CROSSLJP: AN ASYMMETRIC CROSS-VIEW FRAMEWORK FOR LEGAL JUDGMENT PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Legal judgment prediction (LJP) uses case facts to predict applicable provisions or judicial outcomes. Prior work has improved LJP through legal pretraining, statutory knowledge, precedents, and label dependencies, yet prediction often relies on a single task-adapted representation. This representation must capture case context and distinguish applicable provisions, but gives the predictor no separate access to pretrained features unchanged by legal supervision. We introduce CrossLJP, an asymmetric cross-view framework for multi-label legal provision prediction that preserves a frozen semantic view alongside an adapted decision view. Cross-attention uses the semantic view to query the decision view, allowing pretrained case features to guide the weighting of adapted features. Across three benchmarks derived from real-world court cases in Chinese, Korean, and English, CrossLJP achieves the highest mean Exact Match (EM), Micro-F1, and Macro-F1 among the evaluated models. These results support preserving pretrained case features and using them to guide prediction from task-adapted features.

1 INTRODUCTION

Legal Judgment Prediction (LJP) predicts applicable legal provisions, charges, sentences, or other judicial outcomes from case facts (Xiao et al., 2018; Zhong et al., 2018). We focus on multi-label legal provision prediction, where several provisions may apply to the same case. A model must identify the provisions supported by the facts while avoiding missing and unsupported predictions. Effective prediction requires preserving the broader factual account and identifying fine-grained distinctions that determine the applicable provision set. Semantically similar cases may differ in a legally material detail, while cases with different surface content may support the same provision.

Existing approaches address this task using labeled legal cases, often by adapting pretrained encoders to provision prediction (Chalkidis et al., 2020; Xiao et al., 2021). Fine-tuning encourages an encoder to learn features useful for predicting the target provisions. However, this creates a representation-learning tension between preserving pretrained case semantics and specializing for provision prediction. Adapting a single encoder can sharpen distinctions useful for provision prediction, but it also reshapes the representation around provision supervision. Conversely, keeping a pretrained encoder fixed preserves its pretrained mapping but does not itself provide task-specific adaptation. We therefore argue that semantic preservation and decision-oriented specialization are better maintained as separate views with distinct roles in prediction.

Figure 1 illustrates differences in keyword sensitivity between the encoder views. In the methamphetamine case, the frozen view shows higher normalized sensitivity to keywords describing location and delivery details, while the adapted view shows higher normalized sensitivity to the forensic findings identifying the substance and its weight. This observation suggests that the two representations can emphasize complementary aspects of the same case rather than providing redundant features, motivating us to retain both views. The analysis procedure is described in Appendix A.4.

Retaining both views, however, does not specify how they should interact for prediction. Direct concatenation exposes both representations to the classifier but leaves their interaction implicit. This motivates a mechanism that lets one view condition how information from the other is aggregated.

We introduce CrossLJP, an asymmetric cross-view framework that connects complementary representations of the same case before prediction. Rather than relying on either view alone, CrossLJP

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

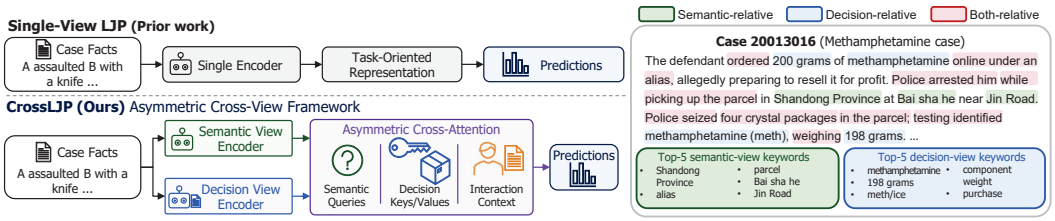


Figure 1: **CrossLJP uses a frozen view to guide the aggregation of adapted features. Left:** A single-encoder classifier predicts from a task-adapted representation. CrossLJP pairs a frozen semantic encoder with an adapted decision encoder. The semantic view provides queries, and the decision view provides keys and values for cross-attention. Prediction uses both encoder representations and the resulting context. **Right:** Perturbation sensitivities for a methamphetamine case, illustrated with English keywords. In this example, location and delivery details rank more highly for the frozen view, while substance and quantity details rank more highly for the adapted view.

retains both representations and constructs an additional interaction context through cross-attention between them. In our main configuration, we pair a frozen Nomic encoder (Nussbaum et al., 2024) with a BGE-M3 encoder (Chen et al., 2024) adapted using LoRA (Hu et al., 2022). Both encoders process the same case facts, and their outputs form the semantic view and the decision view, respectively. These names describe their roles in the model; both views can encode factual context and legally relevant information. The frozen representation contributes directly to prediction while conditioning how features from the task-adapted representation are aggregated. Cross-attention uses the frozen view as queries and the adapted view as keys and values (Vaswani et al., 2017). The resulting interaction context is retained together with both original representations for prediction.

We evaluate CrossLJP on CAIL2018 (Xiao et al., 2018), LBOX (Hwang et al., 2022), and ECtHR-A (Chalkidis et al., 2022), covering Chinese, Korean, and English legal texts. CrossLJP improves both Micro-F1 and exact-match accuracy over BGE-LoRA and direct concatenation of the same frozen and adapted encoder views on all three benchmarks. Compared with Dual-View Concat, CrossLJP improves mean Micro-F1 by up to 2.36 percentage points and mean exact-match accuracy by up to 3.99 percentage points. This controlled comparison evaluates the benefit of cross-view interaction beyond simply providing both representations to the classifier. Ablation studies further examine how integration strategy, encoder configuration, and attention direction affect performance.

Our contributions are as follows:

- We introduce CrossLJP, an asymmetric cross-view framework that retains frozen and task-adapted representations and uses the former to condition the aggregation of adapted features. Both encoder representations remain directly available alongside the learned interaction context.
- We report improvements in Micro-F1 and exact-match accuracy on three legal benchmarks spanning Chinese, Korean, and English. CrossLJP outperforms both BGE-LoRA and a concatenation baseline built from the same two encoder views in the reported runs.
- We examine how the views are used through ablations of integration strategy, encoder configuration, and attention direction. Queries derived from the frozen semantic view with keys and values from the adapted decision view outperform the reverse and bidirectional variants.

