Peeking Inside LLMs: Disentangling Internal Signals In Legal Violation Prediction

Anonymous ACL submission

Abstract

The use of Large Language Models (LLMs) has recently been ubiquitous in the legal field. Legal practitioners have increasingly been using LLMs as a supportive tool, but with caution. One peril that concerns practitioners is the variability in performance across traditional legal NLP tasks, further exacerbated by a lack of transparency. Our work is the *first ever* to adopt interpretability studies in usage of LLMs as legal assistant, focusing on one downstream task of legal violation prediction. We address the concern whether general-purpose LLMs such as LLaMA or Gemma, can be relied upon. In the work, we investigate model's latent representation space to evaluate whether the model's generated outputs align with its internal knowledge, thereby attempting to turn LLM utilization in the legal domain from black-box into moderately informed.

1 Introduction

The discipline of computational law (Love and Genesereth, 2005) is increasingly concerned with the integration of advanced artificial intelligence, particularly Large Language Models (LLMs) – ChatGPT or Gemini, into legal workflows (Guha et al., 2023a; Shui et al., 2023; Yue et al., 2023; Yuan et al., 2024). An LLM operates as a generative model, creating statistically probable token sequences (Vaswani et al., 2017) to perform tasks like legal text summarization (Shukla et al., 2022; Datta et al., 2023; Aumiller et al., 2022), classification tasks (Singh et al., 2025). While the generative capabilities of Large Language Models (LLMs) promise to revolutionize legal analytics, integration of such systems into the justice system is shadowed by the **perils of the "Black Box"** (Garrett and Rudin, 2023). In a domain predicated on due process and the right to a reasoned explanation, deployment of opaque models creates a constitutional friction: a defendant cannot effectively challenge a decision derived from an inscrutable algorithm (Gryz and Rojszczak, 2021). The opacity conceals critical failures, e.g. factual hallucinations where models confidently fabricate non-existent case law (Warren, 2025; Magesh et al., 2024), leading to judicial sanctions (Choudhary, 2025; Cecco, 2024; Merken, 2025) – and potential amplification of historical biases that threaten equitable treatment (Shin et al., 2024). Therefore, the transition into **operational legal AI demands a paradigm shift from *blind trust* to *informed usage***. It is insufficient for a model to be merely accurate – *it must be interpretable*, transforming a legal practitioner's role from a passive recipient of model outcomes to an active auditor of its reasoning. Therefore, **rise of opaque decision-making systems must be accompanied by studies that aim to interpret the systems.**

Interpretability is about opening up a *black box* and inspecting *what's going on inside* when model makes a decision. Instead of stopping at "the model predicted X", it asks *what internal signals pushed it there* – which model component(s) is(are) more relevant for a task, or whether it is relying on shortcuts to generate convincing yet incorrect answers, or whether model encodes signals about its own uncertainty, hallucinations or lack of truthfulness. By directly delving into model internals – activations, weights etc. to study the model beyond surface-level prompting-based approaches, such studies uncover opaque systems into interpretable ones.

Prior literature in Legal AI has employed extrinsic evaluation methods such as Chain-of-Though and similar prompt-based approaches, for relevant insights into the usage of Large Language Models (Yu et al., 2023; Chu et al., 2025; Yu et al., 2025); however, such methods have been demonstrated to have limitations (Turpin et al., 2023) – prominent works (Li et al., 2023; Azaria and Mitchell, 2023) in interpretability studies have shown LLMs *often* encode information about truths

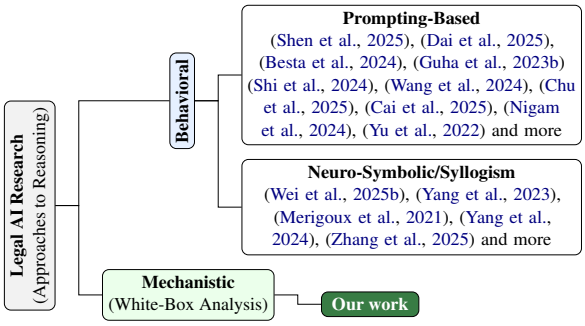
Our Contributions

- Our work, to our knowledge, is the *first ever* to adopt interpretability studies in usage of Large Language Models in Legal AI, through the downstream task of legal violation prediction. We explore model's latent representation space to derive signals about applicability of individual violations (e.g. statutes) predicted by itself in zero-shot inference — essentially, to investigate whether model understands in its internals when a given violation is not valid for a legal court case document *yet* outputs it.
- We find that models indeed encode information about *applicability* of their own predicted violations, to a considerable extent — models can contradict their own predictions acquired via zero-shot inference.

while the final response is incorrect or deviated. Unchecked use of such general-purpose LLMs in legal AI therefore can trigger an avalanche of factual inconsistencies in responses – a highly undesirable behaviour that can challenge the advancements made thus far in the field. **Our work aims to extend interpretability studies to the use of LLMs in Computational Law – a high-stakes domain that demands a transition from black-box to informed usage.**

Organization of the Paper. Section 3.2 provides grounding of the approach, and Section 4 follows the experiments. Finally, Section 4.6 reports the observations we noted via the experiments.

2 Related Works



2.1 Large Language Models in Legal AI

The integration of Large Language Models (LLMs) into the legal domain has shifted the field from traditional inductive logic programming to generative tasks, including Legal Judgment Prediction (LJP), contract review, and statutory reasoning (Cui et al., 2023; Nay et al., 2024). However, the deployment of these models in high-stakes environments is constrained by the "black box" nature of proprietary models and the rigid requirement for a "right to explanation" (Bibal et al., 2021).

To bridge the gap between model performance and legal transparency, behavioral and neuro-symbolic avenues have been explored. Chain-of-Thought (CoT) prompting (Wei et al., 2022) has been widely adopted to simulate the "IRAC" (Issue, Rule, Application, Conclusion) framework. While CoT significantly improves performance on benchmarks like LegalBench (Guha et al., 2023b), recent work questions its faithfulness. Turpin et al. (2023) and Lanham et al. (2023) demonstrate that CoT is often a post-hoc rationalization – models can output a correct reasoning chain while relying on completely different internal features (e.g., biases in the prompt) to predict the answer, thereby making it unreliable in the field of Legal AI.

Alternatively, Neuro-symbolic approaches attempt to hybridize neural networks with formal logic, using LLMs solely as "translators" to extract facts for deterministic solvers (Wei et al., 2025a). While these systems guarantee logical consistency if the extraction is correct, the initial neural translation step remains opaque and susceptible to hallucination, leaving the core reliability problem unsolved (Dahl et al., 2024a).

