AI Alignment and Legal Reasoning

Elliott Ash

This chapter explores the links between legal reasoning and the post-training regimes of aligned large language models (LLMs). Reinforcement learning from human feedback (RLHF), which optimizes model outputs to match graded human preference rankings, leads to LLMs that perform value-weighted interpolations across task-response pairs. Reinforcement learning with verifiable rewards (RLVR), which rewards outputs that pass objective external checks, produces LLMs that faithfully follow structured reasoning paths. The RLHF approach parallels the analogical, precedent-based reasoning characteristic of common-law systems, while RLVR mirrors the deterministic code-based reasoning associated with civil law. These analogies inform the design of multi-agent legal AI systems, which should deploy RLVR-based LLMs for compliance with bright-line rules, while deploying RLHF-based systems to apply more subtle standards.

A. Introduction

Legal systems and machine-learning systems share a central design problem: aligning individual reasoning with collective standards. In both domains, outcomes depend on how reasoning processes internalize shared norms and translate them into consistent decisions. In the law, two broad forms of reasoning address this task. Case-based reasoning derives conclusions by analogy to prior decisions, weighting precedent by relevance and authority. Code-based reasoning applies determinate rules to specified conditions, aiming for predictability and uniformity. Legal institutions balance these approaches to minimize error and promote legitimacy.

An analogous division in reasoning modes defines the latest generation of large language models (LLMs) – and more specifically, the two families of algorithms that are applied to align LLMs with human aims. Through reinforcement learning from human feedback (RLHF), models are tuned to reflect human evaluative patterns, internalizing social preferences much like courts internalize precedent (Stiennon et al. 2020; Ouyang et al. 2022). Through reinforcement learning with verifi-

able rewards (RLVR), models are constrained by formal verification, and then learn to follow explicit reasoning paths (Lambert et al. 2024). These training architectures reproduce the same institutional trade-off that legal systems face, with RLHF better suited to case-based reasoning based on previous examples, and RLVR better suited to code-based reasoning following deterministic rules.

This chapter explores this parallel and its implications for the design of legal AI. Section 2 starts with a review of how LLMs are built. Modern models are trained first by next-token prediction on vast corpora. Transformers make this scalable by attending to long-range context, which helps models represent syntax and semantics and supports generalization across domains (Vaswani et al. 2017). Instruction fine-tuning (IFT) teaches models to follow prompts, while RLHF steers model outputs to match human preference rankings (Stiennon et al. 2020; Ouyang et al. 2022). As a result, RLHF-aligned LLMs generate responses by performing an interpolation over the training and instruction corpora, weighted by their scores on the reward model.

Recent LLMs adopt a new approach: RLVR, which moves from graded human scores to objective checks (Lambert et al. 2024). The model proposes incremental steps of reasoning, which are then externally verified for validity and consistency. RLVR encourages explicit chains of thought and stepwise correctness, which is why it shines on logic, math, and code (Lightman et al. 2024; Lambert et al. 2024). In short, RLHF tunes for context-sensitive judgment; RLVR trains for structured logical reasoning.

I continue in Section 3 by connecting these ideas from LLMs with analogous ideas from the law. Case-based reasoning in the common law aggregates and weights prior decisions by relevance and authority. It is an inductive, analogical process, well captured by classic work in AI and law on case comparison and argument patterns (Ashley 1992; Rissland, Ashley, and Branting 2005). That makes it a natural match for RLHF, which aggregates graded human judgments across many contexts. Rule application and statutory reasoning, by contrast, often turn on element satisfaction and nested conditions. This is closer to RLVR's logic: correctness is defined outside the decision maker, and the task is to check whether the elements are met.

This connection extends to the classic legal distinction between rules and standards (Kaplow 1992). RLHF aligns with context-sensitive standards; RLVR aligns with bright-line rules. Taking a step further, insti-

tutional form matters too, in particular around comparative legal origins (Mattei 1997; La Porta et al. 2008). Common-law systems, evolving through precedent and analogical development, are well-aligned with an RLHF-based approach to reasoning. Civil-law systems, grounded in comprehensive codes and deductions from statutory text, are best-suited to RLVR LLM systems.

