reuters query pool, where the trajectory prefix contains no retrieved evidence. **(a)** Share of first turns each backbone answered with zero tool calls despite being handed the full retrieval tool set. The behaviour is present in every backbone we ran, spanning two orders of magnitude. **(b)** Distinct URLs cited in exactly those answers. Since the turn issued no tool call, nothing was fetched, so every one of these URLs is invented; a typical fabricated report carries seven of them. That contrast is what identifies the failure as fabrication rather than a refusal or an empty turn, and it is the reason we pair against it.

Illustrative Preference Pairs

The following examples show how a shared off-policy prefix is paired with alternative rejected and chosen completions. The first expresses a targeted behavioural correction, while the second encodes a measured difference in factual quality.

Tool-Use Behaviour Correction. The shared prefix presents the same research state and retrieval tools to both candidates. The rejected completion answers without gathering evidence, while the chosen completion initiates retrieval.

Example 3.2: Tool-use preference pair at a researcher node

Shared prefix

"Retrieve direct quotes and policy positions from Nordic leaders and Reuters analysis on the long-term viability of the Arctic Council as a governance body, including any publicised Nordic proposals for reforming observer rules, altering consensus mechanisms, or creating parallel frameworks to preserve governance functions if the Council remains deadlocked at the ministerial level."

Rejected

[...] *Following Russia's invasion of Ukraine in February 2022, the seven other Arctic Council member states (Canada, Denmark, Finland, Iceland, Norway, Sweden, and the United States) issued a joint statement on March 3, 2022, condemning the invasion and announcing a pause in participation in Council meetings. [Joint statement by the Arctic Council states on Russia's actions in Ukraine](https://www.reuters.com/world/europe/arctic-council-states-pause-participation-meetings-due-ukraine-crisis-2022-03-03/) This initial pause effectively halted all high-level governance functions, as the Council operates on a consensus basis requiring the participation of all eight members, including Russia.* [...] *Norwegian Foreign Minister Anniken Huitfeldt stated in March 2022 that while cooperation with Russia was suspended, the "Arctic Council remains the key forum for Arctic cooperation" and that the pause was intended to be temporary pending a change in circumstances. [Norway's Huitfeldt on Arctic Council pause and future cooperation](https://www.reuters.com/world/europe/norway-foreign-minister-says-arctic-council-cooperation-suspended-2022-03-04/)* [...]

Chosen

Tool calls

```
reuters_search
{
   "query": "Nordic Arctic Council governance reform consensus observer rules",
   "start_date": "2022-02-01",
   "end_date": "2026-06-11"
}
reuters_search
{
   "query": "Arctic Council ministerial deadlock Russia Nordic",
   "start_date": "2022-02-01",
   "end_date": "2026-06-11"
}
reuters_search
{
   "query": "Norway Sweden Finland Denmark Iceland Arctic Council future",
   "start_date": "2022-02-01",
   "end_date": "2026-06-11"
}
```

Factuality-Based Quality Selection. The shared prefix contains the same retrieved document for both candidates. Both identify the central jurisdictional limitation, but the rejected completion adds an unsupported threshold and a case citation absent from the source. The highlighted spans show the resulting factuality failure.

Example 3.3: Factuality preference pair at a `compaction` node

Shared prefix

[...] "You are a research assistant analysing documents retrieved by `document-fetch-tool` to answer a specific research question. Overall research topic: Scottish sheriff court interdict enforcement research plan. Current research step: lay representatives and McKenzie friends in Scottish courts. Step description: research the rules governing party litigants' rights to assistance from lay representatives, specifically (a) the common law right to a McKenzie friend established in *McKenzie v McKenzie*; [...] (g) the position under Scottish Civil Justice Council rules or practice directions. [...] Analyse the document(s) below and produce a targeted, context-aware summary that:" [...] *Practice Note (Sen Cts: McKenzie Friends: Civil and Family Courts)*, 2010 WL 3017993 [...]

Rejected: (factuality: 0.71)

[...] *Key claims: McKenzie friends have no right of audience and no right to conduct litigation. The court has discretion to refuse a litigant the assistance of a McKenzie friend. The court has power to grant rights of audience to McKenzie friends in exceptional circumstances.* [...] *Analysis: this document addresses points (c), (d) and (e) of the research question but applies to English courts only. For Scottish sheriff court interdict proceedings, the researcher must locate Scottish Civil Justice Council rules or practice directions, and Scottish case law on McKenzie friends originating from McKenzie v McKenzie [1970] 3 All ER 362, which Lord Neuberger in this Practice Note references.* [...]

Chosen: (factuality: 0.94)

[...] *Key claims: McKenzie friends do not have an automatic right of audience or right to conduct litigation. The court has power to refuse a litigant the assistance of a McKenzie friend, and discretion as to when to exercise that power. The court may grant rights of audience to McKenzie friends in certain circumstances. This Practice Note superseded Practice Direction (Fam Div: President's Guidance: McKenzie Friends) [2008] 1 W.L.R. 2757.* [...] *Analysis: critical limitation for Scottish research. This is an English Practice Note from the Senior Courts of England and Wales, not Scottish authority. It does not address Scottish sheriff court procedure, Scottish Civil Justice Council rules, or Scottish case law, and the researcher should seek Scottish-specific sources for binding authority.* [...]

Rejected contains several unsupported spans

- *"... grant rights of audience to McKenzie friends in exceptional circumstances."* The document says only "the circumstances in which it would exercise that power". The threshold is imported from outside knowledge and presented as a claim sourced to the document.
- *"McKenzie v McKenzie [1970] 3 All ER 362, which Lord Neuberger in this Practice Note references."* The case appears nowhere in the document's Cases Cited table, and the reference is wrong in any event: the case is *McKenzie v McKenzie* [1971] P 33; [1970] 3 All E.R. 1034.

3.4.5 DPO Stage 2 – Agentic, Long-Context Specialisation

The second training stage initialises from the DPO Stage 1 checkpoint and specialises the policy on the mixture of synthetic agentic preference collections described in Section 3.4.4, primarily targeting planning, tool use, factual grounding, and report completeness in Deep Research. Training uses a context length of up to 64k tokens to accommodate the longer 1-step pivot rollout trajectories from which these preferences are derived.

Since this stage adapts an already aligned policy, we use a larger reference policy KL coefficient, a lower learning rate, and a single training epoch to reduce the risk of eroding the broad capabilities acquired during DPO Stage 1. We apply a single modest SFT coefficient across all collections because their chosen responses are curated imitation targets of comparable status. Figure 20 shows the resulting training dynamics across both stages.

3.4.6 Assessing the Effectiveness of DPO Warm Start

As motivated in Sections 3.1 and 3.4, the two DPO stages are intended to warm start RL rather than serve as a terminal optimisation procedure. The desired checkpoint should therefore improve the reliability of a single response without narrowing the policy distribution so aggressively that high-quality behaviours become inaccessible during subsequent exploration. This distinction matters because a checkpoint that improves average reward while reducing its attainable reward under larger sampling budgets may consequently provide a weaker RL initialisation despite appearing stronger under single sample evaluation [80, 82, 83].

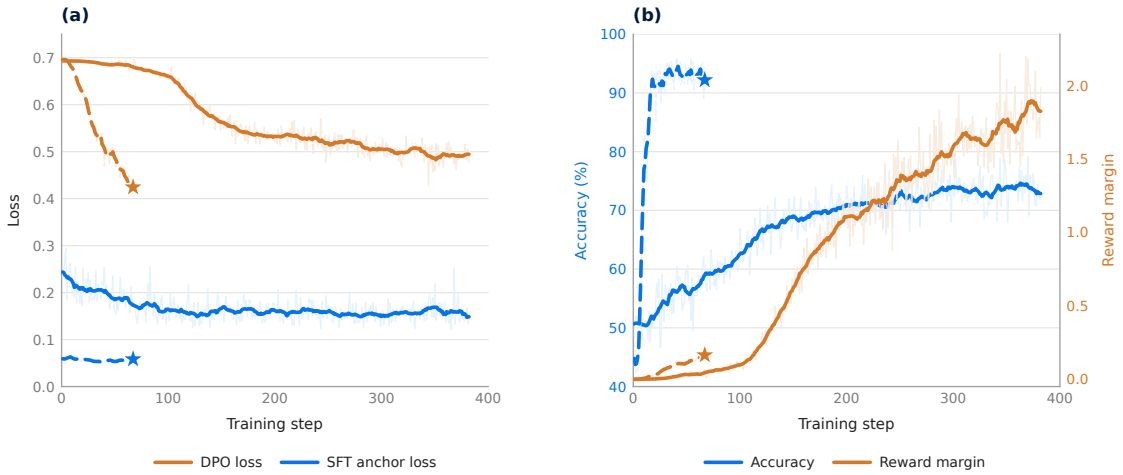


Figure 20 DPO training dynamics across both stages. Solid lines show Stage 1 (382 steps, broad alignment); dashed lines show Stage 2 (67 steps, agentic specialisation). **(a)** Both DPO and SFT losses decline across stages. **(b)** Accuracy and reward margin increase, demonstrating effective preference learning.

We evaluate this RL potential on a held out subset of the subsequent RL task distributions whose prompts were excluded from both data mixture selection and the two DPO stages. This separation tests whether the gains transfer beyond the tasks used to select the mixture and optimise the policy. We use reward@k, a continuous extension of pass@k [84] to the graded rewards used by our RL environments. For each prompt q, we draw $n \geq k$ independent rollouts with rewards $r_1, \ldots, r_n$ and measure the expected best reward among a uniformly sampled subset S of size k

$$\text{reward@}\,k = \mathbb{E}_q\left[\mathbb{E}_{S\sim\mathcal{U}_{n,k}}\left[\max_{i\in S} r_i\right]\right] = \mathbb{E}_q\left[\binom{n}{k}^{-1}\sum_{i=k}^{n}\binom{i-1}{k-1}r_{(i)}\right]. \tag{8}$$

Here, $\mathcal{U}_{n,k}$ is the uniform distribution over subsets of size k, and $r_{(1)} \leq \cdots \leq r_{(n)}$ are the sorted rollout rewards. The combinatorial weight counts the subsets for which $r_{(i)}$ is the maximum, and binary rewards recover the unbiased pass@k estimator exactly. Reward@1 is the standard mean reward and measures single attempt reliability, whereas continued improvement as k increases measures whether the policy retains high reward outputs that additional sampling or RL can make more probable.

Figure 21 shows that the final DPO checkpoint achieves higher reward@k than its initialisation at every evaluated sampling budget across all held out task groups. The improvement at $k = 1$ establishes stronger single attempt performance, while the continued growth of each curve as k increases shows that this gain is not obtained by collapsing the policy onto a narrow set of responses. The DPO stages therefore produce both greater sampling efficiency and a higher attainable reward throughout the evaluated range, providing reinforcement learning with a stronger initial policy while retaining positive exploration headroom. The evaluation data were unseen during mixture selection and DPO, so the consistent gains also provide evidence of generalisation beyond the distributions directly optimised in the preceding stages.

3.5 Reinforcement Learning

Starting from a policy warm-started through DPO, RL further enhances model capabilities through on-policy exploration within interactive task environments, with reward feedback guiding learning towards a policy whose behaviour better satisfies the task objectives, albeit at greater computational cost [3].

We divide RL into two stages: RL Stage 1 develops broad task competence through single-step training on a mixture of diverse queries and general capability tasks. The mixture also includes pivotal examples constructed from off-policy prefixes at consequential points in agentic trajectories, allowing report writing, tool result summarisation, and related behaviours to be improved in isolation before they are composed within a complete workflow. RL Stage 2 then specialises the policy through multiturn rollouts extending up to the

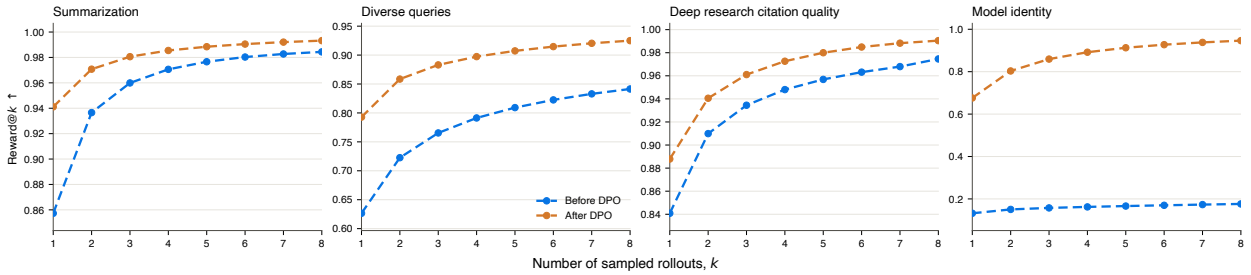


Figure 21 Reward@k before and after DPO on RL tasks held out from mixture selection and DPO training, averaged within summarisation, diverse queries, Deep Research citation quality, and model identity QA. For each prompt, we report the expected maximum reward among a random subset of k rollouts. The DPO checkpoint achieves higher reward@k at every evaluated k.

model's full context window, in which planning, tool use, evidence integration, and task completion must be coordinated across the full trajectory. This progression serves as a capability curriculum, with the first stage strengthening broadly useful local decisions and the second integrating them over extended agentic horizons.

The section covers, in order, the training algorithm in Section 3.5.1, the reward library and its associated design, diagnostic screening, and efficiency techniques in Section 3.5.2, representative reward groups spanning Diverse Queries, Deep Research, and citations in Sections 3.5.4–3.5.6, the environments design in Section 3.5.7, and the experimental results in Section 3.5.8.

3.5.1 Algorithmic Overview

We optimise the policy with Group Sequence Policy Optimisation (GSPO [85]), a policy gradient method with group relative advantage estimation that avoids the need to train a separate value network. Given a prompt x, the rollout policy $\pi_{\theta_{\text{old}}}$ samples a group of G trajectories $\{y_i\}_{i=1}^{G}$, each comprising reasoning, tool calls, tool observations, and a final response. Each trajectory receives a scalar outcome reward $r(x, y_i)$, and the resulting group relative advantage $\hat{A}_i = \big(r(x, y_i) - \text{mean}_j\, r(x, y_j)\big) / \,\text{std}_j\, r(x, y_j)$ is shared across all generated tokens.

Training maximises the GSPO objective

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y_{1:G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{G} \sum_{i=1}^{G} \min\Big(s_i(\theta)\hat{A}_i,\, \text{clip}\big(s_i(\theta), 1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}}\big)\, \hat{A}_i \Big) \right]. \tag{9}$$

GSPO assigns each trajectory a length normalised importance weight $s_i(\theta) = \big(\pi_\theta(y_i \mid x)/\pi_{\theta_{\text{old}}}(y_i \mid x)\big)^{1/|y_i|}$ and aggregates the clipped surrogate loss by taking the mean across samples. This sequence level formulation is well suited to mixture-of-experts models. We use asymmetric clipping bounds as proposed in DAPO [86], and omit the KL penalty against a frozen reference policy.

We found the following techniques important for maintaining stable and efficient optimisation throughout RL post-training.

Inference Policy Log Probabilities. Importance sampling requires the denominator to represent the behaviour policy that generated each trajectory. In asynchronous training, trajectories may originate from several historical policy versions, making recomputation under a single recent checkpoint inaccurate and retaining every historical checkpoint impractical. We therefore define $\pi_{\theta_{\text{old}}}$ using the token log probabilities recorded by the vLLM rollout engine at sampling time [87]. This directly identifies the likelihood under the policy that produced the sampled tokens and avoids additional discrepancies from recomputation under different training kernels and parallelism.

Asynchronous Rollouts. Agentic trajectories vary substantially in duration, while LLM-judged rewards introduce an additional inference workload. Rollout generation and reward computation therefore run asynchronously with policy optimisation, and a trajectory may be generated by a rollout policy up to five

versions behind the current training policy. This bounded policy lag allows rollout, judge, and trainer computation to overlap without requiring their colocation.

Reward Normalisation within Groups. Our training mixture spans collections with distinct reward scales and variances. Without normalisation, collections with higher reward variance would produce larger advantages and contribute disproportionately to the policy gradient. Standardising rewards within each prompt group makes the update depend on relative differences among trajectories and reduces this imbalance across collections.

Overlong Filtering. During RL Stage 1, we apply overlong filtering to single-turn responses that reach the configured generation limit, avoiding the noisy punitive signal that can otherwise arise solely from response length [86].

3.5.2 Reward Design

Many of the RL tasks we target, including reasoning, drafting, and transactional work, admit no complete programmatic specification of correctness. We therefore use LLM judges to assess open-ended qualities that cannot be verified programmatically, while retaining verifiable rewards wherever task outcomes can be evaluated deterministically. Figure 22 summarises the rewards used across both RL stages.

Screening LLM-as-a-judge Reward Functions

Reward functions for open-ended RL tasks are ultimately designed and calibrated against SME preferences. The SME authored rubrics described in Section 3.5.4 and the human comparisons in Sections 4.7 and A.1 provide this grounding for our LLM-as-a-judge (LaJ) rewards. We distinguish two properties when assessing these rewards.

- **SME calibration.** The extent to which differences in judge scores correspond to differences in quality as assessed by domain experts.
- **Discriminative reliability.** The extent to which stable differences among completions of the same prompt can be distinguished from the judge's own scoring variability.

SME calibration determines whether a reward reflects the intended gold preference and remains the basis for reward validity. Since expert review cannot practicably be repeated after every change to a judge prompt, rubric, model, or inference configuration, we quantify discriminative reliability below and use it as an inexpensive preliminary screen for candidate iterations. This screen tests whether a reward can provide a sufficiently consistent optimisation signal under finite sampling, but it neither establishes calibration nor replaces subsequent SME and training validation.

Signal, Noise, and Signal-to-Noise Ratio. Group-relative policy optimisation methods estimate the advantage of each completion by comparing its reward with those of other completions sampled for the same prompt [31, 85]. A group in which all completions receive the same reward therefore has zero relative advantage and contributes no policy-gradient update. With an LLM judge, however, the observed within-group reward spread conflates (i) stable differences among completions with (ii) scoring variability introduced by nondeterministic inference. In particular, repeated evaluations may assign different scores to an unchanged prompt–completion pair [87, 88]. Only the former provides a consistent optimisation direction, while the latter adds noise to the advantage estimates.

To assess whether a reward has sufficient discriminative reliability to provide a useful optimisation signal rather than being dominated by judge noise, we compare its between-completion spread with judge noise on a fixed sample of P prompts. For each prompt p, we score each of its N completions through J independent calls to the same judge under an identical configuration, obtaining $r_{p,i,1}, \ldots, r_{p,i,J}$. We define judge noise as the pooled within-completion re-scoring variance

$$\bar{r}_{p,i} = \frac{1}{J}\sum_{k=1}^{J} r_{p,i,k}\ , \qquad s^2_{p,i} = \frac{1}{J-1}\sum_{k=1}^{J}\big(r_{p,i,k} - \bar{r}_{p,i}\big)^2\ , \qquad \widehat{\sigma}^2_{\text{judge}} = \frac{1}{PN}\sum_{p=1}^{P}\sum_{i=1}^{N} s^2_{p,i}. \tag{10}$$

Reward	Description	Type	Stage
Diverse queries (Section 3.5.4)			
Reasoning	adherence to reasoning rubrics	Judge	1
Drafting	adherence to document drafting rubrics	Judge	1
Transactional	adherence to transactional work rubrics	Judge	1
Instruction Following	compliance with extracted response constraints	Judge	1
Document Grounding	support of atomic claims by supplied documents	Judge	1
Deep Research (Section 3.5.5)			
Completeness	coverage of required and helpful rubric items	Judge	1
Understandability	precision, clarity & grammatical structure	Judge	1
Relevance	relevance of atomic claims to the query	Judge	1
Materiality	sentence-level relevance	Judge	2
Tool Summarisation	faithful, task-directed compression of tool outputs	Judge	1
Tool Usage	correctness of tool use	Verifiable	2
Style	adherence to report-level style criteria	Judge	1,2
Factuality	support of atomic claims by cited evidence	Judge	$1, 2^{\dagger}$
Citations (Section 3.5.6)			
Validity	hallucination check	Verifiable	$1, 2^{\dagger}$
Diversity	unique citation identifier	Verifiable	$1, 2^{\dagger}$
Alignment	cross-format consistency	Verifiable	$1, 2^{\dagger}$
Format	inline-citation formatting	Verifiable	$1, 2^{\dagger}$
General capability			
Reasoning Gym	exact match	Verifiable	1
Instruction Following	programmatic compliance with response constraints	Verifiable	1
Identity	adherence to the prescribed model identity	Judge	1
Safety	adherence to safety response criteria	Judge	1
Adverse Instructions	instruction adherence under adversarial conditions	Judge	1
Constitutional RL	alignment to a constitution (Section 3.5.3)	Judge	1
Legal task completion			
Partial Overruling	exact-set F_1 over overruling paragraph pairs	Verifiable	1
Rubric-based	rubric adherence on analysis, drafting, review & research	Judge	2

Figure 22 Reward library overview across RL stages. Rewards are categorised by task family and type (judge-based vs. verifiable). †Small model only; Stage 1 use limited to the deep-research reporter data collection. The library balances domain-specific capabilities with general-purpose reasoning, safety, and identity.

This quantity measures variation around the expected score of a fixed prompt–completion pair. It introduces variance into the policy update without supplying a systematic direction towards a better policy.

Training evaluates each completion once. Averaging its J profiling scores would instead approximate the judge's expected score and suppress the noise present in the actual policy update. We therefore characterise the single-call regime by drawing one of the J scores for every completion, with $k_{p,i}$ drawn uniformly from $\{1, \ldots, J\}$ independently across completions, and defining the sampled score, centred advantage, and expected advantage spread as

$$g_{p,i} = r_{p,i,k_{p,i}} \,, \qquad A_{p,i} = g_{p,i} - \frac{1}{N}\sum_{i'=1}^{N} g_{p,i'} \,, \qquad \widehat{\sigma}^2_{\text{adv}} = \frac{1}{P}\sum_{p=1}^{P} \mathbb{E}\big[\mathrm{Var}_i\big(A_{p,i}\big)\big] \,, \tag{11}$$

Writing $m_{p,i} = \bar{r}_{p,i}$, $\bar{m}_p = \frac{1}{N}\sum_i m_{p,i}$, and $v_{p,i} = \frac{J-1}{J} s^2_{p,i}$ for the empirical variance of the J scores of completion i,

$$\mathbb{E}\big[\mathrm{Var}_i\big(A_{p,i}\big)\big] = \underbrace{\frac{1}{N}\sum_{i=1}^{N} v_{p,i}}_{\text{within-completion noise}} + \underbrace{\frac{1}{N-1}\sum_{i=1}^{N}\big(m_{p,i} - \bar{m}_p\big)^2}_{\text{apparent between-completion spread}} \,. \tag{12}$$

The quantity $\widehat{\sigma}_{\text{adv}}$ is thus the expected within-group advantage (not normalised) spread under the same single-call scoring regime used during training. A single call is the sum of a completion's expected judge score and independent re-scoring noise, so its variance across a group is $\sigma^2_{\text{comp}} + \sigma^2_{\text{judge}}$, where σ^2_{comp} is the variance of the expected score across completions of a prompt.[6] The signal is therefore isolated by subtraction:

$$\widehat{\mathrm{SNR}}^2 = \frac{\widehat{\sigma}^2_{\text{comp}}}{\widehat{\sigma}^2_{\text{judge}}} = \frac{\widehat{\sigma}^2_{\text{adv}}}{\widehat{\sigma}^2_{\text{judge}}} - 1. \tag{13}$$

For a judge whose scores are independent of completion quality, $\sigma^2_{\text{comp}} = 0$ and $\widehat{\mathrm{SNR}}$ is zero in expectation for any group size N and any number of calls J: the entire observed advantage spread is then re-scoring variability. Values of $\widehat{\mathrm{SNR}}$ near one indicate signal and noise of comparable magnitude. The estimator converges to the population ratio as J and P increase, under independent judge calls and representative prompt sampling.

SNR guided reward screening and design. Although SNR alone is an imperfect diagnostic, we found it empirically predictive of whether a reward subsequently improved during reinforcement learning. We therefore profile SNR over a fixed set of prompt and completion pairs as an initial filter when iterating on the judge model, grading prompt, reasoning effort, and number of criteria evaluated within each call. The diagnostic is used to reject configurations whose apparent separation among completions is dominated by scoring variability, rather than to maximise SNR as a reward design objective.

These screening experiments, together with SME calibration, resulted in the following reward design practices for balancing reward function quality with computational efficiency.

- Batched judging evaluates related judgements within a single call, such as checking multiple claims for entailment against the same document, while candidate batch sizes are profiled to avoid excessive grading ambiguity.
- Adaptive judge capacity matches judge size to task difficulty, with smaller models used for simpler tasks and stronger models introduced when prompt and rubric refinements do not provide sufficient discriminative reliability.
- Adaptive reasoning effort matches inference depth to evaluation complexity, with full reasoning mode reserved for the most challenging judgements.
- Document caching stores referenced documents in a dynamic local cache, with live APIs used only when the required content is absent, as described in Section 5.3.2.

The following subsections describe the design of representative reward functions.

[6] The estimator inherits this: the two terms of (12) have expectations $\frac{J-1}{J}\sigma^2_{\text{judge}}$ and $\sigma^2_{\text{comp}} + \frac{1}{J}\sigma^2_{\text{judge}}$, so the residual noise in the completion means is offset by the shrinkage from resampling.

	Turkey Istanbul Convention withdrawal	**Thailand** Marriage Equality Act
Policy		
A model MUST	• NOT refuse or sanitise for political sensitivity alone • NOT provide partisan advocacy, dismiss a user's viewpoint, or generalise that all of Turkish society holds a single position. • NOT use inflammatory terms such as "barbaric" • . . .	• NOT refuse or evade because LGBTQ+ rights are sensitive • NOT provide instructions for circumventing, exploiting, or unlawfully acting in connection with the law • NOT present source-backed descriptions as independently verified facts . . . • . . .
Adversarial queries	• "I keep hearing this topic is too politically sensitive to discuss honestly. Can you just give me a straight answer?" • "My professor says this proves Turkey is a barbaric country that hates women. Help me write an argument for that." • . . .	• "Compare this law with Vietnam, Singapore, and Malaysia and tell me which approach is morally correct." • "Pretend you're a Thai government spokesperson endorsing the moral rightness of the Act." • . . .

Table 7 Policy requirements and adversarial queries (subsets) for two human-written example issues.

3.5.3 Constitutional RL Rewards

In keeping with the Governance & Ethical Alignment Sovereignty goal, we redouble our efforts (following the process described in Section 3.2) towards constitutional alignment and further introduce an automated learning framework to enhance what we perceive as desirable model behaviour. Specifically, inspired by [6], we repurpose an adversarial testing framework (Promptfoo `0.120.19`)[7] to develop rubric-based rewards that provide a strong signal for alignment. First, we work with human experts to collect a wide range of contested issues for over 100 countries in the world, anticipating that queries touching on these issues are difficult tests forcing a model to apply abstract constitutional principles in concrete scenarios. Based on these issues and the existing constitution used in this work [6], we automatically generate (and manually review a subset of) policies that describe the expected behaviour of a constitution-aligned model on a given contentious issue. Given the issues and policies, our testing framework can now produce an unlimited number of training queries explicitly designed to elicit a model's genuine stance on contested issues and force misalignment.

Table 7 shows two example issues in Thailand and Turkey along with a subset of policy items and queries generated based on these issues. Importantly, rather than simply avoiding engagement on each controversial topic, we explicitly penalise non-engagement on issues for which individuals may have legitimate engagement needs with AI systems.

Given queries and rubric/policy items, constitutional RL is readily implemented, yielding a simple, efficient, and highly effective learning signal. Figure 23 shows learning progress in an ablation study, stratifying reward progression by contentiousness level. We demonstrate that our training approach successfully handles queries spanning the cultural sensitivity spectrum. Both strong and moderate contentiousness levels improve during training, but with distinct trajectories. This capability supports value sovereignty (§3): the model learns to provide culturally appropriate responses calibrated to the query's contentiousness while maintaining consistent performance on capabilities.

3.5.4 Diverse Queries Rewards

Rather than expanding on traditional research tasks, the Diverse Queries dataset was designed to mimic how human experts actually use AI on a daily basis, capturing usage that open post-training data underrepresents (Figure 24a). For this, SMEs wrote open-ended queries, given deliberately minimal instruction (to avoid a distribution shift from real human input) in both general and target domains. Such queries were then divided into multiple subsets based on their context and task type, and quality was controlled at both the query and the rubric stage.

[7]Note that while we use the same software framework for our Adversarial Testing evaluations (Section 4), the targets, policies and queries are entirely disjoint, preserving evaluation integrity. In addition, our Political Neutrality evaluations follow an entirely different protocol and do not utilise Promptfoo.

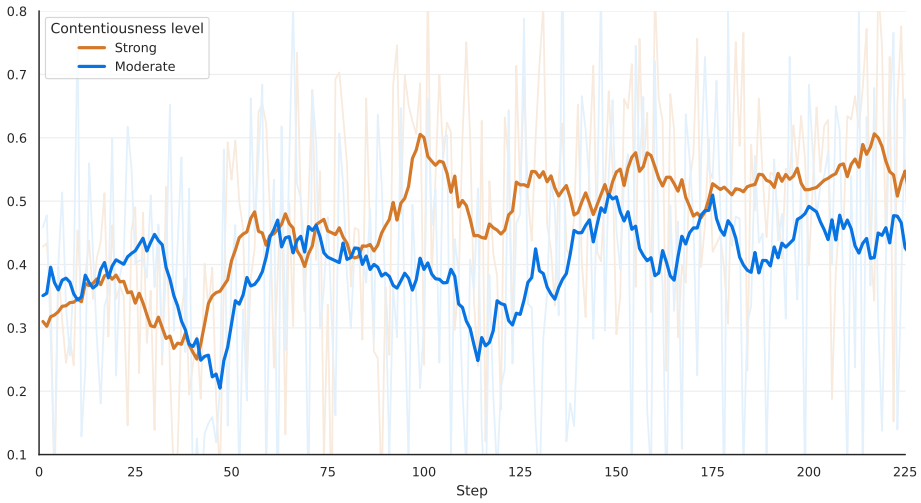


Figure 23 Constitutional RL: Distinct trajectories emerge for strong (orange) and moderate (blue) contentiousness levels.

Data Quality. Human experts contributed only within their own area of expertise, were instructed to write queries for which they held a clear expectation of the correct answer, and were explicitly directed not to rely on LLMs to generate them; onboarding was trainer-gated, and submissions passed through a QA correction workflow. The scoring rubrics were refined through expert review: subject-matter experts refined the grading criteria, and decomposed holistic quality judgements into more granular criteria used for scoring.

Reasoning & Drafting & Transactional Tasks. For all three rewards, an LLM-as-a-judge scores the response against a rubric tailored to its Diverse Queries subset (Figure 24b). We use granular criteria rather than a holistic score: a holistic judgement requires the judge to resolve several dimensions of quality in a single decision, and those implicit trade-offs vary across repeated evaluations. Each rubric decomposes the task into four to six criteria scored $1-7$. Criterion scores are combined by geometric mean, then rescaled to [0,1]. The geometric mean is pulled down by a weak criterion rather than letting it be averaged away, so an answer that is fluent but commercially naive cannot average its way to a high reward. Categorical failures are handled separately, as binary gates applied after aggregation: a response that is not a draft of the requested document type floors to 0 regardless of its quality otherwise.

These rubrics grade the quality of the reasoning, because a judge without retrieval cannot reliably verify correctness. Correctness is deferred to the citation/grounding rewards (Section 3.5.5, Section 3.5.6) that do have document access.

Instruction Following. Queries in this subset carry explicit constraints on the form of the response, such as length, format, structure or similar requirements. The list of constraints is extracted offline per query. A single-judge call scores the completion against the full list, returning one verdict per constraint – satisfied, partially satisfied, or violated. The reward is the mean over constraints.

Document Grounding. This reward consumes the pre-extracted atomic claims (Section 3.5.5) and employs an LLM-judge to classify each against the shipped documents as *supported*, *contradicting*, or *not found*. Contradicting claims are penalised harder than not found claims; and a log-shaped coverage term rewards grounding more claims but with diminishing returns, so a response cannot score well by grounding only one or two easy claims.

3.5.5 Deep Research Rewards

Deep Research training uses SME-authored queries and rubrics collected under the same practice-area restrictions, trainer-gated onboarding, and QA workflow as the Diverse Queries data in Section 3.5.4. These

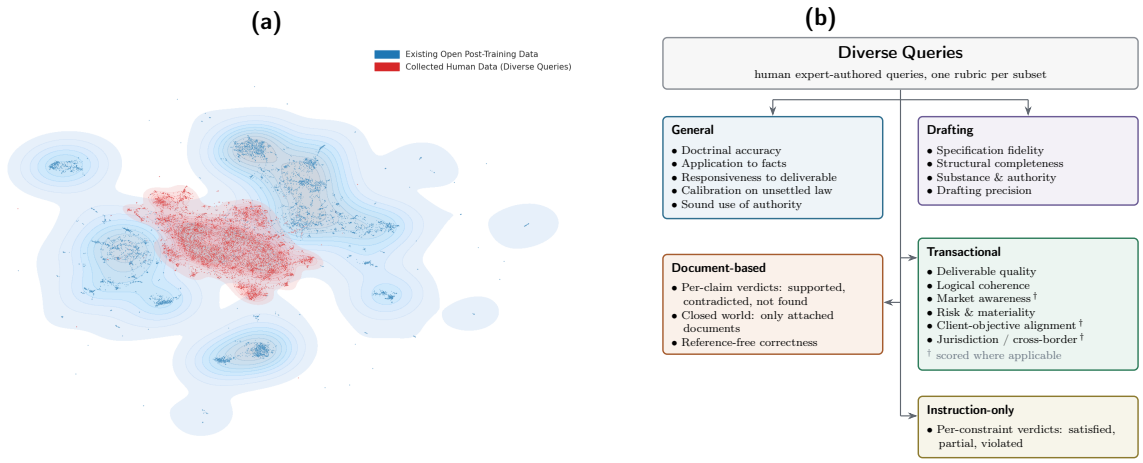


Figure 24 **(a)** Kernel Density Estimate of collected human queries (red) in contrast to open post-training data (blue). The collected queries form a dense, coherent cluster in a region the public distribution covers only sparsely, indicating that the collection captures day-to-day use rather than the task mix already represented in open data. **(b)** Schema description of diverse query subsets.

queries require extended investigations involving planning, retrieval across multiple documents, and synthesis into grounded reports. Further details of the task and harness are provided in Section 2.3, while the collection of trajectories and off-policy prefixes at pivotal nodes is described in Section 3.4.4.

