GraphCompliance: Aligning Policy and Context Graphs for LLM-Based Regulatory Compliance

Jiseong Chung
Seoul National University
Seoul, Republic of Korea
jiseong0529@snu.ac.kr

Ronny Ko
Osaka University
Osaka, Japan
ronny@ist.osaka-u.ac.jp

Wonchul Yoo
Seoul National University
Seoul, Republic of Korea
wchyoo@snu.ac.kr

Makoto Onizuka
Osaka University
Osaka, Japan
onizuka@ist.osaka-u.ac.jp

Sungmok Kim
Seoul National University
Seoul, Republic of Korea
sungmok.kim@snu.ac.kr

Tae-Wan Kim*
Seoul National University
Seoul, Republic of Korea
taewan@snu.ac.kr

Won-Yong Shin*
Yonsei University
Seoul, Republic of Korea
wy.shin@yonsei.ac.kr

Abstract

Compliance at web scale poses practical challenges: each request may require a regulatory assessment. Regulatory texts (e.g., the General Data Protection Regulation, GDPR) are cross-referential and normative, while runtime contexts are expressed in unstructured natural language. This setting motivates us to align semantic information in unstructured text with the structured, normative elements of regulations. To this end, we introduce GraphCompliance, a framework that represents regulatory texts as a Policy Graph and runtime contexts as a Context Graph, and aligns them. In this formulation, the policy graph encodes normative structure and cross-references, whereas the context graph formalizes events as subject–action–object (SAO) entity–relation triples. This alignment anchors the reasoning of a judge large language model (LLM) in structured information and helps reduce the burden of regulatory interpretation and event parsing, enabling a focus on the core reasoning step. In experiments on 300 GDPR-derived real-world scenarios spanning five evaluation tasks, GraphCompliance yields 4.1–7.2 percentage points (pp) higher micro-F1 than the case of LLM-only and RAG baselines, with a reduced tendency toward under- and over-prediction, resulting in a higher recall and lower false positive rates. Ablation studies indicate contributions from each graph component, suggesting that structured representations and a judge LLM are complementary for normative reasoning.

CCS Concepts

• **Information systems → Retrieval models and ranking**.

Keywords

Regulatory Compliance, Large Language Models, Knowledge Graph, Compliance Automation, General Data Protection Regulation (GDPR)

1 Introduction

Automating *regulatory compliance* for web-scale systems has become imperative as services continuously ingest personal data, orchestrate third-party models, and operate across jurisdictions governed by texts such as the General Data Protection Regulation (GDPR) [1]. The task remains difficult because legal norms are densely interlinked, actor- and scope-sensitive, and often hinge on exceptions and derogations that must be traced across articles and recitals—traditionally treated as a logic-heavy verification problem rather than mere text matching [9]. In modern web ecosystems—where platforms broker user data across services and APIs—accountable, auditable compliance is foundational to trustworthy web operation [29].

Effective reasoning about regulatory compliance demands two key capabilities: 1) **semantic understanding** to interpret the nuances of unstructured contexts and 2) **structural reasoning** to navigate scopes, exceptions, and cross-references [3]. Recent large language models (LLMs), along with reasoning pipelines that use them, such as retrieval-augmented generation (RAG), excel at the former but struggle to ensure the verifiability required for the latter due to their black-box nature [16, 24]. Conversely, structured representations like graphs capture structural relationships explicitly but face limitations in interpreting the rich semantic details of natural language [12]. A natural attempt to resolve this dilemma is to leverage graphs to assist LLM reasoning, as in frameworks such as GraphRAG, which use graphs for *enhanced retrieval* [6]. However, such retrieval-centric approaches falter when compliance hinges on deep structural logic. Figure 1 illustrates three recurrent failure modes: missed cross-references, broken decision-tree logic, and checklist conflation. First, for regulations with explicit cross-references (see Figure 1(a)), which serve as key reasoning cues, query-based retrieval often misses the reference chain by focusing on query relevance over inter-chunk relationships [1]. Similarly, when regulations follow a decision-tree structure (see Figure 1(b)), end-to-end LLMs may omit necessary chunks or lose the logical
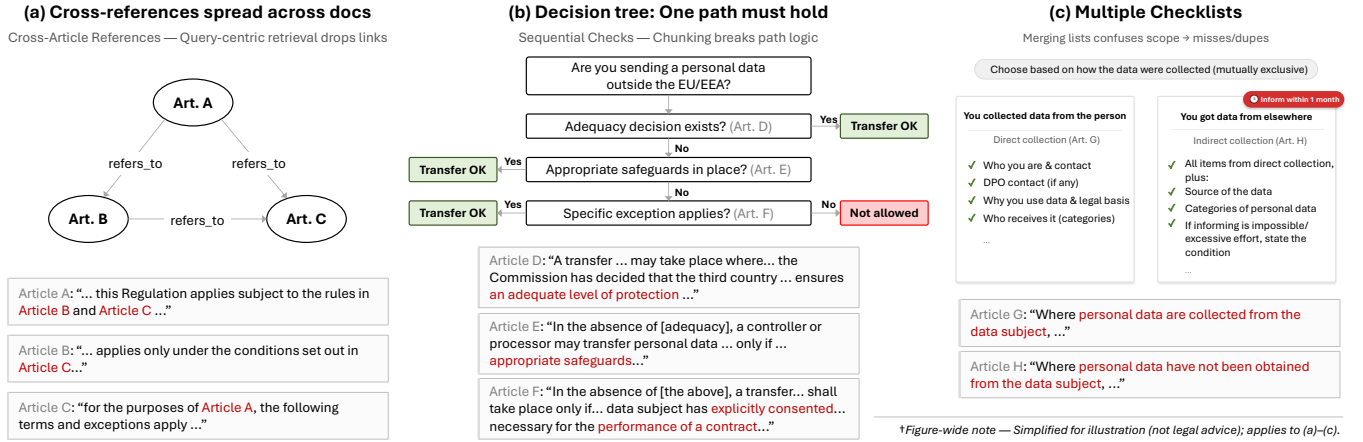
---

*Corresponding author.

Figure 1: Failure cases for existing retrieval/LLM pipelines. (a) Cross-references dispersed across articles/recitals: as explicit references are not co-retrieved, parts of the chain are missing. (b) Decision-tree distribution of provisions: order-dependent yes/no branching is not preserved by keyword/embedding similarity, leading to incorrect end states. (c) Mutually exclusive checklists with a time limit: direct vs. indirect lists are often merged and deadlines dropped, causing omissions or duplicates.

connection between nodes in the multi-hop path [7, 33]. Third, for checklist-style obligations (see Figure 1(c)), ambiguous or complex contexts often confuse LLMs, leading to conflated lists and missed or duplicated checks [1].

