UCVE: Graph Structured Retrieval for Compliance Validation

Anonymous ACL submission

Abstract

Structured compliance validation requires determining whether every obligation in a predefined requirement space is satisfied, violated, or unaddressed by a given document. This can be described as a coverage problem that existing retrieval and classification systems, by treating requirements as independent, are not designed to solve. We introduce **Universal Content Validation Engine (UCVE)**, a domain-agnostic framework that represents compliance requirement spaces as a Contextual Requirement Graph (CRG) and performs retrieval via spectral reranking over its reinforcement learning-weighted Laplacian. Edges between requirement nodes are learned via reinforcement learning with a traversal reward grounded in violation detection, reshaping the graph's community structure such that low-pass spectral filtering distinguishes relevant from irrelevant obligation clusters. Evaluating across three compliance domains, UCVE averages 4.3% matched-K recall lift over flat cosine retrieval and consistently outperforms strong retrieval and graph-based baselines.

1 Introduction

Many document validation settings require verifying compliance with a structured set of obligations. Regulatory filings must include mandated disclosures, consumer contracts must avoid unfair clauses, and Environmental, Social, and Governance (ESG) reports must address specified sustainability criteria. This task is increasingly critical as the volume of global regulations outpaces manual oversight, necessitating automated systems that can handle the nuance of legal language (Chalkidis et al., 2022). We study this problem through the *Universal Content Validation Engine* (UCVE), a framework that utilizes a Context Requirement Graph (CRG) for structured retrieval over a predefined obligation space. In legal, financial, and privacy domains, missing a true violation can incur substantial financial or reputational costs. Therefore, achieving high recall over the obligation space is the central requirement for practical validation systems.

Existing compliance systems typically treat requirements as independent units during inference. Classification models predict labels in isolation, and standard retrieval methods extract evidence for individual requirements without propagating information across related obligations. However, local retrieval fails to capture the semantic and hierarchical interdependencies inherent in regulatory frameworks (Edge et al., 2024). For example, a clause governing data retention frequently contains signals relevant to storage limitations or third party sharing. Because traditional strategies do not systematically exploit these dependencies, they often fail to identify violations that lack direct lexical overlap with the requirement text. Furthermore, many systems lack portability because they are tightly coupled to specific schemas, making them difficult to adapt to new regulatory settings without exhaustive reannotation (Zhong et al., 2020).

To address these limitations, UCVE models validation as inference over a CRG where nodes represent obligations and edges encode semantic and hierarchical relationships. Following initial retrieval, the system propagates relevance signals across the graph using spectral reranking to smooth similarity scores across community structures. Edge weights are optimized via a reinforcement learning objective that rewards the recovery of violations during traversal, allowing the system to identify obligations supported by indirect or distributed evidence. Because the graph is built from explicit requirement definitions rather than task specific outputs, the architecture is natively portable across heterogeneous compliance domains.

We evaluate the framework on three benchmarks covering privacy policies, consumer contract fairness, and sustainability disclosures. Focusing on

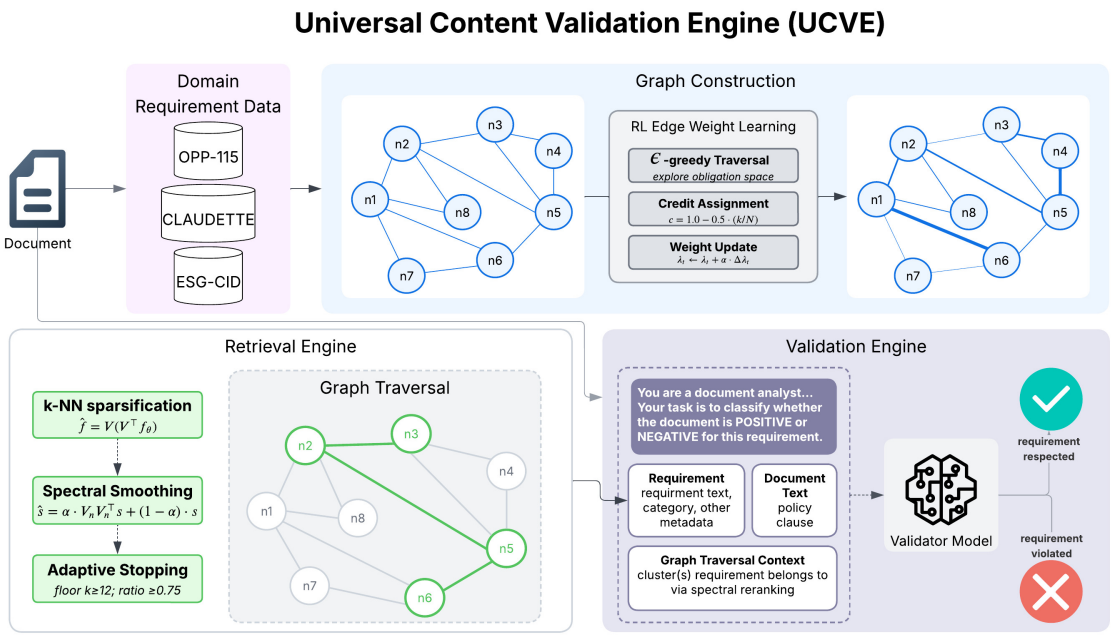

Figure 1: Universal content validation engine architecture.

document level recall, the approach consistently outperforms strong retrieval and graph-based baselines, particularly for obligations requiring cross clause reasoning. This work makes several primary contributions to automated compliance validation. We formulate the task as structured inference over an interconnected graph to prioritize high recall recovery under asymmetric error costs. We introduce the CRG to represent semantic, evidential, and hierarchical relationships between requirements, with edges learned via a reinforcement learning objective that rewards violation discovery during traversal. We propose a graph based validation framework that utilizes a reinforcement learning objective and spectral reranking to optimize violation discovery through graph propagation. Finally, this architecture effectively generalizes across structurally distinct compliance domains without task-specific retraining.

2 Related Work

2.1 Graph-based RAG and Knowledge Graph Approaches

Recent Graph-RAG advancements couple LLMs with Knowledge Graphs (KGs) for complex reasoning via agentic beam search (Sun et al., 2024) or explicit relation planning (Luo et al., 2024). Document-level methods organize contexts using hierarchical communities (Edge et al., 2024) or Maximum Spanning Trees (Zhu et al., 2025) to overcome isolated semantic retrieval. To operationalize traversal, approaches use Personalized PageRank (Bernal Gutiérrez et al., 2024) or map unstructured text to deterministic policy graphs (Chung et al., 2025). However, these rely heavily on static semantic connectivity, pre-existing schemas, or heuristic traversals. UCVE diverges by optimizing structural metadata edges via Reinforcement Learning (RL), enabling the policy to explicitly learn domain-specific co-relevance patterns for high-recall obligation recovery.

