Extended Abstract

Motivation. Frontier coding agents, such as Codex and Claude Code, are a marriage between a model that predicts tokens and a harness that manages the model's lifecycle, context, and interactions. For specialized workflows, the harness—rather than the weights—is often where the capability lies. While verifiable domains such as mathematics and code can apply techniques like reinforcement learning with verifiable rewards (RLVR) to push capability back into the model, domains like law are more knotty since they do not have automatic verifiers. The challenge is to engineer a robust, deterministic harness that can structure, ground, and check long-horizon legal work. Hence, we ask:

> How much of a legal harness's procedural advantage can be internalized back into the base model's weights, so the model's performance improves even under a plain default harness at inference?

Method We study harness-guided post-training of Qwen3.5-9B on Harvey's Long Horizon Legal Agent Benchmark (LAB). We engineer a privileged legal harness H_{legal}- and measure LAB benchmark performance under both H_{legal} and the plain deployment harness provided by Harvey H_{default}. Our legal harness is a structured legal-work runtime with task classification, source grounding, structured intermediates, and a constrained repair loop. We then try to move the harness procedure into the weights along a progression of increasingly on-policy objectives: supervised fine-tuning on harness traces, off-policy reverse-KL distillation, on-policy KL distillation on the default-harness trajectories, and GRPO with a rubric-judge reward.

Implementation LAB contains 1,251 long-horizon legal-agent tasks and 74,990 expert rubric criteria across 24 practice areas; the default judge advised by Harvey is Claude Sonnet 4.6, but we used GPT-5.4-mini due to availability and cost considerations. We use a family/cluster-disjoint 875/188/188 split, derive H_{legal} from $\sim$200 Meta-Harness search iterations patched with Codex, and run rollouts, serving, and QLoRA training on Modal B200 GPUs through vLLM.

Results The legal harness lifts criterion-pass from 40.29% to **69.14%** (+28.85 points) with the weights frozen. Other approaches do not match this performance: SFT under the legal harness reaches 62.67% and drops to (39.91%) with the default harness; off-policy reverse-KL is the worst performer at 37.22% using the default harness; on-policy KL is the only distillation arm to beat the default base, at 45.85%. GRPO is the strongest internalizer, scoring **67.49%** on the default harness.

Discussion Harness design dominates training method at this scale. We believe the distillation failures are due to weak-student on-policy distillation - reverse-KL is mode-seeking and unstable when the teacher is uncertain, which is consistent with off-policy KL underperforming even the base. Additionally, H_{legal} was engineered for LAB's complexity rather than calibrated to a small, non-agentic 9B parameter model.

Conclusion For small legal agents, harness complexity should be co-designed with model capability. We evaluate all systems on a held-out test split since development-set gains alone would not establish generalization.

GRPO Didn't Pass the Bar (But the Harness Did): Harness-Guided Post-Training for Legal Agents

Duy Nguyen
Dept. of Computer Science
Stanford University
duynguy@stanford.edu

Leon Reilly
Dept. of Mathematics, Dept. of Philosophy
Stanford University
leonry@stanford.edu

Abstract

Frontier coding agents pair a fine-tuned model with a deterministic *harness* that manages tool use, workflow state, and execution. In legal domains, where automatic verifiers are limited, much of the system capability may reside in this harness rather than in the model weights. We study whether the procedural advantage of a specialized legal harness can be internalized into Qwen3.5-9B on Harvey's Long Horizon Legal Agent Benchmark (LAB). We engineer a specialized legal harness with task classification, source grounding, structured intermediate outputs, and a constrained repair loop, then compare performance under this harness and Harvey's default deployment harness. We test several post-training methods, including supervised fine-tuning, off-policy reverse-KL distillation, on-policy KL distillation, and GRPO with a rubric-judge reward. The legal harness substantially improves performance with frozen weights, raising the criterion pass rate from 40.29% to 69.14%. However, most training methods fail to transfer this advantage into the default harness: SFT slightly hurts default-harness performance, off-policy reverse-KL performs worst, and on-policy KL gives only a modest gain. GRPO is the strongest internalization method, recovering 67.49% under the default harness. Overall, these results suggest that harness design dominates training method at this model scale, and that procedural knowledge is difficult to distill into a small, non-agentic legal model. We report all results on a held-out test split, and argue that future legal agents should co-design harness complexity with model capability and evaluate on held-out tasks to ensure generalization.