To address specific failure modes in structural reasoning, we propose GRAPHCOMPLIANCE. Our framework constructs two knowledge graphs (KGs) from policy documents and a given context: a **policy graph** that captures the logical structure of regulations, and a **context graph** that formalizes the situational facts. These KGs are aligned by a **compliance gate**, which performs deterministic structural analysis (e.g., reference traversal, exception chaining), before presenting a constrained and simplified problem to the LLM. This approach distinguishes our framework from prior work [6, 11, 35]. While our constructed KGs also enhance retrieval and serve as a knowledge store, their primary function is to act as an active reasoning scaffold: explicit structural lookups—such as traversing cross-references or checking actor attributes—are handled by reliable graph traversal rather than the LLM's general contextual understanding. Consequently, the LLM is reserved for interpreting nuanced semantic information and rendering the final judgment on a pre-analyzed, structured input, tackling cases that contextual similarity or enhanced retrieval alone struggle to resolve (Figure 1).

We instantiate the framework on a benchmark that links the original regulatory text of the GDPR with real or synthetic scenarios, enabling evaluation of compliance or non-compliance of regulatory provisions. GDPR is an EU-wide privacy law governing personal-data processing by controllers and processors; it sets principles and lawful bases and grants data-subject rights (e.g., access, erasure) [1]. We construct a policy graph covering all articles of the GDPR and assess regulatory compliance on a benchmark of 300 real-world-inspired scenarios. Our evaluation includes overall accuracy, error analysis, in-depth ablation studies on our framework's submodules, and extensive comparisons with baselines, demonstrating a 4.1–7.2 pp F1 score gain over existing methods, including RAG and GraphRAG [6, 16].

The primary contributions of this work are as follows:

- **A KG-based LLM framework specialized for regulatory compliance.** A new end-to-end hybrid framework specifically designed to address the structural and semantic gap between normative regulations and real-world contexts.
- **A new methodology based on dual-graph alignment.** A novel approach that models policy and context as separate KGs and uses a 'Plan-anchored compliance gate' to align them, thereby constraining the LLM's reasoning process.
- **State-of-the-art performance through empirical validation.** Comprehensive empirical validation demonstrating a **significant accuracy gain up to 7.2 pp** over strong baselines, with ablation studies attributing these gains to our core structural components.

2 Related Work

2.1 Graph-based LLM Frameworks

A growing body of work integrates graphs with LLMs for evidence organization, cross-document reasoning, and multi-hop retrieval [22]. Representative systems include GraphRAG, which builds entity/community graphs and precomputes community "reports" for query-focused retrieval and summarization [6]; HippoRAG, which couples KGs with Personalized PageRank as a structure-aware long-term memory for multi-hop QA [14]; G-Retriever, which selects task-relevant subgraphs from textual and attributed graphs [11]; MindMap, which aggregates evidence subgraphs prior to LLM-based reasoning [31]; and GraphReader, which conditions the aggregation on graph signals for reasoning-as-reading [19]. Beyond these, KG-centric RAG variants mitigate retrieval gaps and preserve inter-chunk structure via KG-guided expansion [35], text-to-subgraph retrieval [13], and subgraph-size control [17]. In industry, KG-RAG has seen deployment [32]. Inspired by GraphRAG's graph rendering of text, we repurpose the
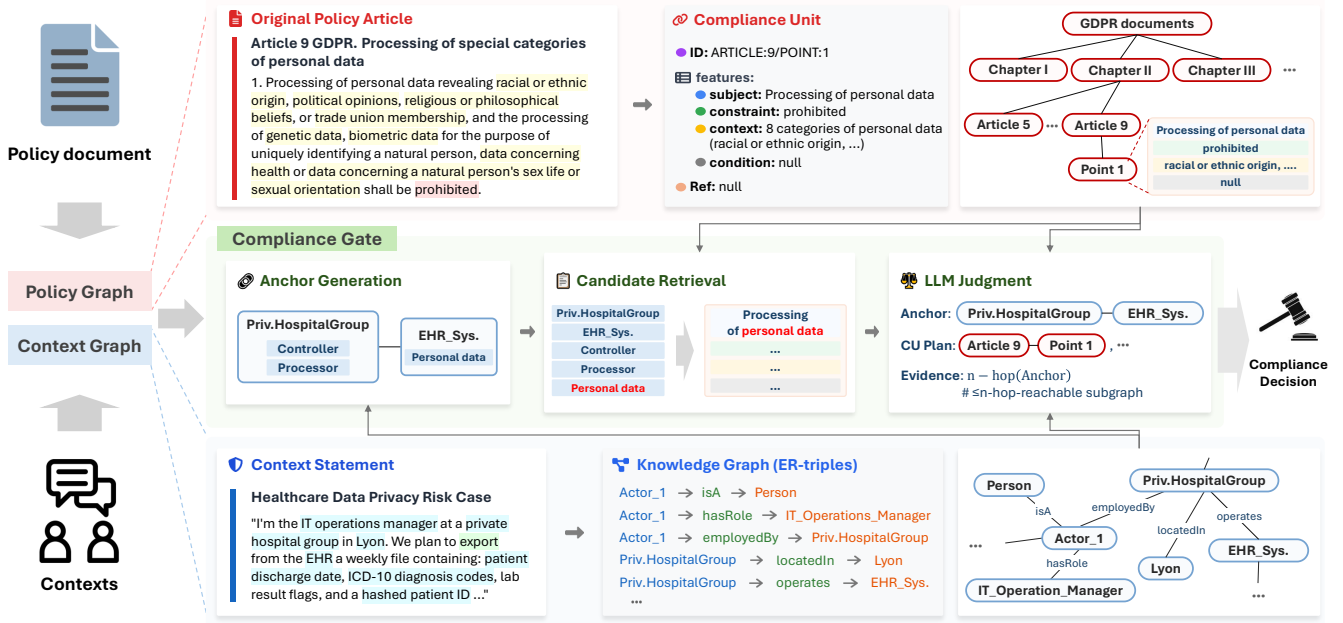
Figure 2: Pipeline overview of GRAPHCOMPLIANCE. Red: *Policy graph construction*; **Blue boxes:** *Context graph construction*; **Green boxes:** *Compliance Gate.*

KG from retrieval to decision scoping: we (i) restrict the context via actor alignment, (ii) execute cross-references/exceptions in-graph, and (iii) ask the LLM only for semantic judgments over a curated policy-derived check plan.

2.2 Graph-Structured Planning and Hierarchical Selection

Beyond flat retrieval, RAPTOR organizes corpora into a recursive summary tree for hierarchical selection [26], while KG planning derives stepwise plans to steer RAG [27, 30]. These lines of work primarily use graphs to structure selection and planning for LLMs. We adopt this external structure for normative texts by modeling regulations and contexts as policy and context graphs, and by executing a deterministic compliance gate (cross-reference traversal, actor alignment) before applying LLM-based semantic matching to a constrained policy-derived check plan.