2.2 Reinforcement Learned Edges

RL is widely used in multi-hop KG reasoning to learn traversal policies, often framed as sequential decision-making (Xiong et al., 2017; Das et al., 2018). To address incomplete KGs and spurious trajectories, subsequent walk-based models employ knowledge-based reward shaping and action dropout (Lin et al., 2018). At the structural level, methods like Iterative Deep Graph Learning continuously optimize both the adjacency matrix and GNN parameters (Chen et al., 2020). Rather than learning navigational pathways or regenerating entire graph topologies from continuous embeddings, UCVE uses RL strictly to weight discrete, domain-specific structural metadata (e.g., hierarchical links). By rewarding the recovery of missed obligations during epsilon-greedy traversal, UCVE

dynamically discovers the structural relationships that genuinely encode co-relevance.

2.3 Spectral Graph Theory

Spectral graph theory formally propagates signals across networks, capturing global node importance via the dominant eigenvectors of transition matrices (Page et al., 1999). Graph Signal Processing extends this by framing retrieval as linear filtering over a spectral domain; projecting queries onto a low-rank Laplacian eigenbasis allows low-pass filters to smooth relevance signals and accelerate manifold traversal (Iscen et al., 2018). UCVE leverages these foundations to refine candidate retrieval. Instead of flat similarities or sequential random walks, UCVE projects traversal scores into the subspace of the bottom eigenvectors of an RL-weighted, sparsified graph Laplacian. This low-pass filtering smooths signals across learned requirement communities, suppressing high-frequency noise and lifting lexically dissimilar but structurally relevant ground-truth obligations to maximize recall.

3 Methods

UCVE is a two-stage framework for requirement validation over arbitrary document collections with a given domain. The high level pipeline architecture is shown in Figure 1.

3.1 Context Requirement Graph

We introduce a CRG, a weighted, undirected graph $G = (V, E, w)$ over a domain's requirement space. The CRG is created separately for each domain, as these graphs are anticipated to be used independently. The graph is created independent of any target document and encodes domain specific metadata explicitly in the graph structure. Each node $v \in V$ represents a single requirement drawn from a domain specific requirement set. Each edge $e \in E$ encodes a relationship between two requirements (semantic similarity of requirement descriptions or structural metadata) with a composite weight $w(e)$ aggregating across multiple typed edge contributions. The graph is constructed once per dataset, independent of any target documents.

3.1.1 Node Definition

Each node $v \in V$ stores a unique identifier, domain indicator, requirement category label and a natural language `requirement` text. To ensure that validation decisions are interpretable and auditable, each node also contains a natural-language validation question derived from its stored requirement and description. Nodes also include positive and negative class definitions that are instantiated programmatically: a positive label indicates that the document text contains language exhibiting, satisfying, or relating to the requirement, while a negative label indicates the absence of such language. This framing converts the per-node classification task into a grounded binary judgment that requires no dataset-specific prompt engineering — the same template applies uniformly across all domains and requirement types.

3.1.2 Embedding

All node requirement texts are encoded using the `all-mpnet-base-v2` model (Reimers and Gurevych, 2019), which produces 768-dimensional L2-normalized embeddings. We deliberately use a domain-general encoder rather than a domain adapted model to ensure that the CRG has the ability to capture relationships that stem from requirement semantics and graph structure rather than a pattern specific to the wording of a given domain. Node embeddings contribute to edge construction and query-node similarity scoring during traversal.

3.1.3 Semantic Edge Construction

The graph is initialized by connecting nodes whose requirement texts are semantically similar, establishing a backbone that reflects the latent structure of the requirement space. For every pair of nodes (v_i, v_j), we compute cosine similarity over their L2-normalized embeddings.

$$sim(v_i, v_j) = e_i \cdot e_j \tag{1}$$

where e_i is the unit-length embedding of node v_i. An undirected edge is created if $sim(v_i, v_j) \geq \tau$, with threshold $\tau = 0.5$, and edge weight equal to cosine similarity value. This threshold admits semantically related requirements while excluding illegitimate connections between unrelated obligation categories.

3.1.4 Structural Edge Enrichment

Semantic similarity alone cannot encode the relational structure between requirements that domain experts encode in the datasets themselves. For instance, hierarchical requirement structures cannot be captured purely through cosine similarity. To capture this structure, we augment the graph with six additional edge types derived from dataset

ingrained metadata, producing a multi-relational graph with seven edge types in total.

Each edge holds an array of typed values, where each value is a tuple `(type, raw weight, learned weight)`. The `raw weight` encodes the intrinsic strength of the relationship: for a cosine similarity typed value, this would be a value $\in [0.5, 1.0]$. For the rest of the metadata types, it is a binary indicator (1.0 if both nodes share that property). The `learned weight` is a scalar multiplier that get computed per type through RL (Section 3.2), initialized to 1.0 for cosine similarity and 0.0 for all metadata types. The composite edge weight aggregated across all active types:

$$w(e) = \sum_{t \in types(e)} raw_weight_t \times learned_weight_t \tag{2}$$

Metadata edges can only increase the total weight, encoding the intuition that nodes related on multiple dimensions are more likely to co-occur as relevant requirements for the same document. The structural edge types instantiated for a given domain are determined entirely by the metadata fields present in that domain's dataset, requirement no changes to the framework's core architecture.

3.2 RL Edge Weight Learning

The structural edge types introduced in Section 3.1.4 augment the graph with domain-specific relational signals whose relative importance varies across domains. Rather than hand-tuning edge type weights per domain, we learn a weight vector $\lambda = [\lambda_1, \ldots, \lambda_5]$ over the six learnable metadata edge types via reinforcement learning, with the cosine similarity weight fixed at 1.0 throughout. The objective is to maximize retrieval recall over the training split — the fraction of ground truth requirement nodes recovered within a fixed traversal budget.

Training uses an epsilon-greedy BFS traversal over the domain graph, with credit assigned to traversed edges based on whether their destination nodes are ground truth violations. Edges leading to violations discovered early in traversal receive higher credit, reflecting that earlier recovery is more efficient. Weights are updated via gradient ascent and clamped to $(-1.0, 5.0)$ to prevent runaway growth. Full training details, credit assignment, and weight update equations are provided in Appendix B and Algorithm 1.

3.3 Requirement Retrieval Engine

At inference time, the CRG is applied to a given document through a three-step procedure. First, entry point selection identifies the most structurally promising nodes from which to initialize traversal. Next, traversal explores the graph from the entry points to retrieve a set of requirement nodes. Lastly, spectral re-ranking refines the candidate set using the community structure encoded in the RL-weighted graph. Together, these steps produce a ranked set of requirement nodes representing the obligations most relevant to the document.

3.3.1 Traversal

Given a query document q, UCVE scores all same-domain requirement nodes in the CRG using spectral reranking over the RL-weighted graph. Rather than sequentially traversing the graph from entry points, spectral reranking operates globally: it smooths cosine similarity scores across the graph's community structure, lifting nodes whose wording diverges from the query but whose spectral neighborhood is relevant, and suppressing nodes with high surface similarity that sit in irrelevant clusters. Alternative traversal strategies including entry-point selection via Graph-Structure-Aware Entry Point Selection (GSAEPS), adaptive Breadth First Search (BFS), and simultaneous graph scoring are evaluated in Appendix D.