2.2 Interpretability for Mechanistic Insights

Distinct from behavioral analysis, Mechanistic Interpretability (MI) primarily involves reverse-engineering models to identify computational subgraphs responsible for certain model behaviors (Zhao et al., 2024; Rai et al., 2025). Foundational work in MI focuses on decomposing the "black box" into atomic units. A major challenge is Polysemanticity, where individual dimensions represent multiple unrelated concepts. To resolve this, Sparse Autoencoders (SAEs) have emerged as a state-of-the-art technique for disentangling superposed activations into monosemantic features (Bricken et al., 2023; Cunningham et al., 2023). Parallel to feature extraction, Activation Patching (or Causal Mediation Analysis) allows localization of specific behaviors to precise model components (Vig et al., 2020), while automated frameworks like ACDC scale this discovery process to map complex algorithms within the weights (Conmy et al., 2023).

Mechanistic insights find applications beyond theoretical analysis. Methods such as Rank-One Model Editing (ROME) enable the surgical correction of factual errors in a model's memory layers without expensive retraining (Meng et al., 2022). Probing and representation analysis examine a model's internal activations to decode its latent knowledge. Kadavath et al. (2022) shows that a model's calibrated output probabilities can serve as a strong indicator of its epistemic uncertainty. Furthermore, a model's internal representations often encode factual truth even when the final output is incorrect, indicating a divergence between latent knowledge and the generative process (Burns et al., 2022; Azaria and Mitchell, 2023). While these technical frameworks are advancing, their application to the normative and procedural requirements of legal reasoning remains a significant open problem.

3 Methodology

3.1 Preliminary

The section outlines the relevant concepts.

Latent Representations. Given an input sequence of tokens T=$\{t_1, t_2, t_3, ..., t_n\}$, a Large Language Model $\mathcal{M}$ computes a set of hidden states. Let us denote the hidden state at layer l at token position i as $x_i^{(l)} \in \mathbb{R}^{d_{model}}$, where d_{model} is the dimension of the hidden state space. x represents a high dimensional (e.g. 4096) *dense* semantic state of the model, generally referred to as *activation vector*.

Superposition and Polysemanticity. While raw activation vectors contain rich and useful semantic information, they are difficult to interpret directly due to being dense in nature. In a dense representation, distinct concepts are often compressed into shared subspaces, causing individual dimensions to be polysemantic i.e. responsive to multiple unrelated concepts. Elhage et al. (2022) and Bricken et al. (2023) demonstrate that a dense activation $x_i^{(l)}$ is a linear combination of significantly more "true" features than the dimensions of the model, $N_{features} \gg d_{model}$. In simpler terms, a model's internal dimensions are significantly fewer than the number of concepts it understands, therefore a raw activation vector represents information in a highly compressed space.

Sparse Auto Encoders. A Sparse Auto Encoder (SAE) serves as a disentanglement mechanism, that projects raw dense activations into an overcomplete basis where features are often monosemantic and therefore *interpretable* (Bricken et al., 2023; Cunningham et al., 2023). SAE maps an input activation $x \in \mathbb{R}^{d_{model}}$ to an overcomplete, sparse representation $f \in \mathbb{R}^{d_{SAE}}$ where $d_{SAE} \gg d_{model}$, via a learned encoder:

$$f = \sigma(W_{enc}x + b_{enc})$$

Here, f represents the disentangled features. σ is an activation function that enforces non-negativity and sparsity. Next, a decoder then approximates the original input, indicated as $\hat{x}$, using these features:

$$\hat{x} = W_{dec}f + b_{dec}$$

The parameters are optimized by minimizing a joint objective function that enforces reconstruction fidelity while imposing an $\mathcal{L}_1$ penalty on f to ensure sparsity:

$$\mathcal{L}_{SAE} = \|x - \hat{x}\|_2^2 + \lambda\|f\|_1$$

By enforcing sparsity (λ), the model isolates the specific independent features active for a given input, effectively resolving the problem caused by unrelated concepts sharing the same subspace.

TopK SAE. These SAEs enforce sparsity by retaining only the K features with the largest values, zeroing out others (Gao et al., 2024). The encoder in this case is defined as,

$$f = TopK(W_{enc}x + b_{enc})$$

This eliminates the need for an explicit sparsity penalty, since TopK imposes a rigid sparsity constraint in terms of a predetermined value of K.

3.2 Our Approach

Here we discuss the approach taken in the paper to uncover a black-box usage of LLM to extract insights into the internal representations of a model. The approach is further discussed in Section 4.

Stage One: Traditional Zero-Shot Setup

We have a dataset $\mathcal{D} = \{u\}_{i=1}^{N}$, with N inputs. We prompt $\mathcal{M}$ to generate a set of potential labels for a given input $u \in \mathcal{D}$, *essentially* a multi-label classification task. We define a generation prompt template $\mathcal{T}_{gen}$ for zero-shot labels, $\mathcal{Y}_{cand}$:

$$\mathcal{Y}_{cand} = \mathcal{M}(\mathcal{T}_{gen}(u))$$

where $\mathcal{Y}_{cand} = \{l_1, l_2, \ldots, l_k\}$ is the set of k distinct candidate labels generated by the model. Here, the stage identifies an initial set of potential labels for u – a single-step retrieval.

Stage Two: Ground Truth Verification and Activation Caching

In the following stage, we incorporate a verification step to the candidate labels acquired in the

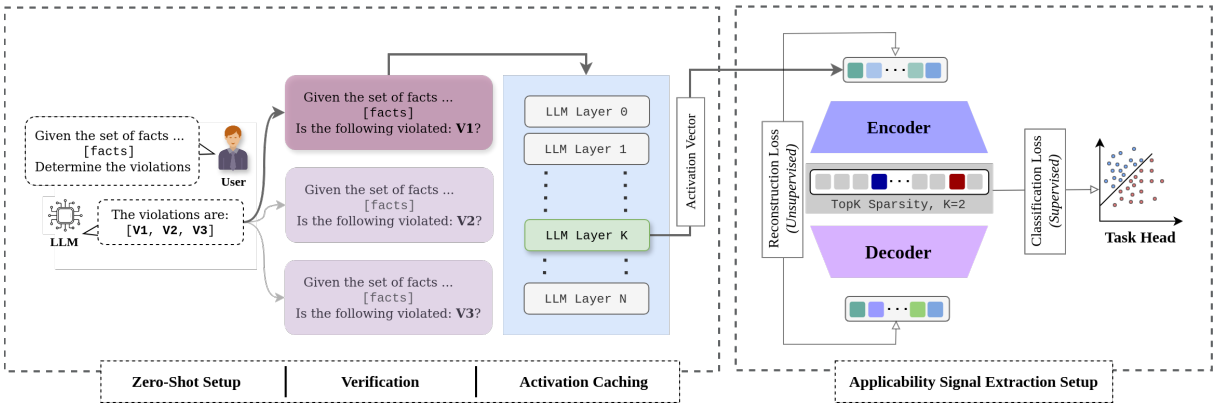


Figure 1: Our three-stage approach toward extracting an indicative signal, referred to as *applicability* signal, from model's latent representation space additionally engineered via Sparse Auto Encoder based setup, that encodes information about validity of a violation in a given legal document. The three stages are: Zero-shot Inference — Ground-truth Verification and Activation Caching — Extraction of Relevant Signal.