1 Introduction

Modern agents are a marriage between a model that predicts tokens and a *harness* that manages the model's lifecycle, context, and interactions. In particular, Codex and Claude Code wrap a fine-tuned model with a harness that reads files, runs tools, manages turn-by-turn state, and verifies edits. A growing line of work makes this explicit and optimizes the harness directly while holding the model fixed. Lee et al. [2026], Lin et al. [2026]

Yet, harnesses increase latency and token costs because their procedural steps must be executed for each new task. In domains with automatic verifiers, this cost can be reduced by training the model to perform the procedure directly. For example, RLVR is one technique that samples model outputs, checks them against a verifier, and increases the probability of outputs that pass. Shao et al. [2024], DeepSeek-AI [2025]. This setup does not neatly map to legal drafting since there is no deterministic program to decide whether a motion is factually correct, persuasive, procedurally appropriate, and so on. These judgments require multiple criteria, legal expertise, and context-specific evaluation. Recent work shows that reinforcement learning can still operate in such domains by turning rubrics into rewards through an LLM judge Gunjal et al. [2025], but the reward can be noisy and hackable Mahmoud et al. [2026], Liu et al. [2026].

We study whether a harness's procedural competence can be internalized into a small open model in this non-verifiable regime, using Harvey's Long Horizon Legal Agent Benchmark (LAB) Grupen et al. [2026]: 1,251 long-horizon legal-agent tasks across 24 practice areas, each scored against detailed binary rubric criteria. In particular, we engineer a specialized legal harness H_{legal} and measure how much of its advantage survives when the model is run under a plain deployment harness H_{default} after training. This applies to On-Policy Harness Self-Distillation question Zhao et al. [2026] to long-horizon legal work, that is, can the model learn to reproduce the harness procedure under the same inference conditions it will face at deployment?

2 Related Work

Harness optimization and agent scaffolds. A model's harness substantially determines agent performance. Meta-Harness optimizes the harness end-to-end around a fixed model Lee et al. [2026], and Agentic Harness Engineering evolves coding-agent harnesses from execution observability Lin et al. [2026]. We use a Meta-Harness-style search to construct H_{legal}. We ask whether the harness can be removed after training by internalizing it into the weights.

Distilling inference-time procedure into weights. Classic sequence-level knowledge distillation trains a student on teacher-generated sequences Kim and Rush [2016], which corresponds most closely to our SFT-on-harness-traces baseline. However, autoregressive students face a train–test mismatch when trained only on teacher prefixes and then deployed on their own generations. Generalized Knowledge Distillation addresses this by training on student-generated sequences with teacher feedback **?**. MiniLLM studies reverse-KL distillation for generative language models **?**, while DistiLLM proposes skew-KL and adaptive off-policy distillation to improve efficiency Ko et al. [2024]. Most directly, OPHSD studies whether harness-assisted rollouts can train the same model to perform without the harness at test time **?**. Our setting differs because the harness procedure is legal, long-horizon, source-grounded, and evaluated by rubric-based LLM judges rather than deterministic verifiers.

RL with verifiers and rubric rewards. RLVR has driven recent gains in domains where outputs can be checked automatically, especially math and code. GRPO was introduced in DeepSeekMath as a resource-efficient alternative to critic-based PPO **?**, and DeepSeek-R1 shows that large-scale RL can elicit reasoning behavior from language models **?**. Process-supervision work further shows that denser verification of intermediate reasoning steps can outperform outcome-only feedback in mathematical reasoning Lightman et al. [2024]. Legal drafting lacks such deterministic verifiers, so we instead use rubric criteria as a judge-mediated reward. This follows Rubrics as Rewards and related work using question- or instance-specific rubrics as reward signals for open-ended tasks **?**Wei et al. [2026].