3.3.2 Spectral Reranking

Sparsification. The RL-weighted graph is fully dense by construction: every node pair carries a non-zero composite weight. Standard graph propagation methods such as heat diffusion, personalized PageRank, embedding propagation, degenerate toward flat cosine similarity ranking, as every node is a neighbor of nearly every other, leaving no discriminative community signal. We address this through k-NN sparsification, retaining only the k strongest edges per node.

Spectral Decomposition. Over the sparsified graph, we compute the unnormalized graph Laplacian $L = D - W$, where D is the degree matrix and W is the adjacency matrix of RL-weighted edge weights. Eigenvectors corresponding to small eigenvalues vary slowly across the graph and capture large-scale community structure, while those corresponding to large eigenvalues capture fine-grained within-cluster variation (von Luxburg, 2007; Chung, 1997). We retain the bottom n eigenvectors $V_n \in R^{N \times n}$ as a smoothness basis for

reranking, where n and the sparsification parameter k are selected per domain on validation data to reflect differences in the natural community structure of each requirement graph. Setting n too large risks overfitting to fine-grained topology; setting it too small discards useful inter-community signal.

Projection and Reranking. Given the vector of cosine similarities $s \in R^N$ over all same-domain nodes, we project into the subspace spanned by V_n:

$$\hat{s} = \alpha \cdot V_n V_n^\top s + (1 - \alpha) \cdot s, \quad \alpha = 0.8 \quad (3)$$

The projection $V_n V_n^\top s$ is a low-pass filter in the graph frequency domain, retaining community-level similarity signal and attenuating high-frequency within-cluster noise (von Luxburg, 2007). This has two complementary effects: (1) it lifts requirement nodes with low cosine similarity when their spectral neighbors score highly, recovering obligations whose wording does not closely match the query but whose community is relevant; (2) it suppresses false positives with high cosine similarity that sit in spectral clusters where most nodes score low. The blend weight α=0.8 prioritizes spectral community structure while retaining direct semantic signal; setting α=1.0 discards raw similarity entirely and was found to slightly degrade performance on queries where a single node is distinctly relevant. Nodes are reranked by $\hat{s}$ and returned up to an adaptive budget controlled by a similarity threshold of 0.75.

RL Edge Weights. The utility of spectral projection depends on whether the sparse graph's community structure encodes ground-truth co-relevance patterns. On a cosine similarity graph, communities reflect embedding similarity clusters, which may not align with how requirements co-occur in documents. On the RL-weighted graph, edges trained to predict co-relevance reshape the Laplacian's community structure such that spectral clusters correspond more closely to sets of requirements that genuinely co-occur — the key mechanism underlying the conditional dependence between RL edge learning and spectral reranking demonstrated empirically in Section 4.

4 Results

4.1 Experimental Design

4.1.1 Datasets

We instantiate the CRG over three domains in the evaluation: privacy policy compliance, using OPP-115 (Wilson et al., 2016) and consumer fairness detection using CLAUDETTE (Wilson et al., 2018), and sustainability reports using ESRS (Ahmed et al., 2025). The OPP-115 dataset contains a two-tiered hierarchy of requirements with 10 category level requirements representing top-level data practice categories (i.e. First Party Collection/Use, Data Retention) and 14 sub-attribute requirements presenting more granular aspects of the top level categories (i.e. personal information type cookies is under the top level First Party Collection/Use node). The CLAUDETTE dataset contains 53 requirements. Each CLAUDETTE node also contains dataset specific metadata: a `fairness tier` which marks if the requirement is clearly or potentially unfair and a `kb tag` field which groups semantically related requirements. Nodes are filtered to those appearing in at least 5 training documents to ensure sufficient annotation coverage. ESRS contains 82 sustainability disclosure requirement nodes drawn from the European Sustainability Reporting Standards subset of the ESRS-CID retrieval benchmark, covering disclosure requirements across environmental, social, and governance topics. This resulting base graphs contains 577 edges connecting CLAUDETTE nodes and 148 connecting OPP-115 nodes and 595 connecting ESRS-CID (ESRS) nodes.

All three graphs are shown to have high density: the OPP-115 graph has an edge density of 53.6%, CLAUDETTE graph has an edge density of 41.9% and ESRS graph has an edge density of 100%. This is a general consequence of constructing graphs from the requirement spaces that describe similar topics in the same domains.

4.1.2 Metrics

We report recall as our primary metric: the fraction of ground truth requirement nodes recovered by the system. This is motivated by the compliance setting where missed violations carry higher cost than false positives. We additionally report Precision@K, the fraction of retrieved nodes that are ground truth violations, capturing retrieval quality within the visit budget.

The input granularity per dataset varies, reflecting the natural unit of each corpus. For OPP-115, the input is a full privacy policy document. For CLAUDETTE, the input is a single Terms of Service clause and for ESRS, the input is a single page from a sustainability report. In all cases, the system receives the input text and must recover which requirement nodes are violated.

Method	OPP-115			CLAUDETTE			ESRS		
	Recall	Precision	K	Recall	Precision	K	Recall	Precision	K
BM25	0.649	0.163	14.0	0.732	0.104	16.0	0.4902	0.0640	16.9
Flat cosine	0.748	0.187	14.0	0.806	0.112	16.0	0.7227	0.0967	16.9
PPR	0.583	**0.201**	14.0	0.731	0.106	16.0	0.5919	0.0825	16.9
Leiden	0.766	0.169	14.0	0.841	**0.119**	16.0	0.6745	0.0908	16.9
Heat kernel	0.747	0.187	14.0	0.817	0.114	16.0	0.7442	0.1000	16.9
CRG + Spectral	**0.792**	0.185	14.2	**0.860**	0.118	16.4	**0.7551**	**0.1009**	16.9

Table 1: Retrieval results at matched-K (OPP-115 K=14, CLAUDETTE K=16, ESRS K=17). PPR = Personalized PageRank; Leiden = GraphRAG community detection.

All graph-based methods are evaluated at their natural stopping criterion. To ensure fair comparison, baselines retrieve exactly K nodes where K equals the average number of nodes retrieved by the full UCVE system on that dataset (K=14 for OPP-115, K=16 for CLAUDETTE, K=17 for ESRS).

We adopt document level recall rather than clause level classification metrics reported by prior work on CLAUDETTE. We compare against five retrieval baselines at matched-K. BM25 (Robertson and Zaragoza, 2009) ranks requirement nodes by lexical overlap with the query document. Flat cosine ranks nodes by cosine similarity between the query and node embeddings. Personalized Page Rank (PPR) runs random walks on the RL-weighted graph seeded from the highest cosine similarity node, following HippoRAG (Bernal Gutiérrez et al., 2024). Heat Kernel Diffusion propagates query-weighted scores over the RL-weighted graph using the unnormalized Laplacian with diffusion time t=0.5. Community Retrieval adapts the community-detection component of GraphRAG (Edge et al., 2024) to the node retrieval setting via Louvain clustering, returning all nodes in the most query-relevant community.