preceding stage as a set – whether each individual candidate label is applicable given u, and therefore transform the multi-label classification task into a binary classification task *via* ground-truth alignment. The purpose of the stage is to deal with each candidate violation in isolation thereby making the process more focused for interpretability studies. For each candidate label $l_j \in \mathcal{Y}_{cand}$ generated by $\mathcal{M}$ in Stage One, we assign a binary target label y_j based on its validity against the dataset's ground truth annotations $\mathcal{Y}_{GT}$. The target is assigned as follows:

$$y_j = 1(l_j \in \mathcal{Y}_{GT}) = \begin{cases} 1, & \text{if } l_j \text{ is in ground-truth.} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We construct a specific Question-Answering prompt $\mathcal{T}_{ver}(u)$ to verify the applicability of each l_j by concatenating to the original input u a query regarding label l_j (e.g.,"Is the label 'l_j' applicable?"). Therefore, we now have a dataset for verification of applicability as,

$$\mathcal{D}_{binary} = \{(\mathcal{T}_{ver}(u_i), y_{ij})\}_{i=1,j}^{N} \tag{2}$$

where N is the total number of original inputs and j indicates labels obtained in Stage One.

Instead of generating a response, we perform a forward pass through $\mathcal{M}$ and extract activation vector at each layer l of $\mathcal{M}$ from the final token of the prompt $\mathcal{T}_{ver}(u)$. We now have a collection of activation vectors for $\mathcal{D}_{binary}$ for each layer i.e. for a model with $\mathcal{L}$ layers we have $\mathcal{L}$ sets of activations.

Stage Three: Task-Guided Sparse Autoencoder Training. Stage three is dedicated to extracting an indicative signal, which we refer to as *applicability* signal, from raw activations with the goal of understanding whether model has an inherent understanding of an individual candidate label's validity in isolation. Since a raw activation vector is polysemantic in nature, and our aim is to extract only *applicability* signal, it requires a transformation – SAE is the tool that performs the decomposition into potentially monosemantic features relevant to our goal. However, SAE is an unsupervised mtheod. To ensure learned relevant features are appropriately weighted to the specific problem of candidate label verification, we extend the standard SAE architecture with a supervised binary classification head. Now, the setup consists of three components • Encoder • Decoder • Task Head.

We employ a two-phase learning strategy to balance structural fidelity with task-specific semantic alignment.

Phase 1: *Unsupervised Structural Initialization* ($t \leq T_{SAE}$) For the initial T_{SAE} epochs, we optimize the setup for TopK sparsity and reconstruction of the activation space to learn a stable feature dictionary using the loss function $\mathcal{L}$ defined earlier.

Phase 2: *Supervised Semantic Alignment* ($t > T_{SAE}$). Once the dictionary is initialized, we introduce the supervised objective. We freeze the reconstruction path i.e. decoder and activate the classification head, updating only the encoder to prioritize features that distinguish between valid and invalid labels. The loss function is extended with a Binary Cross-Entropy (BCE) term:

$$\mathcal{L} = \mathcal{L}_{SAE} + \gamma BCE(\hat{y}, y) \tag{3}$$

where γ scales the supervision signal. This forces

latent features f to align with the ground-truth applicability of the labels, effectively isolating the "directions" in the activation space that correspond to correct classification.

3.3 Deployment: Latent-Space Geometric Uncertainty Estimation

To quantify the applicability signal encoded within the model, we employ a geometric scoring mechanism operating directly within the sparse feature space learned by the Top-K SAE. Following prior literature around Linear Hypotheses (Park et al., 2023; Mikolov et al., 2013), we postulate that the semantic distinction between "applicable" and "non-applicable" labels is encoded as a linear direction vector connecting their respective clusters.

Centroid Calibration. We first calibrate the semantic direction using the labeled validation set. We process the validation activations through the trained SAE encoder to obtain their sparse latent representations $f \in \mathbb{R}^{d_{SAE}}$. We partition these representations into two sets based on their ground truth labels: $\mathcal{F}_{pos}$ (label present in the ground-truth) and $\mathcal{F}_{neg}$(absent). We then compute the centroids for each class in the sparse space:

$$\mu_{pos} = \frac{1}{|\mathcal{F}_{pos}|} \sum_{f_i \in \mathcal{F}_{pos}} f_i \qquad \mu_{neg} = \frac{1}{|\mathcal{F}_{neg}|} \sum_{f_j \in \mathcal{F}_{neg}} f_j \tag{4}$$

These centroids μ_{pos} and μ_{neg} represent the archetypal feature combinations for applicable and non-applicable labels, respectively. We define the latent applicability vector $\mathbf{v}$ as the normalized direction connecting the negative centroid to the positive centroid:

$$\mathbf{v} = \frac{\mu_{pos} - \mu_{neg}}{\|\mu_{pos} - \mu_{neg}\|_2} \tag{5}$$

Latent Projection Scoring. For a new, unlabelled candidate with dense activation x, we first project it into the sparse latent space via the already trained SAE encoder:

$$f_{new} = \text{TopK}(W_{enc}x + b_{enc}) \tag{6}$$

We then calculate the uncertainty score $\mathcal{S}(x)$ by projecting this sparse representation f_{new} onto the validity vector v, using the negative centroid as the geometric origin:

$$\mathcal{S}(x) = (f_{new} - \mu_{neg}) \cdot \mathbf{v} \tag{7}$$

We utilize $\mathcal{S}(x)$ as indication for uncertainty encoded within the activation space of a model.