2 Related Work

LJP predicts legal provisions, charges, sentences, and other judicial outcomes from case facts (Feng et al., 2022b). Early studies made predictions from case text or structured metadata (Aletras et al., 2016; Katz et al., 2017). Neural network models combined case facts with relevant law articles for charge prediction (Luo et al., 2017) and modeled dependencies among law articles, charges, and sentencing decisions (Zhong et al., 2018). Other studies used reading comprehension, multi-perspective reasoning, and iterative question answering for legal prediction (Long et al., 2019; Yang et al., 2019; Zhong et al., 2020a). Legal-domain pretrained models address legal terminology, while long-document encoders and text compression help process lengthy case facts (Chalkidis et al., 2020; Xiao et al., 2021; Lou et al., 2026). Further work uses legal attributes, distinguishes between similar provisions, or matches facts to provisions (Hu et al., 2018; Xu et al., 2020; Ge et al., 2021). Events and rationales have been used to guide prediction (Feng et al., 2022a; Wu et al., 2022). Other approaches model relations between cases, retrieve precedents, or apply legal rules (Liu et al., 2022;

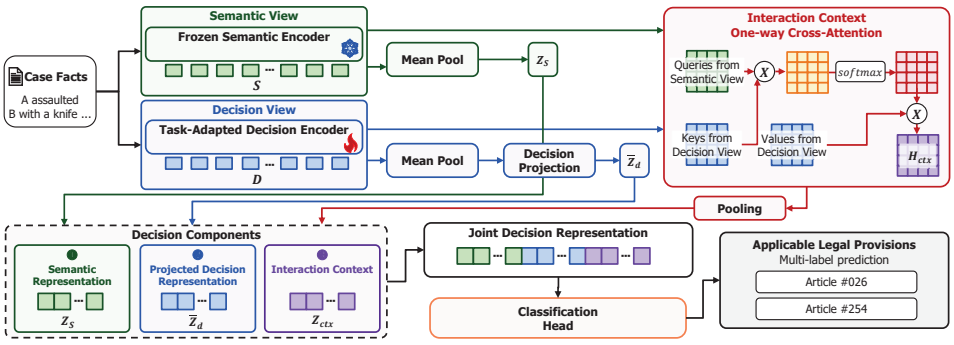


Figure 2: Overview of CrossLJP. A frozen semantic encoder and a task-adapted decision encoder process the same case facts. Learned projections form semantic queries and decision keys and values for cross-attention. The classifier combines the semantic document summary, the projected decision document summary, and the interaction context to predict applicable legal provisions.

Wu et al., 2023; Zhang et al., 2025). Despite these advances, many encoder-based approaches predict from a representation shaped by downstream supervision.

Pretrained encoders can be used as fixed feature extractors or adapted to a downstream task. The relative effectiveness of these approaches depends on the task (Peters et al., 2019). LoRA changes encoder representations through low-rank updates while keeping the original pretrained weights fixed (Hu et al., 2022). Side-Tuning combines the outputs of a frozen pretrained network and a trainable side network through a learned weighted sum (Zhang et al., 2020). Ladder side-tuning passes intermediate activations from a frozen backbone to a smaller trainable side network (Sung et al., 2022). BERT-fused NMT incorporates frozen BERT representations into translation encoder and decoder layers via attention, using translation states to query BERT outputs (Zhu et al., 2020).

Multi-view representation learning studies how to learn representations from multiple views of the same instance (Rastogi et al., 2015; Wang et al., 2015; Tosh et al., 2021). Attention-based models use one sequence to weight and combine representations from another (Parikh et al., 2016; Seo et al., 2017; Tsai et al., 2019). Within LJP, cross-attention has been used to combine case facts with external legal charge knowledge (He et al., 2023). CrossLJP applies cross-attention to the output chunk representations of two encoders processing the same case facts. Queries are computed from the frozen view, while keys and values are computed from the adapted view, allowing frozen features to condition the aggregation of adapted features before document-level pooling. The classifier receives both independently pooled encoder outputs together with the pooled cross-attention context for multi-label legal provision prediction.

3 Method

3.1 Problem Formulation

We formulate legal judgment prediction (LJP) as multi-label classification over a predefined set of legal provisions. Let $\mathcal{Y} = \{1, \ldots, L\}$ denote the candidate set, where each $l \in \mathcal{Y}$ corresponds to one provision. Given the facts of a case X, the goal is to identify the subset $Y \subseteq \mathcal{Y}$ that applies to the case. Because several provisions may apply at once, we represent Y as a multi-hot vector $\mathbf{y} \in \{0,1\}^L$, whose l-th entry is

$$y_l = \begin{cases} 1, & \text{if } l \in Y, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Given a training set $\mathcal{D} = \{(X_i, \mathbf{y}_i)\}_{i=1}^N$, a model f_θ estimates the probability each provision applies,

$$\hat{\mathbf{y}} = f_\theta(X) \in [0,1]^L, \tag{2}$$

where $\hat{\mathbf{y}}_l$ is the estimated probability for provision l. Conventional fine-tuning forms this prediction entirely from a representation shaped by the provision labels. Our formulation keeps the benefits of this adaptation but does not require prediction to rely on the adapted representation alone. Instead, it retains a second representation that remains unchanged by downstream supervision and allows the model to use both when deciding which provisions apply.

3.2 CROSSLJP

Figure 2 summarizes CrossLJP. A frozen semantic encoder and a task-adapted decision encoder produce two representation sequences from the same case facts. The semantic view retains its pretrained mapping, while the decision view is adapted through provision supervision. We connect the sequences through one-way cross-attention, using semantic queries to guide the aggregation of decision features. The resulting interaction context is provided to the classifier together with an independent document summary from each encoder. The classifier thus has direct access to both views as well as features whose weighting depends on their interaction.

Frozen and Adapted Encoders. Let E_s and E_d denote the semantic and decision encoders. In our main configuration, we instantiate E_s with Nomic Embed (Nussbaum et al., 2024) and E_d with BGE-M3 (Chen et al., 2024). Nomic's pretrained parameters ϕ_s remain fixed. For BGE-M3, we keep the pretrained backbone parameters ϕ_d fixed and optimize LoRA parameters ψ in its query and value projections (Hu et al., 2022). These updates allow the decision representation to change under provision supervision. Semantic and decision describe the encoders' learning roles; both receive the same facts and can encode information relevant to the provision labels. We evaluate alternative encoder pairings in Appendix A.5.

Each encoder represents a case as a sequence whose positions correspond to text chunks. Input construction, including tokenization and segmentation, is specified in the implementation details. For encoder $e \in \{s, d\}$, let $\mathbf{u}_{e,t,k} \in \mathbb{R}^{d_e}$ denote the hidden state of token k at valid sequence position t, and let $a_{e,t,k}$ indicate whether that token is valid. We obtain the representation at each position by masked mean pooling:

$$\tilde{\mathbf{z}}_{e,t} = \frac{\sum_k a_{e,t,k}\mathbf{u}_{e,t,k}}{\sum_k a_{e,t,k}}. \tag{3}$$

The semantic vectors are additionally normalized to unit ℓ_2 norm, while the decision vectors retain their pooled values. Stacking the resulting vectors as rows gives

$$Z_s = E_s(X; \phi_s) \in \mathbb{R}^{T_s \times d_s}, Z_d \quad = E_d(X; \phi_d, \psi) \in \mathbb{R}^{T_d \times d_d}, \tag{4}$$

where $d_s = 768$ and $d_d = 1024$. The encoder notation includes the aggregation and normalization specified above. T_s and T_d denote the sequence lengths, including padding when cases are batched. The sequences may differ in length, and their interaction does not require a predefined correspondence between positions.