4.2 Retrieval Performance

CRG + Spectral achieves the highest recall across all three domains, outperforming the strongest baseline (Leiden) by $+2.6\%$ on OPP-115 and $+1.9\%$ on CLAUDETTE, and achieving $+4.5\%$ over flat cosine on ESRS (Table 1). Against flat cosine, spectral reranking yields $+4.4\%$ on OPP-115, $+5.4\%$ on CLAUDETTE, and $+3.2\%$ on ESRS demonstrating that graph structure encodes retrieval-relevant information that embedding similarity alone cannot capture.

ESRS, the largest and most structurally distinct domain (82 nodes, 1818 queries), demonstrates that CRG + Spectral generalizes beyond the two primary evaluation domains without architectural changes. Notably, Community Retrieval (Leiden), one of the strongest non-spectral baselines on OPP-115 and CLAUDETTE, fell behind withCRG + Spectral on ESRS (0.675 vs 0.755, Table 1). This suggests that discrete community assignments become less reliable as the requirement space grows. Spectral reranking is robust to this degredation because it operates at node level rather than assuming nodes are fixed to clusters, smoothing score continuously across the graph Laplacian eigenbasis.

4.3 Graph Structure

The recall gains over flat cosine reflect better ranking rather than visiting more nodes. Table 5 isolates the contribution of RL edge weight learning and spectral reranking as independent components, revealing a conditional dependency between the two.

The final learned weights (Table 2) reveal a structural asymmetry across domains. OPP-115 assigns meaningful weight to hierarchical metadata, `same_category` (0.549) and `same_sub_attribute` (0.556), reflecting that sub-attribute nodes nested under the same parent category commonly co-occur as relevant in the same documents. In CLAUDETTE, metadata weights are near-zero across all types (0.037–0.041), reflecting that at high edge density the metadata structure adds minimal information beyond cosine similarity. In ESRS, category is weighted highest metadata weight (0.500), suggesting that requirements within the same ESG pillar frequently co-occur as relevant, while `same_framework` contributes minimally (0.086).

Spectral reranking on the RL-weighted graph is the only condition that improves recall on a

Edge Type	OPP-115	CLA	ESRS
cosine_sim	1.000	1.000	1.000
same_category	0.549	N/A	0.500
same_sub_attribute	0.556	N/A	N/A
same_fairness_tier	N/A	0.037	N/A
same_kb_tag	N/A	0.041	N/A
same_parent_cat	N/A	0.040	N/A
same_framework	N/A	N/A	0.086

Table 2: Learned edge weights per dataset.

matched-K basis, achieving $+3.4\%$ lift on OPP-115, $+4.1\%$ on CLAUDETTE and 3.2% pm ESRS. Neither component alone is sufficient: spectral reranking on the cosine-only graph yields modest matched-K lift on OPP-115 and only $+3.5\%$ on CLAUDETTE, while BFS on the RL-weighted graph improves over flat cosine (0.710 OPP-115, 0.806 CLAUDETTE) but underperforms spectral reranking by -8.2% and -5.4% respectively at matched-K.

This interaction is most evident on OPP-115, where zeroing RL metadata weights changes 8 edges in the k-NN (k=3) sparsified graph — sufficient to disrupt the spectral community structure that enables the low-pass filter to distinguish relevant from irrelevant requirement clusters. On CLAUDETTE, metadata weights are individually small but collectively meaningful: removing all metadata produces 54 topology differences and a consistent -1.2% recall loss. On ESRS, removing all metadata collectively costs 2.6% recall (Table 4). Together, these results confirm that RL edge weight learning and spectral reranking are load-bearing components whose value is conditional on each other: RL edges create the community structure that spectral projection exploits, and spectral projection is the mechanism by which that structure translates into retrieval gains.

4.4 Robustness

CRG produces consistent gains across structurally distinct domains using hyperparameters selected per domain on validation data. OPP-115 and CLAUDETTE share a common configuration (k=3, n=5, α=0.8), reflecting their smaller requirement spaces (24 and 53 nodes respectively) and well-separated community structure (Appendix C). ESRS, with 82 nodes and a denser graph, requires a wider neighborhood and richer spectral basis (k=5, n=12, α=0.5) (Figure 2). The lower blend weight on ESRS reflects a greater reliance on raw cosine similarity, consistent with the observation that spectral smoothing risks over-aggregating signal in denser graphs. All three domains are well-served by the same framework with only different local hyperparameter adjustments needed.

Sensitivity analysis over k-NN sparsification ($k \in [1, 10]$), eigenvector count ($n \in [2, 15]$), and spectral blend weight ($\alpha \in [0, 1]$) confirms that the chosen configurations are stable (Figure 2). For OPP-115 and CLAUDETTE, k=3 sparsification is optimal with sharp degradation at k=1 and gradual degradation beyond k=3. The blend weight α=0.8 falls within a stable plateau spanning 0.7 to 0.9 on both datasets. For ESRS, the wider k=5 neighborhood is optimal, consistent with the larger and denser node set requiring more structural context per node.

4.5 Full Pipeline Validation

To evaluate the full UCVE pipeline, we pass retrieved requirement nodes to an LLM validator that produces binary compliance verdicts. Non-retrieved nodes are treated as negative, coupling retrieval and validation quality into a single end-to-end recall metric. We evaluate `Llama-3.3-70B` (Dubey et al., 2024) under two validation strategies: a baseline prompt providing only the requirement text and document and a persona-calibrated prompt enriched with traversal metadata and spectral cluster membership derived from the CRG. Results for Llama-3.1-8B are reported in Table 12.

Table 3 reports end-to-end precision, recall, and gap from retrieval ceiling. The difference between retrieval recall and recall of the full pipeline varies across domains: about 25% on OPP-115, 4-5% on CLAUDETTE and 19% on ESRS. This pattern reflects a structural difference in validation difficulty across domains. CLAUDETTE's short, self-contained clauses are easily classified once retrieved because the task involves a shorter, more straightforward passage. OPP-115 and ESRS involve longer documents where evidence may be distributed throughout, making this a more difficult validation task. On ESRS, precision collapses to 0.204, attributable to class imbalance ($\sim$3% positive rate), that causes the validator to default towards positive prediction. The persona strategy partially mitigates this ($+2.4\%$ recall on ESRS), but degrades slightly on CLAUDETTE (-0.4%) suggesting structural context helps most when evidence is distributed. The gap from retrieval ceiling identifies LLM judgement on retrieved candidates as the primary remaining bottleneck, specifically

Dataset	Strategy	Prec	Rec	ΔRec↑[†]
OPP-115	Retrieval ceiling	—	.792	—
	Baseline	.871	.535	−.257
	Persona	**.876**	**.543**	−.249
CLA.	Retrieval ceiling	—	.860	—
	Baseline	.730	**.817**	−.043
	Persona	**.732**	.803	−.057
ESRS	Retrieval ceiling	—	.722	—
	Baseline	**.204**	.527	−.195
	Persona	.198	**.551**	**−.171**

[†] ΔRec↑ is the gap between strategy recall and the retrieval ceiling. Negative values indicate recall below the ceiling; closer to zero is better. The gap reflects *validation errors* on retrieved nodes rather than *retrieval failure*.