2.3 LLM-based Automation of Legal and Regulatory Compliance

LLMs have been used to analyze privacy policies and compliance artifacts. PolicyGPT frames privacy-policy classification using ChatGPT/GPT-4, and Rodríguez et al. report large gains with ChatGPT/LLaMA 2 for scalable policy analysis [23, 28]. Domain-specialized legal LLMs (SaulLM-7B, Lawma-8B) show benefits across analysis, classification, and generation [4, 5]; PPGen targets the automatic generation of GDPR-compliant policies [25]. For verification, PrivComp-KG combines an LLM (via RAG) with a GDPR KG to align snippets to articles and execute SWRL checks, while Compliance-to-Code compiles KG/schema-derived requirements

into executable checks [8, 18]. In software contexts, Alecci et al. link Android code behaviors to privacy requirements [2], and Hassani studies LLM-based requirement extraction and conformance checking across the GDPR and DPAs [10]. Most prior systems operate on isolated text segments with limited cross-article/recital reference tracking. We differ by (i) adopting a policy-agnostic KG schema (premises and CUs), (ii) performing policy-guided context normalization via strong/weak hypernyms, and (iii) executing a compliance gate that performs meta-scope checks, actor alignment, constraint/ condition tests, and cross-reference traversal before the LLM's semantic judgment.

3 Methodology

This section presents the three components of GRAPHCOMPLIANCE— Policy Graph Construction, Context Graph Construction, and Compliance Gate Reasoning. Figure 2 illustrates the end-to-end pipeline; pseudocode for the main procedures appears in Appendix C.

Overview. (i) *Policy Graph Construction* converts regulatory text into premises and compliance units (CUs), each formalized as $\{subject, constraint, context, condition\}$, and links CUs with cross-reference edges. (ii) *Context Graph Construction* extracts entity–relation (ER) triples from the scenario and maps entities to policy-guided hypernyms. (iii) The *Compliance Gate* groups policy-relevant entities into *anchors* (actor/data/system). For each anchor, it retrieves a top-K CU Plan, performs listwise LLM judgment using an "evidence window," applies a reference-edge override for exceptions, and aggregates results at the article level.

3.1 Policy Graph Construction

The objective of this subsection is to detail our three-stage process for converting unstructured regulatory text into a structured **policy graph** (G_P): text classification, rule formalization, and relational linking.

First, the regulatory text is segmented into semantic units (e.g., articles, clauses) and classified as either a contextual **premise** or an actionable **compliance unit (CU)**. Here, a *premise* denotes non-deontic definitional or interpretive material—such as terms, role definitions, scope statements, and purposes—that the system must know to read the code but that is not itself judged for (non)compliance; for consistency, we assign premises at the article level. Each CU is then formalized into a 4-tuple schema, $r = \langle S, \Theta, \Pi, \kappa \rangle$, representing its subject (S), constraint (Π), context (κ), and conditions (Θ) [9, 18]. We further distinguish *actor-CUs*, which encode obligations, prohibitions, or permissions addressed to a role-bearing actor (e.g., controller, processor, recipient) and constitute the units actually judged in a case, from *meta-CUs*, which specify applicability (temporal/territorial scope, role qualification, covered processing) and therefore gate whether an actor-CU should be considered; meta-CUs are evaluated first and are not reported as standalone violations. This classification and typing are performed once, offline, by an LLM.

Finally, the CU nodes are interconnected by identifying cross-references to form the graph structure. Our two-pronged approach uses regular expressions for **explicit references** (e.g., *"Article 5"*) and a small LLM to resolve **implicit, relative references** (e.g., *"paragraph 1"*). This process creates relational edges like REFERS_TO, transforming the flat text into a policy graph (G_P) that faithfully preserves the regulation's logical structure.

3.2 Context Graph Construction

The objective of this subsection is to detail the process of converting unstructured real-world contexts—such as incident reports or logs—into a structured **context graph** (G_C) that can be aligned with the policy graph.

The first step in building G_C is to extract entities and their relations as ($subject, predicate, object$) ER triples [12] from the unstructured text. For this task, we employ the LLM-based ER-triple extraction method proposed in **GraphRAG** [6].

The next step is **hypernym mapping**, which links the extracted entities to the formal terms of the policy. While such conceptual mapping is typically handled implicitly by an LLM's internal representations, we add this as an explicit process to make the mapping transparent and stabilize downstream reasoning. To achieve this, we use the previously built policy graph as retrieval information to guide the mapping.

This policy-guided normalization proceeds as follows. Let H denote the vocabulary of policy-level hypernyms derived from the policy graph. For each context entity e, we retrieve the top-M policy fragments via dense retrieval and elicit candidate hypernyms, yielding a small set $H_e \subseteq H$. Each candidate is treated as a proposal $r = (e, h(r), \text{frag\_id}(r), \text{src}(r))$ and is assigned an LLM-generated confidence score $s(r) \in [0, 1]$. A proposal is marked **STRONG** if its supporting fragment is a *Premise*, and **WEAK** otherwise. Let R_e denote the set of proposals associated with entity e.

Finally, since multiple proposals may exist for a single entity, their scores are aggregated for each unique hypernym label using a max-pooling approach that provides a bonus to STRONG proposals. The aggregated confidence for a hypernym h (per entity e), denoted $\widehat{s}_e(h)$, is

$$\widehat{s}_e(h) = \min\left\{1, \max_{r \in R_e : h(r) = h} \left(s(r) + \beta \mathbf{1}\{\text{STRONG}(r)\}\right)\right\}, \quad (1)$$

where r is an individual proposal, $h(r)$ is its hypernym label, and $s(r)$ is its LLM-generated confidence score. β is a small bonus hyperparameter (we set $\beta = 0.3$ in experiments). $\mathbf{1}(P)$ is the indicator function: 1 if P is true, and 0 otherwise. We retain the top-N hypernyms per entity according to $\widehat{s}_e(\cdot)$ (we use $N = 5$). We denote

$$\Phi_N(e) = \text{Top–}N\left(\left\{ (h, \widehat{s}_e(h)) : h \in H_e \right\}\right), \quad (2)$$

where Top–N returns the N highest-scoring *ordered* pairs (sorted by $\widehat{s}_e(h)$; ties broken by STRONG>WEAK, then lexicographic h). Through this process, the context graph (G_C) is finalized, with a structure that includes the ER triples and, as a feature for each entity, a list of normalized hypernyms.

3.3 Compliance Gate

The **compliance gate** is the core reasoning engine that takes the constructed policy graph (G_P) and context graph (G_C) as inputs to produce a final compliance judgment. The overall pipeline consists of three main stages: **(1) candidate retrieval and re-ranking**, **(2) LLM-based judgment and exception handling**, and **(3) final decision aggregation**.