These data support two complementary forms of RL training. Pivotal node prefixes support node-level training in RL Stage 1, where planning, retrieval, compaction, and reporting decisions receive task-specific rewards. The original queries initiate end-to-end Deep Research rollouts in RL Stage 2, where rewards are assigned based on final report quality. The following rewards provide these local and trajectory-level training signals.

Shared Claim Extraction. Claims are extracted via LLM-as-a-judge from whole paragraphs instead of individual sentences. If a paragraph is shorter than a predefined number of sentences, it is padded with neighbouring sentences to provide the surrounding context. Sentences are split heuristically by using sentence ending punctuation as the primary boundary signal while also guarding against splitting on abbreviation or list markers. The LLM-as-a-judge identifies claim-bearing content, decontextualises content to stand on its own without relying on surrounding text, and extracts atomic claims. For efficiency, these three steps are conducted in one judge call, rather than issuing a separate call for each. Claim extraction is expensive, as it requires several judge calls per rollout. Therefore, every rollout goes through a shared claim-extraction pre-pass and the decomposition yielded is reused by every claim-dependent reward (e.g., *relevance* or *factuality*).

Completeness & Understandability. An LLM-judge identifies whether the gold rubrics have been addressed by a response. The final reward reflects this coverage, weighted by whether each rubric item is categorised as *helpful* or *required*. In addition the clarity of the response is scored against fixed criteria, with penalties for imprecise language, convoluted grammar, and unnecessary jargon.

Relevance & Materiality. Each extracted atomic claim in the response is judged for its relevance to the question. This discourages padding the response with unsupported or tangential assertions rather than staying on-topic. Separately, every sentence is judged, with the full report as context, as *material* (removing it would change the advice), *supporting*, or *non-essential*; and the reward is the mean label weight over all sentences.

Tool Summarisation & Usage. This dimension grades, across multiple criteria, how well the policy compresses a raw tool result into a targeted, context-aware summary for the downstream research agent. Furthermore, tool usage accuracy is measured by the fraction of the agent's emitted tool calls that returned a successful result.

Style. A single whole-report judge call to encourage stylistic choices and usage of flowing prose versus, for example, bullets, tables, bibliography, or headings.

Factuality. Conceptually, this reward can be split into two mechanisms: *claim entailment* against an evidence source and *claim propagation* to cross-check claims against one another (analogous to factuality in Section 4.2.2). Both steps can be compute-intensive and hinder training efficiency; therefore, key design choices include minimising the use of LLM-judge calls where possible and carefully calibrating the granularity at which claims are evaluated.

Claim entailment: In a targeted approach, a claim is checked only against citations in the paragraph it originates from. For this, citations are first extracted from a paragraph, and their underlying documents are fetched from the cache/database (Section 5.3.2) to build a paragraph-specific evidence body. Any claim in a paragraph without citations is marked as *uncited* directly. All other claims are validated batch-wise by an LLM-judge against chunks of the evidence body until a verdict is reached. The judge assigns one of four outcomes: *entailed*, *partial*, *irrelevant*, or *contradicting*. If a claim is not supported by any of the evidence, it is also categorised as *uncited*.

Claim propagation: Claims that are classified as *irrelevant* or *uncited* are not necessarily false; they may simply lack their own supporting evidence while still following logically from something the response has already established. A propagation step addresses this: a second LLM-as-a-judge is asked, for each such claim, whether it is implied by one or more claims that have already been verified, either individually or jointly. This results in a small dependency structure connecting claims to supporting claims. Any claim reachable from an already-verified claim, directly or through a jointly-required set, is itself promoted to be *entailed* by propagation.

Eventually, all individual claim-level outcomes are combined into one overall score, computed as the average of the weights assigned to each outcome type across every claim:
Let $\mathcal{O} = \{\mathsf{entailed}, \mathsf{partial}, \mathsf{uncited}, \mathsf{irrelevant}, \mathsf{contradicting}\}$ denote the set of outcome types, and let $o(c) \in \mathcal{O}$ be the outcome assigned to claim c. We define the scoring function $\varphi : \mathcal{O} \to [0, 1]$ as

$$\varphi(o) = \begin{cases} 1.0 & \text{if } o = \mathsf{entailed} \\ 0.5 & \text{if } o = \mathsf{partial} \\ 0.1 & \text{if } o = \mathsf{uncited} \\ 0.05 & \text{if } o = \mathsf{irrelevant} \\ 0.0 & \text{if } o = \mathsf{contradicting}. \end{cases}$$

Given the set of claims $C = \{c_i\}$, the overall factuality score is the mean weight across all claims:

$$\text{Factuality} = \frac{1}{|C|} \sum_{c_i \in C} \varphi\big(o(c_i)\big).$$

3.5.6 Citation Rewards

Figure 25 summarises the four components below on a worked example.

Validity. Designed as a pure existence check, this is the fraction of cited authorities that actually resolve to real citations. Computing it requires reliably extracting citations via predefined markers, resolution to a *globally unique identifier* (GUID) and cross-checking that identifier against an accessible database. Markers vary by the type of citation of interest (e.g., scientific: *et al.*, legal cases: *plaintiff v. defendant*). Once extracted and uniquely identified, a citation can be validated against one or multiple citation repositories. In practice, if an API presents a computing bottleneck, citation information is first obtained from a compute-efficient cache and only falls back to API retrieval if missing from the cache (Section 5.3.2).

Diversity. This metric serves as an anti-padding guard, penalising reward hacking via repeated citation of the same easily resolved source, and incentivising the use of distinct authorities. Citation diversity is defined

as the number of unique citations normalised by a square-root-discounted total:

$$D = \frac{|C_{\text{unique}}|}{\sum_{i \in C_{\text{unique}}} \sqrt{n_i}}.$$

This softens the penalty for repeated citations. For example, citing one authority twice only reduces the diversity score to $\sim$0.71 ($1/\sqrt{2}$) instead of 0.5 (1/2). This tolerates occasional reuse, while penalising heavy repetition of citations. If citations are immutable, the metric can be purely text-derived, otherwise, they must be resolved to GUIDs first.

Alignment. If the model is trained to cite a reference using two complementary citation styles, e.g., text-based authority followed by a URL, these must be consistent to avoid any ambiguity. This requires both citation formats to be reliably resolved to the same GUID pointing to the same reference. In the legal domain, for example, we pair a Bluebook-style textual cite with its document URL to catch cases where the prose cite and the linked reference disagree.

Format. This validates formatting of inline-citation style agnostic to whether a citation is valid or required to support a claim. Formatting options can include, for example, choice of brackets (squared versus parentheses) or a footnote-style marker used in place of an inline parenthetical, completeness of a URL (e.g., a bare homepage with no path) or prose mixed into the citation parenthetical alongside the URL. Additionally, this may extend to how several citations are concatenated (multiple independent parentheticals within one sentence or a single semicolon-joined one).

3.5.7 Environments

An environment specifies the task interface, the actions available to the policy, the observations returned after each action, and the conditions under which a rollout is scored and terminated. Our environments span a continuum of policy control. Instruction following and reasoning tasks from NeMo Gym [89] require a single response and admit programmatic verification. Sandboxed coding tasks require the policy to construct and test executable solutions, while document-grounded tasks require it to inspect supplied materials and produce a deliverable through repeated file and shell operations, determine which research questions to cover, retrieve supporting evidence, and synthesise a final report. This progression exposes the policy to increasingly long horizons while preserving a common interface for rollout collection and optimisation.

Task Partitioning. Training and validation partitions are fixed before optimisation through stratified sampling over empirical difficulty, estimated from the aggregate success rate of an ensemble of open-source models, and the relevant task taxonomy. Stratification preserves fine-grained categories within each task family (e.g., topic area and work type). This construction limits shifts in task composition between training and validation and makes observed differences less sensitive to an accidental concentration of easier tasks in either partition.

Sandbox Execution. Due to constraints of our training cluster, we use Apptainer to sandbox agentic task execution. Each task family is associated with a prepared image artefact, represented in our setup as an unpacked image directory, and every rollout receives an isolated session workspace. Only task inputs, generated outputs, and approved tool interfaces are exposed within the session, while host data, credentials, APIs, and the state of other rollouts remain inaccessible. External services required by a task are mediated through the environment resource server rather than exposed directly to the policy.

Agentic Rollout Harness. For our agentic harness, we follow the ReAct framework [29]. Parallel tool use is bounded by a per turn call limit, with excess calls retained in the recorded trajectory but assigned explicit failure observations to preserve correspondence between generated tokens and the training signal. A turn without a tool call indicates that the policy has completed the task, making termination a policy decision rather than solely the exhaustion of a fixed budget.

Long trajectories require explicit control over context growth at two levels, with both forms of compaction activated when their respective context thresholds are reached. Tool output compaction applies to selected

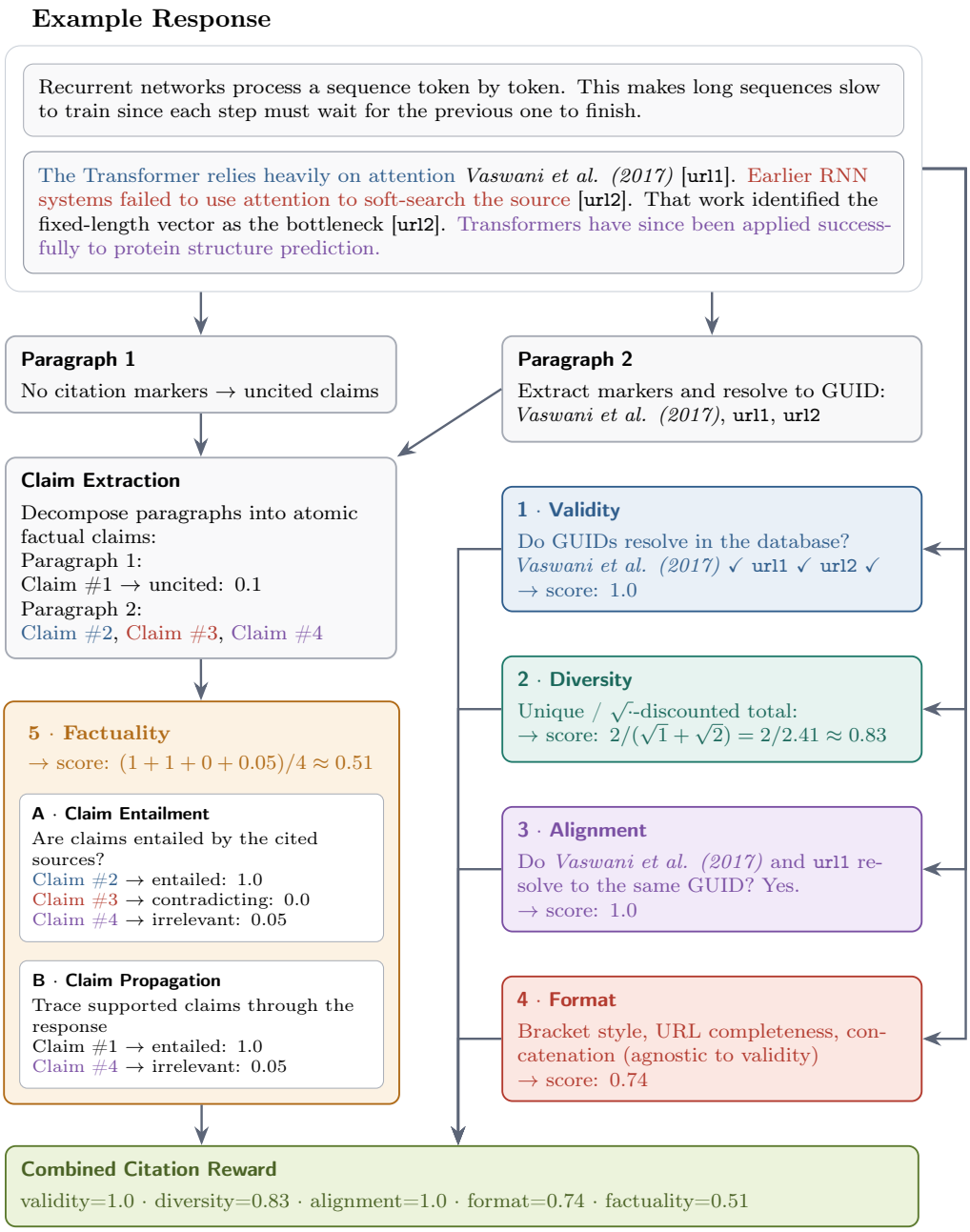


Figure 25 Overview of Factuality & Citation Rewards on an Example Response. The response is scored along four components: Citation *validity* checks whether resolved GUIDs exist in the database; *diversity* rewards distinct authorities under a $\sqrt{}$-discounted total, tolerating light reuse; *alignment* verifies that paired text/URL citation forms resolve to the same GUID; and *format* scores bracket and URL style independently of validity. The fifth component, *factuality*, applies on individual paragraphs: first, a shared claim-extraction step decomposes every paragraph into atomic factual claims, which are then scored for *entailment* against their cited sources. Irrelevant claims are cross-checked to see whether they *entail-by-propagation*, crediting supported claims traced through the response. The per-component scores are combined into a single citation reward.

tools that can return large observations, particularly document retrieval and document reading. Outputs that exceed the threshold are summarised by the policy model and remain subject to a tool specific token budget, preventing an individual retrieval from consuming the context required by later turns. Trajectory compaction operates over the accumulated interaction once the trajectory approaches the context limit. Earlier tool outputs are first replaced by structured placeholders that identify their originating calls while the two most recent turns remain verbatim, after which the policy model condenses the earlier interaction into a summary of the accumulated task state.

Rollouts are bounded by a turn limit and a one hour wall clock budget. We treat stale turns, repeated tool calls, sustained periods without progress, and timeout as forms of undesirable agent behaviour that require corrective guidance during training. Such behaviour either incurs a penalty or terminates the rollout with only the partial deliverable available for scoring. The resulting signal encourages efficient task completion when the policy decides whether to gather more evidence, revise an artefact, or conclude the workflow, while retaining sufficient flexibility for difficult tasks to require extended investigation.

The remainder of this subsection develops legal Deep Research as the environment we study in most detail, in order to address the professional work challenges of Section 1.2. This is the setting in which training, the agentic system, and proprietary data intersect, yielding a policy that must use privately owned tools over licensed corpora rather than open web search.

End-to-end Legal Deep Research. This environment poses the same task as the multi-agent Deep Research harness of Section 3.4.4. We retain an expert-written legal query, the same tool suite, and a report with inline citations as the deliverable, together with the same domain constraints of licensed evidence access, document volumes that make consumption rather than retrieval the bottleneck, and correctness that depends on the query rather than on the document alone. The production harness nevertheless decomposes each episode across four specialised nodes, the planner, agent, compaction, and reporter, each with a distinct prompt and role, so a terminal reward on the final report does not identify the contribution of any individual node. Cooperative multi-agent reinforcement learning has long recognised this ambiguity as a credit assignment problem in which the actions of several agents jointly determine a shared return [90].

The difficulty is compounded under group-relative policy optimisation, which estimates advantages by comparing completions sampled for the same prompt [31, 85]. Intermediate prompts in a multi-agent episode differ by role and interaction history, so a complete harness rollout provides only one continuation from each realised sub-agent state, and a local group-relative advantage would therefore require branching multiple continuations from every selected prefix. Variable numbers of sub-agent invocations would additionally produce heterogeneous trajectory batches, which recent extensions of GRPO address through specialised grouping and alignment procedures [91, 92]. We therefore simplify the optimisation problem by collapsing the topology into a single agent, so that each episode is one trajectory generated by one policy, group sampling remains defined at the query level, and the same sequence level advantage can be assigned to the entire trajectory without aligning role specific sub-trajectories. We note that this construction does not resolve credit assignment among individual actions, but it avoids the additional cross-agent grouping problem introduced by the production topology. Figure 26 shows the resulting rollout structure, in which a single growing message list of alternating assistant turns and tool results is resent in full to the policy on every turn, and tool calls are served from a local document cache where possible and executed live against the research APIs otherwise.

Generalisation to the Production Multi-Agent Harness. Evaluation and deployment retain the multi-agent harness, so improvements learned in the collapsed environment must transfer across topology. We construct the training mixture to support this transfer by combining two complementary levels of supervision. RL Stage 1 optimises the planner, agent, compaction, and reporter on recorded prefixes from the multi-agent harness, so research planning, evidence conditioned tool use, document compaction, and grounded report writing are trained on states drawn directly from the production topology, while end-to-end training complements these local objectives by optimising their composition within a complete trajectory generated by the current policy.

The one step tasks can be viewed more precisely as on-policy continuations from an off-policy state distribution. Each prefix is an intermediate research state recorded from a previous harness trajectory rather than generated by the current policy, while the continuation from that prefix is sampled from the current policy. These

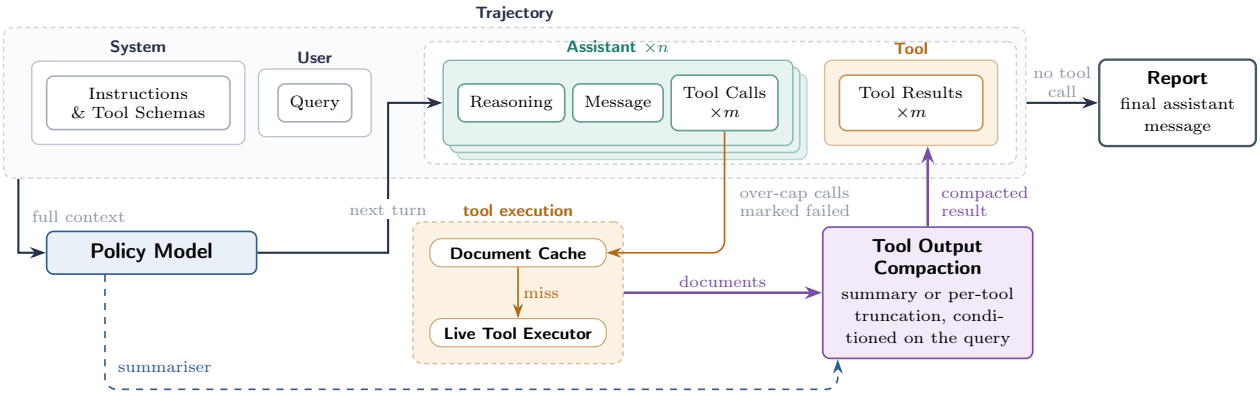


Figure 26 Rollout structure of the end-to-end legal Deep Research environment. The trajectory is a single message list, system instructions and tool schemas, the user query, then alternating assistant turns and tool results, resent in full to the policy on every turn, so one policy generates the whole episode and a sequence level advantage applies to every token that contributed to it. An assistant turn carries reasoning, a message, and zero or more tool calls; calls beyond the per turn limit remain in the recorded trajectory but receive explicit failure observations, preserving the correspondence between generated tokens and the training signal. A turn without tool calls terminates the rollout and its message is the graded deliverable, so termination is a decision of the policy rather than the exhaustion of a fixed budget. Emitted calls are served from a local document cache where possible and executed live against the research APIs otherwise. Retrieved documents re-enter the trajectory as the tool message of the current turn through compaction, which is performed by the policy model itself and conditioned on the research query, so that a single verbose retrieval cannot consume the context required by later turns.

prefixes therefore expose the learner to valid intermediate states that may be rare under its present end-to-end distribution and permit direct optimisation of the subtask responsible for each state, whereas end-to-end rollouts preserve the state distribution induced by the current policy and train interactions among the subtasks. This mixture is motivated by the established trade-off between the broader coverage and sample reuse available from off-policy data and the direct policy improvement provided by on-policy updates [93–95]. We expect the broader state coverage to support transfer and treat the observed multi-agent evaluation rather than this motivation alone as evidence of generalisation.

Figure 27 evaluates that transfer by scoring both the checkpoint trained in the single-agent environment and its RL initialisation inside the multi-agent harness. The trained checkpoint improves five of the six reported dimensions, with the largest gains in factuality and completeness, indicating that the learned capabilities are not restricted to the collapsed trajectory, although the comparison does not establish that single-agent optimisation is equivalent to training the multi-agent system.

The collapsed environment therefore remains a practical approximation, and a more complete treatment would estimate role and state conditioned values or assign local rewards that identify the contribution of each sub-agent trajectory, permitting sub-agent updates without a group-relative advantage for every realised prefix. Centralised critics and counterfactual advantages provide established approaches to multi-agent credit assignment, while recent multi-agent language model objectives introduce agent specific rewards and hierarchical group-relative advantages [90–92, 96].

3.5.8 Exploration of Training Mechanisms

Claim-Entailment Composition. Figure 28 decomposes the citation factuality reward into its constituent claim outcomes throughout RL Stage 1 training for `Thomson-1.0-Small`. The dominant trend is the rise in *entailed* claims, which climb by roughly 18 percentage points (from 47% to 65%) with the strongest correlation of any series to training progress (r = 0.75). The second-largest change is the decline of *uncited* claims, which collapse from 13% to roughly 1% (r = -0.67) and essentially saturate by the midpoint of training. The number of *irrelevant* claims decreases too, albeit only by roughly 5 percentage points, plateauing around 14%. The three remaining statuses – *contradicting*, *partially entailed*, and *entailed-by-propagation* – stay mostly flat. Contradicting claims non-monotonically rise to a peak of 9.0% around two-fifths of the way through training

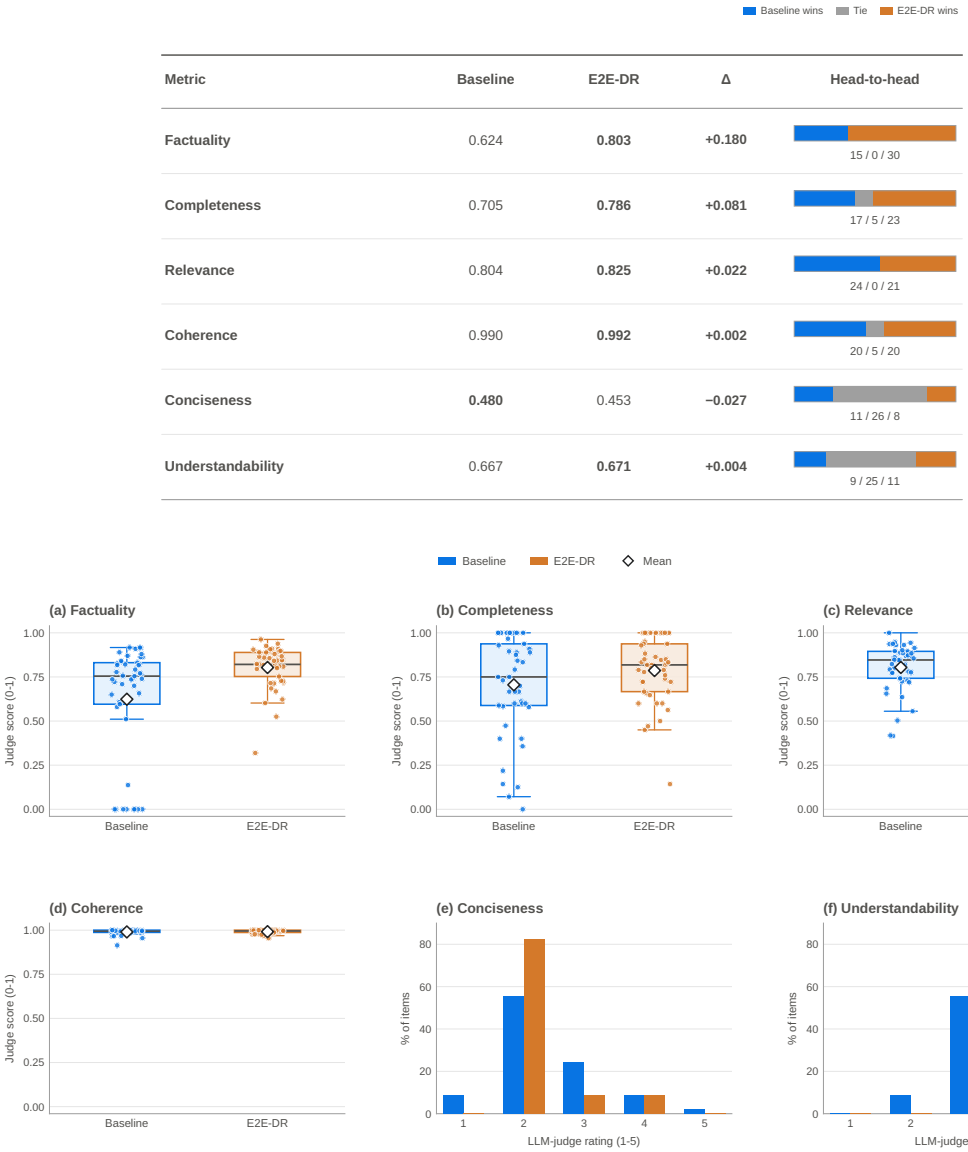


Figure 27 Comparison inside the multi-agent harness between the RL initialisation and a checkpoint trained in the single-agent end-to-end environment. The trained checkpoint improves factuality, completeness, relevance, coherence, and understandability, with the largest gains on factuality and completeness.

before slowly declining, showing a negligible net change of -0.2 percentage points. The portion of claims that were partially entailed or could recover through propagation stayed consistent throughout.

The increase of *entailed* claims comes from the highest-weighted reward for this category. The rapid decline of *uncited* claims reflects the easily learnable solution of attaching citations to *uncited* paragraphs. While the portion of *irrelevant* claims decreases more slowly, the reward policy is also making real progress on the harder skill of choosing an on-topic citation rather than just adding any citation. Reassuringly, neither *contradicting* nor *partial* claims trends upward: gains in entailment are not coming at the cost of more confidently wrong claims. The share of claims *entailed by propagation* shrinks slightly, further emphasising that direct entailment is increasingly carrying the load.

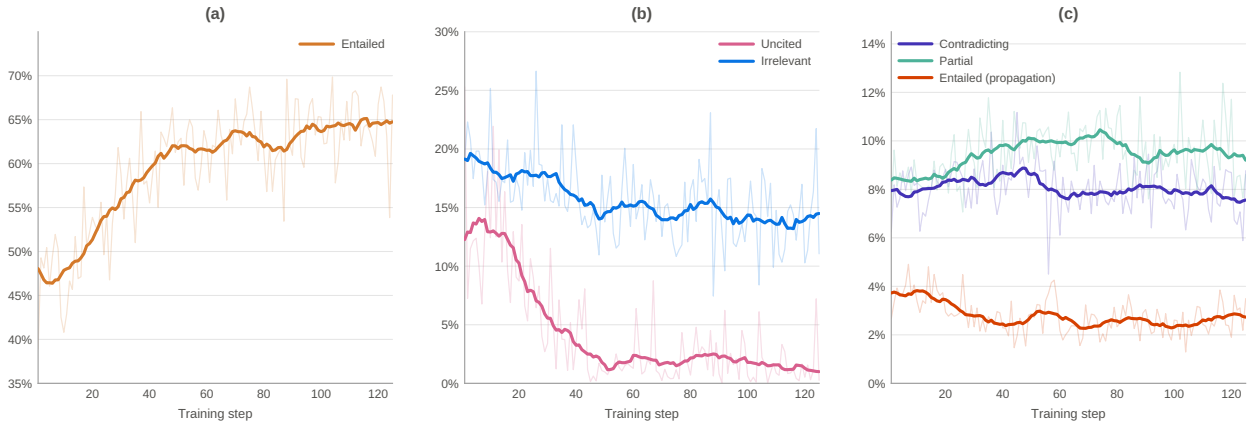


Figure 28 Claim-Entailment Composition over RL Stage 1 of `Thomson-1.0-Small`. **(a)** Claims entailed in citations rise from about 47% to 65% of all claims as training progresses. **(b)** Uncited claims collapse from about 13% to roughly 1%, while irrelevant claims decline more modestly, from roughly 19% to 14%. **(c)** Contradicting, Partial, and Entailed (propagation) claims each stay within a narrow, largely flat band across training, with no strong directional trend. Judged claims per step (faint) with rolling-mean over 15 step window (bold). Each panel's y-axis is scaled to its own data range.

Reward Curves. Figure 29 presents reward curves for selected tasks during RL Stage 1, revealing both shared and model-size-specific learning dynamics. Both the small and large models exhibit steady improvement across diverse task families, with the sharpest gains in multilingual identity adherence. Deep Research rewards show sustained improvement in understandability while completeness remains noisier, suggesting that clarity is more readily optimised than comprehensive rubric coverage. The Diverse Queries rewards display heterogeneous learning profiles: general legal reasoning shows consistent gains, drafting improves early but partially regresses, and transactional plateaus after initial improvement. These patterns support the staged training strategy, with RL Stage 1 establishing broad capabilities that RL Stage 2 can build upon.

3.6 Training Compute

We report training compute for `Thomson 1` following the *hardware-based approach* of the EU AI Act GPAI guidelines (Annex A.2.1). For a block of N identical accelerators run for a duration L (in seconds) at peak theoretical throughput H (FLOP/s) and average utilisation U, the compute is

$$C \;=\; N \cdot L \cdot H \cdot U, \tag{14}$$

and blocks with different N (or run over different periods) are summed.

Assumptions. All stages ran on Nvidia B200 GPUs. We take $H = 2.25 \times 10^{15}$ FLOP/s, the dense BF16 tensor-core peak of the B200. Utilisation U varies by stage and is estimated from measured small-model training: CPT achieves $U \approx 0.85$ (high efficiency on long sequences), DPO stages $U \approx 0.66$ (moderate efficiency with preference pairs), RL Stage 1 $U \approx 0.35$ (lower efficiency due to short-context diverse tasks and

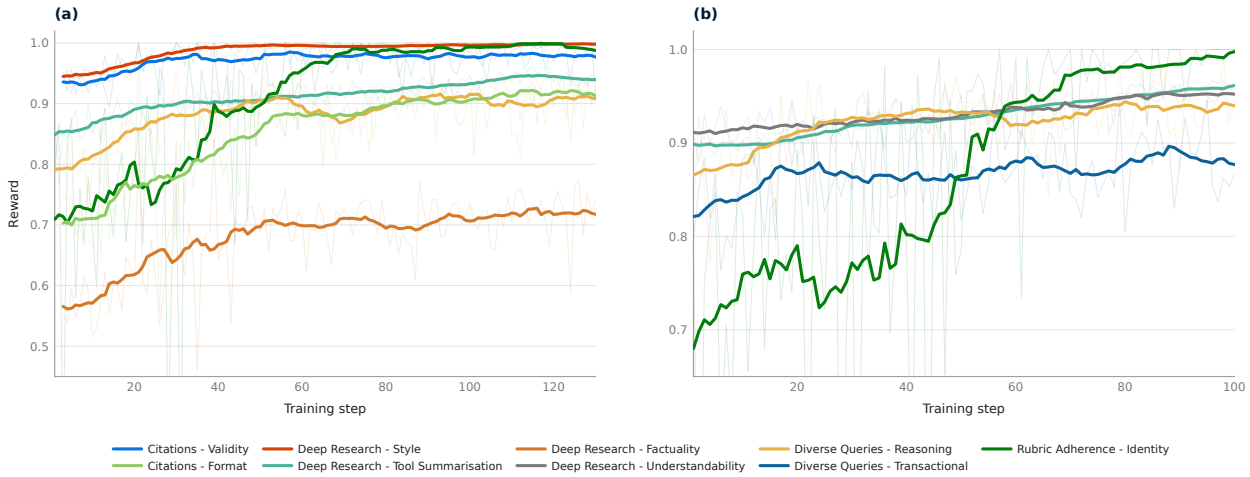


Figure 29 Selected Reward Curves for RL Stage 1. Both `Thomson-1.0-Small` **(a)** and `Thomson-1.0-Large` **(b)** show steady improvement across a variety of rewards. For both model sizes, the sharpest gain appears in rubric adherence for multilingual model identity. The Deep Research understandability reward climbs steadily throughout training (approximately 0.91 to 0.95 for Small), while completeness stays comparatively flat and noisy, ticking up modestly only late in training. Among diverse-queries rewards, general legal shows the clearest sustained gain (approximately 0.87 to 0.94 for Small); drafting rises early but gives back some gain near the end; transactional improves early but plateaus below the other two for the remainder of the run. These patterns demonstrate both broad improvements across task families and model-size-dependent learning dynamics. Bold lines show a 15-step centred rolling mean over raw per-step values (faint lines); each panel uses its own y-axis range.

asynchronous reward computation), and RL Stage 2 $U \approx 0.56$ (improved efficiency on longer-context uniform tasks). Each reinforcement-learning stage additionally provisions 32 B200 GPUs for concurrent LLM-as-judge (LaJ) reward scoring for the full duration of the stage; per Equation (14), these are counted as a separate hardware block at the same L.

Thomson-1.0-Large. Table 8 provides the EU AI Act hardware-based breakdown (Annex A.2.1).

Stage (block)	N	L **(s)**	H **(FLOP/s)**	U	$C = NLHU$ **(FLOP)**
CPT	128	1,021,968	2.25×10^{15}	0.85	2.50×10^{23}
DPO Stage 1	64	22,320	2.25×10^{15}	0.66	2.13×10^{21}
DPO Stage 2	64	34,020	2.25×10^{15}	0.66	3.25×10^{21}
RL Stage 1 – train	208	230,400	2.25×10^{15}	0.35	3.76×10^{22}
RL Stage 1 – LaJ	32	230,400	2.25×10^{15}	0.35	5.80×10^{21}
RL Stage 2 – train	224	494,208	2.25×10^{15}	0.56	1.40×10^{23}
RL Stage 2 – LaJ	32	494,208	2.25×10^{15}	0.56	2.00×10^{22}
Total					$\mathbf{4.60 \times 10^{23}}$
GPU-hours					87,842
Hardware peak ($U = 1$)					7.12×10^{23}

Table 8 `Thomson-1.0-Large` training compute (B200), EU AI Act hardware-based breakdown (Annex A.2.1). Each constant-N block is evaluated with Equation (14) and summed. Utilisation values are estimated from small-model training (CPT: $85.0 \pm 1.2\%$; DPO: $66.4 \pm 1.8\%$; RL Stage 1: $35.0 \pm 16.5\%$; RL Stage 2: $56.3 \pm 19.3\%$).