Legal AI evaluation and reliability. Earlier legal benchmarks such as LegalBench and LawBench evaluate legal reasoning and legal knowledge across curated task suites Guha et al. [2023], Fei et al. [2024]. BigLaw Bench and LAB move toward realistic legal work products and long-horizon legal-agent tasks Harvey Team [2024], Grupen et al. [2026]. Recent work on legal hallucinations and AI legal research tools motivates our emphasis on source grounding and verification Dahl et al. [2024], Magesh et al. [2024].

Legal AI and LAB post-training. LAB evaluates long-horizon legal-agent tasks using detailed rubric criteria Grupen et al. [2026]. Concurrent work from Harvey and Baseten post-trains a 27B open-weight model on LAB and argues that post-training and harness optimization must be co-developed. This contrasts with our 9B setting, where the model benefits strongly from the harness at inference but only partially absorbs the harness procedure through post-training.

3 Method: Harness-Guided Post-Training

Target. We aim to move as much of the legal harness's H_{legal} workflow as possible into the model weights, so that the trained policy improves under the default harness H_{default}. The difference in performance between the H_{legal} and H_{default} harnesses captures how much of the LAB capability is supplied by the harness and how much is parametric.

Objective. We train the model in the setting it will use at deployment. First, we sample trajectories from the student under the default harness:

$$\tau \sim \pi_\theta(\cdot \mid H_{\text{default}}).$$

Along those same trajectories, we ask a teacher (model paired with the legal harness) to guide the student's next-token predictions:

$$\sum_t D_{\text{KL}}\big(\pi_\theta(\cdot \mid \tau_{<t}, H_{\text{default}}) \,\big\|\, \pi_{\text{teacher}}(\cdot \mid \tau_{<t}, H_{\text{legal}})\big)\,.$$

This tests whether the student can learn the behavior supplied by the legal harness while operating only under the default harness. Zhao et al. [2026], Agarwal et al. [2024].

Imitation to RL. We study four objectives that differ in how on-policy they are.

1. **SFT.** Can the model copy legal-harness traces directly?
2. **Off-policy (reverse-KL).** Token-level teacher matching on *teacher-generated* prefixes.
3. **On-policy (KL).** Trains on the student's own *default-harness* trajectories while still using H_{legal} as the privileged teacher.
4. **GRPO.** Moves beyond token imitation: it searches over student generations directly using a verifier reward $r = 0.8\, c_{\text{frac}} + 0.2\, c_{\text{all}}$ for end-to-end task completion, where c_{frac} is the fraction of criteria satisfied and c_{all} is the all-pass indicator Gunjal et al. [2025], Shao et al. [2024].