The reasoning process starts by retrieving candidate CUs from G_P for each factual **anchor** in G_C. We use a simple three-part bi-encoder score for an anchor a and a CU c: (i) similarity between the anchor's entity features and the CU's subject, (ii) similarity between the anchor's hypernyms and the CU's subject, and (iii) a small bonus when any hypernym overlaps the CU's subject terms. Formally,

$$\begin{aligned} S(a,c) = \, &w_{\text{ent}} \langle \mathbf{v}_{\text{ent}}(a), \mathbf{v}_{\text{subj}}(c) \rangle \\ &+ w_{\text{hyp}} \langle \mathbf{v}_{\text{hyp}}(a), \mathbf{v}_{\text{subj}}(c) \rangle \\ &+ w_{\text{bonus}} \mathbf{1}\{ H(a) \cap \text{Subj}(c) \neq \emptyset \} \end{aligned} \quad (3)$$

Here, $H(a)$ is the set of hypernyms attached to a; $\mathbf{v}_{\text{ent}}(a)$ and $\mathbf{v}_{\text{hyp}}(a)$ are embeddings of the anchor's entity and hypernym features; $\mathbf{v}_{\text{subj}}(c)$ is a pre-cached subject embedding of c; $\mathbf{1}(P)$ is the indicator function; and $\text{Subj}(c)$ is the subject-term set of c. Taking the top-K_1 CUs by $S(a, c)$ yields the broad candidate set $C_a^{(1)}$.

Next, this candidate set is refined by a more powerful but computationally intensive cross-encoder that jointly processes $q(a)$ and $d(c)$ to model deep interactions, following the standard re-ranking practice in graph-augmented retrieval [11]. We construct

$$\begin{aligned} q(a) &= [\texttt{predicate}; \texttt{actor\_type}; \texttt{object\_type}], \\ d(c) &= [\texttt{subject}; \texttt{constraint}; \texttt{condition}], \end{aligned} \quad (4)$$

from the anchor and the candidate CU, respectively. The result of this funneling process is a concise and highly relevant **CU Plan**—a curated list of rules $\{P_i\}_{i=1}^K$ prepared for the final judgment.

In the second stage, the generated CU Plan and an **evidence window**—a local subgraph of G_C centered on the anchor—are passed to an LLM for judgment. The initial judgment is performed in a listwise

Table 1: End-to-end compliance judgment performance on our GDPR benchmark. We compare our `GraphCompliance` framework against various baselines across different underlying LLMs. The best performance for each metric is highlighted in bold.

Category	Method	Top-K	Micro F1	Macro F1	Micro F2	Macro F2	MCC	LLM Rater
Raw LLM	GPT-4o	—	44.5	47.2	37.9	42.2	39.7	52.70
	GPT-4.1	—	44.9	47.5	39.2	42.3	41.0	55.41
	GPT-5-thinking	—	49.8	50.8	41.7	44.2	46.6	59.01
RAG (Top-K)	GPT-4o	8	43.8	44.8	36.0	38.4	40.9	59.31
	GPT-4o	30	42.9	44.0	34.9	37.5	40.1	57.48
	GPT-4.1	8	49.5	51.1	43.4	45.6	44.7	60.23
	GPT-4.1	30	49.2	50.9	43.3	45.2	43.3	62.47
	GPT-5-thinking	8	50.6	51.6	44.1	46.5	47.9	68.77
	GPT-5-thinking	30	50.8	52.0	45.3	47.0	48.1	72.70
	Llama3-8B Instruct	8	22.5	21.7	21.1	20.6	15.3	18.18
	SaulLM-7B (GDPR Inst.)	8	22.8	23.2	39.2	37.4	22.1	20.55
	Lawma-8B	8	21.9	22.6	19.1	19.4	17.1	13.27
GraphRAG	GPT-4.1 (local, neighborhood)	8	41.0	40.1	43.4	43.8	31.8	51.38
	GPT-4.1 (global, community summary)	8	47.5	46.8	48.3	49.8	37.7	59.28
GraphCompliance	**GPT-4o**	8	**51.7**	**49.9**	**51.0**	**50.8**	**44.0**	**63.87**
	GPT-4.1	8	**55.4**	**52.9**	<u>63.0</u>	<u>59.7</u>	**48.8**	**76.62**
	GPT-5-thinking	8	<u>57.1</u>	<u>55.4</u>	**62.4**	**58.8**	<u>49.5</u>	<u>79.85</u>
	Llama3-8B Instruct	8	**26.6**	**24.1**	**23.9**	**23.2**	**18.0**	**22.19**
	SaulLM-7B	8	**28.4**	**26.7**	**43.5**	**41.0**	**24.7**	**23.40**

Notes. — indicates not applicable (no retrieval). *Top-K* is the number of retrieved chunks.

fashion, where for each anchor, the evidence window $W(a)$ and the *entire* CU Plan list are provided as a single input to a judgment function J:

$$J : \big(W(a), \{P_i\}_{i=1}^K \big) \mapsto \{ (\hat{y}_i, s_i, \mathsf{why}_i, \mathsf{evid}_i) \}_{i=1}^K. \quad (5)$$

In the first LLM call, the judge holistically considers relationships among candidate rules and, for each CU i, returns a compliance label ($\hat{y}_i$), a confidence score (s_i), a rationale (why_i), and evidence (evid_i) simultaneously. Using a concise judge prompt, we restrict reasoning to the retrieved ANCHOR and CONTEXT WINDOW, prioritize explicit contradictions while allowing strongly implied ones, and forbid inference from silence (ambiguous or out-of-scope cases → INSUFFICIENT/NOT_APPLICABLE). To handle the complexity of regulatory reasoning, we introduce a crucial post-processing step for any judgment initially deemed NON_COMPLIANT. For each violated CU (c), we first compute its **reference closure** $\mathcal{R}^{(c)}$— the set of all CUs reachable by traversing reference edges in G_P. A second LLM call then determines whether any CU within this $\mathcal{R}^{(c)}$ constitutes a valid exception that overrides the initial violation. This override mechanism provides a practical implementation of defeasible logic:

$$\hat{y}_c' = \begin{cases} \text{COMPLIANT}, & \text{if } \exists r \in \mathcal{R}^{(c)} \text{ s.t. IsException}(r, W(a)) = \text{true}, \\ \hat{y}_c, & \text{otherwise.} \end{cases} \quad (6)$$

Finally, we aggregate the (post-override) judgments to a single verdict per article using a *violation-first* rule: if any CU linked to an article is labeled NON_COMPLIANT, we report the highest-confidence violation; otherwise, we return the highest-confidence remaining label.

4 Experimental Evaluations

This section presents our experimental design for validating GRAPH-COMPLIANCE. We first introduce the core research questions, describe the dataset, define the evaluation protocol, summarize the baselines, and conclude with the experimental setup. Implementation details are deferred to Appendix B.1.