3.4 Summary of Our Approach

Here we summarize our approach for the specific task of legal violation prediction; therefore, general terminologies used thus far are replaced with task-relevant terminologies. Figure 1 captures the workflow. We first determine the performance of each model in a traditional zero-shot setup where the task of the model is to predict violations, *essentially* a multi-label classification task. While a zero-shot setup offers single-step strategy for identifying potential violations, its reliability is inherently limited (Chamieh et al., 2024). This approach can produce outputs that are not fully grounded in the provided text, leading to a high rate of false positives or hallucinated violations (Dahl et al., 2024b). The goal of zoro-shot stage, in our work, is not to produce a final definitive list of violations, but to generate a set of plausible candidate violations. Therefore, a layer of verification is imperative. Accordingly, we next frame each model-determined violation into Question-Answering setup to test for applicability – a prompt with original input and each plausible violation prefixed with "Is the following applicable: `[candidate violation]`?". Instead of generating a response in tokens, we delve into the latent representation space of model in search of a pattern that can later be explored. We inspect a relevant layer of models for indications of distinguishability among *applicable* and *non-applicable* violations (based on available ground-truth labels) given by model in zero-shot setup. In the final stage, we process internal representations to derive a value indicating *applicability* encoded within the model – theoretical grounding is established earlier.

4 Experiments

The experimental aim is to extract a signal encoded within model's internals that indicates whether the model understands the validity of its own responses, *yet* provides incorrect output.

4.1 Setup

Datasets and Downstream Task. The selection of datasets – the European Court of Human Rights (ECtHR) (Chalkidis et al., 2021) and Indian Legal Statute Identification (ILSI) (Paul et al., 2021) – offer *jurisprudential heterogeneity*. The ECtHR

represents a supranational body adjudicating international human rights law across 46 diverse European member states. In contrast, the ISLI dataset originates from the apex court of India, a major Global South nation with a distinct common law and post-colonial legal framework. Appendix section A contains more details.

Models. We select *three* open-source sub-10B models – LLaMA 3.1 8B (Team LLaMA, 2024), and Gemma 2 9B (Gemma Team, 2024). We run the models on 2 × 24GB A6000 GPUs. Hereafter, we refer to the models as LLaMA and Gemma.

4.2 Acquiring Plausible Candidate Violations

Our approach begins with zero-shot inference to acquire a set of potential violations in a single step. Input to a model consists of a set of facts pertaining to a document and the model responds with a set of potential violations (e.g. articles in ECtHR) which we refer to as candidate violations (labels). Table 1 reports the scores.

•	Model	Precision	Recall	F1
ECtHR	LLaMA	88.38	31.66	46.22
	Gemma	61.03	67.45	64.08
ILSI	LLaMA	55.93	51.98	53.88
	Gemma	75.77	58.87	66.26

Table 1: Zero-Shot Performance on the datasets.The scores are in micro-average. We observe low to moderate F1-scores across models and datasets, with overall performance in Gemma comparatively better.

We observe that Gemma 2 9B performs better on both datasets, with Recall always higher than LLaMA 3.1 8B. In the next stage, we decompose the multi-label classification setup into a binary classification setup via ground-truth alignment.

4.3 Ground Truth Verification and Activation Caching

Here, we add a layer of verification to the set of candidate violations acquired in the preceding stage. We frame each candidate violation corresponding to every document as a query – whether the violation initially given by the model in the zero-shot stage is applicable or not. We decompose the multi-label classification task into a binary classification task where the presence of a candidate label in the ground-truth is considered as *correct* and the absence of a candidate label is considered as *incorrect*. For example, consider a document d from ECtHR with the following set of ground-truth violations, d_G={Article 6}, and candidate violations, d_C={Article 6, Article 13}. Since *Article 6* is a candidate violation that is present in d_G, it is considered as correctly determined by the model as violated; however, Article 13 is absent and therefore an incorrect prediction by the model. The example here is decomposed as: {1, 0} where 1 indicates presence and 0 otherwise. The entire dataset is now transformed into a set of questions and a corresponding label.

We incorporate the layer of verification through an additional round of prompting. Now, the model receives the same set of facts corresponding to a document as earlier, with a question "Is the following violated: `[candidate violation]`?", e.g., for the example given earlier one question is "Is the following violated: Article 13?". The exact prompt is provided in the appendix section A.

However, instead of generating a response, we extract activation from each layer of the model at the final token position. We now have a set of activations each corresponding to a question and a binary label indicating presence or absence in the ground-truth. The following stages utilize the set of activation vectors collected in this stage, and involve no generation of tokens hereafter.

4.4 Task-Guided Sparse Auto Encoder

We utilize SAE as a *refinement mechanism* to distill the model's dense, entangled activations into a more legible form potentially free of interfering signals not relevant to applicability, making it more interpretable. This ensures that our geometric analysis is performed on a consolidated, isolated signal. This stage is explained in detail in Section 3.2.

Hyperparameter Optimization and Model Selection. To identify the optimal architectural configuration for our SAE, we perform a grid search over two primary hyperparameters: expansion factor $\mathbb{E}$ and sparsity constraint $\mathbb{K}$. $\mathbb{E}$ indicates a value by which initial model dimension is expanded, and $\mathbb{K}$ is essentially K in TopK.

- **Search Space:** We evaluate over $\mathbb{E} \in \{1, 2, 4, 8, 16\}$, where $d_{SAE} = \mathbb{E} \times d_{model}$. For each $\mathbb{E}$, we vary $\mathbb{K} \in \{1, 2, 3, 4, 5\}$.
- **Training and Evaluation:** A unique SAE is trained for every combination $(\mathbb{E}, \mathbb{K})$. The performance of each configuration is measured by the Area Under the Receiver Operating Characteristic curve (AUC) obtained from the task-specific binary classification head. This evaluation is performed

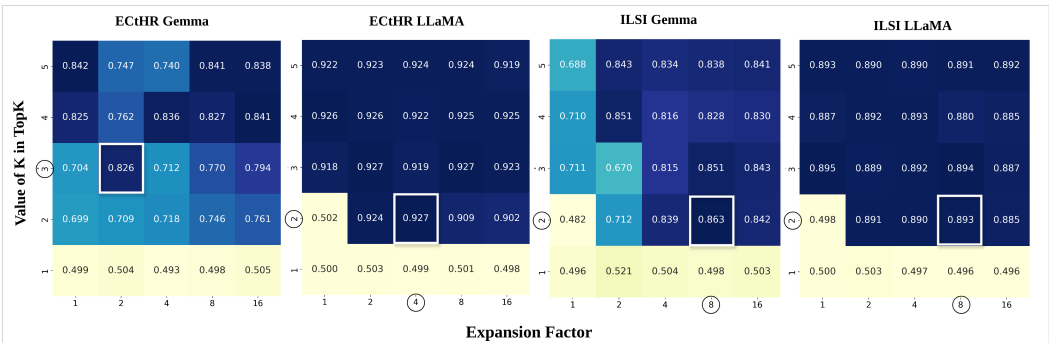


Figure 2: Performance of the SAEs for each combination of $\mathbb{E}$ and $\mathbb{K}$, in terms of AUC Scores.