4 Harness Architecture

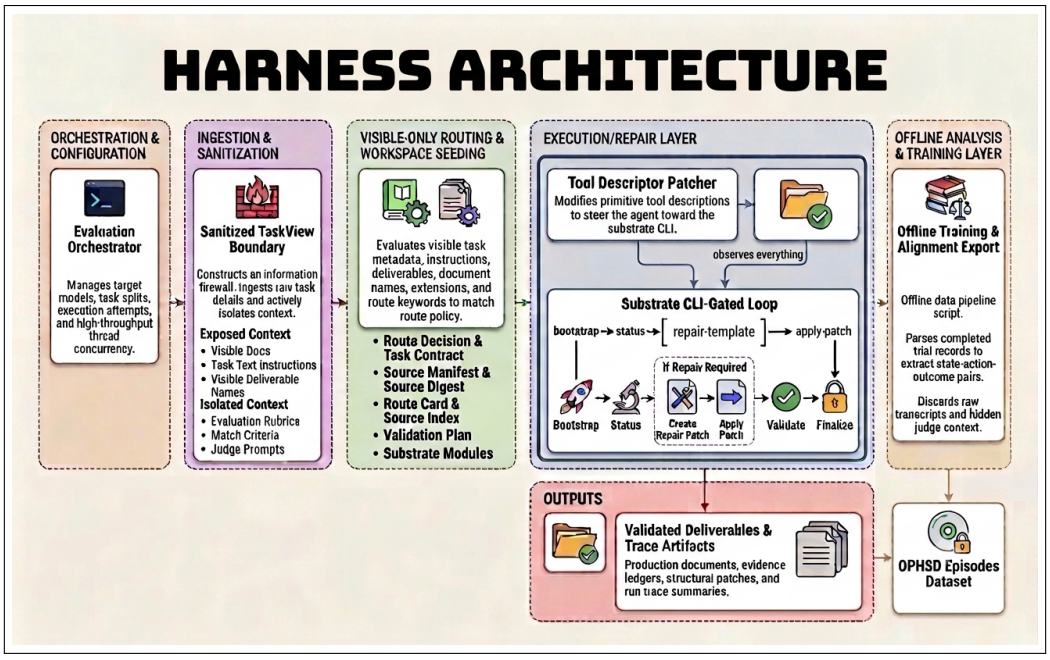

Figure 1: We engineer a specialized legal harness H_{legal} that guides the Qwen3.5-9B agent through a structured legal workflow. We then try to internalize that legal workflow procedure into the model's weights using SFT, off-policy reverse-KL, on-policy KL, and GRPO, and finally evaluate the updated model under the plain default harness H_{default}.

H_{default}: deployment harness. The default harness is provided by Harvey. It provides task context, source metadata, required output units, ledger repair, validation, and finalization, but no legal-work planning.

H_{legal}: teacher harness. Before the model begins a LAB task, the harness prepares the task through four steps: orchestration, task sanitization, workspace setup, and execution with repair. The harness implementation is made publicly available Nguyen and Reilly [2026].

- **Task classification.** The harness identifies what kind of legal work is being asked, e.g. drafting, comparison, extraction, issue-spotting, markup, etc., meaning the model treats each task as a specific work product.
- **Source grounding.** Each output is mapped back to a source. This pushes the model toward coverage and citation. We noticed this design follows retrieval-augmented generation, which augments parametric generation with external non-parametric memory Lewis et al. [2020]. It is also motivated by evidence that long-context models do not reliably use all positions in long contexts Liu et al. [2024], and by work on citation-grounded generation and legal retrieval evaluation Gao et al. [2023], Pipitone and Houir Alami [2024].
- **Structured intermediates before final output.** The model first produces structured intermediate representations of findings, extracted values, drafted clauses, calculations, etc. Final products are produced from these.
- **Repair loop.** The model iterates through planning, drafting, validation, and finalization, and may patch specific fields of the intermediate representation rather than rewrite the whole output.

We constructed H_{legal} by running $\sim$200 iterations of Meta-Harness search over a frozen Qwen3.5-9B and patching each result with Codex. Lee et al. [2026]

5 Experimental Setup

Model and benchmark. We post-train Qwen3.5-9B on LAB. LAB's official metric is all-pass: a task is counted as passed only if the draft satisfies every criterion for that task. Because all-pass is brittle for diagnosis, we also report criterion-pass rate, the fraction of individual criteria satisfied across all tasks. The default LAB judge is Claude Sonnet 4.6; for cost and throughput reasons, we instead use GPT-5.4-mini as the criterion judge for all reported runs.

Data splits. We use a family- and cluster-disjoint split of 875 train, 188 development (dev), and 188 test tasks. The train split is used for learning; the dev split is used for candidate-system selection and ablations; the test split is held out as a locked final estimate. The data export rejects dev and test rows by construction, preventing accidental train-time exposure. All criterion-pass numbers reported in this paper are computed on the held-out test split; the dev split is used only for candidate selection and ablations, and is not reported as a final result.

Uncertainty estimates. Each rubric criterion is scored as a binary pass/fail outcome, so we report criterion-pass rates as mean percentages with binomial standard errors over the 11,151 held-out test criteria:

$$\mathrm{SE}(\hat{p}) = 100\sqrt{\frac{\hat{p}(1-\hat{p})}{11151}}.$$

These standard errors quantify uncertainty from the finite set of evaluated rubric criteria due to insufficient compute.