Research Questions (RQs). To systematically evaluate our framework, we designed experiments guided by the following RQs:

- **RQ1 (Accuracy and Robustness):** Does GRAPHCOMPLI-ANCE outperform baselines, and are the gains consistent across underlying LLMs?
- **RQ2 (Submodule contribution):** Which of the proposed modules contributes most to the overall performance?
- **RQ3 (Submodule fidelity):** Are the generated graphs sufficiently accurate and robust so as not to bottleneck end-to-end performance?
- **RQ4 (Prompt sensitivity):** How sensitive is the framework's performance to variations in the Compliance Gate's prompts?
- **RQ5 (Case-specific analysis):** How does performance vary across different regulatory topics (e.g., GDPR chapters)?

Policy and Benchmark Dataset. We focus on the EU General Data Protection Regulation (GDPR) as our primary regulatory corpus and provide a brief overview in Appendix A. Although many policy texts are publicly accessible, high-quality violation materials are scarce and often encumbered by confidentiality and redistribution restrictions—even for non-commercial research—making it challenging to adopt alternative benchmark datasets. Purely synthetic

narratives risk label drift and undermine external validity. Accordingly, we curate **GCS-300**[1], a semi-synthetic benchmark of 300 scenarios grounded in publicly documented enforcement decisions and official guidance. Each scenario is (i) anchored by citations to its source, (ii) anonymized and minimally abstracted to remove identifying or sensitive details while preserving the legal facts, and (iii) post-screened to remove outliers where anonymization could blur the core lawful basis or violation theory. This pipeline yields reproducible labels without compromising research ethics, but it also makes broad, multi-policy benchmarking challenging in practice. We therefore treat GDPR as a focused, high-fidelity testbed and leave cross-regulation evaluations to future work; the framework itself is policy-agnostic by design.

Evaluation Protocol. We assess performance using a combination of quantitative and qualitative metrics. For quantitative evaluation, we report standard classification metrics, including macro-F1 and micro-F1. To reflect practical utility in human-in-the-loop compliance environments—where minimizing false negatives is critical—we also adopt the **F2-score** ($\beta = 2$) as a key metric, which weighs recall twice as heavily as precision. Because quantitative metrics alone are insufficient to capture the quality of compliance reasoning, we additionally employ an LLM-based rater—following approaches such as [15, 34]—to qualitatively evaluate whether model rationales reflect sound legal reasoning. An ensemble of three strong reasoning models scores alignment to the ground-truth violation articles and lawful basis; scoring details are provided in Appendix E.

Baselines. We structure comparisons along two orthogonal axes: *system design* and *model family*. On the system side, we consider (i) a *raw LLM* setup in which the GDPR text is provided directly in the context window, (ii) a *vanilla RAG* pipeline that retrieves the top-8 most relevant chunks per prompt, and (iii) a *GraphRAG*-style pipeline that builds a document-level graph over the GDPR and retrieves multi-hop node/community summaries under the same retrieval budget for parity. On the model side, we evaluate *GPT-like* closed-weight families and *7–8B* open-weight models [5, 20, 21]. For Lawma-8B [5], which is tuned for multiple-choice prompting, we reformulate the retrieved evidence into a multiple-choice query (candidate articles/options) to match its interface. All baselines share the same decision schema and task prompts (adapted only to input format), and decoding is deterministic (temperature=0.0). Prior work that directly targets our setting—full-scope, article-level regulatory compliance adjudication from unconstrained scenario narratives with explicit violation attribution—remains scarce. Existing regulatory LLMs [4, 5] are primarily designed for legal QA/summarization rather than regulation-wide compliance gating, while GraphRAG is a retrieval/organization framework rather than a task-specific compliance judge. We therefore report comparisons against the closest applicable systems under a unified evaluation protocol.

Experimental Setup. All experiments are conducted on our **GCS-300** benchmark using the GDPR as the policy text. Unless otherwise

---

[1]Due to research-ethics constraints, the GCS-300 dataset itself cannot be released. To support reproducibility and transparency, Appendix D documents the full construction protocol, and we additionally release a limited set of illustrative samples that can be publicly shared.
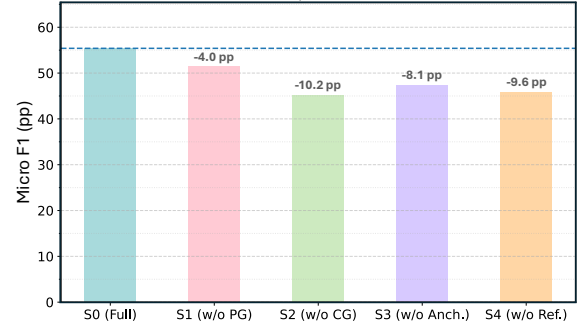

Figure 3: Ablation results on micro-F1. S0 (Full) is shown as both a bar and a dotted baseline for easy comparison. S0: Full model, S2: without Policy Graph, S3: without Context Graph, S4: without Anchoring mechanism, S5: no reference traversal

specified, **GPT-4.1** in a zero-shot setting serves as the default underlying LLM for all RQs to ensure consistency [21]. All graph-based methods utilize the same pre-constructed, single-version Policy and Context Graphs. To ensure a fair comparison, all models and baselines share the same core prompts, with minimal adaptations only to accommodate differences in their input schemas. An exception is made for GPT-5–based systems to account for their distinct, overly conservative response patterns; we uniformly add an emphasis prompt to encourage predicting a violation when the evidence is clear and to minimize free-form reasoning beyond the retrieved evidence.

4.1 RQ1: Accuracy and Robustness

We evaluated the compliance *judgment* accuracy of GRAPHCOMPLIANCE on GCS-300 against Raw LLM, Vanilla RAG (Top-K=8/30), and a GraphRAG-style pipeline, across both general-purpose and domain-aligned LLMs. To ensure validity and fairness, we use a unified decision schema in a *zero-shot* setting with deterministic decoding (temperature=0.0) and identical prompt scaffolding; all graph-based methods share the same pre-generated policy/context graphs. Retrieval budgets are matched, and Top-K is tuned on a held-out validation set; no model-specific prompt tuning is applied. When the GDPR corpus exceeds the context window, the raw-LLM baseline runs in a *multi-turn packing* mode.

Results (Table 1) show that GRAPHCOMPLIANCE achieves clear improvements across all models compared to a strong RAG baseline: on GPT-like models, macro-F1 improves by +2–6 pp, F2 by +12–20 pp (micro/macro), and MCC (Matthews correlation coefficient) by +1–4 pp; on 7–8B open-weight models, macro-F1 rises by +2–4 pp and F2 by +3–4 pp. GraphRAG does not yield a meaningful uplift over RAG on this task; in head-to-head comparisons, GRAPHCOMPLIANCE outperforms GraphRAG by up to +12.8 pp macro-F1 and up to +19.6 pp F2 (and +11.1 pp MCC under Top-K=8 with GPT-4.1). These trends indicate that structuring the problem and constraining reasoning with explicit, typed graph evidence provides more reliable gains than scaling model size or merely graph-summarizing retrieval; the effect holds for domain-aligned LLMs as well. GRAPHCOMPLIANCE shows its largest gains on the F2, which weights recall twice as much as precision ($\beta = 2$). This indicates a substantially

Table 2: Cycle-consistency isomorphism scores for Policy graph and Context graph construction.