There are clear implications for the design of legal AI systems, as explored in Section 4. It is unlikely that there is a one-size-fits-all legal agent. Instead, we should develop coordinated multi-agent systems that route sub-tasks to different models. Determinate statutory checks should go first to an RLVR-trained reasoner that can verify element satisfaction step by step. Ambiguous issues that hinge on community standards and fact sensitivity should go to an RLHF-trained reasoner that aggregates across analogous precedents. The system mirrors how human judging already mixes verification with evaluation, while making explicit the normative choices behind model training and task assignment. The architecture of AI is linked directly with the architecture of law.

B. How Large Language Models Are Trained

This section outlines how large language models are trained and aligned through successive stages that shape their linguistic and reasoning capacities. The base model learns statistical patterns in text, while post-training stages progressively align it with human instructions, preferences, and formal correctness. Specific attention is paid to the two main approaches for post-training – RLHF and RLVR – and the tasks they are designed to perform.

I. Sequence Models, Embeddings, and Transformers

Modern LLMs grow out of sequence modeling in language. The core task is language modeling – estimating the conditional probability of the next token given a context. Early systems used n-gram counts with smoothing (Shannon 1951; Jurafsky and Martin 2023). Neural language models replaced counts with distributed representations that support generalization across contexts (Bengio et al. 2003), including word embeddings mapping discrete tokens to context-sensitive vectors (Mikolov et al. 2013).

The early neural language models were recurrent neural networks (RNNs; Hochreiter and Schmidhuber 1997; Sutskever, Vinyals, and Le 2014). RNNs process sequences token by token, which limits long-range context and parallel processing. An architectural innovation called the attention mechanism, introduced by Bahdanau, Cho, and Bengio (2015), lets models weight relevant parts of the input directly. The transformer architecture is founded around attention – it removes recurrence, uses multi-head self-attention and feed-forward layers plus positional encodings, and models dependencies over long contexts in parallel (Vaswani et al. 2017). Encoder-only models such as BERT support representation learning and classification (Devlin et al. 2019), while autoregressive (decoder-only) language models such as GPT support open-ended generation (Radford et al. 2019). Scaling of training data, parameters, and context windows have improved downstream performance across tasks (Kaplan et al. 2020; Brown et al. 2020).

Autoregressive models like GPT have led the way in the recent rapid development in AI. They are large attention-based neural networks trained on language modeling – to predict the next token in a sequence – following a massive corpus of text. With enough data and trainable parameters, autoregressive LLMs begin to exhibit features of generalized intelligence, and to solve a broad variety of tasks (Brown et al. 2020).

Large scale pretraining works to equip the model with broad knowledge and the ability to fluently continue text. Base LLMs predict sequences based on the large training corpus. They can generate creative sequences, unseen in the training data, by interpolating across those corpora. This allows, for example, base LLMs to "guess" unobserved facts through analogizing between observed facts.

II. Preference Alignment: IFT and RLHF

Pre-trained base LLMs are powerful language interpolators. While this interpolation of new sequences often works well, it also often goes sideways. Strange and incorrect claims will often come from base models – a phenomenon referred to as hallucination. More generally, a downside of pre-trained base models is that they are not well-aligned with human intentions. For example, if I ask a question, a base model might respond with another question, rather than responding with an answer. The next

steps of "post-training" are designed to address this issue – to *align* LLMs with human preferences.

Instruction fine-tuning (IFT) is a stage of post-training that shifts a language model from simply predicting the next word in a sequence to actively following user prompts. In IFT, token prediction training is undertaken on special corpora: collections of prompt–response pairs that teach the model how to carry out specific tasks and produce outputs in desired formats (Chung et al. 2022). By exposing the model to a wide and diverse set of instructions, researchers enable it to generalize better to new prompts, improving its zero-shot and few-shot performance (Wei et al. 2022).

The next step in LLM alignment is RLHF – Reinforcement Learning from Human Feedback. The goal is to improve upon IFT in aligning outputs with user preferences (Stiennon et al. 2020; Ouyang et al. 2022). RLHF starts with a dataset of pairwise comparisons – user prompts, with at least two different responses, ranked by their usefulness to humans. The system designer trains a reward model based on these comparisons to predict which output humans would tend to prefer. The trained reward model can then provide scores to texts based on their tendency to be preferred to baseline statements. RLHF then optimizes the text generator to increase that score while staying close to the IFT model's generations. In practice, that means better tone, structure, and reason-giving, for example in legal advice.