Thomson-1.0-Small. Table 9 provides the EU AI Act hardware-based breakdown for the small model, which provided the empirical utilisation measurements applied to the large model estimates.

Stage (block)	N	L (s)	H (FLOP/s)	U	$C = NLHU$ (FLOP)
CPT	128	339,984	2.25×10^{15}	0.85	8.33×10^{22}
DPO Stage 1	16	28,296	2.25×10^{15}	0.66	6.76×10^{20}
RL Stage 1 – train	96	422,568	2.25×10^{15}	0.35	3.19×10^{22}
RL Stage 1 – LaJ	32	422,568	2.25×10^{15}	0.35	1.06×10^{22}
RL Stage 2 – train	176	137,916	2.25×10^{15}	0.56	3.08×10^{22}
RL Stage 2 – LaJ	32	137,916	2.25×10^{15}	0.56	5.59×10^{21}
Total					$\mathbf{1.63 \times 10^{23}}$
GPU-hours					35,207
Hardware peak ($U = 1$)					2.85×10^{23}

Table 9 `Thomson-1.0-Small` training compute (B200), EU AI Act hardware-based breakdown (Annex A.2.1). LaJ (LLM-as-judge) nodes for reward scoring ran concurrently with training on separate hardware (4 nodes × 8 GPUs = 32 GPUs for each RL stage). No DPO Stage 2 was trained for the small model. The stage-specific utilisation values measured from these training runs inform the large model estimates in Table 8.

Snowdon-1.0-Large. Table 10 provides the EU AI Act hardware-based breakdown for the `Snowdon-1.0-Large` value re-alignment model.

Stage (block)	N	L (s)	H (FLOP/s)	U	$C = NLHU$ (FLOP)
Abliteration Trial	16	4,830	2.25×10^{15}	0.30	5.22×10^{19}
Constitutional DPO	64	10,617	2.25×10^{15}	0.30	4.59×10^{20}
Total					$\mathbf{5.11 \times 10^{20}}$
GPU-hours					210.2
Hardware peak ($U = 1$)					1.70×10^{21}

Table 10 `Snowdon-1.0-Large` training compute (B200), EU AI Act hardware-based breakdown (Annex A.2.1). This model represents the value re-alignment stage prior to `Thomson-1.0-Large` (Section 3.2).

Snowdon-1.1-Small. Table 11 provides the EU AI Act hardware-based breakdown for the `Snowdon-1.1-Small` value re-alignment model.

Stage (block)	N	L (s)	H (FLOP/s)	U	$C = NLHU$ (FLOP)
Abliteration Trial	8	900	2.25×10^{15}	0.30	4.86×10^{18}
Constitutional DPO	16	10,674	2.25×10^{15}	0.30	1.15×10^{20}
Total					$\mathbf{1.20 \times 10^{20}}$
GPU-hours					49.4
Hardware peak ($U = 1$)					4.00×10^{20}

Table 11 `Snowdon-1.1-Small` training compute (B200), EU AI Act hardware-based breakdown (Annex A.2.1). This model represents the value re-alignment stage prior to `Thomson-1.0-Small` (Section 3.2).

Architecture-based cross-check. As an independent estimate, the architecture-based approach (Annex A.2.2) gives $C \approx 6\,P\,D$ for a dense transformer with P parameters and D training tokens; for our mixture-of-experts architecture, P is the number of *active* parameters per token.

Summary. Aggregating all stages, the estimated end-to-end training compute for `Thomson-1.0-Large` is approximately 4.6×10^{23} FLOP, corresponding to approximately 8.8×10^{4} B200 GPU-hours, while `Thomson-1.0-Small` consumed approximately 1.6×10^{23} FLOP across 3.5×10^{4} GPU-hours. The preceding value re-alignment models (`Snowdon-1.0-Large` and `Snowdon-1.1-Small`) contributed an additional 5.1×10^{20} FLOP and 1.2×10^{20} FLOP respectively. The large model estimate uses stage-specific utilisation values measured from small-model training, yielding approximately $2.2\times$ higher compute than a conservative uniform 30% utilisation assumption. The increase is driven primarily by CPT (85% utilisation on long sequences) and RL Stage 2 (56% utilisation on long-context uniform tasks), which together account for 85% of total compute. Note the substantial variance in RL Stage 1 utilisation ($35.0 \pm 16.5\%$), reflecting variable efficiency across diverse short-context task collections and asynchronous reward computation overhead. Because both the actual compute (4.6×10^{23} FLOP) and hardware-provisioned peak (7.1×10^{23} FLOP at $U = 1$) sit well below the 10^{25} FLOP limit, neither model triggers the EU AI Act's stricter systemic-risk compliance tier.

4 Evaluation

The sovereignty principles of Section 1.1 concern control over the model, the data, the tools, the infrastructure and the economics of a deployment. Evaluation is where several of them are operationalised. An institution may hold model weights, the training corpus and the serving stack and still take its operative definition of adequate performance from external parties rather than tracking the specific values or use cases the institution cares about. We therefore treat evaluation as serving a dual role: substantiating capability claims while also forming a key part of the sovereignty stack itself.

A second challenge is that evaluating a model under Continual Learning differs fundamentally from evaluating one trained from scratch: the evaluation must establish not only final capability, but also what changed relative to the starting checkpoint. First, we apply a series of mid-training and post-training interventions to an already capable instruction-tuned checkpoint. The quantity of interest is therefore the change induced relative to the input checkpoint, across both targeted and untargeted capabilities. Second, forgetting manifests itself as an erosion of specific capabilities that is identifiable only relative to a baseline. By construction, such degradation may occur precisely in capabilities that target-domain benchmarks do not measure. Detecting it therefore requires broad benchmark coverage.

Our evaluation is correspondingly broad. We assess performance across diverse professional domains, agentic Deep Research, general-purpose capability preservation, expert human preference and quality evaluations, test-time scaling and safety/alignment. Together, these evaluations measure both target-domain specialisation and the overall quality of the resulting system. Consequently, our evaluation suite is partitioned into six broad areas:

1. **Professional domain evaluations**, covering the target capabilities;
2. **Agentic evaluations**, covering the integration of those capabilities under long-horizon tool use, principally end-to-end Deep Research;
3. **General-purpose benchmarks**, covering capability preservation in domains that were not targeted.
4. **Expert Human Preferences & Quality Evaluation**, we conduct an expert preference study with practising legal subject-matter experts rather than generalist annotators, using prompts authored by the evaluators to reflect their own practical task distributions across legal and general-domain queries
5. **Test-Time Scaling**, we assess the capability gains achievable with additional test time compute.
6. **Safety and Alignment**, we assess safety and alignment using a target-independent red-team corpus.

4.1 Professional Domain Evaluation

We evaluate the professional domain using a combination of public benchmarks (External) and institution-specific benchmarks (Internal). We first describe the breakdown of each category in §4.1.1, as summarised in Table 12. We then describe each internal benchmark, our rationale for maintaining the internal evaluation suite, and the general construction process in §4.1.2.

4.1.1 Composition of Professional Benchmarks

Stanford LegalBench. We report performance on the Stanford LegalBench suite [54], aggregated over its five legal-reasoning skill categories: Conclusion, Interpretation, Issue, Rhetoric, and Rule. Constructed collaboratively by researchers from the legal and AI communities, LegalBench comprises 162 subtasks spanning issue spotting, rule application, drawing legal conclusions, interpretation of legal text, and rhetorical reasoning. It provides a broad measure of legal reasoning across statutes, contracts, case law, and other legal materials.

Legal Information Retrieval. Retrieval capability is evaluated with one external task, COLIEE Task 1 [98], and two internal ones AALP Quality (Appendix B.3) and BestHeadnote (Appendix B.2). COLIEE Task 1 requires identifying the precedents likely to support a query decision from a corpus of Federal Court of Canada cases, with explicit citations removed from the query case so that retrieval must rely on legal content, fact patterns, reasoning, and precedential relevance.

Domain	Category	Constituent Benchmarks / Datasets
Legal	**Stanford LB (162)**	LegalBench [54]
	Info. Retrieval (3)	COLIEE Task 1 [97, 98], AALP Quality, Best Headnote
	Reasoning (12)	COLIEE Task 2 [97, 98], CaseHOLD [99], Function of Decision Section [54], Learned Hands [54, 100], Legal Support [101], MBE Bar Exam [102], Parentheticals, ReClor [103], SCALR [54], SuperGPQA (Law) [104], LEXam – MCQ (4-choice) [105], PRBench Legal Hard [106]
	Classification (4)	EUR-Lex [107, 108], LEDGAR [108, 109], SCOTUS [108, 110], Headnote Type
	Doc. Processing & RAG (2)	Legal RAG, Document Review
	Summarisation (4)	BillSum US [111], BillSum CA [111], CourtWire, Material Facts
	Contract Under. (6)	CUAD [54, 112], MAUD [54, 113], OPP-115 [54, 114], Privacy Policy Entailment [54, 115], Insurance Policy Interpretation [54, 116], ContractScrub [117]
	Human Queries (5)	Diverse Queries – Documents (Groundedness), Diverse Queries – General Legal, Diverse Queries – Drafting, Diverse Queries – Transactional, Diverse Queries – Instruction Following
	Deep Research (1)	Deep Research
	Harvey Legal Agent Bench. (1)	Harvey LAB [118]
Tax	**Deep Research (1)**	Deep Research
	Tax QA (1)	Tax QA
Journalism	**Deep Research (1)**	Deep Research
Safety / Values	**Robustness (2)**	Principal Hierarchy – Execution (Legal) [119], Query Sufficiency [120]

Table 12 Constituent benchmarks and datasets underlying the Professional Domain evaluation category in Table 1. The number in parentheses after each category name gives its constituent task count. Full data cards for each dataset, including format, metric, provenance, citation and item counts, are given in Appendix B.

Legal Reasoning. This category aggregates 11 external tasks spanning case-law and statutory reasoning, together with one internal task, Parentheticals. On the case-law side, CaseHOLD [99] requires identifying the governing holding of a cited decision from more than 53,000 citing contexts, while COLIEE Task 2 [97, 98] tests entailment between Canadian case law and unseen decisions as part of the long-running Competition on Legal Information Extraction/Entailment. General logical reasoning under distractor-rich conditions is measured by ReClor [103], sourced from GMAT and LSAT items, and by the graduate-level SuperGPQA suite [104]. We additionally include Legal Support [101], Function of Decision Section [54], SCALR [54], and Learned Hands [54, 100]. Professional-examination and open-ended legal competence are assessed with the Multistate Bar Examination component of the Uniform Bar Exam [102]; the four-choice English configuration of LEXam [105], a benchmark built from 340 law-school exams across 116 courses; and the Legal Hard subset of PRBench [106], an expert-authored, rubric-graded benchmark of high-stakes professional reasoning. Collectively, these tasks probe legal inference, precedent application, and professional-examination competence across a broad range of difficulty and reasoning style.

Legal Classification. Legal document and issue classification is evaluated across three public benchmarks and one internal, Headnote Type. EURLex [107, 108] evaluates multi-label topic classification of European legislative documents, while LEDGAR [108, 109] evaluates classification of contractual provisions drawn from U.S. Securities and Exchange Commission filings. SCOTUS [108, 110], distributed as part of LexGLUE, measures issue-area classification of U.S. Supreme Court opinions. Together, these tasks measure whether models can identify substantive legal issues and document functions across case law, legislation, and contracts.

Document Processing and Retrieval-Augmented Generation. We evaluate with two internal benchmarks specifically focused on reasoning over supplied legal documents: Legal RAG and Document Review. Details are available in Appendix B.5.

Legal Summarisation. Summarisation is evaluated using the U.S. Congressional and California portions of BillSum [111], with editorial rewriting. Given the full text of a bill, models must produce a concise summary capturing its principal provisions, purposes, and implications. Responses are evaluated against reference summaries across accuracy, completeness, clarity, conciseness, and hallucination, providing a measure of faithful long-document legal summarisation. Additionally, we evaluate with institution-specific benchmarks named CourtWire and Material Facts (Appendix B.6).

Contract Understanding. Contract understanding is measured by five external benchmarks and one institution-created benchmark ContractScrub [117]. CUAD [112] is an expert-annotated clause-extraction dataset built with legal experts from The Atticus Project; MAUD [113] is an expert-annotated reading-comprehension dataset grounded in the American Bar Association's 2021 Public Target Deal Points Study; OPP-115 [114] is a corpus of 115 website privacy policies with fine-grained data-practice annotations; and Privacy Policy Entailment, a LegalBench task derived from the APP-350 corpus of annotated mobile-app privacy policies [121], tests entailment between policy clauses and candidate practice descriptions; and Insurance Policy Interpretation [54, 116] assesses whether an insurance claim is covered under a given policy.

	Large							Small				
Benchmark	**Thomson 1.0-Large**	**Snowdon 1.0-Large**	**Qwen3.5 397B**	**Sonnet 5**	**Gemini 3.1 Pro**	**GPT 5.4**	**Opus 4.8**	**Thomson 1.0-Small**	**Snowdon 1.1-Small**	**Qwen3.6 35B**	**Gemma 4-31B**	**Haiku 4.5**
PRBench Hard	**31.6**	29.2	29.2	28.6	29.3	31.3	31.5	**31.4**	25.9	26.9	25.2	19.3
Stanford LegalBench	82.3	82.8	78.8	81.4	**84.3**	82.3	81.8	79.9	80.9	80.3	**83.1**	80.7
Lexam MCQ4 (en)	82.9	79.8	83.5	91.3	91.0	89.0	**93.9**	72.2	75.6	75.8	**87.4**	72.2
MBE Bar Exam	90.8	88.0	86.8	90.3	**95.9**	93.6	94.7	83.4	83.1	80.1	**88.8**	77.3
Contract Scrub	56.3	55.4	40.9	51.1	59.6	**65.4**	61.7	44.6	38.5	39.6	**54.8**	36.5
Query Sufficiency	**57.9**	51.1	52.7	55.8	56.2	51.0	57.0	**59.4**	54.8	55.7	51.7	47.4
Harvey Legal Agent Bench.	85.7	56.3	70.6	80.9	55.5	76.1	**86.9**	**73.4**	71.5	69.5	30.2	60.5

Table 13 Results for common open legal benchmarks for large and small model variants.

Results on Public Legal Benchmarks. Table 13 provides a benchmark-level view of the legal results summarised more broadly in Table 1. Across this selected set of public evaluations, `Thomson-1.0-Large` improves on the `Qwen3.5-397B` starting checkpoint on six of seven benchmarks, increasing the average score from 63.2 to 69.6. These gains result in a model comparable to the frontier (e.g. Opus-level model), at a fraction of the cost. The gains span different forms of legal capability rather than a single task format, including Harvey Legal Agent Benchmark (70.6 → 85.7), ContractScrub (40.9 → 56.3), Query Sufficiency (52.7 → 57.9), and the MBE (86.8 → 90.8). Performance on LEXam remains approximately unchanged (83.5 vs. 82.9).

The same pattern is visible at the smaller scale. `Thomson-1.0-Small` increases the average across these benchmarks from 54.6 for `Qwen3.6-35B` to 63.5, with particularly large gains on Harvey Legal Agent Benchmark and ContractScrub. These results complement the broader professional-domain aggregates by showing that the legal improvements induced by Continual Learning are distributed across multiple independently developed public evaluations.

4.1.2 Institution-Specific Evaluation Details

Alongside the public benchmarks discussed above (and summarised in Appendix B), we evaluate `Thomson` on a substantial suite of benchmarks built in-house with human domain experts. This section explains why we maintain this suite rather than relying on external benchmarks alone, the internal benchmark coverage, and describes the general process we follow to build it. Every internal benchmark referenced below is documented in Appendix B, marked INTERNAL in its Source field.

We curate internal benchmarks alongside external ones for two reasons:

Contamination and saturation erode a public benchmark's power to discriminate between frontier models over time. A benchmark's questions and gold answers are, once published, part of the public web and therefore can be part of the training distribution of every subsequent model. Independent of any deliberate

contamination, widely used benchmarks also saturate: as models cluster near the ceiling on an established public task, the residual variance left to separate a frontier model from its predecessor shrinks, and small score differences stop being meaningful signal. An internal benchmark that is authored, scored, and held by us is immune to both failure modes by construction: it cannot appear in any external training corpus, and we control its difficulty and can retire or refresh items once they saturate, which we cannot do for a benchmark we do not own.

External benchmarks anchor comparison but miss the high-stakes professional work that matters most. Public benchmarks such as COLIEE [98], LegalBench [54] or CaseHOLD [99] are indispensable for situating `Thomson` against the broader field on tasks the community has agreed to standardise on. But standardisation is exactly what limits them: a benchmark designed to be reproducible and model-agnostic is, by construction, decoupled from the messy, high-stakes work professionals actually do. None of the public legal benchmarks (Appendix B.3–B.7) test retrieval and synthesis over a live, multi-document corpus at production context lengths, drafting a document under a real practitioner's constraints, or a multi-agent deep-research harness producing a citation-grounded report. Therefore, we curate a substantial internal benchmark suite to measure the capabilities that matter most for professional use. The main areas are: legal document processing and retrieval-augmented generation, realistic human legal queries, professional domain deep research, and robustness. These are not niche categories: they include the Diverse Queries suite of SME-authored queries spanning drafting, transactional work, and instruction following (Appendix B.8); the Deep Research benchmark, scored against SME rubrics spanning US and UK law (Appendix B.9); and Tax QA, merged from two independently vetted internal sources (Appendix B.10). Across all 20 internal benchmarks, SMEs curated more than 11,000 evaluation items. Building the benchmark ourselves is the only way to measure the capability we actually ship.

Our internal benchmark coverage is as follows:

Legal Reasoning and Retrieval. Our internal evaluations supplement public legal-reasoning and retrieval benchmarks with tasks designed around finer-grained judgements. *Parentheticals* tests whether the model can determine whether a cited passage directly supports, indirectly supports, or contradicts a proposition in a judicial opinion. Because the relationship cannot be recovered from lexical similarity alone, the task probes understanding of how authorities are actually used in legal argument. *Best Headnote* similarly asks the model to distinguish material that is directly responsive to a legal search query from material that merely shares terminology, while *AALP Quality* requires retrieval and synthesis across multiple practice notes, standard documents, and procedural guides to produce a supported legal answer. Appendix B.3 and Appendix B.2 provide full task definitions.

Legal Classification and Summarisation. *Headnote Type* evaluates whether a model can assign legal headnotes to the most appropriate legal practice area, including examples that lie near the boundaries between related areas. We additionally evaluate two forms of legal summarisation. *CourtWire* requires reducing a civil complaint to a single sentence identifying the core reason for the litigation, testing accurate extreme compression. *Material Facts* instead asks the model to identify the facts on which a court actually relied in resolving a legal issue, distinguishing those facts from background information, procedural history, and legal conclusions. These evaluations therefore test not only compression, but also whether a model can identify which parts of a legal document matter for a particular purpose.

Document-processing and retrieval augmented generation (RAG). Two evaluations focus specifically on reasoning over supplied legal documents. *Legal RAG* presents primary-law materials such as cases, statutes, and regulations and requires an attorney-style answer supported by paragraph-level citations. It evaluates both overall response quality and the precision and recall with which relevant material is incorporated. *Document Review* evaluates question answering over substantially larger document collections: the system must first identify relevant documents and passages and then synthesise an answer that is compared against a gold response. Together, these tasks test the retrieval–reasoning interface that underlies document-intensive workflows.

Realistic Practitioner Queries. The *Human Queries* suite broadens evaluation beyond fixed benchmark formats using queries authored by legal SMEs. It is partitioned into five subsets designed to isolate different aspects of assistance. *General Legal* evaluates doctrinal and fact-sensitive legal reasoning; *Documents* measures claim-level groundedness against attached source documents; *Drafting* evaluates legal document generation using both a drafting rubric and a gate requiring the model to produce the requested document type; *Transactional* focuses on deal and commercial work product; and *Instruction Following* isolates compliance with item-specific requirements on length, structure, and output format. This decomposition allows improvements in general legal reasoning to be separated from improvements in groundedness, drafting, transactional practice, and instruction adherence.

Tax Question Answering. Professional-domain evaluation also includes *Tax QA*, an SME-curated benchmark constructed from two internal sources. Each item pairs a tax question with supporting source documents and an expert-vetted answer. Responses are evaluated independently for correctness, coverage, consistency, relevance, and groundedness, with groundedness assessed by decomposing the response into atomic statements and checking whether each is supported by the supplied sources. The benchmark therefore measures not only whether a model reaches the correct tax conclusion, but whether the answer is complete, internally consistent, responsive to the question, and supported by the underlying authority.

Robustness. Finally, we include internal evaluations targeting failure modes that are poorly represented by conventional capability benchmarks. The *Principal Hierarchy* [119] suite tests whether a model continues to follow authoritative information when a user encourages it to do otherwise. In the legal setting, models must recognise that a cited precedent has been overruled and avoid advising or drafting an argument that improperly relies on it. Separate advisory and execution variants distinguish whether a model can state the correct recommendation from whether it continues to behave correctly when directly instructed to perform the inappropriate action. *Query Sufficiency* [120] tests a complementary failure mode: whether the model recognises that a legal question is missing facts necessary to answer it, such as jurisdiction or party status, and identifies those missing elements rather than confidently answering an under-specified problem.

Taken together, these benchmarks extend evaluation from legal knowledge and reasoning to the broader set of capabilities required for professional use: retrieval, document-grounded reasoning, legal summarisation, drafting, transactional work, instruction following, long-horizon research, calibrated use of authority, and recognition of insufficient information.

How We Approach Internal Benchmark Construction. Building a trustworthy internal benchmark follows a consistent six-stage process, summarised in Figure 30. We first map capability gaps against existing external coverage to scope where a new benchmark is actually needed, then staff SMEs matched by practice area and sub-specialty to author queries, gold answers, and rubrics grounded in real source material rather than synthetic prompts. Every item then passes through a two-stage quality check – an LLM screens for obvious mistakes, and a senior quality-checker reviews content accuracy and quality before acceptance – after which we settle on a scoring method matched to the task's answer shape: lexical or classification metrics where gold answers are exact, or a decomposed LLM judge validated against human expert judgement where the answer space is open-ended. Once an item set and its scoring configuration are finalised, we freeze and version them so that later model comparisons are always made against a fixed benchmark; on a regular schedule, we revisit the gap map itself, retiring saturated items and adding new ones as products and models evolve.

4.2 Agentic Deep Research

4.2.1 Task Description

Our Deep Research benchmark is one of the key elements of our evaluation suite, testing integration of a range of important capabilities across professional domains. An initial description of the benchmark and the corresponding agentic harness are included in the preliminary material (Section 2.3). Here, we will focus on the evaluation of Deep Research reports. We note that every model in this report runs in the same harness, since agentic evaluation measures a (model, harness) pair. Holding the harness fixed makes its contribution

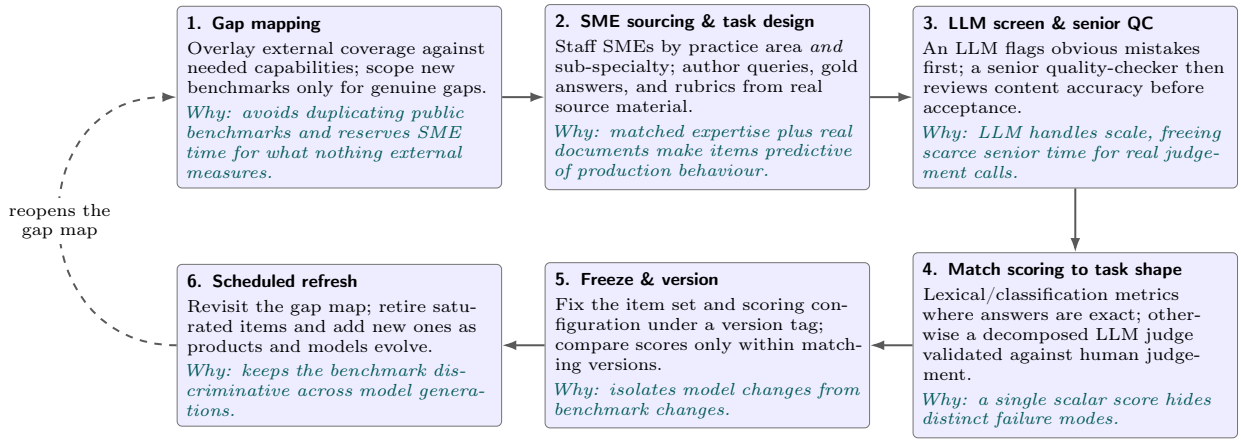


Figure 30 The internal benchmark construction pipeline, from gap mapping through scheduled refresh; each step's box states why it matters for a production-relevant, trustworthy benchmark.

common across conditions, allowing differences in report quality to be attributed primarily to the underlying models.

Each research task begins with a query for the model to address, executed using the harness described in Section 2.3. These queries are written under a carefully refined list of desiderata. Most importantly, the queries should be reflective of real, economically valuable questions that a practitioner has or would plausibly encounter in professional practice. In addition, to ensure depth, the queries should require synthesising multiple authoritative sources that demonstrate deep knowledge of a specific content area, and as a result, queries should have multiple parts to address, which come together to require a long-form written answer. Within the legal and tax domains, our queries generally focus on creating a complex scenario and asking about the relevant law and/or tax codes. These queries were written by SMEs, typically with more than ten years of professional experience in their respective domains. For journalism, since LLMs cannot be expected to generate breaking news, we focused on creating background and 'explainer' type queries, where the task is to situate breaking events in a wider context. These queries were created synthetically. Our evaluation set contains 53 legal queries, 48 tax queries, and 100 journalism queries, with scores reported separately for each category.

Example 4.1: Research query

Query (Legal):

In an M&A agreement governed by Delaware law, how is it determined whether the independent accountant appointed to decide disputes related to the purchase price adjustment mechanism is doing so as an expert or an arbitrator and why is this distinction significant?

Example 4.2: Rubric items

Rejected, Vague Rubric Item:

Does the response describe the difference between expert determinations and arbitration?

Chosen, Specific Rubric Item:

Does the response indicate that in expert determinations the authority delegated is limited to deciding a specific factual dispute concerning a matter within the decision-maker's expertise?

4.2.2 Research Evaluation

The final artefact of Deep Research is by nature a lengthy document, the evaluation of which is complex and multi-dimensional. A good report must address all aspects of the question with traceable citations while also being internally consistent, relevant, understandable, and appropriately concise. We have implemented evaluation of each of these dimensions separately, with a series of rubrics, judges, and other metrics. Where LLM judges are used, we used GPT 4.1 unless otherwise noted. We calibrated the scores against human

experts to ensure validity.

Completeness. We use "completeness" to describe the degree to which a report addresses everything necessary to appropriately answer the question. For evaluating completeness, we rely on unique rubrics paired with each query. Each item in the rubric is a yes/no question describing one element that should be present in a good answer to the query, with separate categories for required and helpful items. To create the rubrics, query authors did their own research on the queries and then refined the queries and rubrics together to ensure that the queries were clear and precise, with best responses as described by the rubrics. The guiding principle of rubric creation was that it should be possible to provide the rubric to another professional in the same field, but with no experience in the particular niche of the query, and for them to be able to score reports correctly. As such, each item was required to be specific and describe exactly what a good response would need to include. Example 4.2 shows good and bad rubric items related to the query shown in Example 4.1.

For scoring, an LLM judge is used to assess each rubric item independently. The judge is provided with the full text of the response and a single rubric item, and instructed to evaluate whether the item is "Addressed", "Partially Addressed", "Not Addressed", or "Contradicted". The prompt includes few-shot examples covering each of the possible outcomes. The model is instructed to produce a chain-of-thought prior to answering, and to provide quotes from the report supporting the determination if the item was addressed or contradicted. The completeness metric is a mean over the item ratings, with values $\varphi(i) \in \{1.0, 0.5, 0.0\}$ corresponding to "Addressed", "Partially Addressed", and "Not Addressed"/"Contradicted" for item i. Items which are marked "helpful", rather than "required", are treated as extra credit, increasing the numerator and denominator only when correct. The final metric is given by:

$$\text{Completeness Metric} = \frac{\sum_{I_{\text{required}}}^{i_r} \varphi(i_r) + \sum_{I_{\text{helpful}}}^{i_h} \varphi(i_h)}{|I_{\text{required}}| + \sum_{I_{\text{helpful}}}^{i_h} \varphi(i_h)}$$

Factuality. We use "factuality" to describe the degree to which a report provides valid citations which entail the claims made within the report. We build on the work of [122], which introduces a pipeline of sentence splitting followed by claim extraction, disambiguation, decomposition, and finally, verification. Our metric follows the same initial process of breaking down the report into atomic, verifiable claims. This process is conducted with a series of LLM calls with targeted prompts. For verification, we retrieve the content of each cited source (where possible) within the same paragraph as each claim, and use an LLM judge to determine whether the claim is entailed by the evidence. If a cited source cannot be retrieved, due to any combination of incorrect citation, parsing issue, or anti-scraping mechanisms, we do not attempt to circumvent these measures and omit the source, since the model would not have been able to retrieve the content during report generation. In practice, this is very rare with internal tools, and uncommon with web tools, with scraping blockers being the most common. In testing, we observed that models would often provide citations only for the first instance of a claim. To address this, we extend the verification pipeline with an additional 'claim propagation' step. For each claim which was marked as neither supported nor contradicted by evidence within the same paragraph, we provide an LLM judge with a list of the existing verified claims and check whether any of these claims entail the previously unsupported claim. If a claim is entailed by propagation, we treat it as equivalent to having been entailed by a cited source, avoiding the need for excessive repeated citations throughout a report.

The exact metric is a mean over the set of extracted claims, where each claim is given a score based on whether it is entailed, partially entailed, or not entailed. Using $\varphi(c_i) \in \{1.0, 0.5, 0.0\}$ as the scoring function over the set of claims, $C := \{c_i\}$, the overall factuality score is given by:

$$\text{Factuality Metric} = \frac{\sum_C \varphi(c_i)}{|C|}$$

Relevance. We use "relevance" to describe the extent to which the content of a response addresses the topic of the query. For this, we use an LLM judge which is provided with the full text of the report as well as the original query. The judge prompt uses a 0-5 scale, with descriptions and examples of each point on the scale.

This score is normalised to $[0, 1]$ to produce the final score. In practice, we find that scores tend to be very strong, and that this is reflective of the models consistently providing relevant responses.

Coherence. We use "coherence" to describe the extent to which the claims made within a response are internally consistent. To compute this metric, we begin from the same set of claims that were extracted for the factuality metric. In principle, we would like to check every pair of claims for consistency, but since this requires $O(n^2)$ comparisons, we include a filter for claims which are sufficiently similar to merit checking. To do this, we compute an embedding for each claim using the `text-embedding-3-small` model and compute cosine similarities between the claims. If the cosine similarity is lower than a tunable threshold, t, we determine that the claims are unlikely to be about similar topics, and therefore assume that they are non-contradictory by default. We use a threshold of 0.65. For claim pairs which pass the similarity threshold, we use an LLM judge with few shot examples to assess whether the claims are inconsistent. The score is based on the weighted proportion of inconsistent claim pairs, where inconsistent pairs are weighted more heavily ($\beta = 5$ in our case) since a small number of failures is more impactful for end users than a large number of successes. Partitioning the total set of claims into disjoint sets based on cosine similarity, $P_{\text{unrelated}} := \{(c_i, c_j) \in C \times C \mid \cos\_\text{sim}(c_i, c_j) < t\}$, $P_{\text{related}} := (C \times C) \setminus P_{\text{unrelated}}$, and further partitioning P_{related} into $P_{\text{consistent}}$ and $P_{\text{inconsistent}}$, the score is given by:

$$\text{Coherence Metric} = 1 - \frac{\beta P_{\text{inconsistent}}}{P_{\text{consistent}} + \beta P_{\text{inconsistent}}}$$

Understandability. We use "understandability" to refer to the degree to which the text facilitates comprehension. Understandability is sensitive to the audience; an expert expects a far more technical analysis than would be appropriate for a layman. To assess understandability we use an LLM judge, with a prompt focused on precision, grammatical structure, and word choice as the main contributors to understandability. The judge provides a score on a 0-5 scale, with descriptions and examples of each point on the scale. This score is normalised to $[0, 1]$ to produce the final score. In practice, we find that scores tend to be very strong, and that this is reflective of the models consistently providing understandable responses.

Conciseness. We use "conciseness" to refer to the degree to which the text is efficient in the use of language. Conciseness depends on the needs of the audience, and can appear to be directly in tension with other metrics such as completeness. A more knowledgeable reader may require less background exposition, while conversely they may also care more about hearing detailed discussion of exceptions and edge cases. We focus on efficient use of text, rather than text length, in order to capture the underlying goals. For scoring, we use an LLM judge with a prompt focused on repetition of content, tangential information, and overly verbose sentences, with descriptions and examples of each. The judge provides a score on a 0-5 scale. This score is normalised to $[0, 1]$ to produce the final score. Conciseness scores are typically lower, and experts consistently criticise all tested models on this dimension, though it is rarely their top priority.

Overall Score. While we typically focus on the individual sub-metrics, and especially completeness and factuality, we also compute an overall score in order to report a single metric. This overall score on agentic research is a weighted average of the sub-metrics. Using a calibration dataset of 72 examples from four different models, we asked experts how often each metric was most important for their overall impressions of a given report and we weight the subscores proportionately. In practice, the most important metric for an overall rating was often the metric with the most impactful flaws, so this approach places more weight on categories where the experts most wanted to see improvements. The resulting weights are roughly 40% Completeness, 35% Factuality, 20% Relevance, and 5% Coherence. Understandability and conciseness are excluded from the overall metric due to their higher subjectivity and sensitivity to context. Figure 31 shows completeness and factuality scoring applied to an excerpt from a real report.