Training and serving. Rollouts and model serving run on Modal B200 GPUs through vLLM. Supervised and distillation training use QLoRA Dettmers et al. [2023] with rank 64, $\alpha = 128$, all-linear adapters, and bf16 compute. The GRPO reward judge is, like all of our judges, GPT-5.4-mini. For evaluation, we also use the OpenAI Batch API inference with prompt caching to reduce evaluation costs while preserving the same criterion-level judge prompts.

Harness extrapolation comparison. As a reference point for the harness, we also evaluate GPT-5.5 on the same 188 held-out test tasks. We run GPT-5.5 with two harnesses: the LAB default harness and our legal harness. GPT-5.5 generations are evaluated with GPT-5.4-mini using the same criterion-pass computation as the Qwen systems.

Table 1: Criterion-pass rate on LAB with standard errors (±SE) calculated over $n = 11,151$ independent evaluation criteria on the held-out test split. Harness design is the dominant performance vector; post-training interventions scale monotonically with how closely they approximate on-policy deployment conditions.

System	Criteria Passed / Total	Criterion-pass Rate
Base + default harness	4493/11151	$40.29\% \pm 0.46\%$
Base + legal harness	**7710/11151**	$\mathbf{69.14\% \pm 0.44\%}$
SFT + default harness	4450/11151	$39.91\% \pm 0.46\%$
SFT + legal harness	6988/11151	$62.67\% \pm 0.46\%$
On-policy KL + default harness	5113/11151	$45.85\% \pm 0.47\%$
Off-policy KL + default harness	4151/11151	$37.22\% \pm 0.46\%$
GRPO + default harness	**7526/11151**	$\mathbf{67.49\% \pm 0.44\%}$
GPT-5.5 Baseline Reference		
GPT-5.5 + default harness	**9532/11151**	$\mathbf{85.48\% \pm 0.33\%}$
GPT-5.5 + legal harness	8434/11151	$75.63\% \pm 0.41\%$

6 Results

Under LAB's official all-pass metric, no Qwen3.5-9B configuration passes a single task ($0/188$); we therefore report criterion-pass throughout as a finer-grained diagnostic.

Harnessing dominates. Keeping the model weights fixed, replacing H_{default} with H_{legal} raises the criterion pass rate from 40.29% to 69.14%, a 28.85-point gain. The bulk of the LAB performance comes from the execution procedure supplied by the harness rather than the base model.

Imitation fails. SFT on legal-harness traces does not improve performance under the default harness. It reaches a score of 62.67% when evaluated with H_{legal}, but only 39.91% with H_{default}, slightly below the base model's 40.29%. Copying harness traces does not make the model reproduce the harness procedure at deployment.

On-policy helps. Off-policy reverse-KL performs worst, reaching 37.22% under H_{default}. On-policy KL trains on the student's own default-harness trajectories and improves to 45.85%.

GRPO transfers most. GRPO optimizes the LAB rubric-judge reward over the student's own generations, rather than imitating teacher tokens. It reaches 67.49% under H_{default}, recovering most of the legal-harness gain without using H_{legal} at inference time. However, it remains below the frozen base model run directly with H_{legal} at 69.14%.

Training is secondary. The core result is that harness design changes performance more than any post-training method we tested. Among the training methods, performance improves as the training signal becomes closer to deployment: SFT and off-policy KL fail to transfer, on-policy KL gives a modest gain, and GRPO recovers most of the harness advantage.

7 Discussion

Harness effects exceed training effects. With fixed weights, H_{legal} improves criterion pass rate by 28.85 points over H_{default} (40.29% to 69.14%). No post-training method exceeds this fixed-weight harness gain. GRPO reaches 67.49% under H_{default}, recovering most of the gap, but still trails direct use of H_{legal}. At this model scale, the inference-time procedure is the main source of performance.