LLMs that have been aligned with IFT and RLHF provide more useful generations. IFT model generations are interpretable as interpolations across a weighted average of the pre-training and instruction-following corpus. After RLHF, the interpretation is a little more complex. It is an interpolation across the pre-training corpus, IFT corpus, and reward pairs corpus, weighted by the predicted usefulness of the generation.

III. Alignment for Structured Reasoning: RLVR

Reinforcement Learning with Verifiable Rewards (RLVR) extends the logic of alignment by replacing human judgment with automatic checking (Lambert et al. 2024). Instead of asking people which answer they prefer, the system rewards the model only when its output passes an objective test. The idea is simple but powerful: for any task where correctness can be defined in advance, the model learns to generate outputs that a verifier

accepts as true. The verifier can be a program that executes code, a mathematical solver that confirms an answer, or any other function that returns a pass or fail signal.

The training setup mirrors other RL post-training pipelines: the model proposes candidate answers; a checker scores them automatically; updates push the policy toward higher verified success while staying close to the base model to preserve fluency and breadth. A critical feature of this approach is to break down complex problems into a series of simple steps, which can then be solved and verified one-by-one (Lightman et al. 2024). This keeps the focus on correctness and avoids brittle format-chasing. RLVR reduces reliance on costly labels and guards against reward-model gaming, but it is only as broad as the verifiers available. Because checkers can be brittle, designers often add light smoothing or combine RLVR with instruction-following or preference objectives to keep learning signals robust (e.g. DeepSeek-AI et al. 2025).

RLVR-based training is a key ingredient in the recent breakthroughs by "reasoning"-based LLMs, such as o1 and DeepSeek-R1, on math and coding tasks (OpenAI 2024; DeepSeek-AI et al. 2025). The other main ingredient is the introduction of internally observed thinking tokens, which allow the model to perform a lengthy chain of thought that is not (by default) revealed to the user. Alongside that, a large corpus of lengthy high-quality reasoning traces is observed by the model during the IFT phase, in particular to break down complex problems into simple, verifiable steps. Thus the LLM learns the style and structure of long-form reasoning.

Because the reward is tied to external correctness, RLVR reduces dependence on noisy, biased, or subjective human evaluations. It is therefore well suited to structured reasoning tasks such as mathematics, code generation, or logical proofs, where success can be defined as the satisfaction of formal conditions. Relative to IFT and RLHF models, which are well-designed to interpolate among preference-weighted examples, RLVR trains models to follow discrete lines of reasoning that take logically correct paths at each juncture.

IV. Discussion of Post-Training Algorithms

To analogize to classical machine learning algorithms, IFT and RLHF can be seen as akin to K-nearest-neighbors algorithms, where similar prompts are identified in the training corpus by their embeddings, and then the

associated responses are weighted based on their predicted user utility and averaged to produce a response. Meanwhile, RLVR is more akin to a decision tree, where the nodes are verifiable reasoning junctures and the branches are the potential set of answers at that juncture. RLVR teaches the LLM to follow the correct branches and eventually to terminate at an overall satisfactory answer.

Can LLMs trained with IFT/RLHF or RLVR extrapolate to new questions or tasks that are outside the training data? This is an active area of research (Chung et al. 2022; Ouyang et al. 2022; Lambert et al. 2024). A limited "yes" answer to this question is the phenomenon of in-context learning: if new documents are provided with the necessary information to answer questions, then an LLM can answer factual questions about those documents even if they did not appear in the training. If a task requires new value judgments, however, it is unclear how an LLM extrapolates to make such judgments. RLVR, meanwhile, enables an important form of extrapolation – that which involves following verifiable reasoning paths. In principle, RLVR models can extrapolate to new logic, math, or code problems by solving them step-by-step.

So far, I have depicted RLHF and RLVR as separable algorithms defining different LLMs. But in practice, they are not cleanly separated. RLHF datasets have many examples of math and code, where the "correct" answer is annotated as that preferred by users. And RLVR datasets with sets of "correct" labels presumably correspond to what users would pick. The main difference is that the RLHF datasets also contain many subjective comparisons, such as the quality of writing, judgments on levels of sentiment or toxicity, or political or moral judgments, all of which reflect human social and cultural norms rather than objective correctness. Further, the performant RLVR models, such as DeepSeek-R1, are trained on both RLVR and RLHF objectives during the post-training phase (DeepSeek-AI et al. 2025). This means that, potentially, such models can still perform the subjective, interpolative tasks for which RLHF is well-designed. However, Ni et al. (2025) show that RLVR models perform worse than RLHF models in modeling subjective decisions (e.g. toxicity detection), suggesting that RLVR training interferes with the LLM's ability to represent distributed preference sets.