	Policy Graph			Context Graph	
# iter.	Semantic	Structural	# iter.	Semantic	Structural
1	0.8749	0.9998	1	0.9132	0.9224
2	0.8703	0.9998	2	0.8927	0.9082
3	0.8687	0.9998	3	0.8886	0.9195
4	0.8691	0.9998	4	0.8742	0.9015
5	0.8691	0.9997	5	0.8516	0.8993

Table 3: Semantic similarity (T_0 vs T_1') averaged over all noise operators.

δ	Mean semantic similarity	95% CI
0.01	0.8706	[0.866, 0.875]
0.03	0.8563	[0.849, 0.863]
0.05	0.8509	[0.841, 0.861]
0.10	0.8231	[0.808, 0.838]
0.20	0.7653	[0.743, 0.788]

lower miss rate on true violations, i.e., the system is less likely to overlook high-risk non-compliance. In human-in-the-loop compliance workflows, such recall-oriented performance is desirable: by reliably surfacing high-probability violations, the framework supports risk-aware triage and shortens downstream audit and remediation cycles.

4.2 RQ2: Submodule Contribution (Ablation Study)

This RQ tests that performance gains arise from the *combination* of modules rather than a single component. We conduct ablations by removing one module at a time and measuring the degradation. To ensure fairness when providing raw text as input, we construct a "dummy graph" with the text as node content, preventing penalties from prompt-schema differences. The variants of GRAPHCOMPLI-ANCE are:

- **S1 (Raw policy):** Replace the Policy Graph with the raw regulatory text, chunked at the *point* level.
- **S2 (Raw context):** Replace the Context Graph with the raw context description, treating the entire text as a single anchor.
- **S3 (E2E on graphs):** Provide the full Policy and Context Graphs as textual input, but without our structured anchoring mechanism.
- **S4 (No reference traversal):** Disable explicit traversal of cross-references between CUs.

As summarized in Figure 3, all proposed components contribute critically to the final performance (further numerical details in Appendix B.2). The largest drop (−10.2 pp) occurs in S2, where the Context Graph is replaced with raw text; recall declines with the loss of the highlighting effect from subgraph-based anchoring,

Table 4: Paraphrase sensitivity on GPT-4.1. Lower F_1 range indicates higher robustness. All values are micro-F1 scores.

Model	Worst	Mean	Best	F_1 range ↓
Raw	32.3	42.1	45.5	13.2
RAG	42.3	46.3	48.1	5.8
GraphCompliance	53.4	54.6	59.9	6.5

which clarifies explicit hypernym information and isolates individual entities/actions. The sizable drop in S4 (−9.6 pp) confirms that, for reference-dependent regulations such as the GDPR, explicit reference linking via graph traversal is highly effective. The smaller drop in S1 reflects that the anchoring effect from the Context Graph remains, while the S3 result suggests that information overload without anchoring harms reasoning. Overall, the superiority of GRAPHCOMPLIANCE stems from synergistic effects—particularly (i) subgraph-based anchoring that clarifies actions and (ii) reference traversal that follows the regulation's logical flow.

4.3 RQ3: Submodule Fidelity

This RQ independently validates the fidelity of the two intermediate representations (the graphs), demonstrating that the final judgment is built on a solid foundation rather than being jeopardized by low-quality intermediate steps. Because a large-scale, gold graph is unavailable, we design proxy evaluations. We first conduct a **reconstruction test** to evaluate information capture in a single pass $T_0 \rightarrow G_0 \rightarrow T_1$, where T_0 is the initial text, G_0 the generated graph, and T_1 the text reconstructed from G_0. We then extend to a **cycle-consistency test** by iterating the transformation ($T_k \rightarrow G_k \rightarrow T_{k+1}$) to check stability/invariance. Information preservation is measured along two dimensions: (1) *Semantic Isomorphism*, the similarity between T_0 and T_k, and (2) *Structural Isomorphism*, a comparison of graph-level statistics between G_0 and G_k. For semantic similarity between two sentence sets A_c and B_c, we use a symmetric max-similarity score s_c:

$$s_c = \frac{1}{2}\left(\mathbb{E}_{a \sim A_c}\left[\max_{b \in B_c} \cos(a, b) \right] + \mathbb{E}_{b \sim B_c}\left[\max_{a \in A_c} \cos(a, b) \right] \right). \quad (7)$$

where $\cos(\cdot, \cdot)$ denotes cosine similarity between unit-normalized sentence embeddings. Finally, we validate that our isomorphism scores are meaningful with a **noise injection test**: we corrupt G_0 to obtain G_0' using a mixture of operators—randomly deleting a fraction δ of edges, adding spurious edges, and altering CU attributes—reconstruct text T_1' from G_0', and measure the drop in semantic similarity relative to T_0.

As shown in Table 2, both graphs exhibit high cycle-consistency. For the policy graph, the semantic score shows negligible degradation, starting at 0.8749 and remaining stable at 0.8691 after five cycles, while the structural score remains near perfect (> 0.9997). The context graph also shows high stability, with semantic similarity stabilizing at 0.8516 after an initial drop from 0.9132. The significance of these scores is corroborated by the noise-injection results in Table 3: injecting just 10% noise ($\delta = 0.10$) lowers the semantic score to 0.8231, notably below the Policy Graph's noise-free
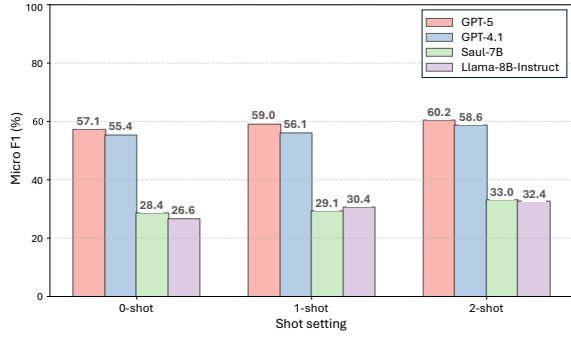
Figure 4: Few-shot dependency across different underlying models, measured in micro-F1 score.

score after five cycles (0.8691). These findings indicate that graph-generation fidelity/robustness is sufficient and does not bottleneck end-to-end performance, supporting the reliability of the gains in RQ1 and RQ2.

4.4 RQ4: Prompt Sensitivity

This RQ validates that high accuracy in RQ1 is not an artifact of a single "golden" prompt. We test robustness to prompt variations in the Compliance Gate from two perspectives. First, to measure syntactic robustness, we paraphrase the core judgment prompt into several semantically equivalent variants and quantify the variability using a F_1 *range*, defined as $F_1^{\text{best}} - F_1^{\text{worst}}$ on micro-F1. Second, to assess the role of in-context learning, we compare zero-shot, one-shot, and two-shot settings while holding the graphs and context fixed.