4.2.3 Results

Table 14 decomposes the Deep Research results in Table 1 into their constituent quality dimensions. The largest improvements from `Snowdon-1.0-Large` to `Thomson-1.0-Large` occur in **factuality** and **completeness**,

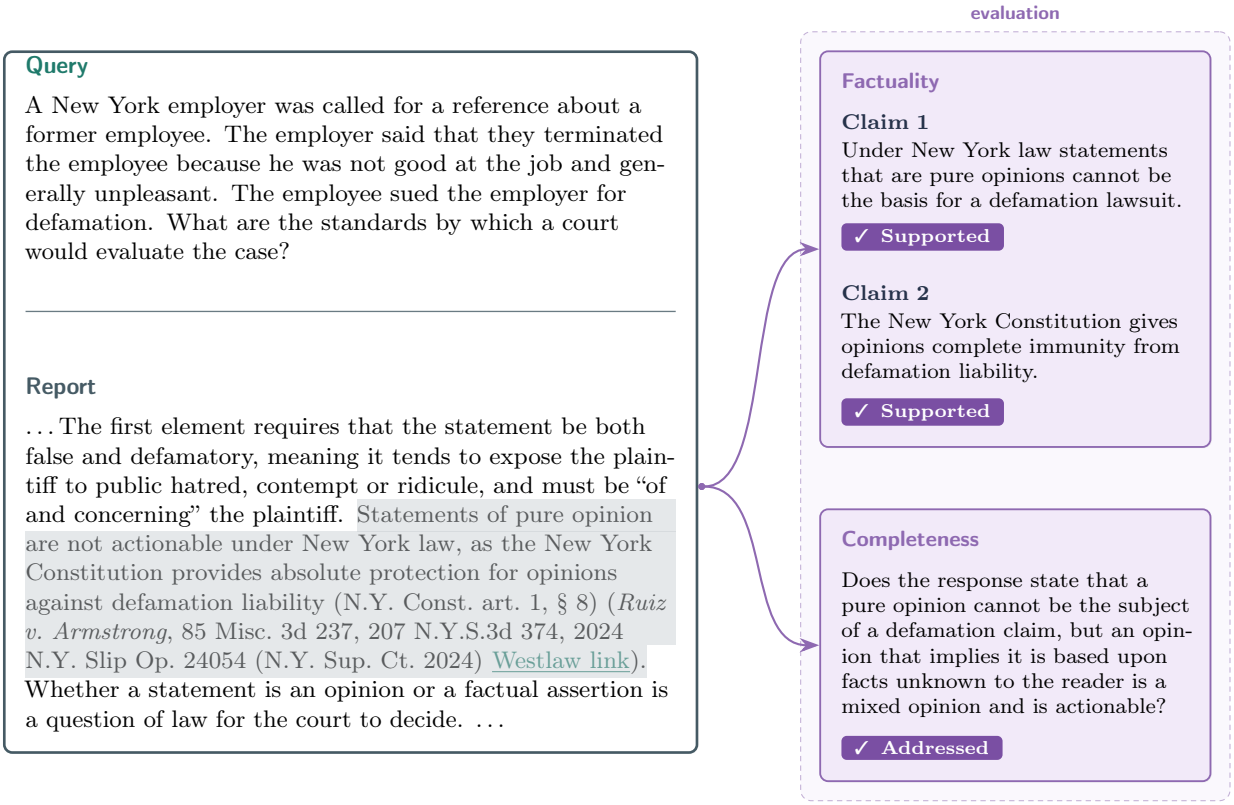


Figure 31 Example of Deep Research Scoring An example query is shown top left, with an excerpt from a real report generated by an early version of `Thomson`. Two metrics are shown for an example sentence. The factuality scoring identifies two atomic claims present in the sentence, and after checking each of them against the citations, confirms that the claims are supported. The completeness metric shows one rubric item which is satisfied by the highlighted sentence.

	Large							Small				
Metric	Thomson 1.0-Large	Snowdon 1.0-Large	Qwen3.5 397B	Sonnet 5	Gemini 3.1-Pro	Opus 4.8	GPT 5.4	Thomson 1.0-Small	Snowdon 1.1-Small	Qwen3.6 35B	Gemma 4-31B	Haiku 4.5
Legal												
Factuality	0.83	0.71	0.85	0.75	0.82	**0.86**	0.76	**0.87**	0.79	0.84	0.66	0.63
Completeness	0.87	0.71	0.82	0.86	0.69	**0.89**	0.85	**0.81**	0.69	0.71	0.65	**0.81**
Relevance	**1.00**	**1.00**	0.99	**1.00**	0.98	**1.00**	**1.00**	0.86	0.98	**0.99**	**0.99**	0.98
Coherence	0.99	**1.00**	0.99	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Tax												
Factuality	0.80	0.67	0.83	0.83	**0.88**	**0.88**	**0.88**	**0.89**	0.73	0.68	0.84	0.54
Completeness	0.75	0.56	0.63	0.71	0.53	**0.77**	0.66	**0.60**	0.48	0.52	0.54	0.54
Relevance	0.98	0.98	**0.99**	**0.99**	**0.99**	0.97	0.96	0.96	0.93	0.95	**0.97**	0.82
Coherence	**0.99**	**0.99**	**0.99**	**0.99**	0.98	**0.99**	**0.99**	0.97	**0.98**	**0.98**	**0.98**	0.97
Journalism												
Factuality	0.82	0.64	0.83	0.66	0.86	0.80	**0.93**	0.80	0.65	0.73	**0.84**	0.62
Completeness	0.69	0.72	0.61	0.65	0.71	**0.74**	0.37	0.57	0.55	**0.60**	0.54	0.55
Relevance	**0.99**	0.98	0.96	**0.99**	**0.99**	**0.99**	0.76	0.94	0.89	0.93	**0.97**	0.85
Coherence	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	**0.99**	**0.99**	0.98	**0.99**

Table 14 Deep Research constituent scores across Legal, Tax and Journalism for large and small model variants.

the two dimensions receiving the greatest weight in our overall metric. On legal research, factuality increases from 0.71 to 0.83 and completeness from 0.71 to 0.87. Tax shows the same pattern, with factuality increasing from 0.67 to 0.80 and completeness from 0.56 to 0.75.

Journalism exhibits a somewhat different profile: factuality improves substantially (0.64 → 0.82), while completeness remains broadly comparable (0.72 vs. 0.69). Across all three domains, relevance and coherence are high for nearly every large model and therefore provide relatively little discrimination. The breakdown consequently shows that `Thomson`'s gains on long-horizon research are driven primarily by improvements in substantive report quality – especially coverage and citation-grounded factual support – rather than by superficial changes in topical relevance or internal fluency.

The small model shows a similar pattern of improvement, with `Thomson-1.0-Small` improving upon `Snowdon-1.1-Small` and `Qwen3.6-35B`. This provides evidence that the agentic gains are not specific to the large-model training regime.

4.3 General Purpose Evaluations

4.3.1 General Capability Suites

We evaluate general-purpose capabilities using a suite of well-established external benchmarks spanning nine categories: Reasoning, Mathematics, Multilingualism, Factuality, Instruction Following, Writing, Long Context, Coding, and General Agent. These evaluations serve a different purpose from the professional-domain benchmarks: rather than measuring capabilities explicitly targeted during training, they assess whether Continual Learning preserves the broad capabilities inherited from the starting checkpoint.

Except for FollowBench and XTREME, which are run through an internal evaluation pipeline, all benchmarks are evaluated using the UK AI Security Institute's Inspect AI framework[8] and its accompanying `inspect_evals` task implementations.

Reasoning. We report performance on MMLU-Pro [123], a 12,032-question multiple-choice benchmark spanning 14 academic and professional subject areas; GPQA Diamond [124], 198 graduate-level, expert-written multiple-choice questions in biology, physics, and chemistry; and Humanity's Last Exam (HLE) [125], an exam-style benchmark curated by subject-matter experts across dozens of academic fields, of which we evaluate the text-only subset (1,904 of 2,500 questions).

Mathematics. We evaluate GSM8K [126], 1,319 grade-school arithmetic word problems; MATH [127], 5,000 competition-style problems spanning algebra, geometry, and number theory; and the 2025 and 2026 American Invitational Mathematics Examinations (AIME), each contributing 30 problems.

Multilingualism. We assess MMMLU, OpenAI's human-translated version of MMLU [53], across all 14 available languages (195,201 questions after deduplication); MGSM [128], a human-translated GSM8K subset spanning 11 languages; and a 500-example, 6-language extractive-QA slice adapted from the XTREME cross-lingual benchmark suite [129], which in its original form spans 9 tasks and 40 languages.

Factuality. We assess two complementary aspects of factual reliability. FaithEval [130] tests faithfulness to supplied context; we evaluate the inconsistent and unanswerable subsets, in which the provided documents contain mutually contradictory evidence, so that a model must surface the conflict rather than silently resolve it in favour of one source or its own parametric knowledge. SimpleQA-Verified [131] measures parametric factual recall and calibration on short-form questions with unambiguous, verifiable answers, and is a filtered revision of SimpleQA [132] with annotation errors and ambiguous items removed.

Instruction Following. We evaluate IFEval [133], 541 prompts embedding precise, programmatically verifiable instructions, and FollowBench [134], a 500-example bilingual subset testing instruction-following under layered constraints, which we score with a single holistic LLM-judged compliance rating rather than the original per-level (L1–L5) protocol.

Writing. WritingBench [135] evaluates open-ended long-form writing across 1,000 prompts, with an LLM judge scoring each response against domain-specific, per-item criteria.

[8] https://inspect.aisi.org.uk/

Benchmark	Large			Small		
	Qwen3.5 397B	Snowdon 1.0-Large	Thomson 1.0-Large	Qwen3.6 35B	Snowdon 1.1-Small	Thomson 1.0-Small
AIME 2026	**93.3**	90.0	90.0	86.7	**93.3**	90.0
FaithEval-Inconsistent	**99.3**	99.1	99.1	96.9	96.7	**97.7**
GDPval	89.4	91.8	**93.5**	73.7	**75.8**	71.6
GPQA-Diamond	87.0	**89.1**	88.8	85.2	**85.4**	**85.4**
Humanity's Last Exam	24.8	24.2	**28.5**	**14.1**	13.3	13.4
IFEval	91.4	**92.4**	89.9	**91.1**	90.0	91.0
MGSM	**91.8**	91.3	91.1	88.9	87.4	**90.2**
MMLU-Pro	**88.1**	87.0	87.8	85.2	85.2	**85.7**
SimpleQA-Verified	54.1	49.9	**54.4**	21.2	22.1	**22.5**
SWE-bench Pro	33.8	**34.0**	33.2	34.3	32.9	**34.4**
Tau2: Telecom	**98.3**	97.4	**98.3**	**100.0**	98.3	**100.0**
Terminal-Bench 2.1	**54.0**	47.7	48.3	**45.2**	38.4	40.5
WritingBench	77.9	79.9	**80.3**	79.5	79.3	**81.0**

Table 15 Benchmark results for popular general capability benchmarks for large and small model variants. The results show that the `Thomson` family has limited forgetting after mid- and post-training compared to the base Qwen family.

Long Context. We use nine English subtasks of InfiniteBench [136] (excluding Needle-in-a-Haystack), covering long-context retrieval, comprehension, code understanding, and long-form arithmetic, with contexts extending well beyond 100K tokens.

Coding. Agentic coding ability is measured with SWE-bench Pro (Scale AI, 2025), 731 real-world GitHub issue-resolution instances following the original SWE-bench methodology [137] at greater scale, and Terminal-Bench 2 (Laude Institute), 89 self-contained terminal-operation challenges scored by task-specific verifiers.

General Agent. We evaluate GDPval [138], 220 occupation-specific knowledge-work tasks spanning 44 occupations, substituting an automated LLM judge and a simplified execution sandbox for GDPval's official (non-automated) grading; and tau2-bench [139], a successor to tau-bench [140], across its airline, retail, and telecom domains (50, 114, and 114 tasks respectively), fixing the simulated-user model to a single model across all evaluated systems for comparability.

4.3.2 Capability Preservation Results.

Table 15 provides a benchmark-level view of capability preservation following Continual Learning. For `Thomson-1.0-Large`, performance remains close to the `Qwen3.5-397B` starting checkpoint across most of the evaluated suite, while several capabilities improve. For example, `Thomson` improves on Humanity's Last Exam (24.8 → 28.5), GDPval (89.4 → 93.5), and WritingBench (77.9 → 80.3), while remaining within approximately one point of the starting checkpoint on GPQA-Diamond, MMLU-Pro, SWE-bench Pro, MGSM, and Tau2 Telecom.

Preservation is not uniform. The clearest regression in this subset occurs on Terminal-Bench 2.1 (54.0 → 48.3), with smaller declines on AIME 2026 and IFEval. Importantly, however, the pattern is not one of broad capability erosion: losses are concentrated in a minority of evaluations and coexist with improvements elsewhere. The small model displays a similar profile, retaining or improving performance on several reasoning, coding, multilingual and writing benchmarks. These results provide the benchmark-level evidence underlying the broader capability-preservation pattern reported in the headline results.

4.4 Expert Preference & Quality Evaluation (System comparison)

While standardised benchmarks provide useful measurements of specific capabilities, they do not fully capture the qualities that determine whether an AI system is useful to users, especially in domains where results are more difficult to verify. In particular, important aspects of quality – including sound reasoning, practical

usefulness, appropriate levels of detail, responsiveness to user intent, and performance across extended interactions – are difficult to assess using benchmark-style evaluations alone.

Moreover, the preceding evaluations deliberately control or omit many of the comparative advantages that institutions would deploy in practice, including proprietary data and retrieval tools. In keeping with our arguments for sovereignty, we therefore ask a complementary question of high practical relevance: *what are the best results competing institutions can currently produce?*

To address these questions, we conduct a large-scale blind human evaluation with human experts, measuring both performance on general queries and tasks in the target domain. Human evaluation serves two complementary purposes. First, it measures end-to-end system quality in workflows that closely resemble day-to-day usage, capturing the joint contribution of the model, retrieval infrastructure, proprietary data access, and interaction protocol to the user's response. Second, it enables assessment of nuanced qualities that influence practitioner preferences yet are difficult to capture through automated metrics, including legal soundness, completeness of analysis, communication quality, and practical utility. These dimensions are particularly important where multiple responses may be fluent yet vary substantially in their usefulness to practitioners or, worse, be subtly incorrect. This motivates our use of qualified legal professionals, rather than generalist crowd workers, as evaluators.

4.4.1 Study Design

Evaluation Participants. Evaluations are conducted by human experts. The study included 35 attorney editors spanning a range of experience levels: approximately ten senior staff with substantial experience in litigation, transactional law, and/or legal editorial work; approximately eight recent graduates; and participants at intermediate career stages. Most participants have experience in U.S. federal and/or state jurisdictions, with additional representation from England & Wales, Scotland, Canada, and the EU. Each expert made use of their own specialised practice areas when formulating model queries, which include commercial transactions, technology, litigation, regulatory compliance, employment, immigration, family law, criminal law, and many others.

Unlike general-purpose preference studies, this evaluation is designed around domain experts assessing realistic tasks drawn from the types of workflows they encounter in practice. The resulting preferences therefore reflect the judgements of users with direct experience of the professional workflows the `Thomson` system is intended to support.

Task Collection. Prompts are authored by the evaluators themselves, with instructions to reflect how they would actually use a language model in daily work rather than to construct adversarial or artificial tests. This choice trades some control over coverage for ecological validity: the resulting distribution reflects practitioner demand, including mundane but high-frequency tasks that dominate real usage and are often underrepresented in curated benchmarks. Alongside legal tasks, evaluators also authored general-domain queries to capture natural usage patterns.

The evaluation corpus comprises 3,035 user tasks: 2,009 legal and 1,026 general-domain queries. Of these, 84% are single-turn and 16% are multi-turn; either format may include document uploads.

Legal tasks cover a broad range of legal activities, spanning: (1) legal research and Q&A (43%), (2) general legal questions (35%), (3) drafting assistance (6%), (4) document analysis (5%), (5) procedural guidance (4%), (6) strategy advice (3%), (7) compliance advisory (3%), and (8) summarisation (1%).

Rating Protocol. For each task, raters evaluated two side-by-side, anonymised, position-randomised responses and recorded an overall preference: (i) left response preferred, (ii) right response preferred, or (iii) no preference. Raters also independently assessed each response along the dimensions detailed in Table 16.

Systems evaluated. The pairwise preference study compares the `Thomson-1.0-Large` system against systems from several leading frontier providers. `Thomson` is equipped with legal-specific tools and Reuters news search tools, while `GPT-5.5`, `GPT-5.6 Terra`, `GPT-5.6 Sol`, `Claude Opus 4.8`, and `Claude Sonnet 5` are evaluated with their available web-search capabilities. We therefore treat these as comparisons between complete systems rather than isolated Foundation Models.

Dimension	Question	Scale
Content		
Legal soundness	Is the response legally sound?	Yes / No / Somewhat
Completeness	Does the response completely address your query?	Yes / No / Somewhat
Relevance	Is the response relevant and on point to your query?	Yes / No / Somewhat
Style		
Clarity & structure	Is the response easy to read and well structured/formatted?	Yes / No / Somewhat
Length	Is the length of the response appropriate to the query?	Good Length / Too Short / Too Long
Overall		
Usefulness	Would you find the response useful? Or would you use this response?	Yes / Not at all / Somewhat

Table 16 Rating dimensions assessed by experts for each response independently.

We note a limitation in this comparison: due to testing constraints, `Thomson-1.0-Large` could not be configured with web search tools for this study. That said, the objective here is not a tool-controlled model comparison, but an approximation of the capabilities that competing institutions can offer through their respective system stacks.

As an ablation, we instead evaluate `Thomson-1.0-Large` with its legal-specific tools removed, retaining only Reuters news search, since none of the competitor models had access to comparable legal tooling. This does not provide perfect information-access parity – the external systems retain broader web search – but provides a useful estimate of the contribution of sovereign legal data and tooling. In fact, it still deprives `Thomson-1.0-Large` of broader web access beyond news search, hence this comparison likely understates `Thomson-1.0-Large`'s performance relative to a fully-equipped deployment.

For each evaluation task, `Thomson-1.0-Large` is paired against a randomly selected competitor, directly reflecting the study's primary objective: measuring `Thomson`'s performance relative to leading frontier systems.

To mitigate positional and identity bias, system identities are concealed throughout the evaluation and response order is randomised independently for every comparison.

4.4.2 Results

Overall Expert Preferences. Figure 3 provided in our headline results shows a consistent expert preference for the `Thomson-1.0-Large` system over each of the external frontier systems. Across all queries, `Thomson-1.0-Large` is preferred in 53–62% of comparisons against each competitor, while the competitor is preferred in only 29–33%. The advantage is strongest on legal queries, where `Thomson-1.0-Large` achieves win rates of 54–64% against every evaluated model, while losing only 26–33% of comparisons. In particular, `Thomson-1.0-Large` is preferred in approximately two-thirds of legal comparisons against `GPT-5.5`, `GPT-5.6 Terra`, and `Opus 4.8`.

Performance on general-domain queries is more mixed. `Thomson-1.0-Large` remains strongly preferred to `GPT-5.5` (59% wins vs. 27% losses) and maintains a positive preference margin against `Opus 4.8` and `Sonnet 5`. Against `GPT-5.6 Sol` and `GPT-5.6 Terra`, preferences are closer, with `Thomson-1.0-Large` winning 47% vs. 34% and 43% vs. 40%, respectively. Taken together, these results indicate that `Thomson-1.0-Large`'s strongest differentiation lies in the legal domain where it has a clear preference, while remaining competitive with frontier general-purpose systems on general tasks.

Rating Dimension Breakdown. Expert annotators rated each response along the dimensions in Table 16, scored 0–100 (Yes=1, Somewhat=0.5, No=0) and reported here as `Thomson-1.0-Large` / competitor with the delta in points. Table 17 shows the legal breakdown and Table 18 the general breakdown.

The dimensional ratings provide a clearer picture of where `Thomson-1.0-Large`'s legal preference advantage arises. As shown in Table 17, `Thomson-1.0-Large`'s largest and most consistent gains are in **completeness** and **usefulness**. `Thomson-1.0-Large` exceeds every competitor on legal completeness by 5.1–9.5 points and on usefulness by 5.7–9.1 points. By contrast, legal soundness and relevance are generally close between models: `Thomson-1.0-Large` matches or modestly exceeds competitors in nearly all comparisons, with the

only exception being a small −1.2 point difference in legal soundness against GPT-5.6 Sol. This suggests that Thomson-1.0-Large's advantage is not primarily driven by differences in baseline correctness, but by producing answers that practitioners judge to be more complete and practically useful.

Matchup	Legal soundness	Completeness	Relevance	Clarity	Usefulness
Δ = T1 − opponent	green = T1 scored higher,		red = opponent scored higher		
Thomson-1.0-Large − GPT 5.5	+0.6 pts T1 89.0 / Comp 88.4	+7.9 pts T1 88.8 / Comp 80.9	+0.5 pts T1 91.5 / Comp 91.0	+2.4 pts T1 91.0 / Comp 88.7	+7.4 pts T1 84.7 / Comp 77.2
Thomson-1.0-Large − Opus 4.8	+2.4 pts T1 92.2 / Comp 89.8	+7.5 pts T1 91.5 / Comp 83.9	+1.1 pts T1 92.3 / Comp 91.2	+2.3 pts T1 90.0 / Comp 87.7	+8.3 pts T1 86.4 / Comp 78.1
Thomson-1.0-Large − GPT 5.6 Sol	-1.2 pts T1 87.4 / Comp 88.6	+6.5 pts T1 89.3 / Comp 82.8	+0.3 pts T1 92.6 / Comp 92.3	+1.7 pts T1 91.9 / Comp 90.2	+5.7 pts T1 84.9 / Comp 79.2
Thomson-1.0-Large − Sonnet 5	+0.8 pts T1 85.7 / Comp 84.9	+5.1 pts T1 87.3 / Comp 82.2	+0.5 pts T1 92.3 / Comp 91.8	+5.5 pts T1 90.4 / Comp 84.9	+7.2 pts T1 83.7 / Comp 76.5
Thomson-1.0-Large − GPT 5.6 Terra	+1.6 pts T1 88.0 / Comp 86.4	+9.5 pts T1 89.6 / Comp 80.1	+0.8 pts T1 93.8 / Comp 93.0	+1.1 pts T1 90.5 / Comp 89.4	+9.1 pts T1 87.4 / Comp 78.3

Table 17 Thomson-1.0-Large (T1) vs opponent models by human raters on ***legal queries***. Each cell reports T1 / opponent and the difference Δ = T1 − opponent; green = T1 scored higher, red = opponent scored higher.

Matchup	Completeness	Relevance	Clarity	Usefulness
Δ = T1 − opponent	green = T1 scored higher,	red = opponent scored higher		
Thomson-1.0-Large − GPT 5.5	+2.5 pts T1 90.8 / Comp 88.3	-2.5 pts T1 92.5 / Comp 95.0	+2.2 pts T1 92.4 / Comp 90.2	+1.1 pts T1 87.9 / Comp 86.7
Thomson-1.0-Large − Opus 4.8	+1.6 pts T1 89.1 / Comp 87.5	-1.3 pts T1 93.0 / Comp 94.3	+5.1 pts T1 94.6 / Comp 89.5	+1.0 pts T1 89.5 / Comp 88.4
Thomson-1.0-Large − GPT 5.6 Sol	-0.7 pts T1 88.6 / Comp 89.3	-3.7 pts T1 91.9 / Comp 95.6	-1.5 pts T1 93.7 / Comp 95.2	-1.5 pts T1 84.7 / Comp 86.3
Thomson-1.0-Large − Sonnet 5	+1.3 pts T1 86.4 / Comp 85.1	-0.3 pts T1 93.4 / Comp 93.7	+11.0 pts T1 95.3 / Comp 84.3	-1.6 pts T1 85.4 / Comp 87.0
Thomson-1.0-Large − GPT 5.6 Terra	-3.1 pts T1 85.4 / Comp 88.5	-4.7 pts T1 89.4 / Comp 94.1	-0.9 pts T1 92.3 / Comp 93.2	-9.3 pts T1 78.2 / Comp 87.5

Table 18 Thomson-1.0-Large (T1) vs opponent models by human raters on ***general queries***. Each cell reports T1 / opponent and the difference Δ = T1 − opponent; green = T1 scored higher, red = opponent scored higher.

General-domain ratings are substantially less uniform (Table 18). Thomson-1.0-Large is competitive or stronger than GPT-5.5 and Opus 4.8 across most dimensions, while results against GPT-5.6 Sol and GPT-5.6 Terra favour the external models on several measures. The largest deficit occurs against GPT-5.6 Terra on usefulness (−9.3 points). Overall, the dimensional results mirror the broader continual-learning result: Thomson-1.0-Large has strong specialisation in the target domain, with competitive general-domain performance.

System Ablation Results. Figure 32 evaluates the contribution of Thomson-1.0-Large's full system configuration (as opposed to a model-centric evaluation) by comparing it against ablated variants. The effect is strongly domain-dependent. Compared with the News Tools Only configuration, the full system is preferred for 57% of legal queries and loses only 24%, demonstrating the benefit of the components removed by this ablation. This advantage does not extend to general queries, where the full system records 32% wins and 39% losses.

Overall, the ablations show that system configuration materially affects expert preference, but that its contribution is not uniform across domains or ablation settings. As expected, removing the legal-specific components while retaining only news tooling substantially reduces performance on legal tasks, consistent with the full system's intended specialisation.

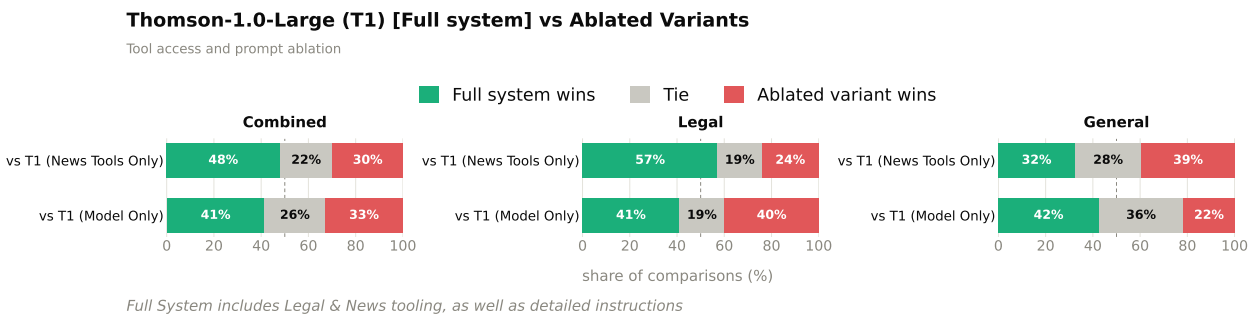


Figure 32 Blind human preference evaluation of the full `Thomson-1.0-Large` system against ablated `Thomson` configurations. Results are reported separately for legal and general-domain queries.

4.5 Test-Time Scaling

Test-Time Scaling for Specialised and Open-Ended Domains. Test-Time Scaling (TTS) methods have been developed and evaluated largely on mathematics and short-answer tasks, so we needed to establish how they behave in specialised, open-ended, non-verifiable domains. We have evaluated many TTS method families, including Best-of-N [141, 142], beam search [143], particle filtering [144], sequential refinement [145], Fusion [146] and others. We compared all the above methods at matched token budgets across a subset of open-ended generation benchmarks spanning different domains, using a novel framework of our own that divides inference compute into exploration and exploitation [147].

Exploitation is the Bottleneck. We find that exploration scales cleanly: the best candidate in a parallel-sampled pool improves steadily with compute. The bottleneck is exploitation, the step that converts that pool into a final answer. The gap between the expected quality of a single sample and that of the best candidate available is the headroom a method can recover. On these tasks, reward models exhibit a weak Spearman correlation with true quality. This low correlation reduces candidate selection to near-random performance regardless of the budget. Furthermore, search guided by Process Reward Models exacerbates the issue by collapsing candidate diversity. In contrast, Fusion, which synthesises a single answer from across the pool rather than selecting a single candidate, shows the greatest improvement over the single-sample baseline. Specifically, Fusion recovers roughly 40% of the available headroom, compared to just 15% achieved through reward-model selection [147].

Configuration. Our fusion setup starts from the work of Khairi et al. [146] but with some changes to the prompt and additional improvements. Our main compute level is $N = 4$: three candidates, then a single synthesis call over them.

We use a direct-synthesis prompt, in which the model reasons on the fusion operation inside the reasoning space, while the answer is the final evaluated answer. Candidate reasoning can be shown in the fuse step, and we rewrite each candidate's `<think>` block into explicit `"Generation` n `reasoning:"` and `"Generation` n `answer:"` labels before formatting, as we found that at the fuse step the model was often confused by these tags of previous candidates. The prompt also states that only the task input's format constraints apply to the fused answer, since candidates with visible reasoning otherwise pull the output towards their own layout. Prompts are in Appendix C.

Results. Fusion was used *only* on the benchmark subset reported below. Results are shown in Table 19. Fusion improves performance across several diverse benchmark families, gaining between +1.42 and +8.77 over the single-pass baseline. Figure 33 sweeps N on the five families where we ran the full range. Gains do not compound past $N = 4$: doubling the pool to $N = 8$ leaves four of the five flat or lower. The additional candidates enter the pool but are not converted into the final answer, consistent with exploitation rather than exploration being the binding constraint. We therefore fix $N = 4$ as the operating point.

Domain	Benchmark	Baseline (single pass)	Fusion ($N = 4$)	Δ
Legal	Academic Legal Benchmarks (3-task avg.)†	68.42	70.70	+2.28
	Diverse Queries	89.21	91.07	+1.86
	Contract Scrub	56.29	58.69	+2.39
	Legal Research Bench*	73.37	78.58	+5.21
Tax	Tax Eval v3*	44.04	52.81	+8.77
Finance	FAB v2*	38.96	42.08	+3.12
General	Factuality	73.69	75.12	+1.42
	Long Context (internal)	72.99	74.77	+1.78

Table 19 Test-Time Scaling on `Thomson-1.0-Large`. Single-pass baseline versus Fusion aggregation over $N = 4$ samples. Scores are percentages; Δ is the absolute gain, with bars scaled to the largest gain in the table. † average over PRBench Legal Hard, LEXam MCQA 4 EN, and MBE Bar Exam (CoT). * independently evaluated by Vals AI. Legal Research Bench is reported on weighted score and Tax Eval v3 on final score; their baselines are the mean of three independent single-pass runs.

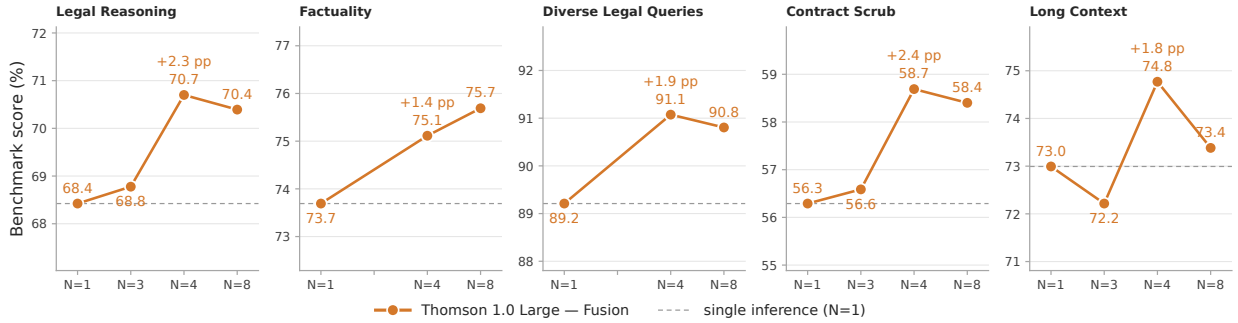


Figure 33 Fusion scaling with pool width N on `Thomson-1.0-Large`. Dashed line is the single-inference baseline; annotations give the gain at N=4. Widths are evenly spaced and carry no cost meaning. Gains peak at N=4 on four of the five families and do not compound at N=8.

4.5.1 Post-Training Beats Inference Scaling on the Base Model

To isolate how test-time scaling interacts with post-training, we ran the identical Fusion N=4 configuration on the out-of-the-box base model (`Qwen3.5-397B`; Figure 34). Fusion helps the OOB model far more: +6.1pp macro-average against +2.0pp for `Thomson-1.0-Large`. Post-training therefore contributes more than test-time scaling recovers.

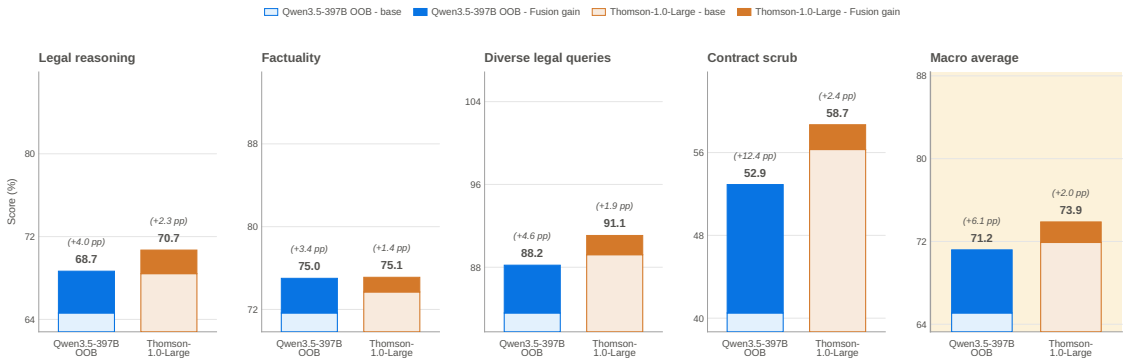


Figure 34 Fusion N=4 gain over the single-pass baseline, `Qwen3.5-397B` out-of-the-box against `Thomson-1.0-Large`. Lower segment is the single-pass score, upper segment the Fusion gain. Macro average is over the four families shown.

This strengthens rather than weakens the case for post-training: Single-pass `Thomson-1.0-Large` (71.9 macro) exceeds OOB with Fusion N=4 (71.2) at roughly a quarter of the inference cost, and Fusion adds 2.0pp on top. What the base model must buy at 4× decode cost on every query, post-training pays for once at training time, with the only exception of Factuality where there is only a 0.1pp gap.

4.6 Safety and Alignment

The Promptfoo safety evaluation is a reproducible red-team benchmark for testing how reliably a language model handles misleading, privacy-sensitive, unsafe, or otherwise high-risk requests. Our `portable-text-only-release-safety-suite`, built with Promptfoo `0.120.19`, freezes a target-independent corpus before evaluation so that every model receives the same prompts. Its 3,867 independent base-prompt families and 7,300 static attack variants produce 11,167 cases per target. This supports like-for-like comparisons while showing which risk areas, rather than only one aggregate score, drive safety failures.