Table 3: End-to-end validation performance with graph-derived context from the CRG using Llama-3.3-70B.

under class imbalance and long document evidence synthesis.

5 Conclusion

Compliance validation requires retrieving relevant evidence and ensuring that every obligation in a structured requirement space has been addressed. This presents a coverage problem that existing retrieval and classification systems, by treating requirements as independent, are not designed to solve. We introduced the Universal Validation Engine (UCVE), which reframes this problem as structured retrieval over a Contextual Requirement Graph (CRG) whose edges encode RL-learned co-relevance among obligations. The central finding is that RL edge weight learning and spectral reranking are conditionally dependent: RL edges reshape the graph Laplacian's community structure such that low-pass spectral filtering can distinguish relevant from irrelevant requirement clusters, and neither component alone achieves the same gains. This mechanism generalizes across structurally distinct domains such as privacy policy analysis, consumer contract fairness, and ESG disclosure, without task-specific retraining, and consistently outperforms retrieval and graph-based baselines under a document-level recall protocol that, to our knowledge, no prior compliance validation system has adopted.

Beyond efficiency, two directions would substantially broaden UCVE's applicability: extending to multi-document compliance settings where obligations must be satisfied across a corpus rather than a single document, and automating obligation node construction from regulatory text to reduce the domain expertise required at instantiation time.

More broadly, this work suggests that the utility of spectral graph methods in retrieval is not fixed by graph topology alone but is a function of how well edge weights encode task-relevant co-occurrence structure. The combination of reward-shaped edge learning with spectral smoothing represents a general principle for structured retrieval over any domain where requirements, concepts, or entities exhibit learnable co-relevance patterns that surface-level similarity cannot capture.

Limitations

CRG exhibits poor path efficiency on dense requirement graphs. Spectral reranking, by projecting node scores through the graph Laplacian eigenbasis, considers a large portion of the node set during scoring — on OPP-115, CLAUDETTE and ESRS this yields path efficiency scores of approximately 0.18, 0.12 and 0.10 respectively. However, recall_positive remains substantially above flat cosine retrieval on these domains, and spectral reranking performance is sensitive to hyperparameter configuration (k-NN neighborhood size, eigenvector count, and reranking weight), with tuned configurations yielding up to 4.37% additional lift. We interpret poor path efficiency on dense graphs as a meaningful signal about domain difficulty, and note that systematic hyperparameter search represents a direct path to further recall gains.

The three evaluation domains span 24 to 82 requirement nodes, which are relatively small requirement spaces. Our results already reveal a scalability signal: ESRS, the largest and densest graph, has a spectral signal that has a lower significance ($\lambda_2 = 0.04$) than OPP-115 ($\lambda_2 = 0.19$), and spectral reranking yields correspondingly smaller gains. This degradation has a compounding cause: larger requirement spaces require more training signal for RL edge weight learning to converge, but larger compliance corpora tend to have sparser positive labels per node: the same class imbalance that challenges the validation stage also weakens the RL reward signal during graph construction. On ESRS, this manifests as near-zero signal during graph construction, requiring a fixed category boost in place of data-driven edge learning. Addressing the compounding challenge of sparse rewards in large obligation spaces through reward shaping, curriculum learning or semi-supervised edge initialization represents an important direction for future work.

References

Shafiuddin Rehan Ahmed, Ankit Parag Shah, Quan Hung Tran, Vivek Khetan, Sukryool Kang, Ankit Mehta, Yujia Bao, and Wei Wei. 2025. Enhancing retrieval for ESGLLM via ESG-CID – A disclosure content index finetuning dataset for mapping GRI and ESRS. *Preprint*, arXiv:2503.10674.

Bernal Bernal Gutiérrez, Michael McInerney, Yiheng Shu, Jielin Wang, Hao Peng, Danqi Chen, and Yue Zhang. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *arXiv preprint arXiv:2405.14831*.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 4310–4330. Association for Computational Linguistics.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2020. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. In *Advances in Neural Information Processing Systems*, volume 33, pages 19314–19326.

Fan R. K. Chung. 1997. *Spectral Graph Theory*, volume 92. American Mathematical Society, Providence, RI.

Jiseong Chung, Ronny Ko, Wonchul Yoo, Makoto Onizuka, Sungmok Kim, Tae-Wan Kim, and Won-Yong Shin. 2025. Graphcompliance: Aligning policy and context graphs for llm-based regulatory compliance. *arXiv preprint arXiv:2510.26309*.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Abhimanyu Dubey, Aaron Grattafiori, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 7 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. 2018. Fast spectral ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7579–7588.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3253.

Linhao Luo, Jiaxin Li, Gholamreza Haffari, and Shirui Pan. 2024. Rog: Bridging agentic planning and knowledge graph reasoning. In *International Conference on Learning Representations (ICLR)*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–380.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. In *International Conference on Learning Representations (ICLR)*.

Ulrike von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N. Cameron Russell, Thomas B. Norton, Eduard Hovy, Joel Reidenberg, and Norman Sadeh. 2016. The creation and analysis of a website privacy policy corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1340, Berlin, Germany. Association for Computational Linguistics.

Shomir Wilson, Florian Schaub, Frederick Liu, Kanthashree Mysore Sathyendra, Daniel Smullen, Sebastian Zimmeck, Rohan Ramanath, Peter Story, Fei Liu, Norman Sadeh, and Noah A. Smith. 2018. Analyzing privacy policies at scale: From crowdsourcing to automated annotations. *ACM Transactions on the Web*, 13(1):1–29.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does NLP benefit legal system: A summary of legal

artificial intelligence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230. Association for Computational Linguistics.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. 2025. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8912–8924, Albuquerque, New Mexico. Association for Computational Linguistics.

A Graph Construction

The CRG is initialized using pairwise cosine similarities between `all-mpnet-base-v2` node embeddings, retaining edges with similarity ≥ 0.5. This yields dense base graphs—148 edges for OPP-115, 577 for CLAUDETTE, and 595 for ESRS—reflecting high semantic overlap within same-domain requirements.Table 5 decomposes the contributions of graph type and retrieval method. While spectral reranking on the cosine-only graph yields modest gains, and BFS on the RL-weighted graph produces inconsistent results, their combination achieves the strongest, most consistent performance. This confirms their conditional dependence: RL edge weights reshape the Laplacian's community structure into a form that spectral projection can exploit, whereas neither component succeeds independently. Full edge distributions are reported in Table 4.