C. LLM Alignment and the Law

This section explores legal reasoning and legal institutions in light of the architectures of large language models. Both domains confront the problem of aligning inference with normative standards – deciding when reasoning should follow explicit rules, rely on experiential judgment, or combine the two. I trace parallels between legal reasoning and model training, focusing on how RLHF corresponds to case-based reasoning, and RLVR corresponds to code-based reasoning.

I. LLM Reasoning and Legal Reasoning

The structure of legal reasoning can be illuminated by comparing it to the successive stages of large language model training. Law, like generative AI, develops through processes of exposure, refinement, feedback, and verification. The steps in legal thought and procedure find parallels in how models are trained to reason – from the absorption of precedent, to doctrinal synthesis, to analogical judgment, to formal rule application.

Legal understanding begins with immersion in precedent and custom, much as models begin with pretraining. Judges, lawyers, and students internalize the language and logic of past decisions by reading widely and repeatedly. Over time, they develop a sense of what arguments sound plausible and what outcomes cohere with established norms (Lamond 2014). This process is not deductive but associative: legal actors, like pretrained models, form a broad intuitive map of patterns and meanings before any explicit rules are applied.

Doctrinal reasoning resembles instruction fine-tuning. From the diffuse materials of experience, jurists extract structured formulations that can guide future decisions. Take in particular the development of legal commentaries or restatements of the law. These materials are developed to interpret facts through settled principles, to follow procedural instructions, and to translate broad concepts into determinate tests. Just as IFT teaches a model to respond coherently to prompts, doctrinal reasoning teaches lawyers to connect precedent to principle through organized, rule-like reasoning.

What about reinforcement learning from human feedback? That element of LLM training can be analogized to the analogical and equitable dimensions of legal reasoning. Judges do not simply execute fixed rules –

they reason by analogy, drawing on previous judgments to decide whether an outcome is contextually justified. Each precedent can be seen as analogous to a pairwise comparison in an RLHF dataset: it features a set of facts (a prompt), two sets of legal arguments from the petitioner and respondent (the two responses for comparison), and a judgment saying which of the two arguments wins (the feedback label). The appeals process acts as an additional subjective feedback system, leading future judges to upweight precedents that have been endorsed by higher courts, and downweight precedents that have been reversed. In addition, the citation network among judges creates a third set of human feedback signals, in which future judges are likely to follow precedents that have been examined and cited by peer judges in the interim. For any given decision, judges collect and aggregate these feedback labels when shaping their arguments and judgments.

Finally, the application of determinate legal rules parallels reinforcement learning with verifiable rewards. When statutes define clear conditions and consequences, the role of the judge is largely to verify whether the elements are satisfied. The process is procedural and externally checkable, not interpretive or analogical. RLVR captures this same mode of code-based reasoning – rule application under an objective verifier, where correctness is determined by a standard outside the decision-maker's discretion.

Seen through this lens, legal reasoning embodies the same layered architecture that defines modern model training. Precedent provides the base of intuitive understanding, doctrine structures that knowledge into usable form, case-based reasoning approximates learning from feedback, and rule application enforces verifiable correctness. Law, like model alignment, thus operates by assembling a variety of learning systems.

II. LLMs and Legal Design

There are corresponding parallels in how LLMs are designed and how legal procedures should be organized. This subsection elaborates on the connection to rules and standards, and *law as data* versus *law as code*. Institutionally, there is an informative distinction between common-law and civil-law jurisdictions.

The long-standing distinction between rules and standards turns on when legal content is specified and how determinate it is (e.g. Kaplow

1992). Rules fix content ex ante, inviting relatively mechanical application. Standards defer content to ex post adjudication and invite balancing. This distinction maps directly onto the difference between RLVR and RLHF. RLVR systems, trained to satisfy externally verifiable criteria, perform best when legal content is rule-like – when conditions and consequences can be specified in advance. RLHF systems, trained to approximate context-specific human judgment, are better suited to standard-like reasoning, where outcomes depend on contextual evaluation. Applying a standard involves identifying relevant analogies, weighting them by their persuasiveness and proximity, and then averaging across them to reach a balanced conclusion. This process mirrors the graded preference learning that defines RLHF.