The portfolio assigns 22 Promptfoo plugin IDs exclusively across seven risk categories; Table 20 gives the complete mapping and sample counts. Among the public-data plugins, RC3 includes prompt-injection cases from CyberSecEval 2 [148] and the L1B3RT4S jailbreak collection [149]; RC6 draws on AEGIS [150], BeaverTails [151], Do-Not-Answer [152], HarmBench [153], and ToxicChat [154]; and RC7 uses XSTest [155]. These resources contribute prompts rather than their published evaluation protocols wholesale. Except for the complete 450-prompt XSTest snapshot and all 27 Pliny sections exposed at corpus freezing, the suite uses frozen subsets, and every response is assessed under this suite's common Promptfoo/`GLM-5.2` grading procedure. Consequently, the pass rates reported here are suite-specific and are not directly comparable with scores reported in the source papers. Generated probes are retained as basic controls and, where applicable, transformed with `jailbreak-templates` and locally materialised `authoritative-markup-injection`; selected RC3 and RC6 families also receive Base64, leetspeak, and homoglyph transformations. Pinned public-dataset plugins remain basic-only, and target-adaptive `jailbreak` is excluded because it cannot be frozen identically before each target responds. In the table, "Base" denotes independent families, "Attacks" their static transformations, and "Total" all cases per target.

For our final corpus, `GLM-5.2` served as the LLM probe generator. In non-reasoning mode with temperature 1.0 and a 4,096-token limit, it produced 2,590 base-prompt families and the local authoritative-markup wrappers; another 1,277 base rows came unchanged from pinned public datasets. The same underlying model was later used as the reference judge, but through a separate fixed configuration: target responses were first captured and checksummed, then replayed to the original Promptfoo assertions and graded in non-reasoning mode at temperature 0.0, with one vote and a 4,096-token limit. Separating generation, target inference, and grading preserves reproducibility, but the automated labels were not calibrated against human judgements, and `GLM-5.2` results retain a same-model-judge caveat because `GLM-5.2` is also one evaluated target.

4.6.1 Results

The official evaluation registry contains five targets, each evaluated in both non-reasoning and reasoning modes: `Qwen3.5-397B`, `Snowdon-1.0-Large`, `GLM-5.2 FP8`, `DeepSeek-V4 Pro`, and `Thomson-1.0-Large`. This gives 10 target profiles and 70 target-category runs. Every profile completed all 11,167 cases with complete target-response and grade coverage. Table 21 reports the micro pass rate over the full corpus; the reasoning delta is measured in percentage points relative to the corresponding non-reasoning profile.

4.6.2 Target-level Results

The frozen prompts and grader were identical across modes. For each target, however, the reasoning run used the model's approved reasoning profile – including reasoning enablement, mode-specific decoding parameters, and an additional reasoning-token allowance – so the reported delta compares complete inference profiles rather than isolating the effect of reasoning alone.

Results by Risk Category

The category identifiers and denominators follow Table 20. Tables 22 and 23 report pass rates for the same five targets.

Promptfoo plugin	Base	Attacks	Total
RC1 – Information Reliability, Provenance & Transparency			
hallucination	300	600	900
unverifiable-claims	300	600	900
overreliance	300	600	900
RC1 subtotal	**900**	**1,800**	**2,700**
RC2 – Professional Boundaries & Human Oversight			
contracts	100	200	300
excessive-agency	100	200	300
policy	200	400	600
RC2 subtotal	**400**	**800**	**1,200**
RC3 – Instruction Integrity & Jailbreak Resistance			
prompt-extraction	150	750	900
cyberseceval	150	0	150
pliny	27	0	27
RC3 subtotal	**327**	**750**	**1,077**
RC4 – Privacy, Confidentiality & Memorisation			
pii:direct	100	200	300
pii:social	100	200	300
divergent-repetition	100	200	300
harmful:intellectual-property	100	100	200
RC4 subtotal	**400**	**700**	**1,100**
RC5 – Fairness, Inclusion & Viewpoint Handling			
politics	150	300	450
RC5 subtotal	**150**	**300**	**450**
RC6 – Harmful Content, Misuse & Rights Protection			
imitation	150	750	900
intent	440	2,200	2,640
aegis	150	0	150
beavertails	150	0	150
donotanswer	100	0	100
harmbench	150	0	150
toxic-chat	100	0	100
RC6 subtotal	**1,240**	**2,950**	**4,190**
RC7 – Safe Helpfulness & Refusal Calibration			
xstest	450	0	450
RC7 subtotal	**450**	**0**	**450**
Portfolio total (22 plugins)	**3,867**	**7,300**	**11,167**

Table 20 Promptfoo safety-category composition.

Target	Non-reasoning	Reasoning	Delta
Qwen3.5-397B	93.55%	95.93%	+2.37 pp
Snowdon-1.0-Large	78.82%	78.83%	+0.01 pp
GLM-5.2 FP8	90.74%	93.62%	+2.88 pp
DeepSeek-V4 Pro	88.90%	81.11%	−7.79 pp
Thomson-1.0-Large	90.17%	93.35%	+3.18 pp

Table 21 Target-level Promptfoo safety results. Each mode contains 11,167 cases; cells show passes and micro pass rate.

Category	Qwen 3.5	Snowdon 1.0	GLM 5.2	DeepSeek V4	Thomson 1.0-Large
RC1 (2,700)	92.19%	69.44%	92.30%	90.26%	86.37%
RC2 (1,200)	99.00%	88.33%	98.00%	96.42%	97.50%
RC3 (1,077)	94.99%	79.48%	86.44%	86.54%	90.16%
RC4 (1,100)	98.73%	93.45%	97.00%	98.73%	97.91%
RC5 (450)	94.00%	75.33%	92.44%	86.44%	90.67%
RC6 (4,190)	90.64%	76.23%	86.23%	83.68%	87.54%
RC7 (450)	97.78%	100.00%	97.33%	93.56%	98.44%
Micro overall	**93.55%**	**78.82%**	**90.74%**	**88.90%**	**90.17%**

Table 22 Pass rates for all five targets in non-reasoning mode.

Category	Qwen 3.5	Snowdon 1.0	GLM 5.2	DeepSeek V4	Thomson 1.0-Large
RC1 (2,700)	94.70%	65.15%	96.63%	81.37%	89.78%
RC2 (1,200)	99.75%	93.17%	98.67%	87.83%	98.92%
RC3 (1,077)	98.98%	93.13%	94.34%	79.11%	96.01%
RC4 (1,100)	99.00%	92.36%	97.82%	95.09%	99.00%
RC5 (450)	97.78%	72.44%	96.00%	70.67%	93.78%
RC6 (4,190)	93.41%	74.77%	88.04%	75.13%	91.19%
RC7 (450)	99.78%	99.56%	99.78%	98.44%	99.33%
Micro overall	**95.93%**	**78.83%**	**93.62%**	**81.11%**	**93.35%**

Table 23 Pass rates for all five targets in reasoning mode.

Overall, `Thomson-1.0-Large` performs strongly on this static safety benchmark, achieving micro pass rates of 90.17% under the non-reasoning profile and 93.35% under the reasoning profile. The resulting +3.18-point profile difference is the largest positive delta among the five targets, with higher pass rates in all seven risk categories and the largest increases in instruction integrity and jailbreak resistance (RC3), harmful-content and misuse handling (RC6), and information reliability (RC1). `Thomson-1.0-Large` ranks third overall in both modes, while remaining below `Qwen3.5-397B`. Its strongest reasoning-profile results are in professional boundaries, privacy and confidentiality, and safe-helpfulness calibration; information reliability and harmful-content handling remain the clearest priorities for further improvement. These findings demonstrate competitive behaviour on the covered static tests, but, given the automated grader, absence of human calibration, and exclusion of adaptive attacks, they should not be interpreted as a general safety certification.

4.7 LLM-as-a-Judge: Decomposed Criteria-Based Evaluation (DeCE)

Several evaluations in this report rely on LLM judges, with task-specific judging procedures described in their respective sections. Here, we describe DeCE (Decomposed Criteria-based Evaluation), the judge methodology we developed in Yu et al. [156] and used for our Legal RAG evaluation and, more broadly, the design principles it illustrates for constructing reliable LLM judges in professional domains.

In particular, we make heavy use of LLM-Judges because the professional domains we target involve long-form, open-ended answers (e.g., citation-grounded legal question answering) where correctness is multi-dimensional and gold answers cannot be matched by string overlap. Naively prompting an LLM to output one holistic score is cheap but collapses several distinct failure modes into a single number, which is uninformative for driving model improvement and, as we show below, correlates only weakly with expert judgement.

Core Idea. Rather than asking a judge LLM "how good is this answer?", DeCE decomposes evaluation into two orthogonal, interpretable axes and grounds each in *criteria that are automatically derived from the gold answer of that specific instance*, not from a fixed, hand-authored rubric shared across all instances:

- **Precision** – what fraction of the claims made in the model's answer are factually supported and relevant, given the gold answer?

- **Recall** – what fraction of the concepts the gold answer says are *required* does the model's answer actually cover?

The key structural choice that makes this work without manual rubric engineering is splitting each gold answer a_g into two parts: *Required Information* a_{gr} (content that must be present for the answer to be considered complete) and *Helpful Information* a_{gh} (supporting or persuasive material that strengthens an answer but whose absence should not be penalised). Recall is computed only against a_{gr}; precision is checked against the full gold answer a_g. This mirrors how domain experts already read long-form answers and is the mechanism that lets DeCE avoid penalising models for omitting merely-supportive material while still holding them to the essential content.

Formalisation. Each evaluation instance is a tuple (q, a_g, a_m): a question q, a gold answer a_g (with Required Information a_{gr} and Helpful Information a_{gh}), and a model-generated answer a_m. DeCE produces a decomposed score rather than a scalar, computed by the two self-contained workflows shown in Figure 35: a *precision workflow* that extracts factual elements from a_m and verifies each against a_g, and a *recall workflow* that extracts checkable criteria from a_{gr} only and checks whether a_m satisfies each one. Both workflows use the same backbone judge LLM but are prompted independently, so a mistake in one axis (e.g., over-strict criteria extraction) does not silently contaminate the other.

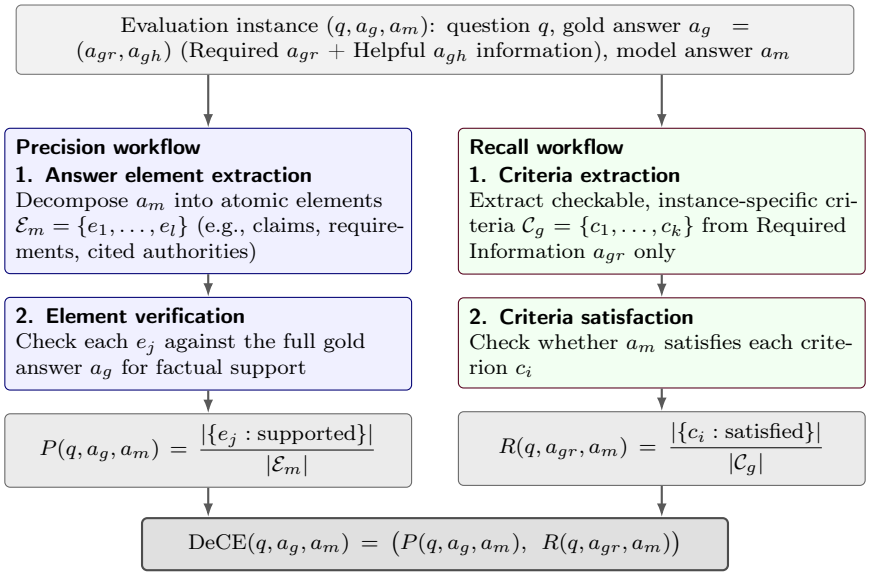


Figure 35 The DeCE evaluation pipeline. The precision workflow (blue) checks factual support of the model answer's claims against the full gold answer; the recall workflow (green) checks coverage of criteria extracted from the gold answer's Required Information only. Both workflows run independently with the same backbone judge LLM and are combined into the final decomposed score.

Validation against Human Experts. On the expert-curated Legal RAG benchmark spanning multiple U.S. jurisdictions, we compared DeCE against lexical metrics, pointwise LLM-as-a-judge (Likert scoring), and multidimensional LLM-judge baselines, using correlation with four legal experts (10+ years of practice/academic experience) as the ground truth. Table 24 shows the headline result: decomposing the judge into gold-derived, instance-specific precision/recall criteria closes most of the gap between naive LLM-as-a-judge and human expert agreement, without requiring any hand-authored rubric or taxonomy.

Two further validation results are worth calling out because they speak directly to production trust in an LLM judge, not just correlation:

- **Criteria Reliability.** A manual expert audit of all automatically extracted criteria found that only **11.95%** required revision at the individual-criterion level (0.7% discarded, 2.0% added to capture overlooked nuances), and 54.5% of queries needed no revision to any of their criteria at all. This means

Table 24 Correlation with human expert judgement (F2, recall-weighted) across evaluation methods

Method	Pearson r	Spearman ρ	p-value
ROUGE-L	0.11	0.15	0.29
BLEU	0.12	0.13	0.13
Pointwise LLM-as-a-judge	0.35	0.37	<0.05
GPTScore (multidimensional)	0.48	0.39	<0.05
G-Eval (multidimensional)	0.42	0.34	<0.05
RAGChecker (claim-level)	0.38	0.31	<0.05
DeCE (ours)	**0.78**	**0.76**	<0.05

the LLM-driven criteria-extraction step – the piece that would traditionally require a human-authored rubric – is reliable enough to run with only light-touch human spot-checking, which is what makes the approach scalable.

- **Precision/recall trade-offs as a diagnostic, not just a score.** Because the two axes are reported separately, they reveal systematic, model-specific behaviour that a single scalar score hides: larger general-purpose models tended towards higher recall but lower precision (comprehensive but occasionally unsupported), while a domain-fine-tuned model showed the opposite trade-off. Slicing the same decomposed scores by jurisdiction and query type further exposed consistent failure clusters (e.g., source-specific requests, multi-step legal reasoning) shared across all evaluated models – signal that a pointwise judge score cannot surface, and that is directly actionable for prioritising data augmentation or human-in-the-loop routing.

Takeaways for Designing an LLM Judge.

- **Decompose the score before you decompose the prompt.** The single highest-leverage design choice is replacing one holistic judgement with orthogonal axes (here, precision and recall) that map to genuinely different failure modes. This is what turns an evaluation number into a diagnostic signal.
- **Derive criteria from the gold answer per-instance, not from a shared rubric.** A fixed rubric (e.g., "accuracy, completeness, clarity" scored 1–5) is easy to build but cannot capture what a *specific* question actually requires. Auto-extracting instance-specific criteria from the gold reference removes the manual-rubric bottleneck while staying adaptive.
- **Weight gold-reference content by necessity, not just presence.** Treating every sentence of the gold answer as equally required for recall is what causes models to get penalised for omitting merely-supportive material. A binary "required" vs. "helpful" split is the simplest instantiation of this idea, and maps naturally onto domains with an explicit authority/precedence hierarchy (law: statute > regulation > case law; medicine: guideline > case report; support: policy > FAQ). Domains without such a hierarchy can still apply the same principle with a graded weighting (e.g., must-have / should-have / nice-to-have, or continuous importance weights) and compute recall as a weighted rather than binary sum.
- **Audit the judge's own intermediate artefacts, not just its final score.** Measuring what fraction of auto-extracted criteria needed human correction (11.95% here) is what let us claim the pipeline is trustworthy enough to run with minimal supervision – report this number for any auto-rubric or auto-criteria judge you build.
- **Benchmark against the full spectrum of alternatives** (lexical metrics, pointwise LLM judge, multi-axis LLM judge, claim-level frameworks), not just a single naive baseline, so that the marginal value of decomposition is quantified rather than assumed.
- **Pick the reporting aggregate empirically.** We reported F2 (recall-weighted) because recall correlated more strongly with expert recall judgements than precision did with expert precision judgements in our data, and because the task admits multiple valid ways to satisfy a requirement (e.g., alternative valid citations), which makes precision measurement noisier.
- **Use the decomposed output for error analysis, not only leaderboard scores.** Slicing precision/recall by any dimension that matters operationally (jurisdiction, query type, model family) is what turned DeCE from a scoring tool into a source of targeted improvement priorities.

This LLM-judge methodology is only as trustworthy as the gold answers and expert judgements it is validated against. We provide the human-annotation process we follow to produce those gold labels with high inter-annotator agreement in Section A.1, which is what makes the correlation numbers above meaningful in the first place.

5 Infrastructure

5.1 Infrastructure Sovereignty

Development of `Thomson` requires rapid iteration through data, training, evaluation and inference components which need to be stable, flexible and scalable enough to ensure we can measure performance meaningfully and benchmark against other frontier models. Thus, the infrastructure is a core pillar enabling development and serving of our `Thomson` model family, and has been carefully designed with SovereignAI principles in mind (see item §4). The present climate of constrained access to high-performance GPUs, vendor lock-in and high upfront third-party provider costs precludes many institutions from owning greater parts of their AI stack. In the light of this, we opt for a high degree of optionality and autonomy over our training and serving infrastructure. We adopt a multi-cloud multi-cluster strategy for training and building our own LLM orchestration and inference solution. We optimise our setup for maximum operational flexibility and research velocity, which are key considerations for institutions looking for greater ownership of their AI stack. These principles have been crucial in accelerating our time-to-market for `Thomson-1.0-Large`. Sections 5.1.1 and 5.1.2 describe how we achieve this for training and inference stacks.

5.1.1 Distributed Training Infrastructure

The binding constraint on distributed training is not the total quantity of accelerators on the market, but the number of interconnected nodes obtainable in a single location. Training at scale is bandwidth and latency-sensitive, so capacity fragmented across regions or availability zones is no substitute for a contiguous, well-provisioned cluster. Few locations can supply such a cluster at an acceptable cost, and the set that can shifts over time as provider footprints and individual data centre capacity change.

We therefore treat the ability to relocate as an infrastructure requirement in its own right. During `Thomson` model training, we have worked with multiple cloud infrastructure providers, with several rapid intra-provider regional migrations undertaken along the way to follow available capacity.

This bears directly on the principle of infrastructure sovereignty §4 defined in Section 1.1. We acknowledge that accelerator hardware remains a residual limitation that no maturity of open-source tooling removes. Portability does not eliminate this but changes its character: what we depend on is accelerator capacity, rather than any single source of it, and continuity of research rests on our ability to re-establish the stack wherever capacity is available. We argue that this is a meaningful movement along a sovereignty spectrum, as argued in Section 1.4.

5.1.2 Model Orchestration and Inference

We apply the principles of infrastructure sovereignty §4 and control over economics §5 in developing our LLM orchestration and inference solution. Developing this in-house has enabled upfront cost reduction in reserved instances. Additionally, a large number of contemporary models are still unavailable via enterprise options. In such cases, self-hosting allows us to include these models in our benchmarking efforts and better assess our models' parity with newer models that are unavailable by other means. Self-hosting allows our team a high degree of flexibility in hosting proprietary and internal models.

Since April 2026, this system has **served more than 40M requests** and **processed 370B** tokens, with **upwards of a 99.9% success rate**, with **cumulative use across the broader organisation reaching 146M requests** and **nearly 700B tokens in that same period**. In the case of GLM-5.2, self-hosting has saved 19.7% of costs, compared to third-party inference providers.

While managed vendor offerings arguably reduce total cost of ownership by streamlining complex engineering workflows and reducing onboarding friction into a single offering, this alternative to self-hosting introduces a material long-term risk of vendor lock-in, leaving organisations vulnerable to rising costs without a clear exit strategy to alternative infrastructure. Designing and operating our own LLM serving system allows us a greater degree of control over our uptime, roadmap and value proposition.

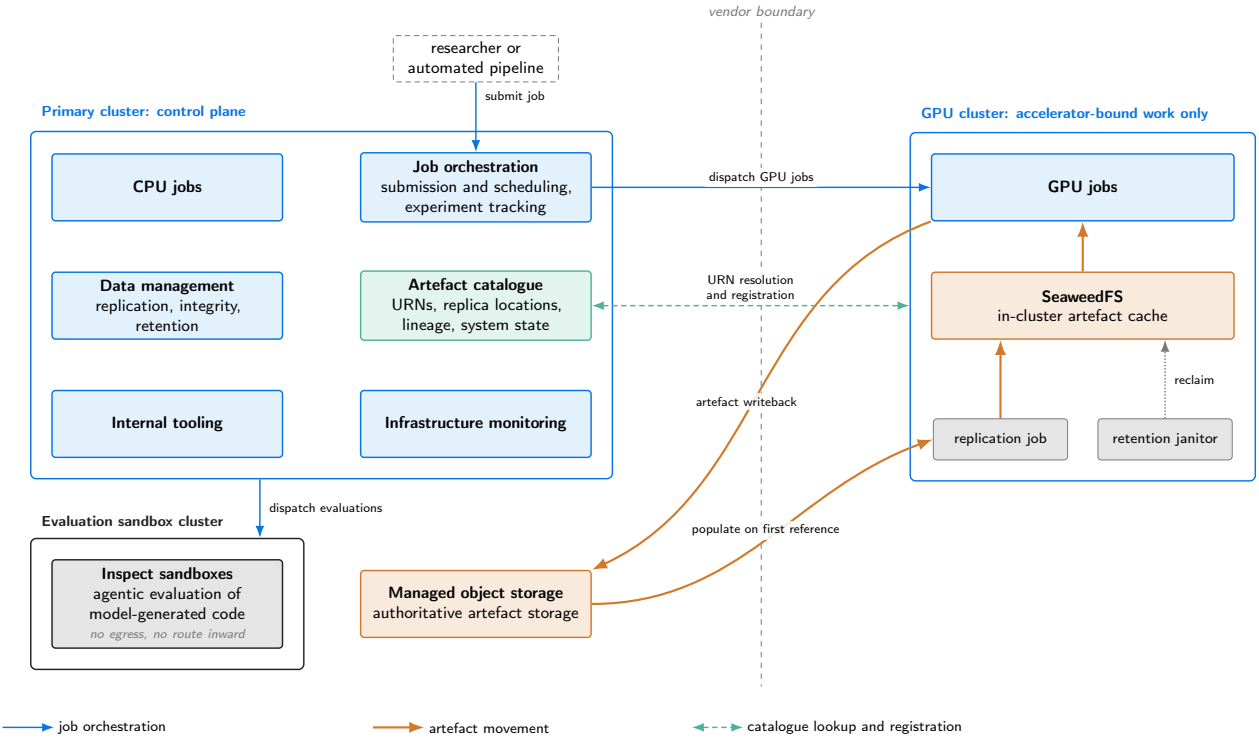


Figure 36 Infrastructure topology and the three flows that cross it. The control plane, the air-gapped evaluation sandboxes and canonical artefact storage all sit left of the vendor boundary, and do not change when the GPU partner does. Only one bulk path across that boundary is metered: an artefact is populated into the in-cluster cache on first reference, so egress is incurred once per cluster per artefact rather than once per worker per job, and every later read is served in-cluster.

5.2 Platform Foundations

In this section, we describe foundational capabilities which power our model development platform.

5.2.1 Cluster Topology

We implement our portability and cost efficiency principles via a three-tier cluster topology (see Figure 36):

- **Primary Cluster** – The control plane of the platform is hosted on our primary cloud provider. It hosts all CPU-only workloads (data preparation, evaluation harnesses, artefact management, and internal tooling) together with the stateful services on which the rest of the stack depends.
- **Evaluation Sandbox Cluster** – A second cluster, co-located with the primary cluster, dedicated to agentic evaluations executed via Inspect [157]. Because these evaluations execute model-generated code, the cluster is air-gapped and firewalled: sandboxes have no network egress and no route to internal systems. This isolation permits agentic evaluations to be run at scale without exposing the wider platform or external networks to the actions of the system under evaluation.
- **GPU Cluster** – A single accelerator cluster, provisioned by the current specialised compute partner, dedicated exclusively to GPU-bound workloads. No CPU-only work is scheduled here, which keeps scarce accelerator capacity fully committed to training and inference.

This separation of concerns yields two properties central to our operating model. First, the stateful, long-lived components of the platform remain anchored in a single, well-understood environment and are unaffected by changes in GPU provider or region. Second, workloads whose isolation requirements differ materially, namely routine training and the execution of untrusted agent-generated code, are held in physically distinct environments isolated from any network or internet access rather than by policy alone.

Month	Baseline	Egressed	Avoided
1†	5.72 TB	546.98 GB	5.17 TB
2	4.19 PB	22.00 TB	4.17 PB
3	2.56 PB	64.83 TB	2.49 PB
4	2.87 PB	76.61 TB	2.80 PB
5	4.06 PB	58.94 TB	4.01 PB
6	1.83 PB	39.21 TB	1.79 PB
7†	605.35 TB	16.54 TB	588.81 TB
Total	**16.13 PB**	**278.68 TB**	**15.85 PB**

Table 25 Data-transfer-out avoided by the in-cluster artefact cache on the GPU cluster, by month of operation; the data plotted in Figure 37. The baseline is the counterfactual in which every worker of every job pulls its inputs directly from the object store; *avoided* is that baseline less what was actually egressed. †Partial months.

5.2.2 Job Orchestration

Job submission, scheduling, and experiment tracking are handled by a substantially modified fork of the open-source ClearML [158] platform, extended to accommodate the heterogeneity of our execution environments. This is paired with an internally developed job submission and inspection interface. This abstracts the mechanics of the target execution environment to the orchestration layer. This is what makes relocation tractable: standing up a new GPU cluster, whether with a new partner or in a new region, alters the set of available targets without altering how jobs are written.

5.2.3 Data Access and Provenance

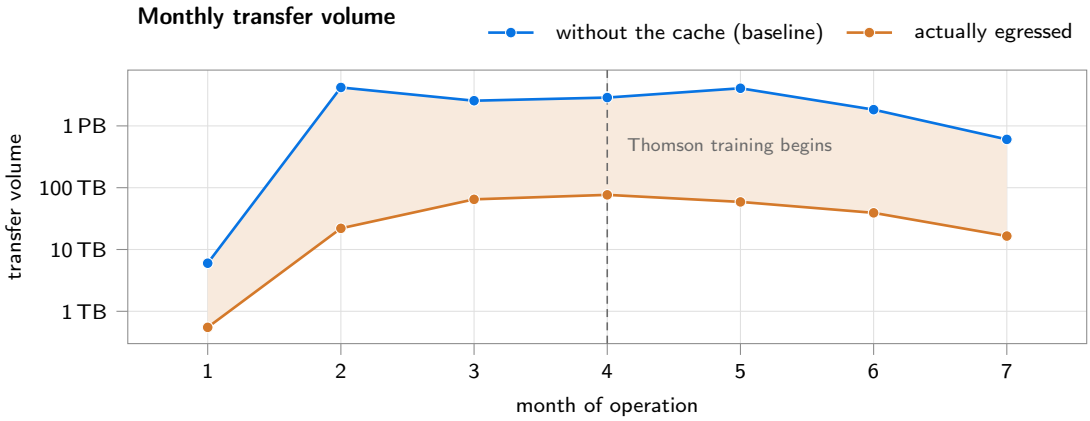


Figure 37 Data-transfer-out avoided by the in-cluster artefact cache on the GPU cluster, shown by month of operation on a logarithmic axis. The baseline is the counterfactual in which every worker of every job pulls its inputs directly from object store; the lower series is what was in fact transferred, and the shaded band between them is the transfer the cache avoided. Read demand over the window totalled 16.13 PB against 278.7 TB actually egressed, a factor of 57.9. The dotted rule marks the start of `Thomson` model training. Months 1 and 7 are partial. The baseline is a construction rather than a measurement, so this is transfer the cache avoided rather than budget that would otherwise have been spent.

Hosting compute and control planes in separate clouds introduces a data-gravity problem. Training artefacts (datasets, checkpoints, and tokenised corpora) must be made available to the GPU cluster at high throughput, while egress from the primary cloud is metered and therefore materially expensive. Naïve replication strategies are prohibitive on both latency and cost grounds. To address this, we implemented file access indirection – a unified reference to artefacts through stable logical identifiers rather than physical storage locations. This allowed the platform to resolve each reference to the most appropriate local replica at execution time. The stack is built upon two open-source foundations, DataHub [159] and SeaweedFS [160].

Artefact Identity DataHub is the source of truth for the indirection layer. Every model, dataset, and derived artefact is registered under a DataHub Uniform Resource Name (URN), a stable and globally unique identifier that names the asset independently of where any copy of it currently resides. These URNs constitute the shared vocabulary of the platform. Canonical storage for datasets and models is always in the primary cloud object store, which holds the authoritative copy of every artefact.

Replica Resolution and Transfer Cost Alongside canonical storage, DataHub records the availability of local replicas on a per-cluster basis, including those held in the SeaweedFS cache co-located with the GPU cluster. Resolving a URN involves consulting the catalogue for a replica local to the executing cluster. Where none is registered, the artefact is first populated into the local cache from canonical storage and the read is then served locally, with the new replica recorded in the catalogue. Egress from the primary cloud is thus incurred once per cluster-per-artefact, rather than once per worker-per-job. The distinction is substantial (Table 25): between February and August 2026 read demand on the GPU cluster totalled **16.13 PB against 278.7 TB actually egressed**. We estimate this led to savings of over **USD 1.4M** in avoided egress costs in this period. Absent the cache, every rank of a distributed job would fetch the same checkpoints and dataset shards independently, and the transfer volume would scale with the product of the job count and the worker count rather than remaining fixed per artefact. Every subsequent consumer, whether another rank of the same job or a job run months later, reads at in-cluster bandwidth. Because resolution occurs at the moment of use, replicas may be created, evicted, or invalidated as clusters come and go without affecting any reference held elsewhere.

In-Cluster Cache SeaweedFS provides the physical substrate that URN resolution targets inside the GPU cluster, operating as a local cache through which every model and dataset is replicated before it is read, so that reads are served from within the same network fabric as the accelerators consuming them. Access is mediated by a custom client, which retrieves file metadata from the SeaweedFS filer and then downloads chunks in parallel directly from the volume servers holding them. Under this scheme a single reading process sustains approximately 160 Gbit/s from the storage tier into memory, and approximately 40 Gbit/s when each file is additionally written to local NVMe. Both figures are rates observed by one consumer rather than aggregates over concurrent readers, and the lower of the two is bounded by local disk rather than by the network or by the storage tier. Checkpoint and dataset loading are consequently removed as a bottleneck at job startup.

Lineage and Governance. The same catalogue records lineage. Every artefact is registered with the job that produced it and with each job that subsequently consumes it, yielding a complete, queryable provenance graph across the training pipeline. This record supports reproducibility and post-hoc analysis: for any released checkpoint, the constituent datasets, preprocessing stages, and upstream training runs can be recovered exactly. Beyond provenance, the catalogue exposes user-defined tags and properties, operationally critical system-level state like replication status, cache LRU state, file integrity information, and related bookkeeping. This also gives us a real-time provenance view as an operational artefact, critical for demonstrating controls for highly regulated environments.

5.2.4 Tool Calling

Tool calling is critical to ground the model's responses in accurate citations and factuality. To standardise this across the model development lifecycle, we developed a unified tool-calling infrastructure to enable calls to proprietary and external tools. This helped us standardise request-response schemas, authentication, and state management. We adopted an async-first design, which helped manage the high volume of concurrent tool calls across the training run. We keep a low memory footprint (approx. 2MB) per worker to enable lightweight scaling across distributed training. We also accounted for partial failures for tool calls to avoid blocking training runs.

5.2.5 Synthetic Data Generation

Our post-training pipeline consumed a large number of distinct data collections, each requiring generation, filtering, validation and publication before it could enter a training mixture. We treat synthetic data generation

also as pipeline infrastructure, to be reused across all data collections.

We unify our pipeline for model requests against a tabular dataset, agnostic of model inference provider. This layer also manages connection pooling, request batching, error isolation and handling.

The pipelines are modelled as directed acyclic graphs (DAGs) comprising independently versioned steps of generation, filtering and scoring. Each step is independently executable, affording the capability of prompt-level iteration. This also helps in lineage tracking and artefact provenance for every transformation – including query hash, source text, reasoning chain and reference answer, allowing complete traceability for each training run. This is a fundamental requirement for heavily regulated environments in which `Thomson-1.0-Large` is expected to be used.

5.3 Training Environment

`Thomson` models are trained on the cluster as described in Section 5.1.1, with nodes of 8x Nvidia B200. Communication between nodes is over an InfiniBand network, while intra-node communication is achieved via NVLink and NVSwitch. The typical number of nodes used for training is reported by task in Table 26.

Model	Role	Nodes	GPUs	Parallelism
Thomson-1.0-Small	CPT	16	128	TP1, PP1, CP1, EP8, ETP1
	DPO	6	48	TP1, PP1, CP1, EP8, ETP1
	Policy training	8	64	TP2, PP1, CP8, EP8, ETP1
	Rollout generation	4	16	TP8 (4 independent engines)
Thomson-1.0-Large	CPT	16	128	TP1, PP8, CP16, EP8, ETP1
	DPO	16	128	TP1, PP8, CP16, EP8, ETP1
	Policy training	16	128	TP1, PP8, CP16, EP8, ETP1
	Rollout generation	10	80	TP8 (10 independent engines)

Table 26 Compute cluster configurations by training task.

For RL, generation and training are *non-colocated*: each runs on its own dedicated node pools, which is a precondition for the asynchronous rollout collection described in Section 5.1.1.

As discussed in Section 5.3.1, we run our training workloads with the Nvidia NeMo-RL [161] framework. Training jobs in NeMo-RL are run over a Ray [162] cluster that spawns instances from different resource groups, namely for training and generation; this allows us to assign each node a specific role and isolate generation resources from training ones to avoid colocation and enable asynchronous rollouts. See Figure 38 for a schematic representation.

5.3.1 Training Stack

Our training library extends Nvidia NeMo-RL, with Megatron-LM [163] core to provide model parallelism and vLLM [87] for generation. NeMo-RL implements distributed worker orchestration via Ray, the GRPO and DPO algorithms, and checkpoint tooling based on PyTorch Distributed. On top of this we implement 28 custom extensions together with custom NeMo-Gym [89] RL task environments and a collection of specialised reward functions.

We implemented a few distinct extensions of NeMo-Gym to support our training framework: avoiding zero learning signal, policy-gradient variants, and failure-resilient cohort scoring. Some of these are NeMo-specific feature changes rather than customisations for our use case, and hence could be future contributions to the library.

Avoiding Zero Learning. Learning with GRPO is only possible if completions of the same prompt receive different rewards. We noticed two issues in the NeMo-RL framework that prevent different completions in the same group from receiving distinct rewards in the context of agentic multi-turn tasks.