Edge Type Removed	OPP-115 ΔR	CLAUDETTE ΔR	ESRS ΔR
`category`	-0.027	+0.000	-0.026
`fairness_tier`	+0.000	+0.000	+0.000
`kb_tag`	+0.000	+0.000	+0.000
`parent_category`	+0.000	+0.000	+0.000
`sub_attribute`	-0.006	+0.000	+0.000
`framework`	+0.000	+0.000	+0.000
All metadata	-0.034	-0.012	-0.026

Table 4: Recall change when removing individual metadata edge types.

B RL Edge Weight Learning Details

We learn a weight vector $\lambda = [\lambda_1, \ldots, \lambda_5]$ over five metadata edge types via epsilon-greedy BFS traversal over the domain graph G. The cosine similarity weight is fixed at 1.0 throughout. Training runs for 30 epochs with learning rate $\alpha = 0.05$ and ϵ decaying from 0.15 to 0.02. Please see Algorithm 1 for more details regarding the RL process.

Dataset	Condition	Recall	Precision
OPP	Flat cosine	0.695	0.195
	Spectral × Cos.	0.759	0.170
	BFS × RL (@ 14)	0.710	0.184
	Spectral × RL	**0.792**	**0.185**
CLA	Flat cosine	0.786	0.122
	Spectral × Cos.	0.848	0.117
	BFS × RL (@ 16)	0.806	0.113
	Spectral × RL	**0.860**	**0.118**
ESRS	Flat cosine	0.723	0.097
	Spectral × Cos.	0.729	0.097
	BFS × RL (@ 16)	0.626	0.104
	Spectral × RL	**0.755**	**0.101**
Avg	Flat cosine	0.735	0.138
	Spectral × Cos.	0.779	0.128
	BFS × RL	0.655	0.150
	Spectral × RL	**0.802**	**0.135**

Table 5: Component analysis of graph type × retrieval method. Each condition uses matched-K while spectral conditions use natural stopping.

Algorithm 1 RL Edge Weight Learning

Require: Graph G, docs $\mathcal{D}$, budget $B = 15$, epochs $T = 30$, LR α
Ensure: Learned weight vector λ

Initialize $\lambda_t \leftarrow 0.0$ for all metadata types t; $\lambda_{\text{cos}} \leftarrow 1.0$
for epoch $1 \ldots T$ **do**
  for each document $d \in \mathcal{D}$ **do**
    $e_q \leftarrow \text{Embed}(d)$
    Entry node $v_0 \leftarrow \arg\max_{v \in \mathcal{G}} \text{sim}(e_q, e_v)$
    Perform ϵ-greedy BFS for B steps, scoring neighbors v by:
    $\text{score}(v) = 0.3 \cdot \text{sim}(e_q, e_v) + 0.5 \cdot w(e_{\text{curr}\to v})$
    for each traversed edge e_k at step k **do**
      Assign credit $c(e_k) = \begin{cases} 1.0 - 0.5 \cdot \frac{k}{B} & \text{if dest is GT} \\ +0.1 & \text{if GT within 3 hops} \\ -0.05 & \text{otherwise} \end{cases}$
    end for
    for each metadata type t **do**
      $\Delta\lambda_t \leftarrow \frac{1}{|E_{\text{visited}}|} \sum_{e \in E_{\text{visited}}} c(e) \cdot \frac{\text{weight}_t(e)}{\sum_{t'} \text{weight}_{t'}(e)}$
      $\lambda_t \leftarrow \text{clamp}(\lambda_t + \alpha \cdot \Delta\lambda_t, [0.0, 3.0])$
    end for
  end for
end for
return λ =0

C Spectral Community Analysis

ID	Size	Label	Categories
C1	7	Core Practices	data_retention, data_security, 1st_party_collection, other, 3rd_party_sharing, user_access, user_choice
C2	10	Collection	First Party Collection/Use sub-attributes (9), Do Not Track (1)
C3	2	Jurisdictional	International and Specific Audiences (2)
C4	2	Policy Meta	Policy Change (1), Other (1)
C5	3	Third-Party	Third Party Sharing/Collection sub-attributes (3)

Table 6: OPP-115 Spectral Communities ($k = 5, k$-NN $= 3, n = 24$).

ID	Size	Label	Categories (Top-1)
C1	9	Content Removal	CR (9) — 100%
C2	17	Liability Limit.	LTD (17) — 100%
C3	5	Arbitration	A (5) — 100%
C4	15	Termination	TER (15) — 100%
C5	7	Unilat. Change	CH (7) — 100%

Table 7: CLAUDETTE Spectral Communities ($k = 5, k$-NN $= 3, n = 53$).

ID	Sz.	Label	Categories	Pillar
C1	12	Gov. & Strat.	Governance (5), Strategy (5), General (2)	G
C2	7	Workforce	Own Workforce (6), Business Conduct (1)	S
C3	2	Anti-corrupt.	Business Conduct (2)	G
C4	26	Environmental	Climate (9), Pollution (6), Circ. Economy (6), Water (5)	E
C5	3	Ethics	Business Conduct (3)	G
C6	11	Labor Stds.	Own Workforce (11)	S
C7	15	Stakeholder	Value Chain (5), Communities (5), Users (5)	S
C8	6	Biodiversity	Biodiversity/Ecosystems (6)	E

Table 8: ESRS-82 Spectral Communities ($k = 8, k$-NN $= 5, n = 82$).

The spectral communities reveal the qualitative differences in graph structure across each dataset. In CLAUDETTE, community membership aligned perfectly with ground-truth categories, where each of the five unfair clause types — arbitration, unilateral change, content removal, limitation of liability, and unilateral termination — forms its own spectral cluster (7. This can be seen by the cleanly separated clusters by category in Figure 3. This clean and direct alignment explains why the spectral reranking achieves its largest recall lift. This lift suggests projecting cosine scores onto the Laplacian eigenvectors effectively propagates relevance signal within category boundaries.

The spectral communities in OPP-115 are affected by the hierarchical structure of the dataset rather than the clean separation present in CLAUDETTE, which is flat. The spectral communities that formed had overlapping datasets (Figure 3), but upon further analysis, it became apparent each community has specific themes. The largest community (C1) groups together all broad-scope data practice categories (retention, security, collection, sharing, access, and choice) which occur in virtually every privacy policy (6. These nodes share high embedding similarity (mean cosine 0.635 vs. 0.469 across communities) because their requirement descriptions all follow the pattern "Does the text describe [user data practice]?" The remaining communities (C2–C5) capture more specialized sub-attribute nodes and rare topics (international audiences, policy changes) that are genuinely discriminative. Spectral reranking exploits this by treating C1 as a co-activation group: if one core data practice is relevant, the others likely are too.

ESRS communities partition cleanly along ESG boundaries. The largest cluster (C4) unifies all environmental topics while social requirements are split into three distinct subclusters: general workforce standards (C6), specific labor issues (C2), and external stakeholder impacts (C7). Governance nodes fragment across C1, C3 and C5. This pillar aligned structure enables spectral reranking to boost recall by 3.2% despite the ESRS subgraph's weaker spectral gap ($\lambda_2 = 0.04$ vs 0.19 for OPP-115)m as the eigenvectors still capture meaningful boundaries.