Each view also provides a document summary independently of the other. Let $m_s \in \{0,1\}^{T_s}$ and $m_d \in \{0,1\}^{T_d}$ indicate valid sequence positions. We define masked mean pooling and the two summaries as

$$\mathrm{Pool}(Z; m) = \frac{\sum_t m_t Z_{t,:}^\top}{\sum_t m_t}, \mathbf{z}_s \quad = \mathrm{Pool}(Z_s; m_s), \qquad \mathbf{z}_d = \mathrm{Pool}(Z_d; m_d). \tag{5}$$

These summaries give the classifier a direct representation of the case from each encoder. The original sequences Z_s and Z_d remain available to the cross-attention module, which computes the interaction context separately.

We project the decision summary to width d_s before classification:

$$\bar{\mathbf{z}}_d = \mathrm{LN}_{\mathrm{align}}(W_{\mathrm{align}}\mathbf{z}_d + \mathbf{b}_{\mathrm{align}}) \in \mathbb{R}^{d_s}, \tag{6}$$

where $W_{\mathrm{align}} \in \mathbb{R}^{d_s \times d_d}$ and $\mathbf{b}_{\mathrm{align}} \in \mathbb{R}^{d_s}$ are trainable, and LN denotes layer normalization. This projection matches the feature dimensions and is optimized through the classification loss, without an auxiliary alignment objective. It acts on the document summary $\mathbf{z}_d$; the interaction module operates on the original sequences through its own query, key, and value projections.

Cross-Attention. Cross-attention allows the semantic view to condition how adapted information is assembled for prediction (Vaswani et al., 2017). The semantic sequence supplies queries, and the decision sequence supplies keys and values. Each semantic query can therefore obtain a different weighted combination of decision features. This assignment gives the views distinct roles within the interaction: semantic features contribute to the attention weights, while the returned features come from the adapted representation.

We project the two sequences into an attention space of width d_b:

$$Q_s = \mathrm{LN}_Q(Z_s W_Q + \mathbf{b}_Q), K_d \quad = \mathrm{LN}_K(Z_d W_K + \mathbf{b}_K), V_d = Z_d W_V + \mathbf{b}_V. \tag{7}$$

Here, $W_Q \in \mathbb{R}^{d_s \times d_b}$ and $W_K, W_V \in \mathbb{R}^{d_d \times d_b}$; biases are broadcast across sequence positions. Query and key projections use separate layer normalizations over the feature dimension at each position. All projection matrices, biases, and normalization parameters are trainable. These parameters are separate from the LoRA updates within BGE-M3 and the decision summary projection in Equation 6. The query and key projections learn the comparison used for attention, while the value projection determines the adapted features returned by that comparison.

We compute attention over valid decision positions:

$$A = \mathrm{softmax}_{\mathrm{row}}\left(\frac{Q_s K_d^\top}{\sqrt{d_b}} + M_d\right) \in \mathbb{R}^{T_s \times T_d}, \tag{8}$$

where the decision mask is

$$(M_d)_{ij} = \begin{cases} 0, & (m_d)_j = 1, \\ -\infty, & (m_d)_j = 0. \end{cases} \tag{9}$$

For semantic query i, A_{ij} specifies the weight assigned to decision position j. The weights sum to one across valid decision positions, and padded positions receive zero weight. We use these weights to construct the interaction context:

$$H_{\mathrm{ctx}} = AV_d \in \mathbb{R}^{T_s \times d_b}, \mathbf{z}_{\mathrm{ctx}} \quad = \mathrm{Pool}(H_{\mathrm{ctx}}; m_s) \in \mathbb{R}^{d_b}. \tag{10}$$

Each row of H_{ctx} contains the decision features aggregated for one semantic query. The decision mask excludes padded keys from attention, while the semantic mask excludes padded query outputs from the final context.

The dependence of the context on both views can be expressed through its effective weights on the decision sequence. Let $n_s = \sum_i (m_s)_i$ and define

$$\bar{a}_j = \frac{1}{n_s}\sum_{i=1}^{T_s}(m_s)_i A_{ij}, \mathbf{z}_{\mathrm{ctx}} \quad = \sum_{j=1}^{T_d} \bar{a}_j (V_d)_{j,:}^\top. \tag{11}$$

Equation 11 rewrites the context computation in terms of one average weight per decision chunk. The coefficients $\bar{a}_j$ are nonnegative, sum to one, and vanish at padded decision positions. Thus, the final context is a weighted summary of projected decision representations. Its content comes from V_d, while the weights depend jointly on the semantic queries and decision keys.

Equation 11 also explains the distinction from independent mean pooling. Writing $n_d = \sum_j (m_d)_j$, a uniform average of the projected decision values is

$$\frac{1}{n_d}\sum_j (m_d)_j (V_d)_{j,:}^\top = W_V^\top \mathbf{z}_d + \mathbf{b}_V. \tag{12}$$

Cross-attention replaces uniform averaging with weights determined jointly by the semantic queries and decision keys.

Training and Prediction. CrossLJP preserves the semantic encoder's pretrained mapping while learning how the task accesses it. In particular, the trainable query projection and its normalization allow provision supervision to shape how preserved semantic features condition decision-side aggregation. The semantic sequence remains fixed for a given input, while its projected queries can change during training. The model can therefore adapt its use of the semantic view through the interaction module.

The final representation contains the semantic summary, the projected decision summary, and the interaction context. We normalize each component separately and concatenate them:

$$\mathbf{h} = \mathrm{LN}_s(\mathbf{z}_s) \,\|\, \mathrm{LN}_d(\bar{\mathbf{z}}_d) \,\|\, \mathrm{LN}_{\mathrm{ctx}}(\mathbf{z}_{\mathrm{ctx}}) \in \mathbb{R}^{2d_s + d_b}. \tag{13}$$

Here, $\|$ denotes concatenation, and the three layer normalizations have separate trainable parameters. The two summaries describe the case independently, while the context provides an additional representation conditioned on their underlying sequences. Concatenation keeps all three components available to the classifier, which learns how they contribute to each provision prediction.