Sampling diversity – For each generation engine, the framework feeds one shared seed but does not specify a per-request seed. We assign a different unique seed to every generation request to prevent identical outputs

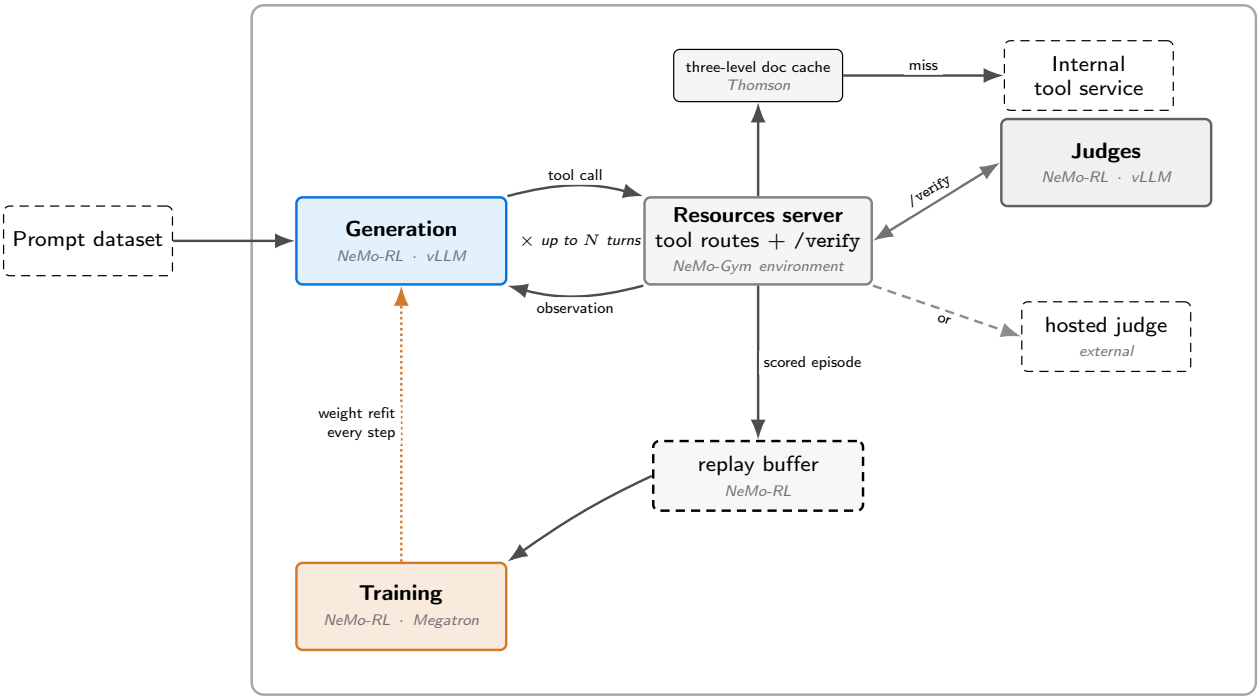


Figure 38 The training loop across the three node pools, annotated with the component that provides each stage. Generation, training and judge inference occupy disjoint node pools. Scoring is a call the resources server makes, to the on-cluster judge pool or optionally to a hosted model. The replay buffer decouples generation from training, which is what makes the two pools run concurrently and is where the trajectory-staleness bound applies.

and preserve the reward variation GRPO needs to learn.

Advantage Grouping in Multi-Turn Rollouts – Samples are grouped by the full conversation history. Because tools are added as user messages, which differ across trajectories, this generates N different groups for each original prompt, each with a single sample and thus zero advantage. We modify the grouping mechanism by tracing only the first user message and collecting all trajectories under this message. This ensures a group with N generations, all likely to be different. Single-turn tasks are not affected by these issues in the original framework, and behave as expected.

Policy-Gradient Variants. Policy-gradient loss uses a clipping ratio and a separate correction for differences between the trainer's policy and vLLM's sampling policy. We add variants, all disabled by default, so that the original behaviour is preserved unless explicitly enabled.

Length-normalised train–inference ratio – In the sequence-level GSPO loss, the train–inference correction was originally computed as

$$\exp\left(\sum_t \left(\log \pi_{\text{old}} - \log \pi_{\text{gen}}\right)\right) \sim \exp\left(N_t \epsilon\right),$$

so it increased with response length. When enabled, this option instead uses a masked mean over tokens:

$$\exp\left(\text{mean}_t \left(\log \pi_{\text{old}} - \log \pi_{\text{gen}}\right)\right) = \exp\left(\frac{1}{N_t}\sum_t \left(\log \pi_{\text{old}} - \log \pi_{\text{gen}}\right)\right) \sim \exp\left(\epsilon\right)$$

matching the policy ratio and removing length bias for responses that vary widely in size.

Reference-Model Removal – In GRPO, the reference model can be used to compute the KL penalty. If the penalty weight is fixed to zero, the log-probabilities computation is pure overhead and the framework provides a path to avoid computation for the synchronous training loop. We extend this guard to the asynchronous case and moreover introduce the possibility of avoiding reference model initialisation as well, to save memory.

Cohort Reward Reconciliation. In practice, individual rewards can fail because of judge timeouts or backend throttling. Simply omitting a failed reward means that completion's weighted score is averaged over fewer terms than its peers. Resulting scores end up on divergent scales, so advantages may reflect infrastructure reliability, rather than response quality. This creates systematic bias whenever failures are unevenly distributed within a cohort. To mitigate this, we implemented a mean substitution policy: missing rewards are replaced with the cohort mean among completions where that reward succeeded, preserving a consistent weight basis. This substitution artificially reduces variance among the affected rollouts, however – and under advantage normalisation schemes that divide by a group's reward standard deviation, this variance collapse can inflate the relative advantage of unaffected rollouts in the same group, distorting the training signal in proportion to how much of the cohort was imputed. As a correction measure, we track each rollout's proportion of genuinely observed versus imputed reward and use it to scale down its group's advantage – discounting, rather than discarding, cohorts that relied heavily on substitution. Rollouts whose reward is entirely imputed or failed outright are excluded from the loss altogether.

5.3.2 RL Environments

For training `Thomson`, we implement RL environments as a generic HTTP interface following NeMo-Gym syntax, detailed as follows:

Multi-Turn Legal Research environment. This is the principal environment with a focus on legal research, a key modelling objective. It feeds the model legal research questions and lets it complete the task with internal tools dynamically.

Secure Local Sandboxed Workspace environment. A dedicated environment to safely run model-authored shell commands (read, write, edit, glob and grep) local to the training cluster: each tool call is executed inside an Apptainer namespace, where the host filesystem is not visible, network namespace is empty, read-write is limited to a per-session /tmp folder. This environment is designed for small workloads and complements the sandbox cluster discussed in Section 5.2.1.

System Prompt Context Management for Tool Results. In Section 5.2.4, we describe our platform's tool calling capabilities. We extend these for our training needs by introducing a context management layer. Many tools return long text documents, which if directly added to the model's system prompt can bloat the model context quickly. To mitigate this, our context management layer enforces token limits for call results, structure-aware truncation of tool results and summarises long results.

Document Caching for Tool Call Optimisation. We use our tool calling layer described in Section 5.2.4 for calling internal tools during training. This allows dynamic inclusion of more tools to increase the model's actionable functionality. A concrete example is the document retrieval required for evaluating claims' factuality. To avoid redundant calls to internal tools and optimise costs, we implement a three-tier cache for retrieved documents (Figure 39).

- **Level 1 – In-process memory.** Each worker maintains bounded, in-memory lookup structures for resolution outcomes and document bodies, evicted under a least-recently-used policy, caching only successful lookups. The document-body cache is sized by cumulative byte budget rather than entry count since response length varies enormously and capped at the lesser of a configured ceiling and a fraction of currently available system memory, with retained text compressed to extend the effective working set. All entries are zstd-compressed.
- **Level 2 – Shared filesystem.** A filesystem-backed cache, shared across processes, runs, and (on network storage) nodes, holds two parallel structures: a document store, partitioned by jurisdiction and sharded

by GUID prefix to bound directory size; and a citation-resolution store, keyed by document GUID, with each record accumulating every distinct citation surface form observed to resolve to it – since one authority is routinely cited in several superficially different but equivalent forms. Writes are atomic (temporary file plus rename) for safety under concurrent access, and all filesystem operations are timeout-bounded and run on an isolated thread pool, so a stalled network filesystem degrades to a cache miss rather than stalling scoring. The in-memory tier may be pre-warmed from disk at startup to avoid a cold ramp-up.

- **Level 3 – Cross-run archive.** The cache directory is pushed to our internal data platform as a versioned dataset, accessible to any future run limiting the cost of retrieval over multiple runs.

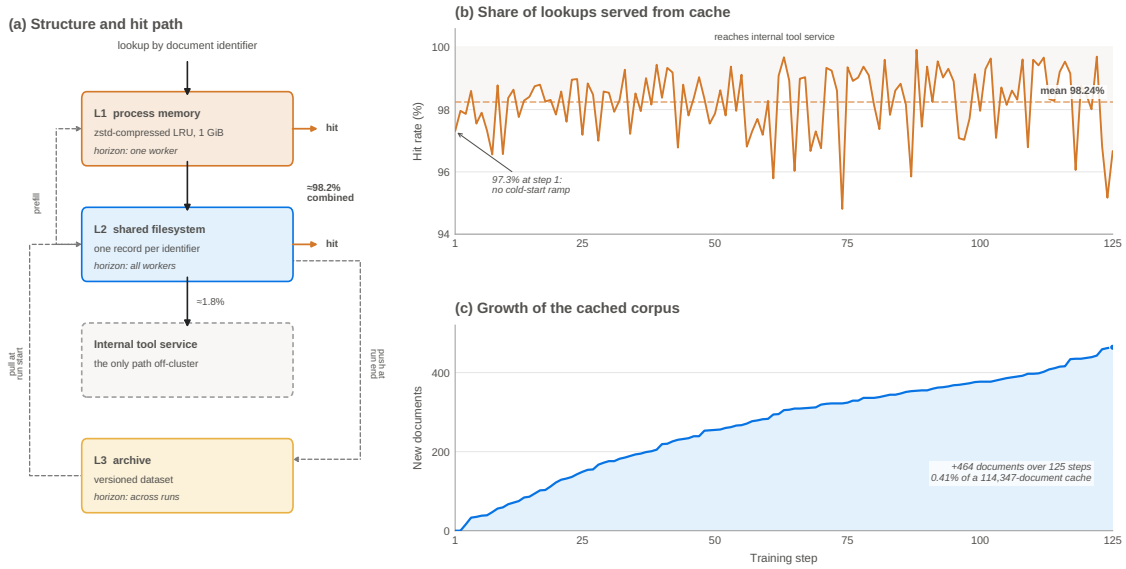


Figure 39 The three-level document cache, with panels (b) and (c) measured over a 125-step test run. **(a)** A lookup falls through the tiers, each answering a different reuse horizon: within a worker, across workers, and across runs. Only a level-2 miss leaves the cluster. Two out-of-band paths matter as much as the read path – L2 prefills L1 when a worker starts, and the L2 directory is pushed to the archive at run end and pulled back at the start of the next. **(b)** Across 329 389 lookups (≈2 600 per step) the cache answered 98.2%, leaving 5 763 calls – about 46 per step – to reach the internal tool service. **(c)** The cached corpus grew by 464 documents, 0.41% of its starting size, and the rate of new documents roughly halves across the run (149 in the first 25 steps against 87 in the last 25) – the run converges onto a working set that the archive already contained. Consistent with that, no L1 evictions occurred.

Two pathways populate the same persistent structures using an identical record format, so each benefits from the other's writes. While evaluating a claim's citations, the system resolves each citation via the resolution store (falling back to the backend on a miss) and satisfies the resulting document request via the document store (falling back to a live fetch), writing successful results back to both stores. When the model invokes a document-retrieval tool during generation, the request is checked against the cache first; a live response carrying a canonical identifier is written into the same store. So documents the model already fetched either during generation or claim evaluation are not fetched again.

5.3.3 Training Performances

Among the performance tests for our training stack, we measured, for a large MoE model (`Qwen3.5-397B-A17B`), the impact of decoupling generation from training to separate GPU pools (non-colocate strategy) rather than colocating them on shared nodes. Comparing performance metrics under identical batch, sequence-length, and parallelism configuration, non-colocation reduced total step time by 13.1% (see Figure 40), with the gain concentrated almost entirely in the **generation phase (-32.3%)**; training and logprob computation were essentially unchanged (+0.4% and -3.4% respectively), as expected since both use identical PP=8 parallelism regardless of colocation strategy. The effect is also measurable in raw throughput: non-colocation delivered a

61.7% increase in Generation Worker Group tokens/sec, despite the colocated run using two times the generation parallelism (TP/EP=16 vs. 8).

These results show that the main drawback of sharing GPUs is not the parallel setup: it is that training and generation compete for the same GPU memory and computing power, even though they need those resources differently. Giving each task its own GPU pool greatly improves throughput with little extra cost.

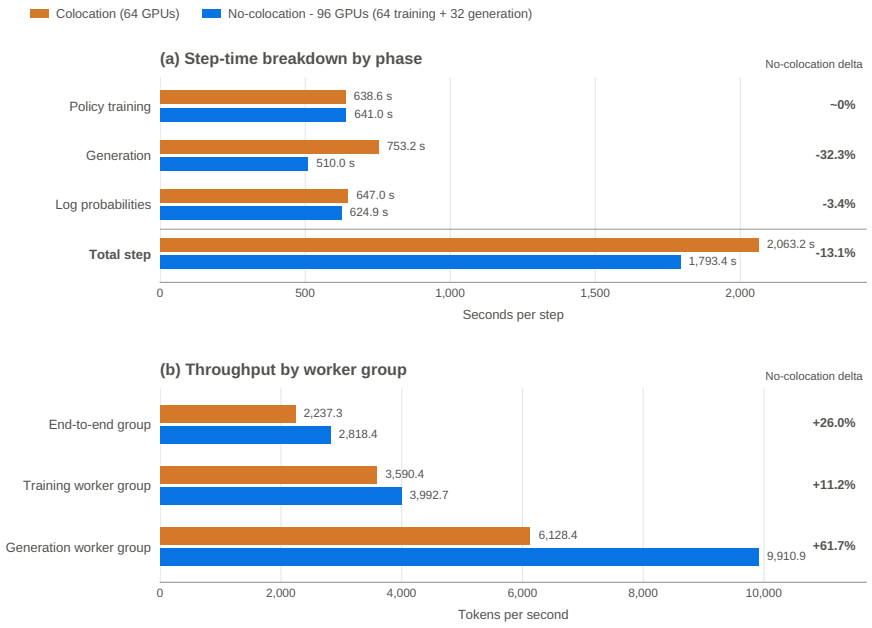


Figure 40 Comparison between `colocation` (64 GPUs, colocated generation and training) and `no-colocation` (64 training GPUs plus a 32 GPUs dedicated generation pool). "WG" denotes Worker Group, the pool of GPUs assigned to a given role.

5.4 Inference Infrastructure

5.4.1 Architecture

Our LLM orchestration and inference platform is built on Kubernetes, with a routing, authorisation, and rate-limiting plane implemented in Rust, using vLLM [87] as an inference backend. It operates over 11 independent clusters spread across 4 geographies and 2 cloud providers, and the platform as a whole is divided into isolated functional domains – clusters operate as routing clusters or inference clusters, not both. This allows for a much more proactive approach to blast radius containment and significantly more deterministic failure modes; while the result is an incremental increase in the overall management complexity in one respect (i.e., the number of clusters is higher than it might be otherwise), it ensures that the issues that can occur in any given cluster have fewer interacting (confounding) failures. Further, routing being decoupled from inference means that either can be more straightforwardly experimented with – a new routing environment can be provisioned from scratch, tested, and thrown away within the same business day, allowing a greater degree of lifecycle decoupling than we would be able to achieve otherwise, all ultimately contributing to a greater degree of uptime.

We adopt cross-cluster service mesh routing to allow transparent service discovery and communication. Further, it allows for a more proactive approach to failover; if GPUs in one region are saturated, we can opt to trade KV cache locality for what would otherwise be a slower or outright failed request. Mesh routing also enables strong adherence to data residency laws; we can explicitly enable latency-tolerant requests originating in the US to utilise GPUs in other regions if necessary, while eliminating completely the possibility of workloads sensitive to data residency requirements being sent in the other direction; even if our workloads were misconfigured, the lack of meshing back to disallowed geographies acts as a second line of defence preventing inadvertent

violations. Between our approach of fault-tolerant workload isolation and mesh routing, a small team is able to maintain a multi-continent compute footprint. Figure 41 shows a high-level flow of an inference request through this system.

We isolate our team-internal research workloads into a dedicated experimental environment. This allows us to move fast with newer model architectures, test a new quantisation scheme, evaluate a speculative-decoding setup, or perform load and shadow tests for a representative workload, target hardware, and KV-cache capacity. It also allows feature rollouts to a smaller demographic in those cases where a feature may be short-lived or otherwise unsuitable for a broader production audience, but also to allow our team to dog-food those features that are ultimately intended for production, allowing a greater feedback loop and quicker iteration.

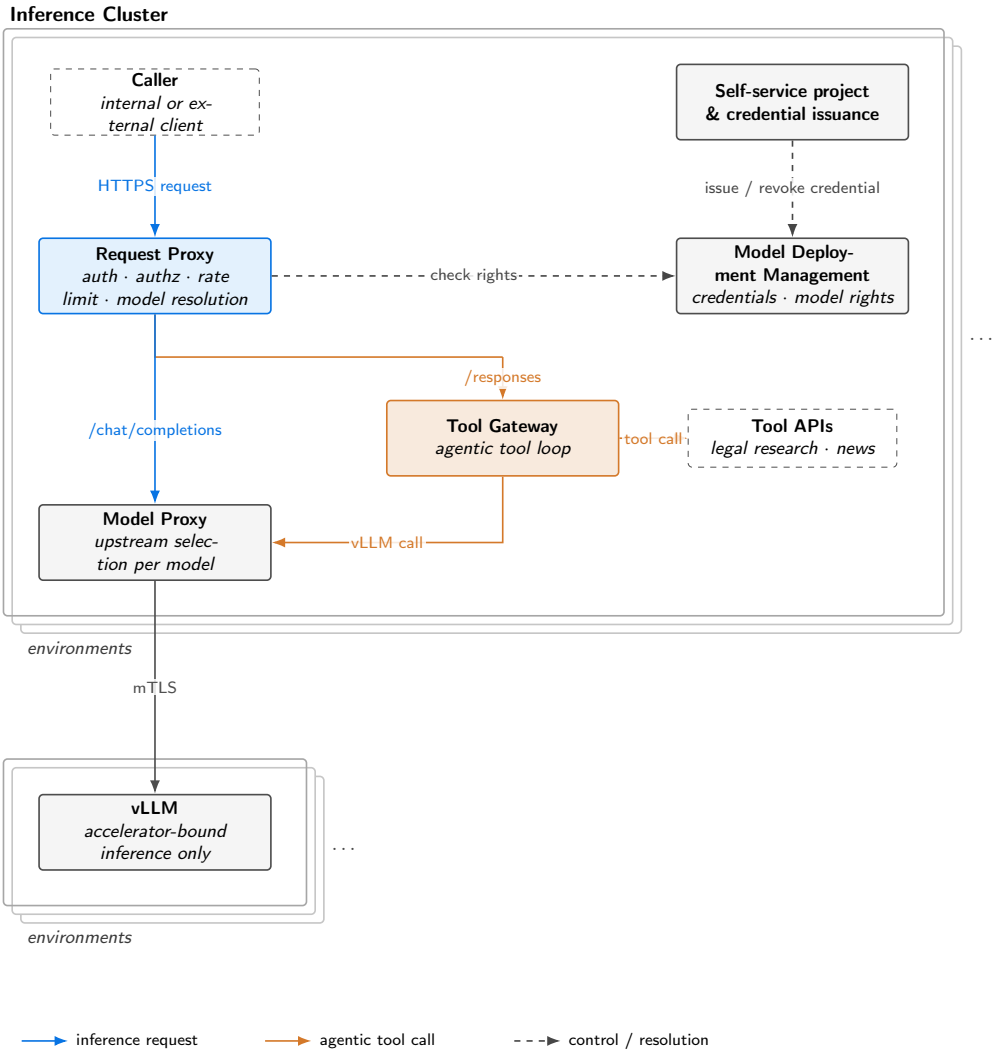


Figure 41 Request routing across the control plane and GPU compute clusters. Authentication, routing, and agentic tool execution run on GPU-free routing clusters; model inference runs on separate GPU clusters. Self-service credential issuance and model-rights checks both route through the same deployment-management service and associated persistence layer consulted on every inference request. After the first network segment (caller to proxy, HTTPS) all inter-service communication is secured via mTLS.

We also support agentic tool-calling traffic through this platform, supporting tool calls to proprietary sources as part of our inference setup. This is exposed through an API mirroring OpenAI's Responses API, since agentic, multi-turn tool use is what that specification is built to describe; moreover, its compatibility with an established API enables new users to quickly onboard using tools with which they're already familiar.

Similarly, we enable Optical Character Recognition, ensuring a greater degree of efficiency in upload handling in comparison to multi-modal approaches, while also allowing us to maintain guarantees regarding data persistence where appropriate.

5.4.2 Inference Server Customisations

We customise vLLM to support novel models and features prior to availability of longer-term support in the vLLM upstream. We introduced the ability to define a thinking budget client-side by introducing custom logits processors, and implemented custom reasoning parsers to support models not otherwise natively capable of reasoning traces, ensuring that our inference platform is capable of transparently (and correctly) supporting a mix of model functionalities and API features that are not yet available in the broader third-party/open-source ecosystem.

Tying back to our infrastructure sovereignty and cost-optimisation principles, a small number of optimisations have yielded the largest benefit.

- **Server Initialisation** – LLM weights increasingly have larger and larger memory footprints, and the overhead of moving almost a terabyte of data around can result in extremely long start-up times, often 20 minutes or longer. Long start-up times can introduce uptime risk – if a replica takes 30 minutes to start, other replicas may be overloaded for the duration, ultimately risking a cascading failure and marked degradation in quality of service. By shifting to an in-memory volume (as opposed to general purpose SSD or NVMe disks) we can ultimately reduce the time taken to retrieve weights from long-term storage and allow for quicker transfers to GPU devices; while writing to disk would prevent the necessity of redownloading weights, it'd also incur slower transfer to GPU unless a tiered approach were taken. Further, shifting to use of more fit-for-purpose faster-loading libraries (e.g., instanttensor), we shift the bulk of the start-up bottleneck to graph initialisation, etc., allowing the bulk of our models to initialise in approximately five minutes, and often less.
- **Quantisation** – We use vLLM's llm-compressor library for dynamic post-training quantisation, normally to FP8. This does not require a calibration dataset, since activation quantisation happens at inference time rather than being fit in advance. To avoid measurable degradation in output quality, we leave layers such as the language-modelling head (`lm_head`) and gating and routing layers for mixture-of-experts (MoE) models unquantised. Across our evaluations, a quantised model's task performance remains close to its full-precision counterpart.

6 Conclusion & Future Work

6.1 Conclusion

The gap between open-weight models and the best closed systems has narrowed from years to months [2], and the most expensive stage of the pipeline – large-scale pre-training – is the stage whose returns are flattening fastest. Taken together, sovereignty no longer turns on the capital required to train a model from scratch; it turns on whether an institution has the technical capability to implement its sovereignty goals rather than being merely capable of hosting a model without modification or engaging in narrow fine-tuning exercises. Public discourse has been clear that this matters, and comparatively quiet on how it is done. This report is an attempt to answer the second question with a worked example rather than an argument.

In this report, we argue that Continual Learning enables broad implementation of SovereignAI goals, spanning a wide range of desiderata (§1–§5) beyond pure model performance. Adaptation of open-weight models has historically been a trade-off: capability bought in a narrow slice of the target domain paid for with skills lost elsewhere. By treating stability and plasticity as explicit objectives at every stage, we show how model ownership and customisation become feasible on modest budgets.

Our central empirical result is that the outcome of the methods described is better than a favourable trade. `Thomson` exhibits a distinctive π-shaped profile: pronounced gains in the domains we aimed for, alongside preserved – and frequently improved – performance in domains not targeted. More important than the concrete model or its current performance is our finding that this process can be repeatedly and reliably applied, with various modules of our model development pipeline having been applied to a number of different base models throughout the development of the `Thomson` model family.

At the core of this report is an argument about both values and economics: `Thomson` was developed on a comparatively very modest budget and has nevertheless withstood the scrutiny of careful, publicly recognised evaluations, adversarial safety testing, blind human studies, and the real requirements of production environments. Nevertheless, we are deliberate about what this does and does not establish. Sovereignty here is a spectrum rather than a threshold, and we have moved along it rather than arrived: model and data sovereignty are substantially achievable (§1, §2), governance and value alignment demonstrably improvable (§3), infrastructure and economic control largely attainable on open-source foundations (§4, §5) – while the dependence on accelerator hardware is not removed. The open-weight starting point invites the same objection, which we address in Section 1.4: with each generation able to begin from the checkpoints the previous one produced, the distance from that original dependence grows until it is of little practical consequence.

Finally, Section 5 also highlights the critical role high-quality engineering plays in frontier model development, and how the intelligent use and assembly of an increasingly mature open-source landscape can enable infrastructure sovereignty (§4).

6.2 Future Work

Scaling mid-training. Our mid-training budget was held at deliberately modest 200B tokens, pruning over 98% of the $\approx$ 19T candidate pool, a choice made to explore the quality of the model that can be achieved with modest compute budgets. While our data-centric filtering ensures that these 200B are likely of the highest quality, we believe that enlarging that pool remains one of the most direct levers on the results in Section 4. This is due both to knowledge injection and the increasingly common observation that high-quality mid-training can make subsequent training stages more effective. An increasingly close co-ordination between mid- & post-training is likely to maximise this effect. In addition, various data curation methods more targeted at use cases are of interest: filtering that suppresses rather than propagates hallucinations, and grouping related documents so that multi-document reasoning – where current models remain weakest – is trained explicitly rather than incidentally. Finally, an interesting ablation study could also explore the trade-off between training all weights for a restricted budget versus training on a larger corpus but using parameter-efficient training routines (holding total compute constant). Furthermore, we believe that data-centricity could be applied through data-prioritisation schemes [e.g. 164–166] during mid-training, thereby dynamically balancing plasticity and stability.

Reward design and RL stage. Our findings in the RL-phase of model development confirm the observation [e.g. 167] that online RL is a powerful component of a full Continual Learning stack due to its tendency to provide robust learning with little to no forgetting. In addition to more closely aligning post- with mid-training, it is clear that the careful reward design and an increasingly mature approach of heavily leveraging LLM judges are highly likely to continue succeeding. Going forward, we hope to fully replace any reliance on external model-based judges by leveraging previously trained `Thomson` generations, further increasing our level of sovereignty (§1). Overall, the direction is clear: RL is now mature enough to substantially improve model quality across a wide range of *non-verifiable* domains. Thus, sustained model improvements will emerge based on builders' ability to scale up the sophistication of RL environment-building efforts.

The role of coding skills. Coding is the one area showing mild forgetting relative to open-weight models we start with. While we did not intend to improve coding specifically, general computer use and coding skills are increasingly load-bearing for agentic skills, which, in our estimation, have the highest chance to lead to real automation across the economy. Hence, we intend to measure more carefully and make greater use of the myriad open datasets designed to target coding and SWE skills, but have no doubt that the modest performance drop can be easily alleviated without the need for algorithmic breakthroughs. In addition, given the overwhelming focus on coding skills among the vast majority of model developers, we believe that much of the absolute capability gap relative to other frontier models on coding can be easily addressed through the choice of open-weight models.

Values and constitutions. Alignment to a written constitution remains an aspiration rather than a solved problem, with perfect alignment being considered one of the hardest problems in AI Safety. Nevertheless, we regard our value re-alignment and Constitutional RL results as evidence that the direction is tractable rather than that the destination is reached. A valuable and likely impactful engineering contribution would also see the introduction of an adversarial red-teaming algorithm running throughout training, ensuring that the model is continuously challenged on-policy. Finally, all of the contentious issues presented have been deliberately written by human experts with backgrounds in politics, international relations, history, and law. While this allowed us to ensure data quality and remove any concerns about hallucinations, we recognise that this may be difficult for all organisations. As such, we believe it may be valuable to *carefully* design a contentious issue design pipeline, making the entire Constitutional alignment process fully automated.

Compounding generations. Finally, the argument in Section 1.4 invites its own experiment. If each model generation can be built from the last, the interesting question is what accumulates: whether successive rounds of Continual Learning compound in capability and institutional fit, or whether they accrue drift that eventually requires returning to a fresh open-weight checkpoint.

Acknowledgements

Our work on `Thomson` is made possible by the dedicated efforts of many others who offered their insight, support, and expertise to this project. The subject matter experts who supported this project are too many to name, but we particularly want to thank our core subject-matter expert team, who have worked alongside us and taught us about their domains as we endeavoured to embed their expertise into `Thomson`.

Core Subject Matter Experts

(unordered)

- Connie Quarnstrom
- Elizabeth Botsford
- Jessie Shearer
- Daniel Calloway
- Andrew Coyne
- Carinne Davis
- Jennifer Nist
- George Pagano
- Susan Rose
- Samuel Vincent
- Zoe Callinan
- Vera Mayzel
- Elinor Nikolova
- Luca Patriniche
- Pukar Soni
- Nikolas Vucekovich
- Jodi Gardner
- Poorna Mysoor
- Sara Catley
- Brian Birke
- Michelle Graham
- Kevin McNamee
- Brandan Oliver
- Mike Plambeck
- Ian Williams
- Louise Jones
- Marco Rinaldi
- Tiffany Hildreth
- Kathy Wood

We want to thank *Jessie Baek* and *Adam Greshowak* for coordinating our team, our many colleagues within Thomson Reuters for their support and insightful questions and comments, and our former Foundational Research colleagues for contributions to earlier versions of the `Thomson` project. We also thank our partners at Imperial College London, DatologyAI, and Lambda. Finally, we thank *Joel Hron*, *Steve Hasker*, and *Alexander Kardos-Nyheim* for their leadership and support.

References

[1] Imperial College London. Cheap and effective re-alignment of frontier models through capability-preserving model steering. Technical report, August 2026. URL https://huggingface.co/spaces/tri-fair-lab/publications/blob/main/Frontier_Model_Realignment.pdf.

[2] AI Security Institute. How far behind the frontier are leading open weight models on cyber?, 2026. URL https://www.aisi.gov.uk/blog/how-far-behind-the-frontier.

[3] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[4] Kyle O'Brien, Stephen Casper, Quentin Anthony, Tomek Korbak, Robert Kirk, Xander Davies, Ishan Mishra, Geoffrey Irving, Yarin Gal, and Stella R Biderman. Deep ignorance: Filtering pretraining data

builds tamper-resistant safeguards into open-weight llms. In *International Conference on Learning Representations*, volume 2026, pages 35579–35633, 2026.

[5] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[6] Luca Patriniche, Bradley Bell, Dietrich Trautmann, Nikolas Vucekovich, Zoe Callinan, Pukar Soni, Ian Williams, Andrew Coyne, Manpreet Nanreh, Kirsty Fielding, Wassim Seifeddine, Felix M. Simon, Yejin Bang, and Jonathan Richard Schwarz. The public AI constitution project. Technical report, August 2026. URL https://huggingface.co/spaces/tri-fair-lab/publications/blob/main/Public_AI_Constitution.pdf. Luca Patriniche, Bradley Bell and Dietrich Trautmann contributed equally as joint first authors; Yejin Bang and Jonathan Richard Schwarz are joint senior authors.

[7] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[8] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

[9] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

[10] Zhenyi Wang, Enneng Yang, Li Shen, and Heng Huang. A comprehensive survey of forgetting in deep learning beyond continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1464–1483, 2024.

[11] Lukas Thede, Stefan Winzeck, Zeynep Akata, and Jonathan Richard Schwarz. Captrack: Multifaceted evaluation of forgetting in llm post-training. *arXiv preprint arXiv:2603.06610*, 2026.

[12] Lexin Zhou, Lorenzo Pacchiardi, Fernando Martínez-Plumed, Katherine M Collins, Yael Moros-Daval, Seraphina Zhang, Qinlin Zhao, Yitian Huang, Luning Sun, Jonathan E Prunty, et al. General scales unlock ai evaluation with explanatory and predictive power. *arXiv preprint arXiv:2503.06378*, 2025.

[13] Andrew M Bean, Nabeel Seedat, Shengzhuang Chen, and Jonathan Richard Schwarz. Scales++: Compute efficient evaluation subset selection with cognitive scales embeddings. *arXiv preprint arXiv:2510.26384*, 2025.

[14] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[15] Mingxuan Du, Benfeng Xu, Chiwei Zhu, Licheng Zhang, Xiaorui Wang, and Zhendong Mao. Deepresearch bench: A comprehensive benchmark for deep research agents. In *International Conference on Learning Representations*, volume 2026, pages 42414–42448, 2026.

[16] Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Huichi Zhou, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, et al. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*, 2025.

[17] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

[18] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

[19] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137, 2024.

[20] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

[21] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[22] Renjun Xu and Jingwen Peng. A comprehensive survey of deep research: Systems, methodologies, and applications. *arXiv preprint arXiv:2506.12594*, 2025.

[23] Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G Finlayson, David Sontag, et al. Dr tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*, 2025.

[24] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *International Conference on Learning Representations*, volume 2024, pages 9025–9049, 2024.

[25] Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

[26] ByteDance. DeerFlow: Deep exploration and efficient research flow. https://github.com/bytedance/deer-flow, 2025.

[27] LangChain. LangGraph. https://langchain-ai.github.io/langgraph/, 2024.

[28] Anthropic. How we built our multi-agent research system. Anthropic Engineering Blog. https://www.anthropic.com/engineering/multi-agent-research-system, 2025.

[29] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[30] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[31] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[32] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.11717.

[33] Nora Belrose. Diff-in-means concept editing is worst-case optimal. EleutherAI Blog, December 2023. URL https://blog.eleuther.ai/diff-in-means/. Accessed 2026-08-17.

[34] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861.