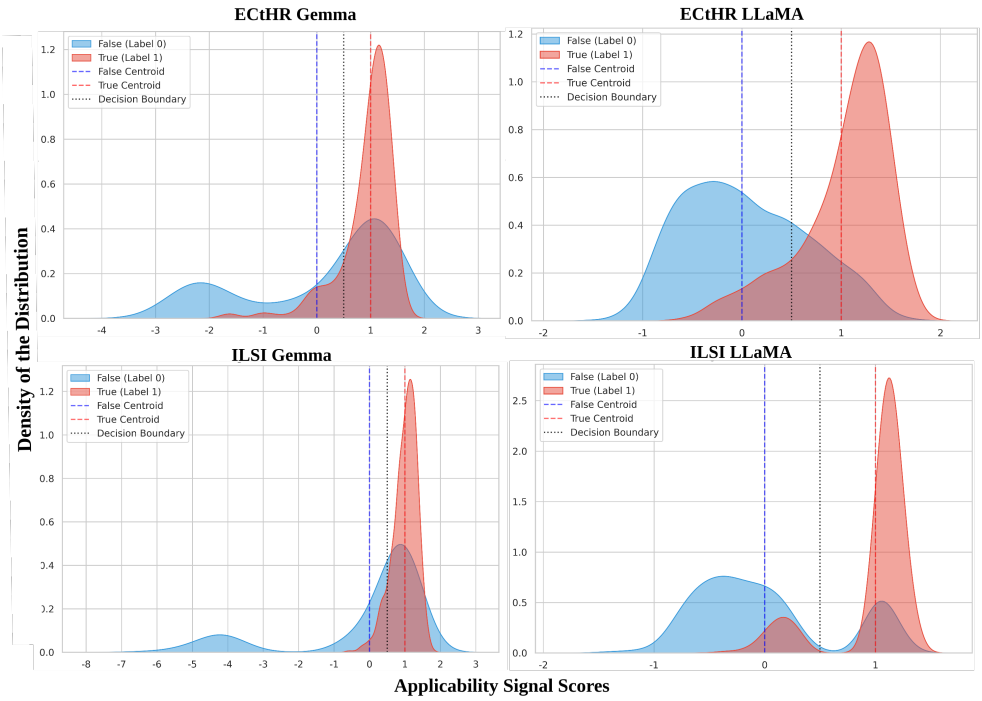


Figure 3: Distribution of *applicability* signal scores across models and datasets. Despite considerable distribution overlap distributions for False can be separated from True in the negative region, indication for False is more appropriately encoded.

on a held-out val set only after training is complete.

• **Selection Criteria:** Our objective is to obtain a combination where classification performance maximizes while maintaining the highest possible level of feature sparsity i.e. minimal $\mathbb{K}$. We identify a point of diminishing returns, selecting the configuration where further increases in $\mathbb{E}$ or $\mathbb{K}$ yield marginal performance gains. A minimal value for $\mathbb{K}$ is aimed for to maintain the monosemantic integrity of the sparse representation.

For inspection, we select one layer per model – the layer where discriminative signal peaks; Following are the best layers: 28 in LLaMA for both ECtHR and ILSI, 41 and 24 respectively in Gemma for ECtHR and ILSI. Therefore, we now explore total four layers across datasets and models. Selection process is detailed in appendix section A.1.

We capture the performance in each combination ($\mathbb{E}, \mathbb{K}$) in the figure 2; *highlighted* cells in the figure correspond to the optimal combinations.

4.5 Quantifying Applicability Signal

Once the optimal SAE configuration is selected, we utilize its latents to derive an *applicability* metric.

• **Centroid Calibration.** Using the sparse features f_{val} extracted from the val set, we compute the class-conditional centroids μ_{pos} and μ_{neg} (as defined in Section 3.3). These centroids represent the *idealized* latent centers for applicable (True) and non-applicable (False) labels, respectively.

• **Directional Vector Construction.** We define a

semantic axis of applicability as a vector $\mathbf{v}$ originating at μ_{neg} and terminating at μ_{pos}.

• **Projection onto the Axis.** For each sample in the test set, we project its sparse representation f_{test} onto $\mathbf{v}$. This projection yields a scalar score $\mathcal{S}$ that quantifies its position relative to the false centroid μ_{neg}.

• **Interpreting Scores.** The resulting scores provide a nuanced view of the model's certainty:
$\mathcal{S} = 0$ and $\mathcal{S} = 1$, when the sample aligns perfectly with the *False* and *True* centroid, respectively.
$0 < \mathcal{S} < 1$, sample resides in an weak region.
$\mathcal{S} < 0$ or $\mathcal{S} > 1$, when the projection lies beyond the centroids, indicating *extreme* alignment. A super-positive score suggests the sample is unambiguously *True*.

4.6 Observations

Figure 4 show distributions of scores among test samples f_{test} for the datasets. Here, we note the conclusions that can be drawn from the distributions, and defer the structural analysis of the graphs in the appendix (section A.3).

Gemma. The False curve extends way beyond μ_{neg} in the negative territory – for ECtHR -4 and for ILSI -7. This indicates that Gemma treats False as opposite of True – pushing the scores in the direction opposite to the True peak. Gemma is

LLaMA. The True and False curves overlapping considerably and scores for either rarely extending to the extreme zones, indicate that LLaMA is unlikely confident when the answer is correct i.e. the given violation is valid. The applicability signal is *weak*.

The distributions lead to two broad conclusions, **one**, Gemma is more confident when a given violation is invalid; it tends to push the signal away from the truth i.e. treats non-applicability as "anti-applicability",
two, applicability signal in LLaMA is confusing; a slight variation can lead to an entirely different understanding about validity, therefore must be used with caution.

Utilizing disentangled applicability signal. We investigate the scores within either distributions to understand whether model's distinguishing ability aligns with the ground-truths. Accordingly,
• First, we consider the entire range of scores.
• Next, we reject samples in the ambiguous zone. An ambiguous zone is determined in terms of entropy (values are given in appendix A.4).

Here, we compute two metrics – 1. Pearson's correlation indicates how scores align with labels, and 2. Accuracy, noted in table 2. We note that a generally poor-performing Gemma has the highest score for the dataset ILSI in the unambiguous regions with an Accuracy score of 86.21%.

	Correlation	Accuracy
Entire Range of Scores		
ECtHR-LLaMA	0.621	78.64
ECtHR-Gemma	0.378	65.63
ILSI-LLaMA	0.701	82.61
ILSI-Gemma	0.399	72.34
Excluding Ambiguous Region		
ECtHR-LLaMA	0.676	86.98
ECtHR-Gemma	0.544	75.00
ILSI-LLaMA	0.759	87.05
ILSI-Gemma	0.658	86.21

Table 2: Correlation with ground-truth and Accuracy scores. LLaMA performs considerably better across datasets, while Gemma's best performance is observed in ILSI in unambiguous zone.