A classifier $g_{\text{cls}} : \mathbb{R}^{2d_s+d_b} \to \mathbb{R}^L$ maps the combined representation to provision logits:

$$\hat{\mathbf{y}} = \sigma(g_{\text{cls}}(\mathbf{h})), \tag{14}$$

where σ is the element-wise sigmoid function. Each output estimates the applicability of one provision, allowing multiple provisions to receive high probabilities for the same case. Attention widths and classifier configurations for each dataset are specified in Appendix A.2.1.

We optimize weighted binary cross-entropy:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{NL}\sum_{i=1}^{N}\sum_{\ell=1}^{L}\Big[w_\ell y_{i\ell}\log\hat{y}_{i\ell} + (1-y_{i\ell})\log(1-\hat{y}_{i\ell})\Big], \tag{15}$$

where $w_\ell > 0$ is computed from label frequencies in the training split and weights the positive term for provision ℓ. The loss supervises the final provision probabilities and jointly optimizes all trainable components using only the provision labels associated with each case.

Training updates the LoRA parameters, the decision summary projection, the cross-attention projections, the normalization layers, and the classifier, while the pretrained parameters ϕ_s and ϕ_d remain fixed. The adapted encoder receives supervision through its document summary and through the attention keys and values. These paths allow the loss to update both the decision features and their contributions to the interaction context. On the semantic side, gradients update the query projection and its normalization without changing the pretrained encoder.

At inference, both views and their interaction context are computed from case facts alone. We predict provision ℓ when $\hat{y}_\ell > \tau_\ell$, using a threshold for each provision. Thresholds are selected using validation predictions and fixed for test evaluation. Optimization settings, class-weight calculation, model selection, and the threshold selection procedure are provided in the implementation details.

4 Experiments

4.1 Experimental Setup

Dataset and evaluation. We evaluate CrossLJP on three benchmarks that predict legal provisions from case facts, each in a different language: CAIL2018, LBOX, and ECtHR-A. CAIL2018 (Xiao et al., 2018) contains Chinese criminal cases and requires predicting the criminal law articles. We use the statute classification task from LBOX (Hwang et al., 2022), which maps Korean case facts to the statutory provisions applied by the court. ECtHR-A (Chalkidis et al., 2022) uses factual paragraphs from cases before the European Court of Human Rights to predict the articles of the European Convention on Human Rights found to have been violated. Each case may have more than one label, and each dataset defines its label set. We use Micro-F1 as our primary measure and report Exact Match (EM) and Macro-F1 in the main table. Micro-F1 aggregates true positives, false positives, and false negatives across all cases and labels. EM requires the predicted label set to match the reference set exactly. Macro-F1 averages per-label F1 over all target labels. For models evaluated under our controlled protocol, all metrics are reported as percentages and summarized as mean $\pm$ standard deviation over five random seeds. Published values are reproduced as reported in the cited studies. Bold scores indicate the highest mean among models evaluated under our controlled protocol. Where reported, Δ denotes a variant's mean score minus that of CrossLJP for the corresponding metric, in percentage points. Checkpoint selection and per-label threshold selection for the controlled models are described in Appendix A.2. Selected thresholds are held fixed during test evaluation.

Baselines.

Table 1 includes results reported by prior work under their original evaluation settings. We also conduct controlled encoder comparisons using a common training framework within each dataset. Frozen BGE-M3 trains a classifier while keeping the encoder unchanged. BGE-M3 Full FT updates all encoder parameters using task supervision, while BGE-LoRA adapts BGE-M3 with low-rank updates. BGE-LoRA serves as our main single-encoder comparison because it uses the same decision encoder and adaptation approach as CrossLJP. Dual-View Concat combines frozen Nomic and LoRA-adapted BGE-M3 by concatenating their document representations before classification. This comparison evaluates the contribution of cross-attention beyond providing both encoder representations to the classifier. Additional variants that change the encoders or their interaction are introduced

Table 1: Classification performance on CAIL2018, LBOX, and ECtHR-A. Published results are reproduced from the cited studies under their original evaluation settings and are provided for context. Controlled baselines and CrossLJP are evaluated using our common experimental protocol. "NR" indicates an unreported metric, and "–" indicates that the dataset was not evaluated.

	CAIL2018			LBOX			ECtHR-A		
Method	**EM**	**Micro-F1**	**Macro-F1**	**EM**	**Micro-F1**	**Macro-F1**	**EM**	**Micro-F1**	**Macro-F1**
Representative Prior Work									
LEMM (Zhong et al., 2020b)	NR	81.56	64.69	–	–	–	–	–	–
ZipLJP (LLaMA3.2-3B) (Lou et al., 2026)	73.5	NR	71.5	–	–	–	–	–	–
PLJP (BERT) (Wu et al., 2023)	NR	NR	56.63	–	–	–	–	–	–
LegalGraphRAG (Chen et al., 2026)	47.9	NR	NR	–	–	–	–	–	–
KoGPT2-base (Hwang et al., 2022)	–	–	–	85.7 ± 0.8	NR	NR	–	–	–
LCUBE-base-zero (Hwang et al., 2022)	–	–	–	85.8 ± 0.8	NR	NR	–	–	–
mT5-small (Hwang et al., 2022)	–	–	–	87.2 ± 0.3	NR	NR	–	–	–
PGR (Extractive) (Rosamma et al., 2026)	–	–	–	–	–	–	50.20	66.16	45.49
Legal-BERT (Hierarchical) (Chalkidis et al., 2022)	–	–	–	–	–	–	NR	70.0	64.0
Longformer (Chalkidis et al., 2022)	–	–	–	–	–	–	NR	69.9	64.7
Single-encoder baselines									
Frozen BGE-M3	58.75 ± 0.26	73.81 ± 0.22	61.13 ± 0.13	67.03 ± 2.57	85.04 ± 0.96	35.12 ± 0.84	50.36 ± 1.78	70.93 ± 0.70	60.19 ± 1.13
BGE-M3 Full FT	70.97 ± 1.79	81.29 ± 0.66	70.72 ± 0.74	76.38 ± 1.57	88.09 ± 1.34	38.95 ± 1.56	56.02 ± 2.48	73.99 ± 1.02	65.56 ± 1.39
BGE-LoRA	63.77 ± 1.72	76.85 ± 1.31	63.41 ± 1.75	85.94 ± 3.25	93.71 ± 1.58	42.01 ± 0.52	56.66 ± 0.59	74.91 ± 0.57	68.25 ± 1.64
Dual-encoder models									
Dual-View Concat	74.06 ± 1.87	83.47 ± 0.92	74.39 ± 0.84	86.68 ± 2.40	94.09 ± 0.78	42.18 ± 0.33	55.32 ± 0.97	74.28 ± 0.44	66.40 ± 2.26
CrossLJP	**78.05 ± 1.05**	**85.83 ± 0.64**	**74.85 ± 0.23**	**87.68 ± 1.74**	**94.48 ± 0.92**	**42.99 ± 0.93**	**57.08 ± 0.87**	**74.93 ± 0.65**	**68.75 ± 1.84**

in the ablation study. Within each dataset, all controlled models use the same preprocessing, data splits, metric definitions, and per-label threshold-selection procedure unless otherwise specified.