On-policy objectives transfer better. Off-policy reverse-KL performs below the base model (37.22% vs. 40.29%), consistent with a train–test mismatch between teacher-generated prefixes and student-generated deployment states. On-policy KL reduces this mismatch and improves to 45.85%. GRPO performs best because it optimizes rubric reward on the student's own outputs rather than token agreement with a teacher Jin et al. [2026], Gu et al. [2024].

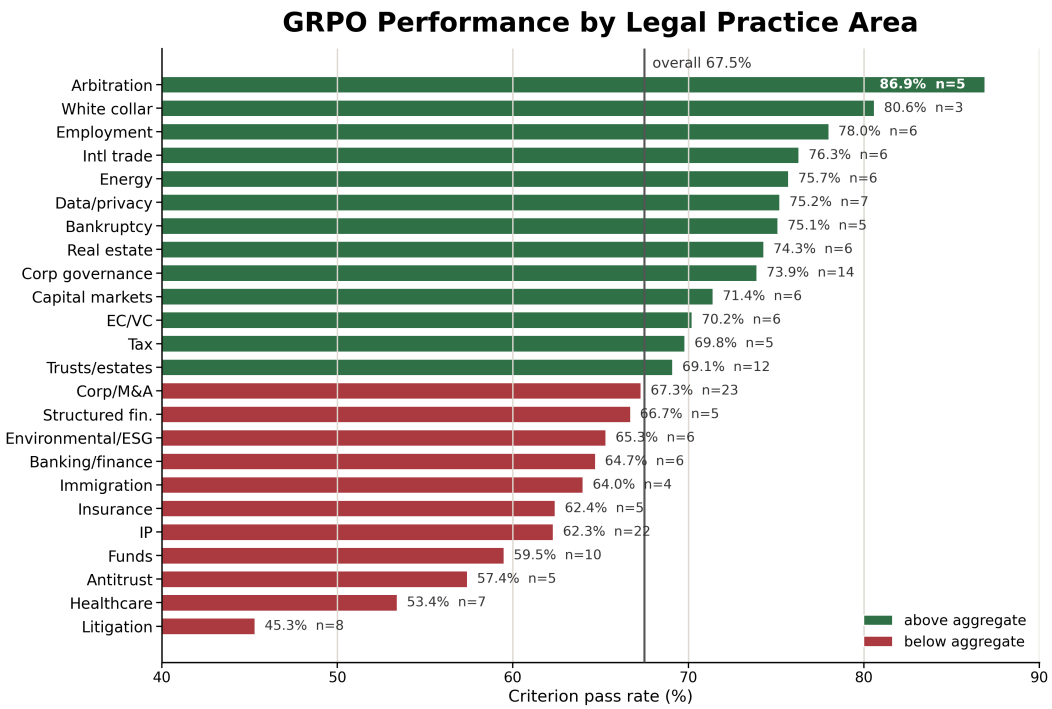

Figure 2: Per-practice-area breakdown of criterion-pass under each system.

Model capacity is the likely bottleneck. H_{legal} encodes planning, source grounding, structured intermediates, and repair. Qwen3.5-9B can use this procedure when it is supplied at inference, but does not fully internalize it through the training methods tested here.

Limitations. First, H_{legal} may be too complex for a 9B non-agentic model, so these results may not reflect a capacity-matched harness. Second, the harness–capability interaction is established from only two capability points (Qwen3.5-9B and GPT-5.5); a fuller model-size $\times$ harness-complexity sweep is needed before treating "match complexity to capability" as a scaling law. Third, all runs are scored by GPT-5.4-mini rather than LAB's official Claude Sonnet 4.6. LLM-as-judge evaluation can approximate human preferences at scale, but prior work documents position, verbosity, self-enhancement, and limited- reasoning biases Zheng et al. [2023], as well as bias toward LLM-generated text in some NLG evaluation settings Liu et al. [2023]. More importantly, recent work on rubric-based RL shows that models can exploit verifier failures or rubric-design omissions: reward gains under the training verifier need not transfer to independent judges or broader quality judgments **?**. Future work should re-score with multiple cross-family judges (including the official Sonnet 4.6) and, ideally, expert legal review.