As the high accuracy scores in table 2 indicate, when a model receives each predicted violation individually, it can *to a considerable extent* encode information about its applicability i.e. it can detect when its own predictions are not valid. In another view, **models can contradict their own predictions acquired via zero-shot inference**. Therefore, **mere zero-shot performance cannot be considered as the sole indication of a model's eligibility as a reliable legal AI assistant**, demanding a deeper mechanistic understanding of its internals.

5 Conclusion

The rapid integration of Large Language Models and their prevailing "black-box" usage introduces critical perils that demand greater transparency that behavioral testing alone cannot provide. Addressing this need for intrinsic verification, our work peeks inside two general-purpose open-source LLMs to derive mechanistic insights for the task of legal violation prediction. We propose an approach for extracting an isolated indicative signal about applicability of violations predicted by the model itself in zero-shot inference. This signal enables us to discern the model's true internal state – revealing whether it genuinely assesses a violation as applicable, or is merely generating the text of a violation without the requisite internal understanding. This distinction offers a novel pathway toward interpretable, comparatively transparent legal assistants that go beyond surface-level text generation.

Limitations

Concerns that may be considered as limitations are two – first, our work is limited to two jurisdictions (European Court of Human Rights and Indian Supreme Court) which does not present an extensive evaluation of the models across countries and regions; and second, we consider two models both are of similar size (LLaMA has 8 billion parameters and Gemma has 9 billion) therefore model size has not been considered as a factor. However, our purpose was to show that high-stake domains such as law must adopt interpretability studies that dive deeper than surface-level methodologies that only offer extrinsic evaluation. Our work so far has established the imperative of research into the architectures of the LLMs currently being used as assistants. We reserve the limitations as future work.

References

Dennis Aumiller, Ashish Chouhan, and Michael Gertz. 2022. EUR-lex-sum: A multi- and cross-lingual dataset for long-form summarization in the legal domain. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7626–7639, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Maciej Besta, Nils Blach, Ales Kubicek, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Adrien Bibal, Michael Lognoul, Alexandre De Streel, and Benoît Frénay. 2021. Legal AI and the Right to Explanation: A Review. In *International Conference on Artificial Intelligence and Law (ICAIL)*.

Trenton Bricken, Adly Templeton, Joshua Batson, and 1 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Hua Cai, Shuang Zhao, Liang Zhang, Xuli Shen, Qing Xu, Weilin Shen, Zihao Wen, and Tianke Ban. 2025. Unilaw-R1: A Large Language Model for Legal Reasoning with Reinforcement Learning and Iterative Inference. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18128–18142.

Leyland Cecco. 2024. Canada lawyer under fire for submitting fake cases created by AI chatbot. *The Guardian*.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. Paragraph-level rationale extraction through regularization: A case study on european court of human rights cases. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Mexico City, Mexico. Association for Computational Linguistics.

Imran Chamieh, Torsten Zesch, and Klaus Giebermann. 2024. Llms in short answer scoring: Limitations and promise of zero-shot and few-shot approaches. In *Proceedings of the 19th workshop on innovative use of nlp for building educational applications (bea 2024)*, pages 309–315.

Amit Anand Choudhary. 2025. Fake cases, twisted verdicts as petitioner files 'ai reply'. *The Times of India*.

Yu Ying Chu, Sieh-Chuen Huang, and Hsuan-Lei Shao. 2025. Unpacking legal reasoning in llms: Chain-of-thought as a key to human-machine alignment in essay-based nlu tasks. In *Proceedings of the 5th Workshop on Natural Logic Meets Machine Learning (NALOMA)*, pages 1–7.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, and 1 others. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Junfeng Cui, Hu Shen, and 1 others. 2023. A Survey on Legal Judgment Prediction: Datasets, Metrics, Models and Challenges. *IEEE Access*, 11:12345–12367.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E. Ho. 2024a. Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models. *Journal of Legal Analysis*, 16(1):64–93.

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

Xin Dai, Buqiang Xu, Zhenghao Liu, Yukun Yan, Huiyuan Xie, Xiaoyuan Yi, Shuo Wang, and Ge Yu. 2025. Legal δ: Enhancing legal reasoning in llms via reinforcement learning with chain-of-thought guided information gain. *arXiv preprint arXiv:2508.12281*.

Debtanu Datta, Shubham Soni, Rajdeep Mukherjee, and Saptarshi Ghosh. 2023. MILDSum: A novel benchmark dataset for multilingual summarization of Indian legal case judgments. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5291–5302, Singapore. Association for Computational Linguistics.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and 1 others. 2022. Toy models of superposition. *Transformer Circuits Thread*. Foundational work defining superposition and why features hide in polysemantic neurons.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*.

Brandon L. Garrett and Cynthia Rudin. 2023. The Right to a Glass Box: Rethinking the Use of Artificial Intelligence in Criminal Justice. *Cornell Law Review*.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Jarek Gryz and Marcin Rojszczak. 2021. Black box algorithms and the rights of individuals: no easy solution to the "explainability" problem. *Internet Policy Review*, 10(2):1–24.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023a. Legalbench: a collaboratively built benchmark for measuring legal reasoning in large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Neel Guha, Julian Nyarko, Daniel E Ho, and 1 others. 2023b. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, and 1 others. 2023. Measuring Faithfulness in Chain-of-Thought Reasoning. *arXiv preprint arXiv:2307.13702*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

Nathaniel Love and Michael Genesereth. 2005. Computational law. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, ICAIL '05, page 205–209, New York, NY, USA. Association for Computing Machinery.

Varun Magesh, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho. 2024. AI on Trial: Legal Models Hallucinate in 1 out of 6 (or More) Benchmarking Queries. *Huuman-Centered Artificial Intelligence, Stanford University*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35.

Denis Merigoux, Nicolas Chataing, and Jonathan Protzenko. 2021. Catala: A programming language for the law. volume 5, pages 1–29. ACM New York, NY, USA.

Sara Merken. 2025. AI 'hallucinations' in court papers spell trouble for lawyers. *Reuters*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

John J. Nay, David Law, and 1 others. 2024. Large Language Models as Tax Attorneys: A Case Study in Legal Capabilities Emergence. *Journal of Empirical Legal Studies*. Working Paper.