4.2 Classification Performance

Table 1 presents published reference results and controlled encoder comparisons across CAIL2018, LBOX, and ECtHR-A. The published values were obtained under the original settings of the corresponding studies and therefore provide broader context rather than controlled head-to-head comparisons. Within our controlled experiments, CrossLJP achieves the highest mean EM, Micro-F1, and Macro-F1 on all three datasets, with Micro-F1 scores of 85.83, 94.48, and 74.93, respectively. We next examine what these controlled comparisons reveal about retaining a frozen representation and integrating it with an adapted representation.

We first examine whether adding the frozen representation provides additional predictive value. Dual-View Concat pairs frozen Nomic with LoRA-adapted BGE-M3 and trains a classifier on their concatenated case representations. On CAIL2018 and LBOX, it improves over the strongest single-encoder baseline across all three metrics. The comparisons are against BGE-M3 Full FT on CAIL2018 and BGE-LoRA on LBOX, with Micro-F1 gains of 2.18 and 0.38 percentage points and corresponding EM gains of 3.09 and 0.74 points. These improvements suggest that features from the frozen encoder remain useful alongside a representation already adapted through legal supervision. The adapted representation learns distinctions relevant to provision prediction, while the frozen representation preserves another encoding of the same case facts. Their combination allows the classifier to draw on features available through either view. The results on these two datasets support keeping pretrained features explicitly accessible during downstream prediction, where they can complement the features learned through task adaptation.

We then examine whether cross-attention improves the combination of the two representations. Both Dual-View Concat and CrossLJP use frozen Nomic and LoRA-adapted BGE-M3, and both train a joint classifier with access to the two case representations. CrossLJP additionally provides an interaction context constructed through cross-attention. Compared with concatenation, it improves Micro-F1 by 2.36 points on CAIL2018, 0.39 points on LBOX, and 0.65 points on ECtHR-A, with higher mean EM and Macro-F1 on all three datasets as well. Since concatenation already provides both encoder views, these gains support the proposed way of incorporating their interaction into prediction. Cross-attention allows the frozen representation to guide how task-adapted evidence is combined. The resulting context captures a use of the two views that depends on their relationship, alongside their separate contributions to classification. The observed improvements are consistent with this intended role: the preserved representation serves both as a source of predictive features and as a basis for consulting the adapted representation. This comparison therefore supports learning how the views interact when forming the final case representation.

Table 2: Ablation study of CrossLJP on CAIL2018, varying the semantic encoder and integration method. All variants use LoRA-adapted BGE-M3 as the decision encoder.

Variant	Semantic encoder	Integration	EM	Micro-F1	ΔMicro-F1
Logit Ensemble	Nomic (frozen)	Logit fusion	70.75 ± 1.91	81.57 ± 0.94	−4.26
Dual-View Concat	Nomic (frozen)	Concatenation	74.06 ± 1.87	83.47 ± 0.92	−2.36
Same Backbone	BGE-M3 (frozen)	Cross-attention	74.62 ± 1.40	84.30 ± 0.78	−1.53
Both Adapted	Nomic (LoRA)	Cross-attention	77.04 ± 1.88	85.48 ± 0.77	−0.35
CrossLJP	Nomic (frozen)	Cross-attention	**78.05 ± 1.05**	**85.83 ± 0.64**	0.00

The EM and Macro-F1 results clarify the significance of these improvements for provision prediction. Relative to concatenation, CrossLJP increases EM by 3.99 points on CAIL2018, 1.00 point on LBOX, and 1.76 points on ECtHR-A. An exact match requires recovering the entire reference provision set without omitting a provision or adding an incorrect one. These gains therefore show that improved aggregate label prediction is accompanied by more cases with completely correct provision sets. Macro-F1 increases by 0.46, 0.81, and 2.35 points, respectively. Because each target label contributes equally to Macro-F1, these gains show that the advantage remains visible in average per-label performance. CAIL2018 exhibits the largest EM and Micro-F1 gains over concatenation, while ECtHR-A exhibits the largest Macro-F1 gain. The benefits thus appear through different aspects of prediction across the datasets. Across three benchmarks, CrossLJP achieves higher mean scores than both Dual-View Concat and the strongest single-encoder baseline on every metric. These comparisons support retaining complementary representations and incorporating their interaction to improve both individual provision prediction and recovery of the complete provision set.

4.3 Effect of View Integration

Table 2 compares three integration methods using frozen Nomic and LoRA-adapted BGE-M3. Logit Ensemble combines per-label prediction scores produced by separate classification heads for the two encoder views. Dual-View Concat supplies both encoder representations to a joint classifier. CrossLJP additionally provides an interaction context computed through cross-attention. These comparisons examine the value of combining the views at the prediction level, using their features jointly, and incorporating their interaction into prediction.

Logit Ensemble obtains 70.75 EM and 81.57 Micro-F1, exceeding BGE-LoRA in Table 1 by 6.98 and 4.72 points, respectively. Dual-View Concat increases EM to 74.06 and Micro-F1 to 83.47, improving over Logit Ensemble by 3.31 and 1.90 points. The gain from concatenation supports making both representations available before the provision predictions are formed. In Logit Ensemble, the frozen view contributes through the outputs of its own classification head. With concatenation, the classifier learns the contribution of the frozen features together with the adapted features. It can therefore use the two representations in combination when predicting each provision. The higher mean scores favor this joint use of features over the evaluated fusion of separate prediction scores.

CrossLJP further increases EM to 78.05 and Micro-F1 to 85.83, gaining 3.99 and 2.36 points over Dual-View Concat. This comparison examines the additional value of an interaction context when both encoder representations are already available to the classifier. In CrossLJP, the context is formed from adapted BGE-M3 features using attention weights computed from both views. The frozen representation consequently contributes both directly to classification and to the way adapted evidence is combined. The gains support learning this interaction alongside the joint use of the two representations. Across the three integration methods, the highest means are obtained when the classifier receives both views together with a context formed through their interaction.

4.4 Encoder Choice and Adaptation

Table 2 also examines the choice and adaptation of the semantic encoder while retaining cross-attention integration. Same Backbone and Both Adapted use LoRA-adapted BGE-M3 as the decision encoder, with semantic queries and decision keys and values.