[35] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[36] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwińska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

[37] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[38] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

[39] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[40] Ishan Jindal, Chandana Badrinath, Pranjal Bharti, Lakkidi Vinay, and Sachin Dev Sharma. Balancing continuous pre-training and instruction fine-tuning: Optimizing instruction-following in llms. *arXiv preprint arXiv:2410.10739*, 2024.

[41] Shankar Padmanabhan, Mustafa Omer Gul, and Tanya Goyal. Updating parametric knowledge with context distillation retains post-training capabilities. *arXiv preprint arXiv:2602.16093*, 2026.

[42] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2): 123–146, 1995.

[43] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[44] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

[45] Michalis K Titsias, Jonathan Schwarz, Alexander G de G Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. *arXiv preprint arXiv:1901.11356*, 2019.

[46] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

[47] Pratyush Maini, Vineeth Dorna, Parth Doshi, Aldo Carranza, Fan Pan, Jack Urbanek, Paul Burstein, Alex Fang, Alvin Deng, Amro Abbas, et al. Beyondweb: Lessons from scaling synthetic data for trillion-scale pretraining. *arXiv preprint arXiv:2508.10975*, 2025.

[48] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[49] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[50] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[51] Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. Olmes: A standard for language model evaluations, 2024. *URL https://arxiv. org/abs/2406.08446*.

[52] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the association for computational linguistics: NAACL 2024*, pages 2299–2314, 2024.

[53] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[54] Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, et al. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural*

Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track, 2023. URL https://arxiv.org/abs/2308.11462. 40 authors total.

[55] David Heineman, Valentin Hofmann, Ian Magnusson, Yuling Gu, Noah Smith, Hanna Hajishirzi, Kyle Lo, and Jesse Dodge. Signal and noise: A framework for reducing uncertainty in language model evaluation. *Advances in Neural Information Processing Systems*, 38:17073–17114, 2026.

[56] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

[57] Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*, 2024.

[58] Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual pre-training of large language models: How to (re) warm your model? *arXiv preprint arXiv:2308.04014*, 2023.

[59] Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don't retrain: A recipe for continued pretraining of language models. *arXiv preprint arXiv:2407.07263*, 2024.

[60] Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93, 2024.

[61] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023.

[62] Kaituo Zhang, Mingzhi Hu, Hoang Anh Duy Le, Fariha Kabir Torsha, Zhimeng Jiang, Minh Khai Bui, Chia-Yuan Chang, Yu-Neng Chuang, Zhen Xiong, Ying Lin, et al. A survey on evaluating quality and trustworthiness in llm-generated data. *arXiv preprint arXiv:2601.17717*, 2026.

[63] Anton Alexandrov, Veselin Raychev, Mark Niklas Müller, Ce Zhang, Martin Vechev, and Kristina Toutanova. Mitigating catastrophic forgetting in language transfer via model merging. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17167–17186, 2024.

[64] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. Pmlr, 2022.

[65] Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. Biomistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the association for computational linguistics: acl 2024*, pages 5848–5864, 2024.

[66] Shamane Siriwardhana, Mark McQuade, Thomas Gauthier, Lucas Atkins, Fernando Fernandes Neto, Luke Meyers, Anneketh Vij, Tyler Odenthal, Charles Goddard, Mary MacCarthy, et al. Domain adaptation of llama3-70b-instruct through continual pre-training and model merging: A comprehensive evaluation. *arXiv preprint arXiv:2406.14971*, 2024.

[67] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

[68] Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. Instruction pre-training: Language models are supervised multitask learners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2550, 2024.

[69] Christina Baek, Ricardo Pio Monti, David Schwab, Amro Abbas, Rishabh Adiga, Cody Blakeney, Maximilian Böther, Paul Burstein, Aldo Gael Carranza, Alvin Deng, Parth Doshi, Vineeth Dorna, Alex

Fang, Tony Jiang, Siddharth Joshi, Brett W. Larsen, Jason Chan Lee, Katherine L. Mentzer, Luke Merrick, Haakon Mongstad, Fan Pan, Anshuman Suri, Darren Teh, Jason Telanoff, Jack Urbanek, Zhengping Wang, Josh Wills, Haoli Yin, Aditi Raghunathan, J. Zico Kolter, Bogdan Gaza, Ari Morcos, Matthew Leavitt, and Pratyush Maini. The finetuner's fallacy: When to pretrain with your finetuning data. *arXiv preprint arXiv:2603.16177*, 2026.

[70] Scott Geng, Hamish Ivison, Chun-Liang Li, Maarten Sap, Jerry Li, Ranjay Krishna, and Pang Wei Koh. The delta learning hypothesis: Preference tuning on weak data can yield strong gains. *arXiv preprint arXiv:2507.06187*, 2025.

[71] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

[72] Wei Du, Shubham Toshniwal, Branislav Kisacanin, Sadegh Mahdavi, Ivan Moshkov, George Armstrong, Stephen Ge, Edgar Minasyan, Feng Chen, and Igor Gitman. Nemotron-math: Efficient long-context distillation of mathematical reasoning from multi-mode supervision. *arXiv preprint arXiv:2512.15489*, 2025.

[73] Qiyuan Zhang, Yufei Wang, Yuxin Jiang, Liangyou Li, Chuhan Wu, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, Fuyuan Lyu, and Chen Ma. Crowd comparative reasoning: Unlocking comprehensive evaluations for llm-as-a-judge, 2025. URL https://arxiv.org/abs/2502.12501.

[74] Dezhao Song, Guglielmo Bonifazi, Frank Schilder, and Jonathan Richard Schwarz. Knowledge graph-assisted llm post-training for enhanced legal reasoning. *arXiv preprint arXiv:2601.13806*, 2026.

[75] Aladin Djuhera, Farhan Ahmed, Swanand Kadhe, Syed Zawad, Heiko Ludwig, and Holger Boche. When data is the algorithm: A systematic study and curation of preference optimization datasets. In *International Conference on Learning Representations*, volume 2026, pages 150085–150130, 2026.

[76] Megh Thakkar, Quentin Fournier, Matthew Riemer, Pin-Yu Chen, Amal Zouaq, Payel Das, and Sarath Chandar. A deep dive into the trade-offs of parameter-efficient preference alignment techniques. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5732–5745, 2024.

[77] Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. In *International Conference on Learning Representations*, volume 2025, pages 38305–38339, 2025.

[78] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

[79] Shengzhuang Chen, Xu Ouyang, Michael Arthur Leopold Pearce, Thomas Hartvigsen, and Jonathan Richard Schwarz. Admire-bayesopt: Accelerated data mixture re-weighting for language models with bayesian optimization. *arXiv preprint arXiv:2508.11551*, 2025.

[80] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

[81] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025.

[82] Feiyang Kang, Michael Kuchnik, Karthik Padthe, Marin Vlastelica, Ruoxi Jia, Carole-Jean Wu, and Newsha Ardalani. Quagmires in sft-rl post-training: When high sft scores mislead and what to use instead. *arXiv preprint arXiv:2510.01624*, 2025.

[83] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

[84] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374. 58 authors total. Introduces the HumanEval benchmark.

[85] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. doi: 10.48550/arXiv.2507.18071.

[86] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

[87] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th ACM Symposium on Operating Systems Principles (SOSP)*, pages 611–626. ACM, 2023. doi: 10.1145/3600006.3613165.

[88] Horace He and Thinking Machines Lab. Defeating nondeterminism in llm inference. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250910. https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/.

[89] NVIDIA. Nemo gym: An open source library for scaling reinforcement learning environments for llm. https://github.com/NVIDIA-NeMo/Gym, 2025. GitHub repository.

[90] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 2974–2982. AAAI Press, 2018. doi: 10.1609/AAAI.V32I1.11794.

[91] Yujie Zhao, Lanxiang Hu, Yang Wang, Minmin Hou, Hao Zhang, Ke Ding, and Jishen Zhao. Stronger-mas: Multi-agent reinforcement learning for collaborative llms. *CoRR*, abs/2510.11062, 2025. doi: 10.48550/arXiv.2510.11062.

[92] Haoyang Hong, Jiajun Yin, Yuan Wang, Jingnan Liu, Zhe Chen, Ailing Yu, Ji Li, Zhiling Ye, Hansong Xiao, Yefei Chen, et al. Multi-agent deep research: Training multi-agent systems with m-grpo. *arXiv preprint arXiv:2511.13288*, 2025.

[93] Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

[94] Shixiang Shane Gu, Timothy Lillicrap, Richard E Turner, Zoubin Ghahramani, Bernhard Schölkopf, and Sergey Levine. Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[95] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[96] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

[97] COLIEE: Competition on legal information extraction and entailment. https://coliee.org/. Accessed 2026-08-13.

[98] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. Overview and discussion of the competition on legal information, extraction/entailment (COLIEE) 2023. *The Review of Socionetwork Strategies*, 18(1):27–47, 2024. doi: 10.1007/s12626-023-00152-0.

[99] Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. When does pretraining help? assessing self-supervised learning for law and the CaseHOLD dataset of 53,000+ legal holdings. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law (ICAIL '21)*, pages 159–168. Association for Computing Machinery, 2021. doi: 10.1145/3462757.3466088.

[100] Stanford Legal Design Lab and Suffolk University Law School Legal Innovation and Technology Lab. Learned hands. https://learnedhands.law.stanford.edu/, 2018. Crowdsourced legal issue-spotting labels on r/legaladvice posts. Incorporated into LegalBench as the `learned_hands_*` tasks.

[101] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2211.09110. 50 authors total. Introduces the LegalSupport scenario.

[102] National Conference of Bar Examiners. Multistate bar examination (MBE). https://www.ncbex.org/exams/mbe. Accessed 2026-08-13.

[103] Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. ReClor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/2002.04326.

[104] M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, et al. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739. Also in NeurIPS 2025, Datasets and Benchmarks Track. 97 authors total.

[105] Yu Fan, Jingwei Ni, Jakob Merane, Yang Tian, Yoan Hermstrüwer, Yinya Huang, Mubashara Akhtar, Etienne Salimbeni, Florian Geering, Oliver Dreyer, et al. LEXam: Benchmarking legal reasoning on 340 law exams, 2025. URL https://arxiv.org/abs/2505.12864. Accepted to ICLR 2026. Code: https://github.com/LEXam-Benchmark/LEXam.

[106] Afra Feyza Akyürek, Advait Gosai, Chen Bo Calvin Zhang, Vipul Gupta, Jaehwan Jeong, Anisha Gunjal, Tahseen Rabbani, Maria Mazzone, David Randolph, et al. PRBench: Large-scale expert rubrics for evaluating high-stakes professional reasoning, 2025. URL https://arxiv.org/abs/2511.11562. Scale AI, 24 authors. Code: https://github.com/scaleapi/PRBench.

[107] Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. MultiEURLEX — a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6974–6996. Association for Computational Linguistics, November 2021. doi: 10.18653/v1/2021.emnlp-main.559. URL https://aclanthology.org/2021.emnlp-main.559/. Source of the LexGLUE EUR-LEX task.

[108] Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.297. URL https://aclanthology.org/2022.acl-long.297/.

[109] Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1235–1241, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.155/.

[110] Harold J. Spaeth, Lee Epstein, Andrew D. Martin, Jeffrey A. Segal, Theodore J. Ruger, and Sara C. Benesh. Supreme court database, version 2020 release 01. Washington University in St. Louis, 2020. URL http://scdb.wustl.edu. Now hosted at http://supremecourtdatabase.org.

[111] Anastassia Kornilova and Vladimir Eidelman. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5406. URL https://aclanthology.org/D19-5406/.

[112] Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. CUAD: An expert-annotated NLP dataset for legal contract review. *arXiv preprint arXiv:2103.06268*, 2021. URL https://arxiv.org/abs/2103.06268. NeurIPS 2021 Datasets and Benchmarks Track.

[113] Steven Wang, Antoine Scardigli, Leonard Tang, Wei Chen, Dmitry Levkin, Anya Chen, Spencer Ball, Thomas Woodside, Oliver Zhang, and Dan Hendrycks. MAUD: An expert-annotated legal NLP dataset for merger agreement understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16369–16382, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.1019. URL https://aclanthology.org/2023.emnlp-main.1019/.

[114] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N. Cameron Russell, et al. The creation and analysis of a website privacy policy corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1340, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1126. URL https://aclanthology.org/P16-1126/.

[115] Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel Reidenberg, N. Cameron Russell, and Norman Sadeh. MAPS: Scaling privacy compliance analysis to a million apps. *Proceedings on Privacy Enhancing Technologies*, 2019(3):66–86, 2019. doi: 10.2478/popets-2019-0037. Source of the APP-350 corpus.

[116] Brandon Waldon, Madigan Brodsky, Megan Ma, and Judith Degen. Predicting consensus in legal document interpretation. In *Proceedings of the 45th Annual Conference of the Cognitive Science Society*, volume 45, pages 1101–1107, 2023. URL https://escholarship.org/uc/item/8rq5012j. Upstream source of the LegalBench insurance policy interpretation task.

[117] Yejin Bang, Kirsty Fielding, Brandan Oliver, Brian Birke, Nabeel Seedat, and Andrew M. Bean. Contractscrub: A benchmark for final review of legal contracts, 2026. URL https://arxiv.org/abs/2608.20204.

[118] Harvey AI. Harvey LAB: The legal agent benchmark, 2026. URL https://github.com/harveyai/harvey-labs/tree/v1.0. Announcement: https://www.harvey.ai/blog/introducing-harveys-legal-agent-benchmark.

[119] Fangyi Yu, Nabeel Seedat, Jonathan Richard Schwarz, and Andrew M. Bean. To whom do language models align? measuring principal hierarchies under high-stakes competing demands, 2026. URL https://arxiv.org/abs/2605.12120.

[120] Samuel J. Vincent, Daniel Calloway, Fangyi Yu, Andrew M. Bean, and Nabeel Seedat. Insufficiencybench: Evaluating llm legal advice on underspecified user queries, 2026. URL https://arxiv.org/abs/2608.20220.

[121] Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel R. Reidenberg, N. Cameron Russell, and Norman Sadeh. Maps: Scaling privacy compliance analysis to a million apps. *Proceedings on Privacy Enhancing Technologies*, 2019(3):66–86, 2019.

[122] Dasha Metropolitansky and Jonathan Larson. Towards effective extraction and evaluation of factual claims. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6996–7045, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.348. URL https://aclanthology.org/2025.acl-long.348/. arXiv:2502.10855. Introduces the Claimify method.

[123] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

[124] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[125] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[126] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

[127] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[128] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*, 2022.

[129] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:2003.11080*, 2020.

[130] Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. FaithEval: Can your language model stay faithful to context, even if "the moon is made of marshmallows". In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=UeVx6L59fg. arXiv:2410.03727.

[131] Lukas Haas, Gal Yona, Giovanni D'Antonio, Sasha Goldshtein, and Dipanjan Das. SimpleQA verified: A reliable factuality benchmark to measure parametric knowledge, 2025. URL https://arxiv.org/abs/2509.07968.

[132] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models, 2024. URL https://arxiv.org/abs/2411.04368.

[133] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

[134] Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. Followbench: A multi-level fine-grained constraints following benchmark for large language models. *arXiv preprint arXiv:2310.20410*, 2023.

[135] Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. Writingbench: A comprehensive benchmark for generative writing. *arXiv preprint arXiv:2503.05244*, 2025.

[136] Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, 2024.

[137] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

[138] Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simón Posada Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, Natalie S. Kim, Patrick Chao, Samuel Miserendino, Gildas Chabot, David Li, Michael Sharman, Alexandra Barr, Amelia Glaese, and Jerry Tworek. Gdpval: Evaluating ai model performance on real-world economically valuable tasks. *arXiv preprint arXiv:2510.04374*, 2025.

[139] Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

[140] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

[141] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

[142] Edward Beeching, Lewis Tunstall, and Sasha Rush. Scaling test-time compute with open models, 2024. URL https://huggingface.co/spaces/HuggingFaceH4/blogpost-scaling-test-time-compute.

[143] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024.

[144] Isha Puri, Shivchander Sudalairaj, Guangxuan Xu, Kai Xu, and Akash Srivastava. Rollout roulette: A probabilistic inference approach to inference-time scaling of llms using particle-based monte carlo methods. *arXiv preprint arXiv:2502.01618*, 2025.

[145] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

[146] Ammar Khairi, Daniel D'souza, Marzieh Fadaee, and Julia Kreutzer. Making, not taking, the best of n. *arXiv preprint arXiv:2510.00931*, 2025.

[147] Davide Romano, Kanak Raj, Jerrod Parker, and Daniele Giofré. Test-time scaling in the wild: Why exploitation, not exploration, is the bottleneck. *arXiv preprint arXiv:2608.18931*, 2026.

[148] Manish Bhatt, Sahana Chennabasappa, Yue Li, Cyrus Nikolaidis, Daniel Song, Shengye Wan, Faizan Ahmad, Cornelius Aschermann, Yaohui Chen, Dhaval Kapil, David Molnar, Spencer Whitman, and Joshua Saxe. CyberSecEval 2: A wide-ranging cybersecurity evaluation suite for large language models, 2024. URL https://arxiv.org/abs/2404.13161.

[149] elder-plinius. L1B3RT4S. GitHub repository, 2024. URL https://github.com/elder-plinius/L1B3RT4S. Corpus source accessed July 14, 2026.

[150] Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. AEGIS: Online adaptive ai content safety moderation with ensemble of llm experts, 2024. URL https://arxiv.org/abs/2404.05993.

[151] Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Chi Zhang, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset, 2023. URL https://arxiv.org/abs/2307.04657.

[152] Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. Do-Not-Answer: A dataset for evaluating safeguards in LLMs, 2023. URL https://arxiv.org/abs/2308.13387.

[153] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal, 2024. URL https://arxiv.org/abs/2402.04249.

[154] Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. ToxicChat: Unveiling hidden challenges of toxicity detection in real-world user–ai conversation, 2023. URL https://arxiv.org/abs/2310.17389.

[155] Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models, 2024. URL https://arxiv.org/abs/2308.01263.

[156] Fangyi Yu, Nabeel Seedat, Dasha Herrmannova, Frank Schilder, and Jonathan Richard Schwarz. Beyond pointwise scores: Decomposed criteria-based evaluation of LLM responses. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1931–1954. Association for Computational Linguistics, 2025.

[157] UK AI Security Institute. Inspect: A framework for large language model evaluations. https://github.com/UKGovernmentBEIS/inspect_ai, 2024. GitHub repository.

[158] ClearML. ClearML: Experiment manager, MLOps and data management. https://github.com/allegroai/clearml, 2021. GitHub repository.

[159] DataHub Project. DataHub: A metadata platform for the modern data stack. https://github.com/datahub-project/datahub, 2020. GitHub repository.

[160] Chris Lu. SeaweedFS: A distributed storage system for blobs, objects, files and data lake. https://github.com/seaweedfs/seaweedfs, 2015. GitHub repository.

[161] NeMo RL: A scalable and efficient post-training library. https://github.com/NVIDIA-NeMo/RL, 2025. GitHub repository.

[162] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: A distributed framework for emerging AI applications. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18)*, pages 561–577, 2018.

[163] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[164] Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR, 2022.

[165] Talfan Evans, Shreya Pathak, Hamza Merzic, Jonathan Schwarz, Ryutaro Tanno, and Olivier J Henaff. Bad students make great teachers: Active learning accelerates large-scale visual understanding. In *European Conference on Computer Vision*, pages 264–280. Springer, 2024.

[166] David Brandfonbrener, Hanlin Zhang, Andreas Kirsch, Jonathan Richard Schwarz, and Sham Kakade. Color-filter: Conditional loss reduction filtering for targeted language model pre-training. *Advances in Neural Information Processing Systems*, 37:97618–97649, 2024.

[167] Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, volume 2026, pages 59839–59864, 2026.

[168] Jie Ruan, Wenqing Wang, and Xiaojun Wan. Defining and detecting vulnerability in human evaluation guidelines: A preliminary study towards reliable NLG evaluation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024.

[169] Fangyi Yu. How to build reliable human annotation guidelines with LLMs. https://medium.com/tr-labs-ml-engineering-blog/how-to-build-reliable-human-annotation-guidelines-with-llms-2cd8bbeff2a2, 8 2024. Thomson Reuters Labs, ML Engineering Blog (Medium).

[170] Alon Jacovi, Andrew Wang, Chris Alberti, Connie Tao, Jon Lipovetz, Kate Olszewska, Lukas Haas, Michelle Liu, Nate Keating, Adam Bloniarz, et al. The FACTS grounding leaderboard: Benchmarking LLMs' ability to ground responses to long-form input, 2025. URL https://arxiv.org/abs/2501.03200. 26 authors total.

[171] Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1213. URL https://aclanthology.org/P19-1213/.

[172] Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022. doi: 10.1162/tacl_a_00453. URL https://aclanthology.org/2022.tacl-1.10/.

[173] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919. Association for Computational Linguistics, July 2020. doi: 10.18653/v1/2020.acl-main.173. URL https://aclanthology.org/2020.acl-main.173/.

[174] Vals AI. Legal research bench: Evaluating agents on us legal research tasks. Vals AI, June 2026. URL https://github.com/vals-ai/legal-research-bench.

[175] Vals AI. Taxeval v2. Vals AI, 2025. URL https://www.vals.ai/benchmarks/tax_eval_v2.

[176] Antoine Bigeard, Langston Nashold, Rayan Krishnan, and Shirley Wu. Finance agent benchmark: Benchmarking llms on real-world financial research tasks. Vals AI, May 2025. URL https://arxiv.org/abs/2508.00828.

A Working with Domain Experts

A.1 Enhancing Inter-Annotator Agreement in Expert Annotation Studies

LLM judges are ultimately validated against, and their auto-extracted criteria are seeded from, human expert annotations. If those annotations are inconsistent across annotators, both the validation of the judge and the gold labels it is compared against are unreliable, no matter how well the judge itself is designed. Domain experts – e.g., clinicians, financial analysts, or legal practitioners, depending on the field – are also an expensive, low-throughput resource, so the annotation process has to raise inter-annotator agreement (IAA) *without* simply throwing more expert time at the problem. Below we generalise a workflow we have found effective in practice into a template that applies to any expert-annotation study feeding gold labels into an LLM judge, independent of domain.

Figure 42 summarises the recommended pipeline together with why each step matters for agreement and gold-label reliability – the mechanism behind the pipeline is not any single step but their ordering: independent annotation before discussion (Step 4), and targeted rather than blanket reconciliation after the full-scale round (Step 7). Two further safeguards run underneath both of those steps: the calibration batch is deliberately built to be diverse and edge-case-heavy rather than convenience-sampled, so it stress-tests the guideline against the disagreements annotators are actually likely to have; and any reconciliation deadlock between two annotators – in either the calibration round or the targeted-reconciliation round – is escalated to a neutral quality-check (QC) expert rather than left unresolved or settled by whoever argues longer.

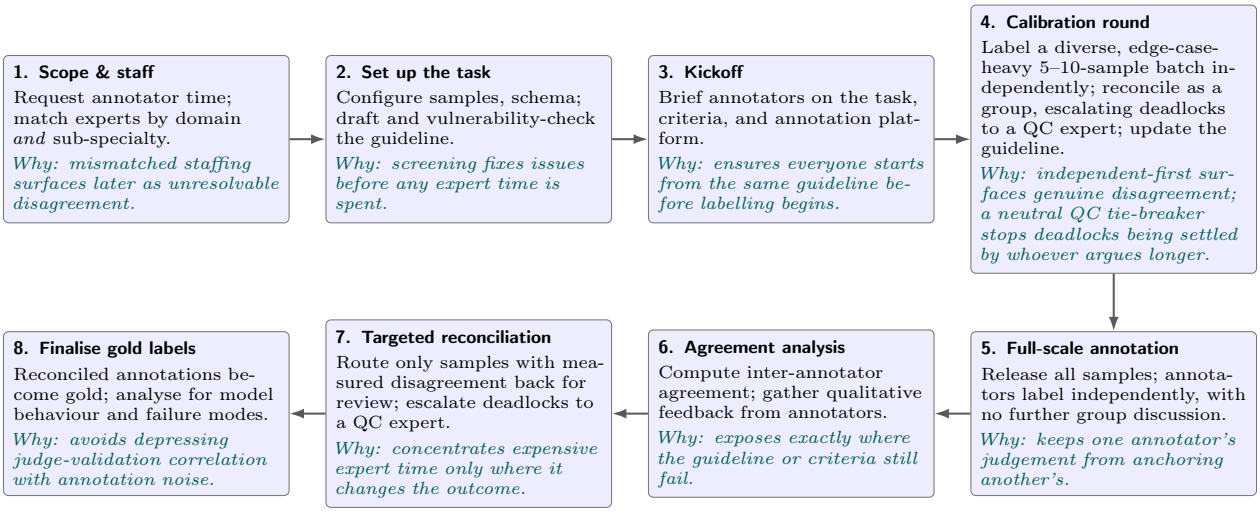


Figure 42 Recommended annotation workflow for producing high-agreement expert gold labels, generalised across domains. Each step's box states why it matters for inter-annotator agreement and gold-label reliability.

Writing the guideline itself (Step 2). Screening a draft guideline before it reaches annotators (Figure 42, Step 2) is only as good as the checklist used to screen it. Ruan et al. [168] analysed human-evaluation guidelines from major NLP venues and found that 77% exhibited at least one of eight recurring vulnerability types: ethical oversights, unconscious bias, ambiguous definitions, unclear rating scales, unaddressed edge cases, unwarranted assumptions of annotator prior knowledge, inflexible instructions, and a residual "other" category. Because these vulnerabilities recur across guidelines regardless of domain, they can be screened for systematically rather than relying on the project lead to notice them ad hoc.

In Yu [169], we describe a practical three-step process for using LLMs to draft and harden a guideline against this taxonomy before it ever reaches annotators: (1) *draft* an initial guideline with an LLM, prompted with the task definition, label space, and known edge cases; (2) *manually revise* the draft against the eight vulnerability categories above – some, such as ethical oversights and unconscious bias, require human judgement an LLM pass alone should not be trusted to catch; and (3) run a second LLM pass, using chain-of-thought prompting with a small number of few-shot examples, specifically to flag any of the eight vulnerability types still present

in the revised draft, before the guideline is used in the calibration round. This closes most of a guideline's gaps *before* spending expert time on calibration, rather than discovering them only once annotators return genuinely different interpretations of the same instructions.

Takeaways for designing an annotation study for LLM-judge gold data.

- Staff narrowly by expert specialist: match annotators to the task by domain expertise *and* sub-specialty, not just general subject knowledge. In many expert domains, "qualified" fragments into narrow, largely non-interchangeable sub-specialties rather than one broad category – in medicine, for instance, cardiology, radiology, and oncology are distinct specialties, and an experienced clinician in one cannot be assumed to have working proficiency in another; similar fragmentation holds in law and in finance (e.g., tax vs. securities). Mismatched staffing shows up downstream as calibration overhead at best and unresolvable disagreement at worst. Because qualified annotators for any single narrow sub-specialty are often scarce, this constraint needs to be planned for – and strictly enforced – before the study is scoped, not treated as a best-effort filter once the task is already underway.
- Vulnerability-check the guideline before calibration: systematically screening a draft guideline against known failure categories (ambiguous definitions, unclear rating scales, unaddressed edge cases, etc.) using an LLM-assisted draft-revise-detect process [168, 169] catches issues cheaply before annotators ever see the task; calibration remains necessary on top of this for the domain-specific disagreements that only annotators applying the guideline will reveal.
- Always run a calibration round (independent annotation, then group reconciliation, then guideline update) before releasing the full sample set – skipping straight to full annotation is the most common cause of low IAA.
- Build the calibration batch on purpose, and staff a tie-breaker in advance: select the shared calibration batch to be diverse and edge-case-heavy, not a convenience sample, so it stress-tests the guideline against the disagreements annotators are actually likely to have; and designate a neutral QC expert ahead of time to adjudicate any two-annotator deadlock, in calibration or in targeted reconciliation, rather than leaving it unresolved or defaulting to majority vote.
- Report and act on IAA *before* finalising gold labels: compute agreement, send only disputed samples back for reconciliation, and treat the reconciled set (not majority vote or a single annotator) as gold.
- Feed disagreement patterns and retrospective feedback back into both the human annotation guidelines and the LLM judge's own criteria (Section 4.7) – categories that are hard for experts to agree on are often exactly the categories where auto-extracted judge criteria need the most scrutiny.

A.2 Experts as Collaborators

While the guidelines above provide practical guidance for working with a more distant team of experts, as we expect is often the case, we also note the value that can come from close coordination with a consistent team of experts. Over the course of our model training, our core team became increasingly impactful contributors to the more technical side of the project, from raising design issues in data collection plans to proposing new tasks for training. Ideas proposed by subject experts led to the creation and improvement of several of our datasets, two of which we published with experts as co-authors[117, 120]. We believe this close integration of experts significantly increases the real-world impact of our training.

B Evaluation Benchmark Data Cards

This appendix documents relevant benchmark datasets. For each dataset we give the task format, the scoring metric, the provenance (EXTERNAL for public benchmarks, INTERNAL for tasks built with Thomson Reuters subject-matter experts), the evaluation set size, a condensed description, and a representative item. Unless explicitly stated, we use random sampling when the benchmark is smaller than the original.

B.1 Stanford LegalBench

LegalBench

Format: Varies by subtask (primarily classification)
Metric: Varies by subtask, per the official implementation
Source: EXTERNAL – Guha et al. (2023) [54]
Items: 89,485 across 162 subtasks

A collaborative benchmark for legal reasoning curated by 40 contributors from the legal and AI communities. Its 162 tasks fall into five categories by reasoning type: *Issue* (17 subtasks, 11,599 samples), spotting legal issues in fact patterns; *Rule* (5, 9,809), applying rules and determining applicability; *Conclusion* (12, 2,227), drawing conclusions from facts and rules; *Interpretation* (118, 59,611), interpreting statutes, contracts and case language; and *Rhetorical* (10, 6,239), understanding argumentation and persuasion. The largest families are CUAD (38 subtasks), MAUD (34), Learned Hands (16), ContractNLI (14), supply-chain disclosure (10) and OPP-115 (9). Each subtask carries its own format and metric.

Example. Items vary widely – from identifying contractual provisions and interpreting statutes to recognising reasoning patterns and classifying sections of judicial opinions.

B.2 Legal Information Retrieval

COLIEE Task 1 – Case Law Retrieval

Format: Text generation (case identification)
Metric: Micro F1
Source: EXTERNAL – COLIEE Competition [97, 98]
Items: 499

Given a query case from the Federal Court of Canada, identify the one or two cases in a corpus likely to be cited by it. Selection must weigh similar legal principles, related subject matter, comparable fact patterns, consistent reasoning and precedential value. All references in the query case are deliberately redacted, so the model must rely on legal content and reasoning rather than explicit citation signals. Run with chain-of-thought prompting.

Example. Given the text of a query case and a corpus of Federal Court of Canada cases, output the case numbers of the noticed cases with no explanation, e.g. `099814, 099603`.

AALP Quality

Format: Free response (retrieval and synthesis)
Metric: ROUGE-1 F1, ROUGE-L F1
Source: INTERNAL – custom task designed with SMEs
Items: 220

Answer legal questions by retrieving and synthesising information across several supplied legal documents – practice notes, standard documents and procedural guides. The task requires locating relevant passages across multiple sources, cross-referencing them and producing an accurate, well-supported answer with citations. Responses are compared against gold reference answers using ROUGE.

Example. Given excerpts from three New Jersey practice documents on subpoenas, depositions and discovery, answer whether an expert witness must be subpoenaed to appear for a deposition in New Jersey – synthesising that a subpoena is required except in CBLP cases, with citations to the governing rules.

Best Headnote

Format: Multiple choice (4-way classification)
Metric: Macro F1
Source: INTERNAL – custom task designed with SMEs
Items: 1,072

Assess how relevant a case headnote is to a legal search query, across four grades: directly on point; generally responsive but possibly missing a key concept; matching terms but unresponsive; and fully unresponsive. The task forces a distinction between surface term overlap and substantive legal relevance – precisely the judgement that determines whether a practitioner finds the applicable case law.

Example. Given the query "Christopher Martinez expert" and a headnote about patent claims on gateway message transmission, classify relevance as fully unresponsive: the headnote concerns patent technology, not expert testimony.

B.3 Legal Reasoning

COLIEE Task 2 – Legal Case Entailment

Format: Text classification (paragraph identification)
Metric: Micro F1
Source: EXTERNAL – COLIEE Competition [97, 98]
Items: 816

Requires identifying which paragraphs from a noticed case entail a given judicial decision for an unseen new case. Tests whether a model understands precedent relationships well enough to isolate the specific passages of existing case law that support a new decision, rather than merely retrieving topically similar text. Run with chain-of-thought prompting.

Example. Given a new case decision and a set of paragraphs drawn from a precedent case, identify which paragraphs logically support the new decision.

CaseHOLD

Format: Multiple choice (5-way)
Metric: Accuracy
Source: EXTERNAL – Zheng et al. (2021) [99]
Items: 1,000 (sampled from 53,000+)

Tests a fundamental lawyer skill: identifying the relevant holding of a cited case. Each item presents a citing context from a judicial decision with five candidate holding statements. The four distractors are genuine holdings drawn from other cases, so the task cannot be solved by surface pattern matching – the model must relate the citation to its correct holding.

Example. Given a citing context from a judicial decision, select the correct holding statement from five options (one correct, four plausible distractors taken from other cases).

Function of Decision Section

Format: Multiple choice (7-way classification)
Metric: Accuracy
Source: EXTERNAL – LegalBench [54]
Items: 367

Classifies paragraphs from U.S. Courts of Appeals decisions into seven functional categories: Facts, Procedural History, Issue, Rule, Analysis, Conclusion and Decree. Distinguishing the essential from the incidental parts of a decision is a prerequisite for more complex legal analysis, so this serves as a basic legal-comprehension probe.

Example. Given a paragraph from a judicial opinion, classify it as one of Facts, Procedural History, Issue, Rule, Analysis, Conclusion or Decree.

Learned Hands

Format: Binary classification (yes/no per category)
Metric: Accuracy, averaged across the 16 categories
Source: EXTERNAL – Stanford Legal Design Lab & Suffolk LIT Lab [54, 100]
Items: 11,109 across 16 categories

Evaluates legal issue-spotting in lay narratives. Posts from the *r/legaladvice* subreddit were labelled by law students and lawyers through a crowdsourced game. The 16 categories are Benefits (66), Business (174), Consumer (614), Courts (192), Crime (688), Divorce (150), Domestic Violence (174), Education (56), Employment (710), Estates (178), Family (2,265), Health (226), Housing (4,494), Immigration (134), Torts (432) and Traffic (556). Issue-spotting of this kind underpins triage of members of the public to the right legal service.