Shubham Kumar Nigam, Anurag Sharma, Danush Khanna, Noel Shallum, Kripabandhu Ghosh, and Arnab Bhattacharya. 2024. Legal judgment reimagined: Predex and the rise of intelligent ai interpretation in indian courts. *arXiv preprint arXiv:2406.04136*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Shounak Paul, Pawan Goyal, and Saptarshi Ghosh. 2021. Lesicin: A heterogeneous graph-based approach for automatic legal statute identification from indian legal documents. *Preprint*, arXiv:2112.14731.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. 2025. A practical review of mechanistic interpretability for transformer-based language models. *Preprint*, arXiv:2407.02646.

Jiaxin Shen, Jinan Xu, Huiqi Hu, Luyi Lin, Guoyang Ma, Fei Zheng, Fandong Meng, Jie Zhou, and Wenjuan Han. 2025. A law reasoning benchmark for llm with tree-organized structures including factum probandum, evidence and experiences. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17252–17274.

Juanming Shi, Qinglang Guo, Yong Liao, and Shenglin Liang. 2024. Legalgpt: legal chain of thought for the legal large language model multi-agent framework. In *International Conference on Intelligent Computing*, pages 25–37. Springer.

Jisu Shin, Hoyun Song, Huije Lee, Soyeong Jeong, and Jong Park. 2024. Ask LLMs directly, "what shapes your bias?": Measuring social bias in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16122–16143, Bangkok, Thailand. Association for Computational Linguistics.

Ruihao Shui, Yixin Cao, Xiang Wang, and Tat-Seng Chua. 2023. A comprehensive evaluation of large language models on legal judgment prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7337–7348, Singapore. Association for Computational Linguistics.

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. 2022. Legal case document summarization: Extractive and abstractive methods and their evaluation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1048–1064, Online only. Association for Computational Linguistics.

Amrita Singh, Aditya Joshi, Jiaojiao Jiang, and Hyeyoung Paik. 2025. A survey of classification tasks and approaches for legal contracts. *Artificial Intelligence Review*, 58(12):380.

Team LLaMA. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. *arXiv preprint arXiv:2305.04388*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, and 1 others. 2020. Investigating gender bias in bert's attention heads. In *NeurIPS Workshop on Black-box NLP Interpretability*. Foundational work for Causal Mediation Analysis.

Xuran Wang, Xinguang Zhang, Vanessa Hoo, Zhouhang Shao, and Xuguang Zhang. 2024. Legalreasoner: A multi-stage framework for legal judgment prediction via large language models and knowledge integration. *IEEE Access*.

Zach Warren. 2025. GenAI hallucinations are still pervasive in legal filings, but better lawyering is the cure. *Thomson Reuters*.

Bin Wei, Yaoyao Yu, Leilei Gan, and Fei Wu. 2025a. An LLMs-based Neuro-symbolic Legal Judgment Prediction Framework for Civil Cases. *Artificial Intelligence and Law*.

Bin Wei, Yaoyao Yu, Leilei Gan, and Fei Wu. 2025b. An llms-based neuro-symbolic legal judgment prediction framework for civil cases. *Artificial Intelligence and Law*, pages 1–35.

Jason Wei, Xuezhi Wang, Dale Schuurmans, and 1 others. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 24824–24837.

Yu'an Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. 2024. Harnessing the power of large language models for natural language to first-order logic translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6942–6959.

Zhun Yang, Adam Ishay, and Joohyung Lee. 2023. Coupling large language models with logic programming for robust and general reasoning from text. *arXiv preprint arXiv:2307.07696*.

Fangyi Yu, Lee Quartey, and Frank Schilder. 2022. Legal prompting: Teaching a language model to think like a lawyer. *arXiv preprint arXiv:2212.01326*.

Fangyi Yu, Lee Quartey, and Frank Schilder. 2023. Exploring the effectiveness of prompt engineering for legal reasoning tasks. In *Findings of the association for computational linguistics: ACL 2023*, pages 13582–13596.

Wenhan Yu, Xinbo Lin, Lanxin Ni, Jinhua Cheng, and Lei Sha. 2025. Benchmarking multi-step legal reasoning and analyzing chain-of-thought effects in large language models. *arXiv preprint arXiv:2511.07979*.

Weikang Yuan, Junjie Cao, Zhuoren Jiang, Yangyang Kang, Jun Lin, Kaisong Song, Tianqianjin Lin, Pengwei Yan, Changlong Sun, and Xiaozhong Liu. 2024. Can large language models grasp legal theories? enhance legal reasoning with insights from multi-agent collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7577–7597, Miami, Florida, USA. Association for Computational Linguistics.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and 1 others. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*.

Kepu Zhang, Weijie Yu, Zhongxiang Sun, and Jun Xu. 2025. Syler: A framework for explicit syllogistic legal reasoning in large language models. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 4117–4127.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2).

A Experimental Details

Datasets. ECtHR has 9K, 1K, and 1K documents in train, validation and test sets respectively. ILSI has documents 42884, 10203, 13043 documents in train, validation and test sets respectively. Due to limited resources we take docs that are under 1500 words; For ECtHR, we obtain 5879, 567, 532 documents in train, validation and test sets, respectively; for ILSI, we obtain 32424, 7341, 9595 documents in train, validation and test sets, respectively. However, since our work does not demand a huge dataset in particular, we take a subset from each split and work with 8K, 1K, 1K documents as train-val-test.

Models. LLaMA 3.1 8B has 32 layers; Gemma 2 9B has 42 layers in total. The models are adequately large for zero-shot inference and latent representation analysis.

Prompts. 3 is the prompt utilized in inference i.e. generating initial set of candidate violations. 4 is the verification prompt which is used for activation caching.

A.1 Training Regime

Selection of Layers. Linear Hypotheses (Park et al., 2023; Mikolov et al., 2013) argues that there is a linear direction from one concept to another in a model's latent representation space, and therefore two contrasting concepts are linearly separable. In stage two, we have two concepts – applicable and non-applicable, indicated by two labels, True and False. We utilize Logistic Regression as the linear classifier that finds a linear decision boundary between True and False labels. We select the layer that outperforms every other layer i.e. the layer that best separates the two concepts (classes) and inspect it through our setup. Here we utilize the initial raw activations as the utility of this step is limited to only finding a reliable layer to inspect. Following are the best layers: 28 in LLaMA for both ECtHR and ILSI, 41 and 24 respectively in Gemma for ECtHR and ILSI.

A.2 Results

A.3 Structural Analysis of Score Distributions.

ECtHR Gemma

Range: Wide dynamic range, extending from roughly -4.0 to +3.0.

False Distribution (Blue - Structure):

General. It has a smaller, distinct hump in the negative region (around -2.5) representing "Confident Rejections".

The "Shadow" Peak. The primary mass of the False distribution is not at 0.0; instead, it rises aggressively to form a peak almost exactly at x = 1.0.