Same Backbone replaces frozen Nomic with a frozen copy of BGE-M3, so the two views originate from the same pretrained model and differ in whether they receive task adaptation. It obtains 74.62 EM and 84.30 Micro-F1, exceeding BGE-LoRA in Table 1 by 10.85 and 7.45 points, respectively. This result supports the value of maintaining frozen and adapted views even when they share the same pretrained backbone. The two roles can provide a useful basis for prediction without requiring different encoder families. Using Nomic for the frozen view further increases EM by 3.43 points and Micro-F1 by 1.53 points in CrossLJP. These results support retaining separate frozen and adapted views and indicate that the choice of frozen encoder can affect performance.

Both Adapted retains the Nomic/BGE-M3 pair and applies LoRA to Nomic in addition to BGE-M3. It obtains 77.04 EM and 85.48 Micro-F1. Keeping Nomic frozen in CrossLJP yields higher means by 1.01 EM points and 0.35 Micro-F1 points, while using fewer trainable parameters. This comparison distinguishes preserving the semantic mapping from learning how its features are used. In CrossLJP, the Nomic encoder retains its pretrained mapping, while the query projection, cross-attention module, and classifier learn from provision supervision alongside the decision encoder's LoRA parameters. The task can therefore learn how to consult the preserved features and combine them with adapted evidence. The higher mean scores support this separation between a fixed source of semantic features and a trainable mechanism for using them.

4.5 Effect of Attention Direction

Table 3 examines how assigning the semantic and decision views to queries, keys, and values affects prediction. All three variants use frozen Nomic and LoRA-adapted BGE-M3, and their classifiers receive both encoder representations together with the attention context. In $S \rightarrow D$, the semantic view supplies queries and the decision view supplies keys and values. In $D \rightarrow S$, these assignments are reversed. Bidirectional attention combines contexts computed in both directions. Each variant supplies a 256-dimensional context, keeping the classifier input width fixed.

Table 3: Attention direction on CAIL2018. S and D denote the semantic and decision views. Arrows point from queries to keys and values.

Direction	EM	Micro-F1
$D \rightarrow S$	73.00 ± 1.46	82.88 ± 0.95
Bidirectional	76.19 ± 1.72	84.76 ± 0.67
$S \rightarrow D$	**78.05 ± 1.05**	**85.83 ± 0.64**

The $S \rightarrow D$ direction used by CrossLJP achieves 78.05 EM and 85.83 Micro-F1. The reverse direction obtains 73.00 EM and 82.88 Micro-F1, giving CrossLJP gains of 5.05 and 2.95 points, respectively. Since both variants retain the two encoder representations as classifier inputs, this comparison concerns the construction of the additional context. With semantic queries and decision keys and values, the preserved representation conditions how task-adapted features are combined. The context is therefore formed from features learned for provision prediction, with their contributions determined through interaction with the pretrained view. The higher mean scores support assigning these distinct roles to the two representations.

Bidirectional attention obtains 76.19 EM and 84.76 Micro-F1. CrossLJP improves on these means by 1.86 and 1.07 points, respectively. The advantage over both the reverse and bidirectional variants supports the asymmetric construction of the context for this encoder pair on CAIL2018. The semantic view provides the conditioning signal, while the decision view supplies the task-adapted features being combined. Because both encoder representations also remain directly available to the classifier, this assignment gives the context a specific role: using the preserved view to guide the selection and combination of adapted evidence.

5 Conclusion

We introduced CrossLJP, an asymmetric cross-view framework that preserves a frozen semantic view and uses cross-attention to connect it with a task-adapted decision view for multi-label legal provision prediction. Across three legal benchmarks in Chinese, Korean, and English, CrossLJP achieves higher mean EM, Micro-F1, and Macro-F1 than the evaluated single-encoder baselines and Dual-View Concat. Ablations on CAIL2018 favor cross-attention over logit fusion and concatenation, and support using queries derived from the frozen semantic view to guide the aggregation of adapted decision features.

Reproducibility Statement

We provide detailed descriptions of the model architecture and training procedure in the main paper, with additional implementation details, dataset preprocessing, hyperparameters, and experimental settings in the appendix. We also report the random seeds and model selection procedures used in our experiments to facilitate reproduction of the reported results.

AI Usage Disclosure

Generative AI tools were used to assist with literature discovery, research ideation and experimental feedback, drafting portions of the manuscript, and improving the clarity and readability of the writing. All AI-assisted outputs were reviewed and verified by the authors. The authors take full responsibility for the methodology, experimental results, analyses, claims, and content of this work.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ computer science*, 2, 2016.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proc. the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.137. URL https://aclanthology.org/2024.findings-acl.137/.

Zerui Chen, Qinggang Zhang, Zhishang Xiang, Zhimin Wei, Linfeng Gao, Xiao Huang, Zhihong Zhang, and Jinsong Su. LegalGraphRAG: Multi-agent graph retrieval-augmented generation for reliable legal reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37455–37484, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1738. URL https://aclanthology.org/2026.acl-long.1738/.

Yi Feng, Chuanyi Li, and Vincent Ng. Legal judgment prediction via event extraction with constraints. In *Proceedings of the 60th annual meeting of the association for computational linguistics (ACL)*, 2022a.

Yi Feng, Chuanyi Li, and Vincent Ng. Legal Judgment Prediction: A Survey of the State of the Art. In *Proc. International Joint Conferences on Artificial Intelligence (IJCAI)*, 2022b.

Jidong Ge, Yunyun Huang, Xiaoyu Shen, Chuanyi Li, and Wei Hu. Learning fine-grained fact-article correspondence in legal cases. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 2021.

Congqing He, Tien-Ping Tan, Xiaobo Zhang, and Sheng Xue. Knowledge-enriched multi-cross attention network for legal judgment prediction. *IEEE Access*, 11:87571–87582, 2023. doi: 10.1109/ACCESS.2023.3305259.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *Proc. International Conference on Learning Representations (ICLR)*, 2022.

Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. Few-shot charge prediction with discriminative legal attributes. In *Proc. the 27th international conference on computational linguistics*, 2018.

Wonseok Hwang, Dongjun Lee, Kyoungyeon Cho, Hanuhl Lee, and Minjoon Seo. A Multi-Task Benchmark for Korean Legal Language Understanding and Judgement Prediction. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022.

Daniel Martin Katz, Michael J Bommarito, and Josh Blackman. A general approach for predicting the behavior of the Supreme Court of the United States. *PloS one*, 12, 2017.

Dugang Liu, Weihao Du, Lei Li, Weike Pan, and Zhong Ming. Augmenting legal judgment prediction with contrastive case relations. In *Proc. the 29th international conference on computational linguistics (COLING)*, 2022.

Shangbang Long, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. Automatic judgment prediction via legal reading comprehension. In *China National Conference on Chinese Computational Linguistics*, 2019.