To this end, we evaluated the prompt sensitivity of the Compliance Gate from two perspectives. First, to measure syntactic robustness, we observed performance variations across several semantically equivalent but syntactically different versions (paraphrases) of the core judgment prompt. To quantify this variance, we measure the difference between the best and worst micro-F1 scores (i.e., $F_1^{\text{best}} - F_1^{\text{worst}}$). This provides a direct and intuitive measure of robustness that can be readily compared across different reasoning frameworks. Second, to assess the impact of in-context learning, we compared the performance of the same judgment task under zero-shot, one-shot, and two-shot settings. This design is valid as it isolates the impact of prompts and examples by keeping the input graphs and context fixed.

Our results indicate that GRAPHCOMPLIANCE is highly robust to prompt variations. As summarized in Table 4, both GraphCompliance and the RAG-aided baseline demonstrated markedly higher stability against prompt paraphrasing compared to the raw LLM baseline. The F_1 range for GraphCompliance was low at 6.5 pp, a level of stability comparable to the robust RAG-aided baseline (5.8 pp). This result supports that the high accuracy of GraphCompliance reported in RQ1 is not an artifact of a single, favorably-tuned prompt, but a robust finding. Meanwhile, the few-shot test results in Figure 4 show that performance consistently improves with the number of shots for all underlying models. This suggests that, in addition to the rich contextual understanding gained from graph

Table 5: Per-chapter performance analysis on GDPR Chapter III and V, comparing Recall (%) and False Positive Rate (FPR, %). Our framework (GraphCompliance) is compared against a Raw LLM baseline across different model scales.

Chapter	Method	Recall (%) ↑	FPR (%) ↓
Ch. V	Ours (GPT-like)	**99.2**	**4.4**
	Baseline (GPT-like)	91.1	52.2
	Ours (7-8B Models)	**84.4**	**28.9**
	Baseline (7-8B Models)	57.1	95.9
Ch. III	Ours (GPT-like)	**97.2**	**37.1**
	Baseline (GPT-like)	77.8	53.6
	Ours (7-8B Models)	**58.3**	**57.1**
	Baseline (7-8B Models)	46.3	92.8

alignment, a small number of examples can serve as a clearer guideline for the final LLM on *how* to perform the compliance judgment itself. In summary, our framework achieves both high accuracy and stability with minimal prompt tuning.

4.5 RQ5: Case-Specific Analysis

This RQ analyzes *where* and *why* our framework excels beyond aggregate metrics by focusing on specific topic clusters. We compare GRAPHCOMPLIANCE to a raw LLM baseline on two representative GDPR chapters: Chapters III (Rights of the Data Subject) and V (International Transfers), across a large model (e.g., GPT-4.1) and 7–8B models (e.g., SaulLM-7B, Llama-3-8B-Instruct). Because a single scenario can trigger multiple articles within a chapter, we treat this as a chapter-level **any-hit classification**: a chapter is positive if any of its articles is correctly identified. Performance is measured using **Recall** and **False Positive Rate (FPR)**; due to the any-hit setting, high recall is expected, making FPR crucial to over-prediction tendencies.

The results, summarized in Table 5, reveal that GRAPHCOMPLI-ANCE consistently achieves higher Recall and substantially lower FPR than the raw LLM baseline across all models and chapters. This advantage is particularly pronounced for **Chapter V**, whose decision-tree-like normative structure is a natural fit for our graph-based representation. Our framework's ability to traverse explicit references results in near-perfect Recall (99.2%) with a very low FPR (4.4%) on large models, whereas the baseline struggles with the complex logic, leading to a high FPR (52.2%). For **Chapter III**, baselines exhibited over-sensitivity to general terms like *'transparency'*, leading to an overly defensive and noisy prediction pattern. In contrast, GRAPHCOMPLIANCE's reliance on specific entity-subject alignment avoids this pitfall. This analysis demonstrates *how* our structured approach improves reasoning: it excels at navigating the explicit logical paths common in regulations and is more robust to the keyword-based distractions that plague text-only models.

5 Conclusions and Outlook

This work proposes GraphCompliance to address the gap between the structural complexity of regulatory texts and the unstructured

nature of real-world contexts. The hybrid framework converts policies and contexts into a Policy Graph and a Context Graph, then aligns them via a Compliance Gate to structurally guide the final LLM-based judgment. Our experiments show that this structured, neuro-symbolic approach significantly improves accuracy, robustness, and fidelity over standard end-to-end baselines, offering a path toward verifiable compliance automation.

Despite these promising results, limitations remain, pointing to important directions for future work. Because the quality of initial graph extraction directly impacts the final judgment, a key challenge is to enhance the automation and robustness of graph construction. We propose two major directions: first, extending the framework to broader regulatory domains such as finance and healthcare to validate its policy-agnostic design; second, designing a more sophisticated agent network to reduce prompt dependence. We believe this work provides a strong foundation for future research into reliable and verifiable neuro-symbolic systems for normative reasoning.

Reproducibility. To preserve double-blind review, we do not include any repository in this submission. Upon acceptance, we will release the source code that reproduce all reported results.

References

[1] 2016. Regulation (EU) 2016/679 (General Data Protection Regulation). Official Journal of the European Union (OJ L 119), 4 May 2016. https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng

[2] Marco Alecci, Nicolas Sannier, Marcello Ceci, Sallam Abualhaija, Jordan Samhi, Domenico Bianculli, Tegawendé F. Bissyandé, and Jacques Klein. 2025. Toward LLM-driven GDPR compliance checking for Android apps. In *Proceedings of the 33rd ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE) Companion*. Association for Computing Machinery, 606–610. doi:10.1145/3696630.3728508

[3] Tara Athan, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner. 2015. LegalRuleML: Design principles and foundations. In *Reasoning Web. Web Logic Rules*. Lecture Notes in Computer Science, Vol. 9203. Springer, 151–188. doi:10.1007/978-3-319-21768-0_6

[4] Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, André F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. 2024. SaulLM-7B: A pioneering large language model for law. *arXiv* (2024). arXiv:2403.03883 [cs.CL] https://arxiv.org/abs/2403.03883

[5] Ricardo Dominguez-Olmedo, Vedant Nanda, Rediet Abebe, Stefan Bechtold, Christoph Engel, Jens Frankenreiter, Krishna Gummadi, Moritz Hardt, and Michael Livermore. 2024. Lawma: The power of specialization for legal tasks. arXiv:2407.16615 [cs.CL] https://arxiv.org/abs/2407.16615

[6] 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* (2024). arXiv:2404.16130 [cs.CL] https://arxiv.org/abs/2404.16130

[7] European Data Protection Board. 2025. International data transfers (SME guide). https://www.edpb.europa.eu/sme-data-protection-guide/international-data-transfers_en