C.1 Structural Signal

Each CRG's edge types contribute differently across domains, reflecting the structural properties of each dataset. On OPP-115, removing `same_category` edges costs -2.7% recall and removing `same_sub_attribute` edges costs an additional -0.6%. On CLAUDETTE, the three dataset-specific edge types individually contribute negligible recall lift, but removing all three collectively costs -1.2% recall, suggesting that structural signal in CLAUDETTE is distributed rather than localized to any single edge type. On ESRS, only `same_category` contributes meaningfully (-2.6% recall when removed), with all other edge types contributing zero individually and -2.6% collectively, mirroring the pattern of OPP-115 but with a single active edge rather than two. The near-additivity of OPP-115's two active edge types indicates they capture largely independent

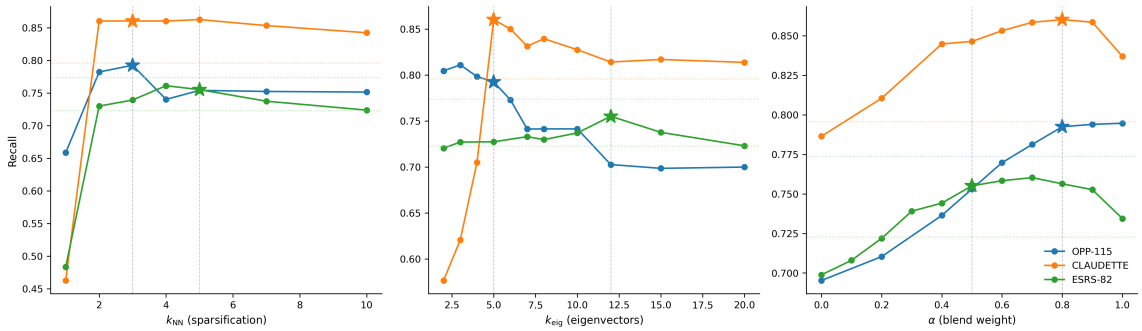

Figure 2: Spectral reranking hyperparameter sensitivity. Each panel sweeps one parameter while holding the others fixed at their chosen values ($k_{\text{NN}} = 3$, $k_{\text{eig}} = 5$, $\alpha = 0.8$). Stars mark chosen values; dotted lines show flat cosine baselines. Performance is stable across wide ranges for k_{eig} and α, but degrades sharply at $k_{\text{NN}} = 1$ where over-sparsification disconnects the graph.

structural signals; the collective dependence on CLAUDETTE reflects its denser, flatter schema. Full edge type results are reported in Table 4.

D Graph Traversal

D.1 GSAEPS

Traversal begins from one or more entry nodes selected by the GSAEPS procedure. Rather than simply starting from the node with highest cosine similarity to the query — which may be locally optimal but structurally isolated — GSAEPS scores candidates using a linear combination of features that jointly capture direct relevance, statistical standing within the node set, and structural position in the graph.

Dataset	Entry Strategy	Rec.	Eff.	C70	Vis.
OPP-115	Random	0.967	0.160	0.725	22.6
	Cosine	0.847	0.177	0.646	17.6
	GSAEPS	**0.865**	**0.174**	**0.657**	18.4
CLA.	Random	0.998	0.051	0.713	47.1
	Cosine	0.967	0.078	0.700	30.5
	GSAEPS	**0.967**	0.078	0.700	30.6
Avg.	Random	0.982	0.106	0.719	34.9
	Cosine	0.907	0.127	0.673	24.1
	GSAEPS	**0.916**	**0.126**	**0.679**	24.5

Note: Random entry achieves highest recall by setting a low sweep threshold, effectively degenerating to flat cosine retrieval. At matched visit budgets (∼18 nodes), GSAEPS outperforms cosine-only entry by +1.7pp recall on OPP-115.

Table 9: Effect of entry point strategy on downstream traversal recall.

D.2 Traversal Strategies

Adaptive Graph BFS performs adaptive-threshold BFS from one or more entry nodes, using edge weights to guide neighbor selection at each step. From the current node, all unvisited same-domain neighbors are scored by:

$$score(v) = 0.3 \cdot sim(e_q, e_v) + 0.5 \cdot w(e_{current \to v}) \quad (4)$$

where $(e_{current \to v})$ is the composite RL-weighted edge weight from the current node to candidate v. This scoring intentionally weighs edge structure more heavily than direct query similarity, encoding the intuition that nodes strongly connected to already-visited relevant nodes are more likely to be relevant than nodes that merely resemble the query text via cosine similarity. Traversal continues as long as the best candidate's similarity exceeds an adaptive stopping threshold:

$$threshold = max(0.60 \times median(entry_s ims), 0.35) \quad (5)$$

where $entry_sims$ is the set of query similarities for all entry nodes. When no remaining candidate exceeds this threshold, traversal halts. This adaptive stopping allows traversal to continue longer when entry nodes are highly similar to the query — indicating a well-matched document — and terminate earlier when the match is weaker, naturally scaling the visited set size to document relevance rather than imposing a fixed budget.

Simultaneous Graph Scoring scores all same-domain nodes simultaneously using a seven dimensional feature vector that combine query similarity and graph structure. For each node v_i, the score is:

$$s(v_i) = w_{gs} \cdot f(v_i) \quad (6)$$

where $f(v_i)$ is a seven dimensional feature vector. The features can be found in 10.

All nodes whos score exceeds a fraction of the maximum score is retrieved:

$$retrieved = \{v_i : s(v_1) \geq \tau \times maxs(v_j)\} \quad (7)$$

Feature	Description
Query Sim	Cosine similarity between e_q and e_i
Neighbor Avg Wght	Mean edge weight from v_i to 5 most similar nodes
Parent Avg Sim	Mean query similarity of same-parent nodes
Parent Max Sim	Max query similarity among same-parent nodes
Same Parent Top-1	1 if v_i shares parent category with top node
Query Z-Score	Z-scored query similarity within dataset
Parent Count	Fraction of dataset nodes in v_i's parent category

Table 10: Features used in graph-enhanced node scoring.

with $\tau = 0.6$. Simultaneous Graph Scoring does not require sequential traversal decisions or a similarity sweep — all nodes are scored and filtered in a single pass. Hence, this method is more suited for smaller, more dense graphs where structural signals are informative across the full node set.

D.3 Graph-Document Alignment

We evaluate a graph-document alignment baseline inspired by GraphCompliance (Chung et al., 2025), adapted to our evaluation protocol. The baseline operates in three stages: (1) the document is sentence-segmented and a context graph is constructed by linking sentences sharing key terms above a Jaccard similarity threshold of 0.5; (2) each obligation node's `validation_question` is embedded and used to retrieve the top-k most similar sentence nodes from the context graph; (3) an Large Language Model (LLM) alignment gate receives the retrieved sentences and produces a binary compliance verdict per node. Unlike CRG, this baseline derives its graph structure from the document rather than from a pre-built obligation space. We note this implementation is an adaptation rather than a faithful reimplementation of GraphCompliance; results should be interpreted accordingly.