Fanghao Lou, Qiqi Wang, Guanyu Chen, Kaiqi Zhao, and Huijia Li. ZipLJP: Zipped Information Processor for Legal Judgment Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40, 2026.

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. Learning to predict charges for criminal cases with legal basis. In *Proc. the 2017 conference on empirical methods in natural language processing (EMNLP)*, 2017.

Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic Embed: Training a Reproducible Long Context Text Embedder. *arXiv preprint arXiv:2402.01613*, 2024.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proc. the 2016 conference on empirical methods in natural language processing (EMNLP)*, 2016.

Matthew E Peters, Sebastian Ruder, and Noah A Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proc. the 4th Workshop on Representation Learning for NLP (RepL4NLP)*, 2019.

Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. Multiview LSA: Representation learning via generalized CCA. In *Proc. the 2015 conference of the North American chapter of the Association for Computational Linguistics: human language technologies (NAACL)*, 2015.

KS Rosamma, Nagamma Patil, and CD Jaidhar. Efficient and fair legal judgment classification via personalized graphrank summarization and class-balanced learning. *IEEE Access*, 2026.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. In *Proc. International Conference on Learning Representations (ICLR)*, 2017.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 12991–13005. Curran Associates, Inc., 2022. doi: 10.52202/068431-0944. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/54801e196796134a2b0ae5e8adef502f-Paper-Conference.pdf.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic learning theory*, 2021.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proc. the 57th annual meeting of the association for computational linguistics (ACL)*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems (NeurIPS)*, 30, 2017.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual E5 text embeddings: A technical report, 2024. URL https://arxiv.org/abs/2402.05672.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *Proc. International conference on machine learning (ICML)*, 2015.

Yiquan Wu, Yifei Liu, Weiming Lu, Yating Zhang, Jun Feng, Changlong Sun, Fei Wu, and Kun Kuang. Towards interactivity and interpretability: A rationale-based legal judgment prediction framework. In *Proc. the 2022 conference on empirical methods in natural language processing (EMNLP)*, 2022.

Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. Precedent-Enhanced Legal Judgment Prediction with LLM and Domain-Model Collaboration. In *Proc. the 2023 conference on empirical methods in natural language processing (EMNLP)*, 2023.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. CAIL2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*, 2018.

Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. Lawformer: A Pre-trained Language Model for Chinese Legal Long Documents. *AI Open*, 2, 2021.

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. Distinguish confusing law articles for legal judgment prediction. In *Proc. the 58th annual meeting of the association for computational linguistics (ACL)*, 2020.

Wenmian Yang, Weijia Jia, Xiaojie Zhou, and Yutao Luo. Legal judgment prediction via multi-perspective bi-feedback network. *arXiv preprint arXiv:1905.03969*, 2019.

Jeffrey O Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-tuning: a baseline for network adaptation via additive side networks. In *European conference on computer vision*, pp. 698–714. Springer, 2020.

Yue Zhang, Zhiliang Tian, Shicheng Zhou, Haiyang Wang, Wenqing Hou, Yuying Liu, Xuechen Zhao, Minlie Huang, Ye Wang, and Bin Zhou. RLJP: Legal Judgment Prediction via First-Order Logic Rule-enhanced with Large Language Models. *arXiv preprint arXiv:2505.21281*, 2025.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. Legal judgment prediction via topological learning. In *Proc. the 2018 conference on empirical methods in natural language processing (EMNLP)*, 2018.

Haoxi Zhong, Yuzhong Wang, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. Iteratively questioning and answering for interpretable legal judgment prediction. In *Proc. the AAAI conference on artificial intelligence (AAAI)*, 2020a.

Huilin Zhong, Junsheng Zhou, Weiguang Qu, Yunfei Long, and Yanhui Gu. An element-aware multi-representation model for law article prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6663–6668, 2020b.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Incorporating bert into neural machine translation, 2020. URL https://arxiv.org/abs/2002.06823.

A Appendix

A.1 Datasets and Splits

We evaluate multi-label legal provision prediction on CAIL2018, LBOX, and ECtHR-A. Each input consists of case facts, and the target is the associated set of provisions. CAIL2018 contains Chinese criminal cases, LBOX contains Korean cases, and ECtHR-A contains English factual paragraphs from cases before the European Court of Human Rights. Table 4 summarizes the dataset configurations.

Table 4: Dataset sizes and target label counts. CAIL2018 counts follow removal of validation and test cases whose facts also occur in training.

Dataset	Language	Train	Validation	Test	Labels
CAIL2018	Chinese	154,592	16,657	31,690	183
LBOX	Korean	2,208	276	276	169
ECtHR-A	English	9,000	1,000	1,000	10

For CAIL2018, we use the existing `train.jsonl`, `valid.jsonl`, and `test.jsonl` files, with `facts` as the input field and `statutes` as the target field. The training split defines a vocabulary of 183 article labels. Encoded targets in all splits contain only labels in this vocabulary. The raw splits contain 154,592 training, 17,131 validation, and 32,508 test cases. Removing validation and test cases whose fact strings exactly match training facts removes 474 and 818 cases, respectively, yielding 16,657 validation and 31,690 test cases. The corresponding exact-match check finds no training-fact overlap in the LBOX or ECtHR-A validation and test splits, so their split sizes remain unchanged.

A.2 Implementation details

We use `nomic-ai/nomic-embed-text-v1.5` as the frozen semantic encoder and `BAAI/bge-m3` as the decision encoder. Each view uses the tokenizer associated with its encoder checkpoint. The semantic tokenizer has a maximum input length of 512 tokens, while the decision tokenizer uses the configured chunk length. For the controlled encoder experiments on CAIL2018, the chunk length is 256 tokens with a 32-token overlap, retaining at most 16 chunks per document.

Threshold selection. For the controlled encoder models, we select a separate threshold for each label using validation probabilities from the selected checkpoint. Candidates comprise all unique validation probabilities for that label, together with its minimum probability minus 10^{-7} and its maximum probability plus 10^{-7}. We choose a threshold that maximizes validation F1, retaining 0.5 for labels with no positive validation examples. Predictions contain labels whose probabilities are strictly greater than their thresholds, including an empty set when none exceed their thresholds. The selected thresholds remain fixed during test evaluation.

A.2.1 Training Configuration

Table 5 summarizes the training configuration for the controlled encoder experiments on CAIL2018.

For the CAIL2018 weighted binary cross-entropy objective, let N be the number of training cases and N_l the number with label l. The positive-class weight is

$$w_l = \begin{cases} 1, & N_l = 0, \\ \min\left(50, \dfrac{N - N_l}{N_l + 10^{-5}}\right), & N_l > 0. \end{cases} \tag{16}$$

All class weights are computed from the training split. Per-device batch sizes and gradient accumulation steps are adjusted to memory requirements while maintaining an effective batch size of 32.