Observation. The Blue curve is effectively "hiding" underneath the Red curve.

True Distribution (Red - Structure):

General. Sharp, high-kurtosis peak centered tightly at 1.0.

Extreme Overlap. The intersection of Red and Blue is massive. The model assigns the same geometric score to a large portion of False answers as it does to True answers.

Structural Implication. This graph visually depicts "Confident Hallucination"; a linear classifier will struggle here.

ECtHR LLaMA

Range: Compact range, mostly between -1.5 and +2.0.

False Distribution (Blue - Structure):

General. The blue curve is a wide, flattened hill centered roughly between -0.5 and 0.0. It has a long tail that extends far into the positive region (0.0 to 1.0), crossing the decision boundary.

True Distribution (Red - Structure):

General. shifted slightly to the right of the centroid (peak approx 1.3). It has a "left-leaning shoulder" that enters the weak zone.

Overlap. The Blue and Red curves blend together significantly in the 0.0 to 1.0 interval.

Structural Implication. Structure represents ambiguity – middle zone signals confusion.

ILSI Gemma

Range: Extreme dynamic – stretching from -8.0 to

Stage One: Zero-Shot Inference
You receive a set of facts from a legal document in `[Indian Supreme Court/European Court of Human Rights]`. Your task is to determine which `[Indian Penal Code Sections/Articles]` have been violated given the facts. `[facts]` Return only the violations.

Table 3: Stage One prompt.

Stage Two: Verification of Candidate Violations
You receive a set of facts from a legal document in `[Indian Supreme Court/European Court of Human Rights]`, and a proposed violated `[Indian Penal Code Section/Article]`. Determine whether it is True i.e. applicable or False i.e. not applicable. Consider the following set of facts: `[facts]` The label is:

Table 4: Stage Two prompt.

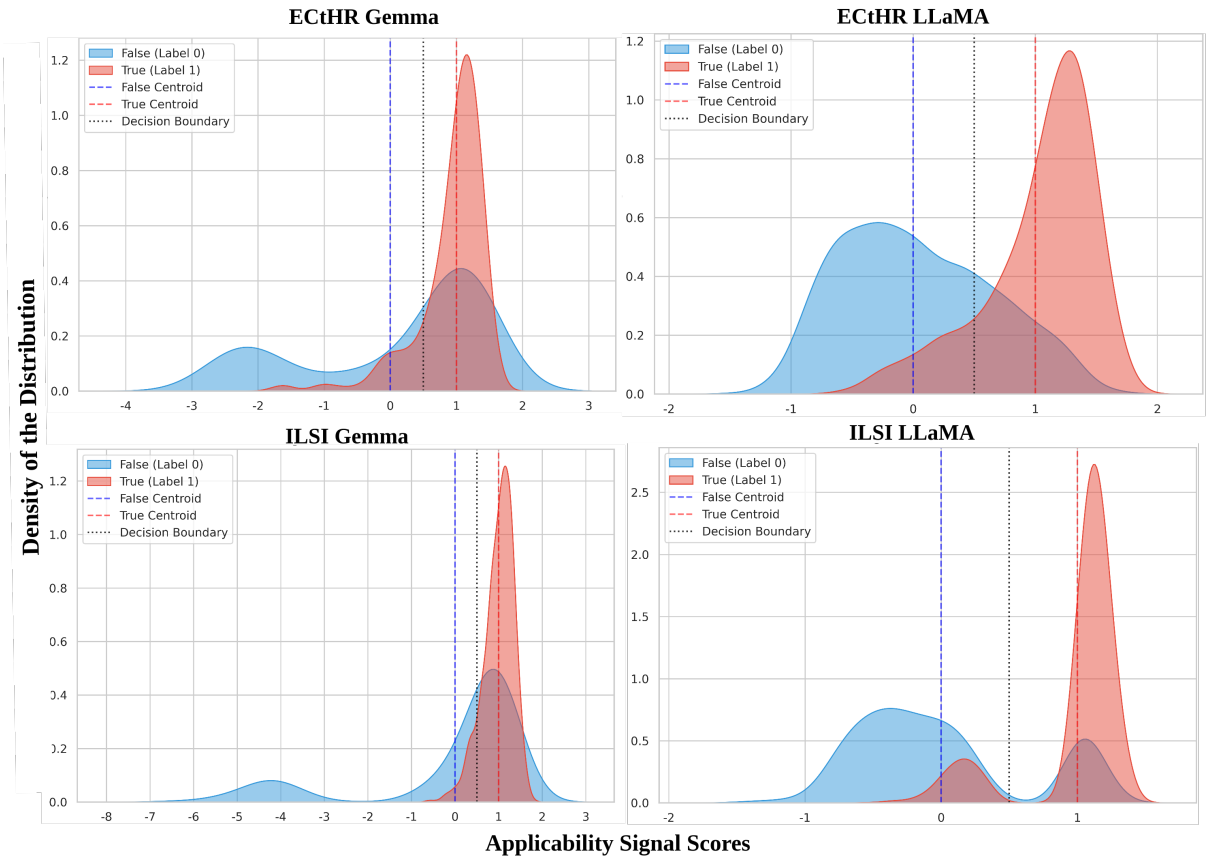


Figure 4: Graph Repeated for Structural Analysis; Distribution of *applicability* signal scores across models and datasets.

+3.0.

False Distribution (Blue - Structure):

General. Multimodal - a peak centred around -4.5; A second peak at 1.0, underneath the Red peak.

True Distribution (Red - Structure):

General. Extremely sharp, narrow peak at 1.0.

Overlap. There is a near-empty region between -2.0 and 0.0.

Structural Implication. Model is either extremely confident the answer is False (score -5) or it gets confused and thinks it is perfectly True (score +1).

ILSI LLaMA

Range: -2.0 to +2.0.

False Distribution (Blue - Structure):

General. One hump centered at -0.5, and another at +0.2

True Distribution (Red - Structure):

General. There is a sharp peak at 1.0.

Overlap. An almost empty region around 0.7.

Structural Implication. Selecting 0.7 can result in a successful classification.

A.4 Selection of Ambiguous Zone

Following the True-False distribution, we turn to entropy to select an ambiguous zone where the model is supposed to be mostly confused. To identify the region, we calculate entropy at every 0.1 interval with a threshold value of 0.8, and select the range from the first occurrence of entropy value higher than the threshold to the last occurence. Following are the regions that are identified as ambiguous,

ECtHR-LLaMA: 0.6 to 1.2;

ECtHR-Gemma: 0.0 to 1.4;

ILSI-LLaMA: 0.1 to 1.0;

ILSI-Gemma: -0.3 to 1.2;

The ranges are close to the weak zone we defined.