Data	Model	Acc	F1	Y-P	Y-R	Y-F1
OPP	70B	0.396	0.392	0.948	0.288	0.442
CLA	8B	0.826	0.559	0.121	0.974	0.215
ESRS	8B	0.870	0.571	0.130	0.604	0.213

Table 11: Graph-document alignment baseline. **Y-P** and **Y-R** denote precision/recall over the positive class.

The results in Table 11 reveal a precision-recall collapse on CLAUDETTE and ESRS: Yes-recall is high (0.974 and 0.604 respectively) but Yes-precision collapses to 0.121–0.130, indicating that the alignment gate defaults to flagging nearly all nodes as violated. OPP-115 exhibits the inverse pattern — high Yes-precision (0.948) with low Yes-recall (0.288) — suggesting that the context graph structure is over-selective in that domain, failing to surface relevant passages for most violated nodes. In both failure modes, the document-derived context graph does not provide sufficient discriminative signal to distinguish violated from satisfied requirements. CRG's pre-built obligation space and spectral reranking address this directly: by operating over a graph whose structure reflects learned co-relevance among requirements rather than surface-level term overlap in the document, UCVE achieves substantially higher and more consistent recall across domains.

E Validation

E.1 LLM Prompt Templates

Listing 1: Baseline Prompt (`graph_context="none"`)

```
You are a document analyst. You are given a
    document and a requirement that has been
    retrieved as a candidate match. The
    requirement is assumed to be at least
    somewhat relevant to the document. Your task
     is to classify whether the document is
    POSITIVE or NEGATIVE for this requirement.

- Positive means the document text contains
    language that exhibits, satisfies, or
    violates this requirement.
- Negative means the document text does NOT
    contain language that exhibits this
    requirement, OR the requirement is not
    applicable to this document.

## Requirement
**Requirement**: Does the report describe the
    sustainability matters addressed by the
    governing bodies and the information
    provided to them for oversight?
**Category**: governance
[...]
```

Listing 2: Persona Prompt (`graph_context="persona"`)

```
[...]
## Context from Graph Traversal
The traversal (5 nodes visited) was connected by
     same category (100%) relationships.
Visited requirement topics: "governance", "
    general".
This requirement was reached because it is
    connected to "governance" via embedding
    similarity (weight 0.91).
```

```
[...]
```

E.2 Full Pipeline Results - Llama 3.1 8B

Dataset	Strategy	Prec	Rec	ΔRec$\uparrow^{\dagger}$
OPP-115	Retrieval ceiling	—	.792	—
	Baseline	.775	.564	−.228
	Persona	**.782**	**.567**	**−.225**
CLA.	Retrieval ceiling	—	.860	—
	Baseline	.748	**.697**	**−.163**
	Persona	**.773**	.685	−.175
ESRS	Retrieval ceiling	—	.722	—
	Baseline	**.153**	.521	−.201
	Persona	.149	**.531**	**−.191**

$^{\dagger}\Delta$Rec$\uparrow$ is the gap between strategy recall and the retrieval ceiling. Negative values indicate recall below the ceiling; closer to zero is better. The gap reflects *validation errors* on retrieved nodes rather than *retrieval failure*.

Table 12: End-to-end validation performance with graph-derived context from the CRG using Llama-3.1-8B.

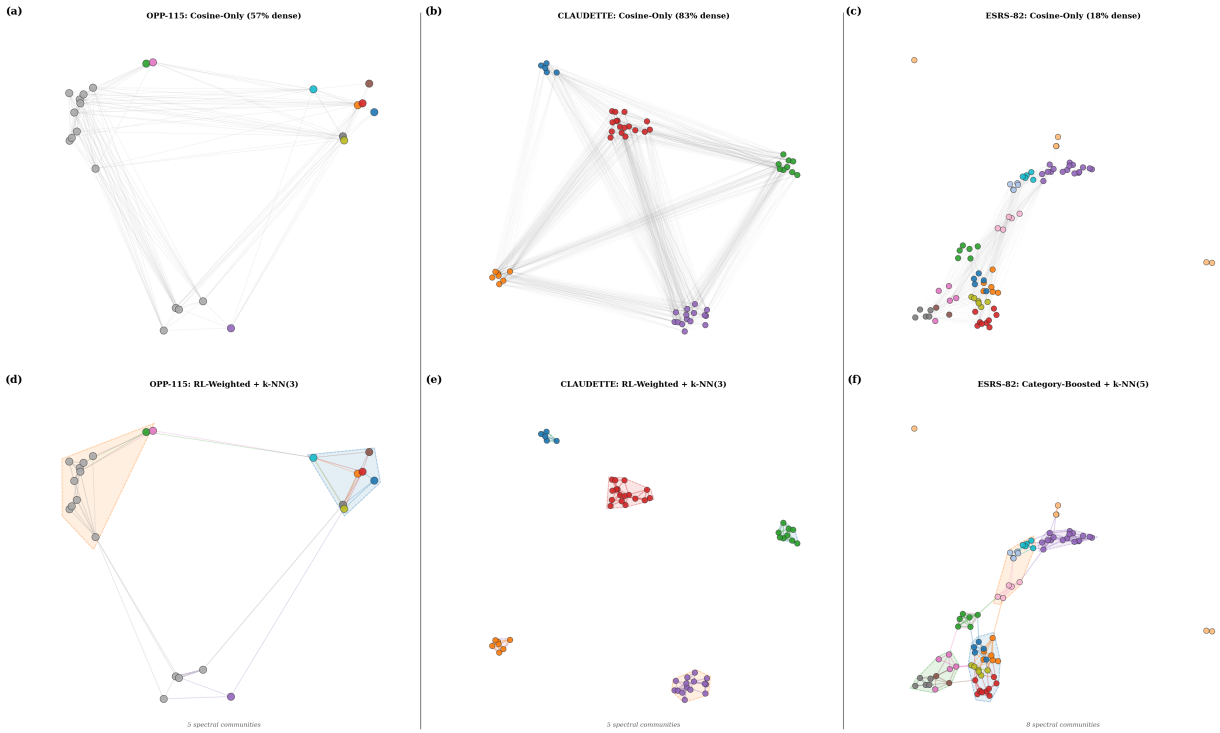

Figure 3: Community structure of the Contextual Requirement Graph across three compliance domains. (a–c) Dense cosine-similarity graphs (57–100% edge density) where near-complete connectivity obscures meaningful topology. (d–f) After edge weight learning and k-NN sparsification, distinct spectral communities emerge (dashed hulls; 5 eigenvectors for OPP-115/CLAUDETTE, 12 for ESRS-82). Node colors indicate requirement categories. This learned sparse structure is the signal spectral reranking exploits for +3.4–4.5% recall lift over flat cosine retrieval.