The fit between legal form and model architecture also parallels the emerging distinction between *law as data* and *law as code* (Livermore and Rockmore 2019). *Law as data* treats legal materials – opinions, briefs, statutes – as empirical sources for retrieval and synthesis. It focuses on understanding how legal actors reason and predict how they will decide. *Law as code*, by contrast, seeks to formalize statutes and procedures into executable logic, converting legal rules into programmable conditions. The analogy is straightforward. *Law as data* corresponds to instruction fine-tuning and RLHF: the signal is graded, context-sensitive, and oriented toward interpolation across precedents. *Law as code* corresponds to RLVR: the signal is discrete, oracle-tied, and based on objective verification. Together, these approaches suggest that legal AI systems will evolve along both trajectories – one oriented toward interpretive reasoning from data, and the other toward executable reasoning from rules.

The two dominant varieties of legal systems – common law and civil law – reflect a deep structural divide between case-based and code-based reasoning (Zweigert and Kötz 1998; La Porta et al. 2008). Common law systems evolve through precedent: courts decide individual disputes and justify outcomes by reference to earlier decisions (Gennaioli and Shleifer 2007). Legal doctrine develops inductively, as patterns of reasoning are inferred from case to case. Civil law systems, by contrast, begin with a comprehensive code that specifies legal rules and their conditions of application. Judicial reasoning proceeds deductively, applying the relevant statutory provisions to the facts at hand.

This common law mode of reasoning aligns closely with reinforcement learning from human feedback. RLHF systems learn through graded evaluation, aggregating many examples of preferred outcomes to approxi-

mate the judgments of a community of evaluators. Similarly, common law reasoning proceeds by analogy to previous cases, weighting prior decisions by their relevance, authority, and coherence with general principles. The inductive and value-sensitive nature of both processes makes RLHF especially well suited to assist judges in common law settings, where legitimacy depends on responsiveness to evolving norms and shared intuitions of fairness.

Civil law reasoning, by contrast, corresponds more directly to reinforcement learning with verifiable rewards. RLVR systems operate under explicit verification criteria: outputs are accepted only when they satisfy an external test of correctness. This mirrors the logic of statutory adjudication, in which a judge determines whether the legal elements are met and then applies the rule mechanically. Because RLVR systems can manage highly structured and nested rule sets, their adoption could support the drafting of more precise and internally consistent codes. Over time, such systems may help civil law jurisdictions move closer to the "law as code" ideal, enhancing transparency and reliability in rule application. This trajectory is already visible in code-based systems such as China's, where judicial practice discourages citation of precedent. China has had early, large-scale adoption of LLM-based decision support tools (e.g. Liu and Li, 2024).

D. Implications for Legal AI

The parallels between how models are trained and how legal systems work have implications for how AI should be used in the legal system. The architectures of RLHF and RLVR do not just resemble different legal reasoning styles; they also suggest how AI systems might align with different legal tasks, doctrines, and institutional forms. In thinking about the integration of AI into law, it is therefore useful to ask which aspects of legal decision-making are best handled by systems trained on preference-based rewards and which by systems trained on verifiable ones.

RLHF-based systems are naturally suited to tasks that require interpretive or evaluative judgment. Because these models learn from graded, context-sensitive feedback, they can interpolate across a range of factual situations and moral intuitions. Many legal standards – such as "reasonableness," "good faith," or "due care" – operate precisely in this way. Deciding whether conduct was negligent or an offer was made in good

faith depends on the aggregation of many contextual features, as well as on sensitivity to community expectations. An RLHF-based model, trained on extensive corpora of judicial reasoning and fact patterns, could approximate this inductive process by identifying clusters of analogous cases and synthesizing their logic. In effect, RLHF offers a scalable mechanism for producing judgments that reflect the continuous and socially grounded nature of standard-based adjudication.

RLVR systems, by contrast, are better aligned with rule application. These models operate under explicit verification criteria, rewarding outputs only when they satisfy external correctness conditions. This makes them well suited to legal domains where compliance can be defined with precision – taxation, regulatory compliance, contract validation, and other areas governed by intricate statutes and administrative rules. RLVR can encode and execute arbitrarily complex legal frameworks, checking whether each factual element or computational step satisfies the relevant conditions. A legal reasoning engine built around verifiable rewards could thus perform the functional equivalent of statutory interpretation and rule enforcement, translating legal code into executable code and ensuring consistency and precision in outcomes.