Table 5: Training configuration for the controlled encoder experiments on CAIL2018.

Setting	Value
LoRA target projections	BGE-M3 query and value projections
LoRA rank / scaling factor	32 / 64
LoRA dropout	0.05
CrossLJP attention width	256
CrossLJP classifier input dimension	1,792
CrossLJP classifier head	LayerNorm, dropout, linear output layer
Training objective	Weighted binary cross-entropy
Optimizer	AdamW
Learning rate	5×10^{-5}
Weight decay	0.01
Warmup	First 10% of optimization steps; linear increase
Subsequent schedule	Cosine decay
Maximum gradient norm	1.0
Precision	BF16
Effective batch size	32
Maximum epochs	100
Validation frequency	After every epoch
Early stopping patience	15 epochs without improvement
Checkpoint criterion	Validation Micro-F1 at threshold 0.5
Seed set	$\{42, 123, 456, 789, 2027\}$
Hardware per run	One NVIDIA RTX PRO 5000 Blackwell, 48 GB

LBOX and ECtHR-A. The main-table controlled experiments on LBOX and ECtHR-A use the same training configuration. We use AdamW with a learning rate of 5×10^{-5}, weight decay of 0.01, and weighted binary cross-entropy. The batch size is 4. We train for up to 90 epochs and evaluate every 300 training steps. We select the checkpoint with the highest validation Micro-F1 and stop training after 18 consecutive validation evaluations without improvement. When LoRA is used, its rank is 64 and dropout is 0.05. Documents are processed using 384-token chunks with a 64-token overlap, retaining at most 12 chunks. CrossLJP uses semantic-to-decision cross-attention, an attention width of 128, and a classifier hidden dimension of 128. We use random seeds $\{42, 123, 456, 789, 2027\}$.

A.3 Performance Metrics

For a dataset with n test cases and L target labels, let Y_i and $\widehat{Y}_i$ denote the reference and predicted label sets for case i. Exact Match (EM) is

$$\mathrm{EM} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\left[\widehat{Y}_i = Y_i\right]. \tag{17}$$

Micro-F1 aggregates the overlap between reference and predicted label sets across cases:

$$F_{1,\mathrm{micro}} = \frac{2 \sum_{i=1}^{n} \left|Y_i \cap \widehat{Y}_i\right|}{\sum_{i=1}^{n} |Y_i| + \sum_{i=1}^{n} |\widehat{Y}_i|}. \tag{18}$$

Macro-F1 averages per-label F1 over all L target labels. Let TP_l, FP_l, and FN_l denote the true-positive, false-positive, and false-negative counts for label l:

$$F_{1,\mathrm{macro}} = \frac{1}{L} \sum_{l=1}^{L} \frac{2\mathrm{TP}_l}{2\mathrm{TP}_l + \mathrm{FP}_l + \mathrm{FN}_l}. \tag{19}$$

A label's F1 is set to zero when its denominator is zero. ECtHR-A training and evaluation use the ten article labels, with no-violation cases retained as empty target sets. Empty reference and predicted sets constitute an exact match. We report all metrics as percentages. For the controlled experiments, prediction sets follow the threshold-selection procedure described in Appendix A.2.

A.4 PERTURBATION ANALYSIS

We use deduplicated CAIL2018 test case 20013016, selected from an automatically generated candidate list in which CrossLJP was correct while the decision-only prediction was incorrect. The model input is the original Chinese fact description; English keyword descriptions are used only for display. Candidate keywords are word or short-span units constructed using Jieba 0.42.1 with `HMM=False`. We preserve predefined expressions and apply rule-based merging to dates, quantities with units, function words, and remaining single-character tokens. Punctuation tokens are excluded from the candidate set. For each candidate, we delete its corresponding span from the original text without inserting mask tokens or replacement strings, and recompute the representation for each encoder view.

For each view $e \in \{s, d\}$, let $z^{(e)}$ denote the original document representation and $z^{(e)}_{-i}$ the representation after deleting keyword i. We compute and normalize representation sensitivity as

$$a_i^{(e)} = 1 - \cos\left(z^{(e)}, z^{(e)}_{-i}\right), \qquad \tilde{a}_i^{(e)} = \frac{a_i^{(e)}}{\sum_{j=1}^{m} a_j^{(e)} + 10^{-12}},$$

where m is the number of candidate keywords. For each view, we display the five keywords with the highest normalized sensitivity. To compare the views, we compute $g_i = \tilde{a}_i^{(d)} - \tilde{a}_i^{(s)}$. Highlights use orange for $g_i \geq 0$ and blue for $g_i < 0$, with color intensity determined by $|g_i|$. The top-five lists are ranked by each view's normalized sensitivity, rather than by g_i.

Table 6: Encoder substitution on CAIL2018. Scores are percentages; parentheses show changes from the original CrossLJP configuration in percentage points. Semantic encoders are frozen; decision encoders use LoRA.

Semantic	**Decision**	**Integration**	**EM**	**Micro-F1**	**Macro-F1**
–	E5	Single encoder	61.46 (−14.25)	76.49 (−8.34)	64.58 (−8.88)
Nomic	E5	Concat	65.18 (−10.53)	78.82 (−6.01)	67.73 (−5.73)
Nomic	E5	CrossLJP	67.74 (−7.97)	80.20 (−4.63)	70.22 (−3.24)
E5	BGE-M3	CrossLJP	**76.96** (+1.25)	**85.49** (+0.66)	**75.09** (+1.63)
Nomic	BGE-M3	CrossLJP (Ours)	75.71	84.83	73.46

A.5 ENCODER SUBSTITUTION

We examine alternative encoder pairings for CrossLJP on CAIL2018, using seed 42 for all configurations (Table 6). For the decision-encoder substitution, we replace BGE-M3 with multilingual E5-base (`intfloat/multilingual-e5-base`) (Wang et al., 2024), while retaining frozen Nomic. Nomic and E5-base inputs use the prefixes `search_document:` and `passage:`, respectively, each followed by a space. We apply LoRA to E5-base with rank 32 and dropout 0.05. The decision representation width changes from 1,024 to 768, while semantic-to-decision cross-attention, the 256-dimensional interaction context, and the three-component classifier input are retained.

With E5-base as the adapted decision encoder, CrossLJP improves over matched concatenation by 2.56 EM, 1.38 Micro-F1, and 2.49 Macro-F1 percentage points, supporting the additional interaction context with a different decision encoder. Replacing frozen Nomic with E5 also yields higher scores than the original Nomic/BGE-M3 configuration in this comparison. These results support applying the proposed asymmetric framework beyond the original encoder pairing.