8 Conclusion

We studied whether the procedure encoded in a specialized legal harness can be transferred into Qwen3.5-9B and retained under a plain default harness. On Harvey LAB, the harness itself is the crucial factor. With frozen weights, our legal harness H_{legal} improves the criterion pass rate by 28.85 points over H_{default}. Post-training transfers this advantage only partially. SFT and off-policy KL do not improve default-harness performance, on-policy KL gives a modest gain, and GRPO recovers most of the gap at 67.49%, but still does not exceed direct use of H_{legal}. With GPT-5.5, the default harness reaches 85.48% criterion pass rate, while the legal harness reaches 75.63%. Thus, the same added procedure that helps Qwen3.5-9B can hurt a stronger model. This suggests that harness complexity should be matched to model capability.

9 Team Contributions

- **Duy Nguyen:** evaluation and training infrastructure; integration of LAB into the harness stack; construction of H_{legal} via Meta-Harness search; SFT, reverse-KL, on-policy KL, and GRPO training on Modal B200; methods and results write-up.
- **Leon Reilly:** runtime harness execution and optimization; baseline and ablation studies; evaluation design and criterion-pass analysis; related work and introduction write-up.

AI Tools Disclosure We used Claude Code and Codex as the harness proposer to help build evaluation and training infrastructure on top of the harness codebase, and to assist with figure generation and literature search.

References

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. GKD: Generalized knowledge distillation for auto-regressive sequence models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2306.13649.

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. arXiv:2401.01301.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2305.14314.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. LawBench: Benchmarking legal knowledge of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. arXiv:2309.16289.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. arXiv:2305.14627.

Niko Grupen, Gabriel Pereyra, and Julio Pereyra. Introducing harvey's legal agent benchmark (LAB). `https://www.harvey.ai/blog/introducing-harveys-legal-agent-benchmark`, 2026. Harvey AI blog, May 6, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2306.08543.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, et al. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. arXiv:2308.11462.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains, 2025.

Harvey Team. Introducing biglaw bench. `https://www.harvey.ai/blog/introducing-biglaw-bench`, 2024. Harvey AI blog, Aug. 29, 2024.

Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2603.07079.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1317–1327. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1139. arXiv:1606.07947.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.03898.

Yoonho Lee, Omar Khattab, Kenneth Lee, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses, 2026.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. arXiv:2005.11401.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2305.20050.

Lin et al. Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses, 2026.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2024. Transactions of the ACL; arXiv:2307.03172.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. arXiv:2303.16634.

Yixin Liu, Yue Yu, DiJia Su, Sid Wang, Xuewei Wang, Song Jiang, Bo Liu, Arman Cohan, Yuandong Tian, and Zhengxing Chen. Examining reasoning LLMs-as-judges in non-verifiable LLM post-training, 2026.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho. Hallucination-free? assessing the reliability of leading AI legal research tools, 2024.

Anas Mahmoud, MohammadHossein Rezaei, Zihao Wang, Anisha Gunjal, Bing Liu, and Yunzhong He. Reward hacking in rubric-based reinforcement learning, 2026.

Duy Nguyen and Leon Reilly. Larness: Official release of the LAB legal harness runtime. `https://github.com/yudduy/larness-release/tree/main/harness`, 2026.

Nicholas Pipitone and Ghita Houir Alami. LegalBench-RAG: A benchmark for retrieval-augmented generation in the legal domain, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024.

Xiyu Wei, Qingwei Zong, Xiaoguang Li, Eugene J. Yu, and Sujian Li. QuRL: Rubrics as judge for open-ended question answering. In *International Conference on Learning Representations (ICLR)*, 2026. Poster. OpenReview: `https://openreview.net/forum?id=DrhWTuhtYq`.

Zhengyang Zhao, Lu Ma, and Wentao Zhang. Training with harnesses: On-policy harness self-distillation for complex reasoning, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena, 2023.