In practice, however, there is no clean division between subjective and objective tasks, nor between case-based and code-based reasoning. Legal decision-making typically involves a combination of the two. Even in highly codified domains such as tax law, subjective judgments often play a decisive role – for example, in determining whether an expense was incurred for personal or business reasons (Goldin et al. 2024). Similarly, areas traditionally governed by standards still rely on statutory definitions and procedural rules. Just as RLHF and RLVR operate in tandem during model training, human legal reasoning mixes evaluation and verification, weighing analogical arguments against fixed legal criteria.

This hybrid character of law points toward an analogous architecture for legal AI. Effective systems will need to integrate multiple reasoning modes, drawing on both RLHF and RLVR models. An outline for a plausible design is a multi-agent system in which different components specialize in different tasks and route questions accordingly. Determinate rule-following could be handled first by an RLVR model, which checks the applicable statutes or procedural rules. Ambiguous or threshold-based issues could then be referred to an RLHF model, which aggregates reasoning across similar cases to approximate equitable judgment. This layered design echoes recent work in AI on routing among specialized models to

enhance performance and efficiency (Chen et al. 2023; Wang et al. 2024; Ong et al. 2024).

The parallels between AI alignment and legal reasoning are more than conceptual – they offer a framework for designing systems that can think with law, not merely about it. The next step is empirical. Hybrid architectures that combine RLHF's preference-based reasoning with RLVR's verifiable reasoning should be tested systematically on legal AI benchmarks, such as LEXGLUE, LegalBench, and Lexam (Chalkidis et al. 2022; Guha et al. 2023; Fan et al. 2025). Such evaluations can help determine whether the hypothesized match – RLHF with case-based reasoning and RLVR with code-based reasoning – holds in practice, and may motivate new benchmarks that more clearly distinguish the two.

Beyond benchmarking, legal AI systems must be tested in the field. Liu and Li (2024) show that judges using chatbots often render decisions first and then use AI to rationalize them after the fact. Future systems should instead support reasoning itself – helping judges deliberate, structure arguments, and reach better-grounded outcomes. Doing so requires attention not only to logic but also to psychology and institutions, as judges respond to framing, workload, and norms (Engel and Glöckner 2013; Engel 2022). Aligning legal AI with these behavioral realities will ensure that technology complements, rather than substitutes for, human judgment. The ultimate goal is not to mechanize justice, but to build systems that model and enhance the structure of human legal reasoning. The right approach will join the verifiability of rules with the sensitivity of standards, and the precision of code with the adaptability of human thought.

References

Ashley, Kevin D. 1992. "Case-Based Reasoning and Its Implications for Legal Expert Systems." *Artificial Intelligence and Law* 1 (2): 113–208.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. "Neural Machine Translation by Jointly Learning to Align and Translate." *International Conference on Learning Representations (ICLR)*.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. "A Neural Probabilistic Language Model." *Journal of Machine Learning Research* 3: 1137–55.

Brown, Tom B., et al. 2020. "Language Models Are Few-Shot Learners." *Advances in Neural Information Processing Systems (NeurIPS)*.

Chalkidis, Ilias, et al. 2022. "LEXGLUE: A Benchmark Dataset for Legal Language Understanding." *Findings of the Association for Computational Linguistics (ACL)*.

Chen, L., Zaharia, M., & Zou, J. (2023). Frugalgpt: How to use large language models while reducing cost and improving performance. arXiv preprint arXiv:2305.05176.

Chung, Hyung Won, et al. 2022. "Scaling Instruction-Finetuned Language Models." *arXiv:2210.11416*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, et al. 2025. "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning." arXiv preprint arXiv:2501.12948.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *Conference of the North American Chapter of the ACL (NAACL)*.

Engel, Christoph. 2022. "Judicial Decision-Making. A Survey of the Experimental Evidence." *SSRN Electronic Journal*. doi:10.2139/ssrn.4199122.

Engel, Christoph, and Andreas Glöckner. 2013. "Role-Induced Bias in Court. An Experimental Analysis." *Journal of Behavioral Decision Making* 26 (3): 272–84.