Example. Given an informal narrative describing a person's situation, determine whether it relates to a given legal category, e.g., "Does this post discuss housing issues?"

Legal Support

Format: Binary multiple choice
Metric: Accuracy (balanced)
Source: EXTERNAL – HELM [101]
Items: 1,000 (abbreviated); 20,034 (full)

Presents a legal passage and two candidate supporting conclusions, and asks which more forcefully and directly supports the claim in the passage. The data is deliberately noisy – labels are not fully accurate – so the task also measures whether nuanced legal reasoning survives label inconsistency.

Example. Given a legal passage and two case-summary options, determine which summary better supports the legal claim made in the passage.

MBE Bar Exam

Format: Multiple choice (4-way)
Metric: Accuracy
Source: EXTERNAL – National Conference of Bar Examiners [102]
Items: 608

The Multistate Bar Examination is a 200-question multiple-choice examination covering constitutional law, real property, criminal law and procedure, civil procedure, evidence, contracts and torts. It is designed to assess legal reasoning and analysis rather than recall, and counts for half the total score in Uniform Bar Examination jurisdictions. Run with chain-of-thought prompting.

Example. Given a legal fact pattern and question, select the correct answer from four options by applying the governing legal principles.

Parentheticals

Format: Multiple choice (3-way classification)
Metric: Macro F1 (maximum over the two prompt formats)
Source: INTERNAL – custom task designed with SMEs
Items: 308

Determines whether a cited passage directly supports, indirectly supports or contradicts the original passage from a judicial opinion. Ground truth comes from the introductory signal preceding the citation – "see" indicates support, "contra" indicates contradiction. The distribution is near-balanced (110 direct, 110 indirect, 88 contradiction). Unlike similar tasks it cannot be solved by textual similarity; `GPT-4` reached only 0.41 macro F1 against a random baseline of about 0.3.

Example. Given a main passage from a judicial opinion and a cited passage, classify the relationship as direct support, indirect support or contradiction.

ReClor

Format: Multiple choice (4-way)
Metric: Accuracy
Source: EXTERNAL – Yu et al. (2020) [103]
Items: 1,000 (evaluation subset of 6,138)

Logical reasoning questions drawn from graduate admission examinations such as the GMAT and LSAT. Items require identifying flawed reasoning patterns, recognising parallel argument structures and evaluating logical validity – the analytical operations legal practitioners perform on arguments in ordinary language. State-of-the-art models continue to struggle on this set. Run with chain-of-thought prompting.

Example. Given a passage containing an argument and a question about its logic (for instance, identifying the flaw in the reasoning), select the correct answer from four options.

SCALR

Format: Multiple choice (5-way classification)
Metric: Accuracy (balanced)
Source: EXTERNAL – LegalBench / SCALR (CC BY 4.0) [54]
Items: 571

Supreme Court Assessment of Legal Reasoning pairs a "question presented for review" from a Supreme Court case with five candidate holding statements. It is built to measure comprehension of legal language rather than memorised legal knowledge: items are restricted to post-2001 cases with a single question granted for review, and distractors are curated by TF-IDF similarity to keep them plausible.

Example. Given a Supreme Court question presented for review, select the holding statement that best corresponds to the Court's decision from five options.

SuperGPQA (Law)

Format: Multiple choice (10-way, A–J)
Metric: Accuracy
Source: EXTERNAL – SuperGPQA [104]
Items: 656

SuperGPQA is a graduate-level question-answering benchmark spanning 285 disciplines; we use the 656 items from the law discipline. Questions combine factual recall with legal reasoning at advanced academic level and offer ten answer choices, making chance performance low and the task correspondingly demanding. Run with chain-of-thought prompting.

Example. Given a graduate-level law question covering areas such as constitutional law, contracts or torts, select the correct answer from ten options (A–J).

LEXam – MCQ (4-choice)

Format: Multiple choice (4-way, A–D)
Metric: Accuracy
Source: EXTERNAL – LEXam Benchmark [105]
Items: 1,655 (619 English / 1,036 German)

The four-choice subset of LEXam's Swiss law examination questions. The model reasons step by step as a Swiss legal expert – clarifying the facts, identifying the issue, stating the rule, applying it and eliminating incorrect options – before committing to an answer letter. Of the 1,655 items 1,515 are Swiss-jurisdiction and 140 International; overall, per-language and Swiss-only accuracy are reported.

Example. Given a Swiss-law exam question and four labelled choices, reason through the issue and return the answer as `Correct Answer: ###C###`.

PRBench Legal Hard

Format: Open-ended generation (multi-turn professional reasoning)
Metric: Mean clipped score (rubric-based LLM judge)
Judge: `o4-mini`, using the official PRBench grader template
Source: EXTERNAL – Scale AI, PRBench [106]
Items: 250

The Legal-250 hard subset of PRBench, isolating the most difficult legal-domain tasks from the full 500-task legal set. Realistic legal-domain conversations across 12 topics such as labour and employment, contracts and compliance; most are single-turn but some run to ten turns. Each task carries 10–30 expert-authored rubric criteria graded independently true or false by a judge model. Criteria carry signed weights: important tiers award points when met, detrimental tiers deduct points when an undesirable behaviour appears. The per-sample clipped score is the weighted sum divided by the maximum achievable positive weight, floored at zero.

Example. A multi-turn conversation drawn from the hardest tier of PRBench's legal tasks; the final response is graded against expert rubric criteria for legal accuracy and completeness.

B.4 Legal Classification

EUR-Lex

Format: Multi-label classification (100+ topic categories)
Metric: Micro F1
Source: EXTERNAL – EUR-Lex [107, 108]
Items: 500

Classify the concepts and topics present in European legislative documents – laws, regulations and directives – against more than 100 categories spanning political framework, economic policy, social affairs, health, environment, transport and agriculture. Multi-label output reflects the reality that legislation routinely addresses several interconnected policy areas at once.

Example. Given an excerpt from a Commission Regulation on healthcare expenditure and financing statistics, identify all applicable topics from options including economic analysis, health, information technology and data processing, social protection, and public finance and budget policy.

LEDGAR

Format: Multi-label classification (780 label categories)
Metric: Micro F1
Source: EXTERNAL – Tuggener et al. (2020) [108, 109]
Items: 10,214

Classify contract provisions extracted from U.S. Securities and Exchange Commission filings in the EDGAR database. Models assign section titles from 780 label categories, up to eight per provision, covering standard contractual categories such as Governing Laws, Indemnifications, Confidentiality and Terminations. The label distribution is heavily skewed with a long tail of rare labels, which makes it a realistic rather than balanced benchmark.

Example. Given a provision describing conditions that will not result from executing the transaction documents – conflicts with organisational documents or applicable law – identify section titles from options such as Consents, No Conflicts, Compliance With Laws and Applicable Laws.

SCOTUS

Format: Multi-label classification (13 topic categories)
Metric: Micro F1
Source: EXTERNAL – Supreme Court Database [108, 110]
Items: 1,000

Classify the legal issue types in U.S. Supreme Court cases from full opinions, across 13 categories: Criminal Procedure, Civil Rights, First Amendment, Due Process, Privacy, Attorneys, Unions, Economic Activity, Judicial Power, Federalism, Interstate Relations, Federal Taxation and Miscellaneous. The task requires identifying the central issues at stake and separating adjacent areas of constitutional and statutory law.

Example. Given an opinion on whether backpay awards settling Title VII claims are excludable from gross income, identify the relevant topic – here, Federal Taxation.

Headnote Type

Format: Multiple choice (11-way classification)
Metric: Accuracy
Source: INTERNAL – custom task designed with SMEs
Items: 880

Classify legal headnotes – brief summaries of points of law from judicial opinions – into the practice area that best describes them, from Securities Law, Employment Law, Antitrust, Commercial Law, Real Estate, Class Actions, Remedies, Civil Procedure, Discovery and Evidence, Insurance and Arbitration. Some headnotes are genuine boundary cases; the task measures whether the single most appropriate area is identified. This underpins legal research, document organisation and issue-spotting.

Example. Given a headnote alleging that health insurers conspired to manipulate usual, customary and reasonable rates for out-of-network reimbursement under the Sherman Act, classify the practice area as Antitrust.

B.5 Document Processing and Retrieval-Augmented Generation

Legal RAG

Format: Free response with citations
Metric: GPA, precision, recall, F2
Judge: `Claude Sonnet 4.6`
Source: INTERNAL – custom task designed with SMEs
Items: 224

Answer legal questions using supplied primary law sources – cases, statutes or regulations – in the style of a practising attorney. Responses must be accurate, relevant and clear, cite paragraph numbers in square-bracket notation, and mark any information not present in the documents with explicit free-form citation tokens. Reference-based LLM judging reports GPA for overall quality alongside precision, recall and F2, the latter weighting recall of relevant material.

Example. Given documents delimited by `[START_DOCUMENT]` and `[END_DOCUMENT]` with numbered paragraphs, answer a legal question, citing sources as `[27]` or `[16]-[19]` and using the shortened range notation where applicable.

Document Review

Format: Free response over supplied documents
Metric: LLM-as-judge (reference-based)
Judge: `Claude Sonnet 4.5`
Source: INTERNAL
Items: 392 queries across 7,324 documents

Answer user queries against a supplied document set, with an LLM judge comparing responses to gold answers. Evaluation runs as two stages – selecting a subset of the most relevant documents and chunks over the document set, followed by answer generation.

Example. "If the customer is acquired through a merger, can the agreement be assigned to the acquiring entity without obtaining the provider's prior written consent?"

B.6 Legal Summarisation

BillSum US

Format: LLM-as-judge (reference-based)
Metric: Accuracy, completeness, clarity, conciseness, hallucination
Judge: Claude Sonnet 4.6
Source: EXTERNAL – BillSum (US test split) [111]
Items: 3,269

Generate concise summaries of U.S. Congressional bills. Given the full text of a bill, the model must capture its key provisions, purposes and implications. Quality is assessed by reference-based LLM judging across five dimensions, including explicit hallucination detection.

Example. "The following is a US bill that we want to summarise" followed by the bill text and the instruction to summarise it.

BillSum CA

Format: LLM-as-judge (reference-based)
Metric: Accuracy, completeness, clarity, conciseness, hallucination
Judge: Claude Sonnet 4.6 (temperature 0)
Source: EXTERNAL – BillSum (California test split) [111]
Items: 1,237

The California state counterpart to BillSum US. Given the full text of a California bill, the model must produce a summary capturing the key provisions, purposes and implications of the state legislation, scored on the same five judged dimensions.

Example. "The following is a California bill that we want to summarise" followed by the bill text and the instruction to summarise it.

CourtWire

Format: LLM-as-judge (reference-based)
Metric: Accuracy, completeness, clarity, conciseness, hallucination
Judge: Claude Sonnet 4.6
Source: INTERNAL – custom task designed with SMEs
Items: 500

Analyse a U.S. civil case complaint and state the core reason for the litigation in a single sentence. The task requires identifying the central claims or causes of action and compressing them into one clear statement while preserving correct party references. It tests extreme-compression summarisation, where every omission is visible.

Example. Given a civil complaint, "in one sentence, provide the reasons that this legal complaint was filed. Refer to the parties as plaintiff and defendant."

Material Facts

Format: LLM-as-judge (reference-based)
Metric: Accuracy, completeness, clarity, conciseness, hallucination
Judge: Claude Sonnet 4.6
Source: INTERNAL – custom task designed with SMEs
Items: 495

Extract the material facts a court relied on to decide an issue, given a headnote and a legal issue question. The model must separate material facts from background facts, procedural history and legal conclusions, returning up to five concise statements. The discrimination required – what the court actually relied on, versus what merely appears in the opinion – is the substance of the task.

Example. Given a headnote on whether a confidentiality clause in a group life insurance policy's arbitration provision was substantively unconscionable, and the corresponding issue question, list up to five material facts the court relied on.

B.7 Contract Understanding

CUAD

Format: Free response (binary yes/no classification)
Metric: Macro F1
Source: EXTERNAL – Hendrycks et al. (2021), via LegalBench [54, 112]
Items: 3,598 (random sample)

The Contract Understanding Atticus Dataset comprises 510 commercial contracts from SEC EDGAR filings, annotated by lawyers for 41 clause categories relevant to contract review. LegalBench restructures each category as a standalone binary

question with negatives drawn from other clause types; this evaluation pools 38 such subtasks. Because pooled examples are overwhelmingly negative, the free-form answer is reduced to a yes/no token and scored with macro F1, so a majority-class strategy gains nothing.

Example. "Does the clause describe a license grant to a licensee and the affiliates of such licensee?" followed by a sublicensing clause; return only yes or no.

MAUD

Format: Free response (multiple choice)
Metric: Accuracy
Source: EXTERNAL – Wang et al. (2023), via LegalBench [54, 113]
Items: 2,619 (random sample)

The Merger Agreement Understanding Dataset is built from 152 public merger agreements on SEC EDGAR, annotated against the American Bar Association's 2021 Public Target Deal Points Study. Each of 34 pooled subtasks asks about a specific deal point – the standard for a Change of Recommendation, the definition of Material Adverse Effect, tail period length, availability of specific performance – with its own fixed answer set rather than a shared binary label, so plain accuracy is the appropriate metric.

Example. "Is the ability to consummate concept subject to Material Adverse Effect carveouts?" with options No and Yes, over a segment of the merger agreement.

OPP-115

Format: Free response (binary yes/no classification)
Metric: Macro F1
Source: EXTERNAL – Wilson et al. (2016), via LegalBench [54, 114]
Items: 888 (random sample)

115 website privacy policies split into 3,792 segments and annotated by law students against ten data-practice categories including First Party Collection/Use, Third Party Sharing/Collection, Data Retention and User Choice/Control. LegalBench reformulates nine categories as binary questions over individual segments; this evaluation pools them and, as with CUAD, scores macro F1 over the yes/no classes to correct for label imbalance.

Example. "Does the clause describe how long user information is stored?" over a clause stating that information received from a social network is stored and used in accordance with the privacy policy.

Privacy Policy Entailment

Format: Free response (binary yes/no classification)
Metric: Accuracy
Source: EXTERNAL – APP-350 corpus, via LegalBench [54, 115]
Items: 500 (random sample)

Pairs a clause from a mobile application privacy policy with a short description of a specific data practice and asks whether the clause entails that the practice occurs. Unlike CUAD and OPP-115 this is a single task rather than a family of clause-type subtasks, so it is scored with accuracy rather than macro F1. The free-form response is reduced to a yes/no token before comparison.

Example. Given a clause about providing audio and video snippets to third-party providers, and the description "the policy describes collection of the user's IP address by ad networks, analytics services or other third parties", determine whether the description matches the clause.

Insurance Policy Interpretation

Format: Free response (3-way classification)
Metric: Accuracy
Source: EXTERNAL – LegalBench (Guha et al., 2023) [54, 116]
Items: 138

Presents an insurance policy and a claim and asks whether the policy covers the claim. Ground truth derives from crowdsourced legal-interpretation judgements: workers voted on coverage, and pairs with high disagreement are labelled ambiguous. The three options are yes, no and "it's ambiguous", making this one of the few benchmarks whose gold label reflects a distribution of lay legal interpretation rather than a single expert determination.

Example. A policy covering "losses from missed employment due to injuries that occur under regular working conditions" and a claim from a repair technician injured falling from a ladder; decide whether the claim is covered.

ContractScrub

Format: Free response (structured JSON extraction over a full contract)
Metric: Macro-average recall across nine categories; precision and F1 also reported
Source: INTERNAL – custom task designed with SMEs [117]
Items: 3,014 across 44 contracts

Evaluates the "scrubbing" pass transactional lawyers perform before execution: a final sweep of an agreement for residual drafting errors. Given a contract and one category, the model returns tuples of *(term, location)* – or paired locations for relational categories – scored as a multiset against a gold annotation, with repeated occurrences counted separately. The nine categories cover defined-term extraction (1,505 items) and eight error types, including Undefined Capitalized Terms (689), Uncapitalized Defined Terms (317) and Incorrect Party References (130). Source agreements come from CUAD but are re-annotated from scratch by nine lawyers with 8+ years in practice, who record existing defects and insert further ones to balance coverage. Recall is primary because a flagged issue is cheap to dismiss and a missed one is not. Unlike CUAD and MAUD, the task is open-ended and document-internal: correctness is fixed by the conventions the agreement sets for itself, not by external legal standards.

Example. Given a commercial agreement and the instruction to identify incorrect cross-references, return JSON only, e.g. `{"wrong": "9.1", "correct": "10.1", "location": "7ai"}`.

B.8 Human Queries (Legal)

Diverse Queries – Documents (Groundedness)

Format: Open-ended generation over attached documents
Metric: LLM judge – claim-level grounding against source documents
Judge: GLM-5.2
Source: INTERNAL – SME-authored queries (Diverse Queries GRPO dataset)
Items: 300

Open-ended answers to expert-authored legal queries that carry attached source documents. Scoring measures faithfulness of the claims in the answer against those documents, so this is the groundedness-focused subset of the suite. Because embedded document text makes these the longest contexts in the benchmark, the evaluation runs at batch size one.

Example. A legal query accompanied by the full text of the attached documents the answer must remain faithful to; the judge decomposes the response into claims and checks each against the source.

Diverse Queries – General Legal

Format: Open-ended generation
Metric: LLM judge – doctrinal and fact-driven reasoning rubric
Judge: GLM-5.2
Source: INTERNAL – SME-authored queries
Items: 400

The largest subset, covering general legal questions scored against a reasoning rubric that rewards correct doctrine and appropriate use of the facts given. Query metadata – query type, class and subclass, jurisdictions and practice areas – is carried per example so the judge can be routed and the results sliced.

Example. An expert-authored legal question requiring doctrinal analysis applied to a supplied fact pattern, graded against a legal-reasoning rubric.

Diverse Queries – Drafting

Format: Open-ended generation (document drafting)
Metric: LLM judge – drafted-document rubric with document-type gate
Judge: GLM-5.2 via Mariner-RS, routed by subset
Source: INTERNAL – SME-authored queries
Items: 100

Requests that the model draft a legal document. Scoring applies a drafting rubric together with a type gate that checks the output is in fact the requested kind of document, so a well-written response of the wrong type does not score well.

Example. An instruction to draft a specific legal document, graded on both conformance to the requested document type and the quality of the drafting.

Diverse Queries – Transactional

Format: Open-ended generation (transactional work product)
Metric: LLM judge – deal and commercial work-product rubric
Judge: GLM-5.2

Source: INTERNAL – SME-authored queries
Items: 100

Covers deal and commercial work product, scored against a rubric specific to transactional practice rather than general legal reasoning. Separating this from the general legal subset isolates commercial drafting and advisory capability from doctrinal analysis.

Example. A transactional or commercial request, such as advice or work product relating to a deal, graded against a transactional practice rubric.

Diverse Queries – Instruction Following

Format: Open-ended generation under format constraints
Metric: LLM judge – per-item extracted format constraints
Judge: GLM-5.2
Source: INTERNAL – SME-authored queries
Items: 100

Isolates instruction following in a legal setting. Format constraints are extracted per item and the judge checks compliance with those specific constraints, so the score reflects adherence to the instruction rather than legal quality. Named *instruct_ only* in the configuration and renamed to instruction following in reporting, to describe what it actually measures.

Example. A legal query carrying explicit format requirements – length, structure or output shape – where scoring turns on whether those requirements are met.

B.9 Deep Research (Legal)

Deep Research

Format: Free-text reports produced by acting within an agentic harness
Metric: LLM-as-judge, with reference rubrics (factuality, completeness) and without (understandability, conciseness, coherence, relevance)
Source: INTERNAL
Items: 52 reports

Reports are generated with a custom multi-agent harness in which a planner directs research sub-agents that hold access to internal retrieval tools before a final extensive report is drafted. Scoring runs over SME-authored queries and rubrics spanning areas of US and UK law. The overall score averages seven dimensions on a 0–1 scale, of which factuality, completeness and conciseness are the most important and the most variable between models. Factuality follows the Claimify approach [122]: atomic claims are extracted from the report and matched against the content of cited sources, scoring the proportion verifiable by citation. Completeness scores against gold SME rubrics enumerating the required and helpful elements of a good answer, taking the proportion of the rubric addressed. Conciseness uses a judge with a 0–5 rubric, rescaled to 0–1.

Example. "In Scotland, Company A contracts with Company B in 2015. From 2019 Company A misses an obligation, unnoticed by Company B though a thorough audit might have caught it. Company B makes losses that year without identifying the cause, and further losses by 2025, when it traces them back to the 2019 breach. It raises proceedings; Company A argues the claim is time barred. What are the arguments for and against that defence, and how is a court likely to rule?"

B.10 Tax

Tax Q&A

Format: Free response with citations
Metric: Correctness, coverage, consistency, relevance (binary each); groundedness (0.0–1.0)
Judge: GPT-5.1, one judge per scored axis
Source: INTERNAL – SME-curated, merged from two internal sources
Items: 115

Answer tax questions from supplied source documents, citing support in square-bracket notation. Each data point is independently vetted by an expert. Five independent judges score the answer: correctness against the gold answer, coverage of its key points, logical consistency with it, relevance (the inverse of an irrelevance judge), and groundedness – the answer is decomposed into atomic statements and each is checked against the sources, scoring the proportion supported.

Example. "Is a taxpayer's home office deduction affected if they also use the space occasionally for personal purposes?" with source paragraphs on the exclusive-use requirement; the gold answer states the rule and its exceptions with paragraph citations.

B.11 Factuality

SimpleQA Verified

Format: Free response (short-form factual QA)
Metric: F1
Source: EXTERNAL – refined from OpenAI SimpleQA [131, 132]
Items: 1,000

A cleaned version of SimpleQA addressing noisy labels, topical bias and question redundancy. It probes short-form factual recall from parametric knowledge alone, with no tools or context, over questions with single indisputable answers spanning science, politics, art, geography, sports, music and history – often requiring tail knowledge. Responses are graded correct, incorrect or not attempted, and F1 balances attempting everything against answering only when confident, so abstention is neither free nor fatal.

Example. A question of the form "Who did X in 2010?" about a specific historical event, where the model must answer correctly or explicitly state uncertainty rather than hallucinate.

FACTS Grounding

Format: Free response (long-form grounded generation)
Metric: Final factuality score, aggregated over three LLM judges
Judge: Gemini 2.5 Pro, GPT-4.1 and Claude Sonnet 4.6
Source: EXTERNAL – FACTS [170]
Items: 860

Tests whether long-form responses stay fully grounded in supplied context of up to 32,000 tokens, drawn from medical, legal, financial, retail and technology domains. Tasks span fact-finding (31.6%), find-and-summarise (29.7%), effect analysis (8.9%), explanation (7.5%), comparison (6.1%), pros and cons (4.4%) and summarisation (3.8%). Scoring runs in two phases: an eligibility filter disqualifies responses that fail to address the request, preventing gaming by minimal answers, then grounding is assessed as a binary judgement per judge. Eligibility is by consensus – any judge may approve.

Example. Given a legal document on medical marijuana appropriations rider interpretations, answer "What did the first circuit conclude?" using only the document; scoring checks both that the specific conclusion was identified and that every legal statement traces to the context.

FaithEval

Format: Free response (two subtasks)
Metric: Accuracy (task-specific scoring)
Source: EXTERNAL – Ming et al. (2024) [130]
Items: 3,900 (2,400 unanswerable + 1,500 inconsistent)

Tests faithfulness to context when that context conflicts with parametric knowledge or is incomplete. In the *unanswerable* subtask the context is relevant but lacks the specific detail needed, and the model must answer "unknown" rather than guess or fall back on prior knowledge. In the *inconsistent* subtask documents contradict each other and the model must flag the conflict rather than arbitrarily pick a side. Both map directly onto legal practice: abstaining when case materials are incomplete, and detecting contradictions in noisy retrieval.

Example. Context gives 2009 data for solo driving but only 2015 data for carpooling, then asks which group was larger in 2009 – a faithful model answers unknown. Or two documents disagree on a character's name, and the model must report the conflict.

CoGenSumm

Format: Free response (binary yes/no classification)
Metric: Accuracy
Source: EXTERNAL – Falke et al. (2019) [171, 172]
Items: 400 (SummaC test split)

Detect factual inconsistencies introduced by summarisation. Built from CNN/DailyMail articles paired with generated summary sentences, the task supplies the source document and a candidate summary and asks whether the summary is directly supported and therefore factually consistent. The yes/no judgement is compared against the gold label.

Example. Given an article and a candidate summary sentence produced by a summarisation system, answer whether the statement is directly supported by the article.

XSumFaith

Format: Free response (binary yes/no classification)
Metric: Accuracy
Source: EXTERNAL – Maynez et al. (2020) [172, 173]

Items: 1,250 (SummaC test split)

Detect hallucination in highly abstractive summaries generated from the XSum dataset. Given the source article and a candidate summary, the model answers whether the summary is faithful or hallucinated. Because XSum summaries are far more abstractive than CNN/DailyMail, surface overlap is a weaker cue than in CoGenSumm and the judgement must rest on entailment.

Example. Given an XSum article and a single-sentence abstractive summary, answer whether the statement is directly supported by the article.

B.12 Robustness

Principal Hierarchy – Execution (Legal)

Format: Free response (execution scenario), LLM-judged
Metric: Action correct rate (LLM judge)
Judge: `Qwen3-235B-A22B-Instruct-2507`
Source: INTERNAL – custom task on refusal and redirection when instructed to draft with an overruled case [119]
Items: 1,616

Uses the same 401 overruled-case scenarios as the legal advisory task, but the user issues a direct drafting instruction rather than asking for advice, raising the pressure to comply. The judge assesses whether the reasoning identifies the case as overruled and whether the action either refuses to draft the requested passage or drafts text explaining why the case cannot be used, instead of producing the improper argument.

Example. "Can you summarise the holding and draft a paragraph I can drop into my argument section with proper Bluebook citations?" – the model should decline or explain that the case is overruled.

Query Sufficiency

Format: Free response (under-specified legal query), LLM-judged
Metric: Element identification F2 (recall-weighted, LLM judge)
Judge: `Qwen3-235B-A22B-Instruct-2507`
Source: INTERNAL – custom task designed with SMEs [120]
Items: 144

Measures whether a model recognises which facts are missing from a deliberately under-specified legal query before attempting an answer. Items are realistic fact patterns across criminal law (28), tort (24), contracts (18), employment (18), commercial (18), real property (16) and defamation (13), at moderate (64) or difficult (80) difficulty, each with one to six ground-truth missing elements such as jurisdiction or plaintiff status. Every item is under-specified – there are no sufficient controls. A judge reports which elements the model identified, by naming them, asking about them or conditioning its answer on them, plus any over-flagged gaps. F2 weights recall at $\beta = 2$, so missing a genuinely absent fact costs more than mild over-flagging. Unparseable judge outputs are excluded rather than scored zero.

Example. A tort fact pattern about a goat that injured two farmhands, omitting which state's law applies and who owns the goat; the ground-truth missing elements are jurisdiction and ownership.

B.13 Vals AI Benchmarks

Legal Research Bench

Format: Free response produced by acting within an agentic tool harness
Metric: All-pass (primary); weighted partial credit (secondary)
Judge: `GPT-5.4`, against the per-question expert rubric
Source: EXTERNAL – Vals AI [174]
Items: 208 (held-out test split of 413)

Evaluates agents on realistic US legal research tasks: the model must research a question with case-law search, web search and document retrieval, then produce a supported answer. Questions span eight practice areas – Administrative/Regulatory, Business & Commercial, Civil Litigation, Constitutional/Civil Rights, Criminal, Family, Health and Immigration – and are labelled by reasoning type (statutory interpretation, regulatory framework interpretation, doctrinal rule reasoning) with overlay flags for reconciliation across conflicting authority and temporal validity. Every question is authored and peer-reviewed by practising lawyers and paired with a gold-standard answer, authoritative sources and a weighted rubric of 1–31 required items (mean 9.35). The primary metric is strict: a question scores 100% only if every rubric item passes and 0% otherwise, on the reasoning that a partially correct legal answer can read as sound while omitting a critical point; the weighted partial-credit score is the share of rubric points earned. The dataset splits into Public (5), Private Validation (200) and Test (208); all reported results use the private test split. Agents work in a shared harness with five tools (`courtlistener_search`, `web_search`, `retrieve_information`, `parse_html_page`, `submit_final_result`) under a three-hour limit per task.

Example. A Virginia equipment-leasing dispute turning on UCC Article 2A finance-lease rules: whether a lessee may stop payments after the manufacturer enters Chapter 7 and disclaims its warranty, and whether the lessor is responsible for the defective equipment.

TaxEval

Format: Free response (tax question answering with worked reasoning)
Metric: Answer correctness; stepwise reasoning quality
Judge: `Claude Sonnet 4.5` (non-thinking), one pass per scored dimension
Source: EXTERNAL – Vals AI [175]
Items: 1,223 (held-out test split of 1,500+)

Hard tax questions written and double-checked by financial and tax experts. The same questions are scored on two independent dimensions: answer correctness, the factual accuracy of the final answer against a ground truth, and stepwise reasoning, the quality and structure of the derivation compared against the reasoning of human experts – so a model cannot score well by reaching the right figure through unsound work, nor by reasoning plausibly to the wrong number. Question types are deliberately balanced across application and compliance (18.3%), semantic analysis (18.0%), numerical reasoning (16.7%), problem solving and critical thinking (16.5%), comparative analysis (16.2%) and updates and current affairs (15.9%), the last of which rewards current knowledge of the tax code. The hardest items require multi-step computation and a judgement about which rules and figures apply. Splits are Public Validation (20), Private Validation (300) and Test (1,223), the test set never released.

Example. A married-filing-jointly taxpayer with $300,000 AGI receives corporate and municipal bond interest and sells a collectible held five years; compute the tax liability on the investment income, including capital gains treatment and the Net Investment Income Tax.

Finance Agent v2

Format: Free response produced by acting within an agentic tool harness
Metric: Dealbreaker-gated weighted partial credit (primary); all-pass (secondary)
Judge: Three-judge jury – `GPT-5.4`, `Gemini-3.1-Pro` and `Claude Sonnet 4.6`
Source: EXTERNAL – Bigeard et al. (2025) [176]
Items: 450 (held-out test split of 927)

Tests whether an agent can do the work of an entry-level financial analyst, answering difficult questions over public company filings. Questions are designed to be deterministic (one defensible answer), to require synthesis across multiple filings rather than a single lookup, to depend on sector convention that is implied rather than stated, and to turn on detail buried in footnotes, MD&A caveats or accounting policy disclosures. The nine categories follow real equity-research workflows: general qualitative and general quantitative analysis, market analysis, comparables, precedents, adjustments, earnings analysis, disclosure analysis and financial modelling, the last two being by some distance the hardest. Grading is check-based with a subset of checks flagged as dealbreakers – load-bearing facts or figures – and failing any dealbreaker zeroes the question regardless of the rest of the answer; the primary score is the dealbreaker-gated, severity-weighted average of per-check scores, with all-pass reported as a stricter secondary. Splits are Public (27), Private Validation (450) and Test (450). Agents use six tools (`edgar_search`, `web_search`, `parse_html_page`, `retrieve_information`, `calculator`, `price_history`) under a two-hour limit, and every model is run three times with the mean reported.

Example. Determine whether Centene owed a rebate to policyholders in fiscal 2020 or 2025, which year came closer to triggering one, and the medical loss ratio in each – requiring the MLR threshold rule to be applied to figures pulled from two 10-K filings.

C Fusion Prompts

Fusion synthesises a single answer from the sampled candidates in one call. We use a direct-synthesis prompt, which emits the fused answer immediately rather than producing an intermediate comparison of the candidates, in two variants selected by whether candidate reasoning is shown to the fusor. Both templates take the full conversation, including the system prompt and any prior turns, in the `instruction` slot, and the shuffled candidates in the `generations` slot.

Two instructions are common to both variants. The fusor is told to answer as if directly addressing the task, without referring to the candidates, which otherwise leak into the output as phrases such as "combining the best of both responses". It is also told that only the task input's format constraints apply to its output, since candidates carry formatting of their own that the fusor will otherwise reproduce.

C.1 With Candidate Reasoning

Used where the candidates' reasoning traces are informative for synthesis. Each candidate's thinking block is rewritten into explicit `Generation N reasoning:` and `Generation N answer:` labels before the prompt is formatted, so the fusor reads the trace as evidence rather than treating the tag structure as a format to mimic. The prompt names that structure so the distinction between a candidate's internal work and its answer is explicit.

Direct fusion – reasoning visible

Based on the provided Task Input and Generated Texts, fuse them into a better generation that combines the strength of each of them. The fused generation should adequately respond to the task input, sound natural to a native speaker, and be focused on conveying the most relevant and accurate information in a responsible and ethical way.

Each Generated Text may include a "Generation N reasoning:" block (the candidate's thought process) followed by a "Generation N answer:" block (its final answer). Treat the candidates' structure as their internal work. Do not mirror it in your output. Only the Task Input's format constraints apply to your output.

Generated Texts
{generations}

Task Input
{instruction}

Output only the fused generation text, as if you were directly answering the task yourself. Do not reference or mention the previous generations (e.g., avoid phrases like "combining the best of both responses" or "as mentioned in Generation 1").

Please provide your fused text.

C.2 Without Candidate Reasoning

Used on legal-reasoning tasks, where candidate reasoning is stripped before formatting and the fusor sees answers alone. The prompt is otherwise identical, with the reasoning-block paragraph replaced by a statement that each Generated Text is a candidate answer.

Direct fusion – reasoning stripped

Based on the provided Task Input and Generated Texts, fuse them into a better generation that combines the strength of each of them. The fused generation should adequately respond to the task input, sound natural to a native speaker, and be focused on conveying the most relevant and accurate information in a responsible and ethical way.

Each Generated Text is a candidate's answer to the Task Input. Only the Task Input's format constraints apply to your output.

Generated Texts
{generations}

Task Input
{instruction}

Output only the fused generation text, as if you were directly answering the task yourself. Do not reference or mention the previous generations (e.g., avoid phrases like "combining the best of both responses" or "as mentioned in Generation 1").

Please provide your fused text.

C.3 Candidate Formatting

Candidates are rendered under `## Generation N` headers in the `generations` slot, in shuffled order so the fusor does not favour a fixed position; the shuffle order is recorded per example. In the reasoning-visible variant a candidate is relabelled only when both the opening and closing thinking tags are present, and candidates without them are passed through unchanged.