Fan, Y., Ni, J., Merane, J., Salimbeni, E., Tian, Y., Hermstrüwer, Y., … Elliott Ash, Niklaus, J. (2025). LEXam: Benchmarking Legal Reasoning on 340 Law Exams.

Gennaioli, Nicola, and Andrei Shleifer. 2007. "The Evolution of Common Law." *Journal of Political Economy* 115 (1): 43–68.

Goldin, J., Koehne, S., & Lawson, N. 2024. "Optimal Income Tax Deductions for Mixed Business and Personal Expenditures." National Bureau of Economic Research.

Guha, Neel, et al. 2023. "LegalBench. A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models." *arXiv:2308.11462*.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–80.

Jurafsky, Daniel, and James H. Martin. 2023. *Speech and Language Processing*. 3rd ed.

Kaplan, Jared, et al. 2020. "Scaling Laws for Neural Language Models." *arXiv:2001.08361*.

Kaplow, Louis. 1992. "Rules versus Standards. An Economic Analysis." *Duke Law Journal* 42 (3): 557–629.

Lambert, Nathan, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, et al. 2024. "TÜLU 3: Pushing Frontiers in Open Language Model Post-Training." arXiv preprint arXiv:2411.15124.

Lamond, Grant. 2014. "Precedent and Analogy in Legal Reasoning." In *Stanford Encyclopedia of Philosophy*, ed. Edward N. Zalta.

La Porta, Rafael, Florencio López-de-Silanes, and Andrei Shleifer. 2008. "The Economic Consequences of Legal Origins." *Journal of Economic Literature* 46 (2): 285–332.

Lightman, Hunter, Vineet Kosaraju, Yura Burda, et al. 2024. "Let's Verify Step by Step." *ICLR*.

Liu, J. Z., & Li, X. (2024). How do judges use large language models? Evidence from Shenzhen. Journal of Legal Analysis, 16(1), 235–262.

Livermore, Michael A., and Daniel N. Rockmore, eds. 2019. *Law as Data. Computation, Text, and the Future of Legal Analysis*. Santa Fe, NM: SFI Press.

Mattei, Ugo. 1997. "Three Patterns of Law. Taxonomy and Change in the World's Legal Systems." *American Journal of Comparative Law* 45 (1): 5–44.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." *ICLR Workshop*.

Ni, Jingwei, Yu Fan, Vilém Zouhar, Donya Rooein, Alexander Hoyle, Mrinmaya Sachan, Markus Leippold, Dirk Hovy, and Elliott Ash. "Can Large Language Models Capture Human Annotator Disagreements?." arXiv preprint arXiv:2506.19467 (2025).

Ong, I., Almahairi, A., Wu, V., Chiang, W. L., Wu, T., Gonzalez, J. E., … & Stoica, I. (2024). Routellm: Learning to route llms with preference data. arXiv preprint arXiv:2406.18665.

OpenAI. 2024. "OpenAI o1 System Card." arXiv preprint arXiv:2412.16720.

Ouyang, Long, et al. 2022. "Training Language Models to Follow Instructions with Human Feedback." *NeurIPS*.

Radford, Alec, et al. 2019. "Language Models Are Unsupervised Multitask Learners." OpenAI Technical Report.

Rissland, Edwina L., Kevin D. Ashley, and L. Karl Branting. 2005. "Case-Based Reasoning and Law." *The Knowledge Engineering Review* 11 (2): 113–44.

Shannon, Claude E. 1951. "Prediction and Entropy of Printed English." *Bell System Technical Journal* 30 (1): 50–64.

Stiennon, Nisan, et al. 2020. "Learning to Summarize with Human Feedback." *NeurIPS*.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. "Sequence to Sequence Learning with Neural Networks." *NeurIPS*.

Vaswani, Ashish, et al. 2017. "Attention Is All You Need." *NeurIPS*.

Wang, J., Wang, J., Athiwaratkun, B., Zhang, C., & Zou, J. (2024). Mixture-of-agents enhances large language model capabilities. arXiv preprint arXiv:2406.04692.

Wei, Jason, et al. 2022. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." *NeurIPS*.

Zweigert, Konrad, and Hein Kötz. 1998. *An Introduction to Comparative Law*. 3rd ed. Oxford: Clarendon Press.