[8] Leon Garza, Lavanya Elluri, Aritran Piplai, Anantaa Kotal, Deepti Gupta, and Anupam Joshi. 2024. PrivComp-KG: Leveraging KG and LLM for Compliance Verification. In *2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*. 97–106. doi:10.1109/TPS-ISA62245.2024.00021

[9] Guido Governatori and Antonino Rotolo. 2010. Norm compliance in business process modeling. In *Semantic Web Rules (RuleML 2010)*. Lecture Notes in Computer Science, Vol. 6403. Springer, 194–209. doi:10.1007/978-3-642-16289-3_17

[10] Shabnam Hassani. 2024. Enhancing legal compliance and regulation analysis with large language models. *arXiv* (2024). arXiv:2404.17522 [cs.AI] https://arxiv.org/abs/2404.17522

[11] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-Retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://proceedings.neurips.cc/paper_files/paper/2024/hash/efaf1c9726648c8ba363a5c927440529-Abstract-Conference.html

[12] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabrina M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge graphs. *Comput. Surveys* 54, 4, Article 71 (2021). doi:10.1145/3447772

[13] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2025. GRAG: Graph retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*. 4145–4157. https://aclanthology.org/2025.findings-naacl.232.pdf

[14] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html

[15] Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the 41st International Conference on Machine Learning (ICML '24) (Proceedings of Machine Learning Research, Vol. 235)*. PMLR, 26396–26415. https://proceedings.mlr.press/v235/lee24c.html

[16] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*, Vol. 33. 9459–9474. https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[17] Mufei Li, Siqi Miao, and Pan Li. 2025. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *International Conference on Learning Representations (ICLR)*. https:

//iclr.cc/virtual/2025/poster/30084 Poster; method: SubgraphRAG.

[18] Siyuan Li, Jian Chen, Rui Yao, Xuming Hu, Peilin Zhou, Weihua Qiu, Simin Zhang, Chucheng Dong, Zhiyao Li, Qipeng Xie, and Zixuan Yuan. 2025. Compliance-to-code: Enhancing financial compliance checking via code generation. *arXiv* (2025). arXiv:2505.19804 [cs.LG] https://arxiv.org/abs/2505.19804

[19] Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, Wenbo Su, and Bo Zheng. 2024. GraphReader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 12758–12786. doi:10.18653/v1/2024.findings-emnlp.746

[20] AI @ Meta Llama Team. 2024. The Llama 3 Herd of Models. *arXiv* (2024). arXiv:2407.21783 [cs.CL] https://arxiv.org/abs/2407.21783

[21] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, et al. 2023. GPT-4 Technical Report. *arXiv* (2023). arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[22] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[23] David Rodríguez, Ian Yang, Jose M. Del Alamo, and Norman Sadeh. 2024. Large language models: A new approach for privacy policy analysis at scale. *Computing* 106 (2024), 3879–3903. doi:10.1007/s00607-024-01331-9

[24] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215. doi:10.1038/s42256-019-0048-x

[25] Pattaraporn Sangaroonsilp, Hoa Khanh Dam, Omar Haggag, and John Grundy. 2024. Interactive GDPR-compliant privacy policy generation for software applications. *arXiv* (2024). arXiv:2410.03069 [cs.SE] https://arxiv.org/abs/2410.03069

[26] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*. doi:10.48550/arXiv.2401.18059

[27] Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, and Wenjie Zhang. 2024. Paths-over-Graph: Knowledge graph empowered large language model reasoning. *arXiv* (2024). arXiv:2410.14211 [cs.AI] https://arxiv.org/abs/2410.14211

[28] Chenhao Tang, Zhengliang Liu, Chong Ma, Zihao Wu, Yiwei Li, Wei Liu, Dajiang Zhu, Quanzheng Li, Xiang Li, Tianming Liu, and Lei Fan. 2023. PolicyGPT:

Automated analysis of privacy policies with large language models. *arXiv* (2023). arXiv:2309.10238 [cs.CL] https://arxiv.org/abs/2309.10238

[29] W3C Data Privacy Vocabularies and Controls Community Group. 2022. DPVO-GDPR: GDPR extension for DPV-OWL. W3C Community Final Specification. https://www.w3.org/community/reports/dpvcg/CG-FINAL-dpv-owl-gdpr-20221205/

[30] Junjie Wang, Mingyang Chen, Binbin Hu, Dan Yang, Ziqi Liu, Yue Shen, Peng Wei, Zhiqiang Zhang, Jinjie Gu, Jun Zhou, Jeff Z. Pan, Wen Zhang, and Huajun Chen. 2024. Learning to plan for retrieval-augmented large language models from knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 7813–7835. doi:10.18653/v1/2024.findings-emnlp.459

[31] Yilin Wen, Zifeng Wang, and Jimeng Sun. 2024. MindMap: Knowledge graph prompting sparks graph of thoughts in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 10370–10388.

[32] Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024. Retrieval-augmented generation with knowledge graphs for customer service question answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. Association for Computing Machinery, Washington, DC, USA, 2905–2909. https://dl.acm.org/doi/10.1145/3626772.3661370

[33] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2369–2380. doi:10.18653/v1/D18-1259

[34] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS '23)* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Article 2020, 29 pages. https://github.com/lm-sys/FastChat/tree/main/fastchat/llm_judge

[35] 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 North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 8912–8924. https://aclanthology.org/2025.naacl-long.449.pdf

A Concise Overview of the GDPR

The General Data Protection Regulation (GDPR; Regulation (EU) 2016/679) is the European Union's comprehensive legal framework governing the processing of personal data of natural persons. The instrument comprises 173 recitals and 99 articles arranged into 11 chapters, with definitions in Article 4 and extensive cross-references that structure interpretation across provisions. At a high level, the GDPR articulates foundational principles for lawful processing, enumerates justifications for processing, confers enforceable rights on data subjects, prescribes organisational and technical obligations for controllers and processors, regulates international transfers, and establishes independent supervision and remedies. The table below summarises the chapter structure and principal subject matter without methodological commentary.

Table 6: GDPR chapters and principal subject matter (articles in parentheses).

Chapter (Arts.)	Principal subject matter
Ch. 1 (1–4)	General provisions; subject matter; material and territorial scope; core definitions (personal data, processing, controller/processor, etc.).
Ch. 2 (5–11)	Principles of processing (lawfulness, fairness, transparency, purpose limitation, minimisation, accuracy, storage limitation, integrity/confidentiality, accountability); lawful bases; consent; special categories; criminal data.
Ch. 3 (12–23)	Rights of the data subject: information, access, rectification, erasure, restriction, portability, objection; safeguards for automated decision-making and profiling.
Ch. 4 (24–43)	Controller and processor obligations: governance, contracts, records, security of processing, breach notification, data protection by design/default, DPIA, DPO, codes and certification.
Ch. 5 (44–50)	Transfers to third countries or international organisations: adequacy decisions, appropriate safeguards (e.g., SCCs, BCRs), derogations, onward transfer conditions.
Ch. 6 (51–59)	Independent supervisory authorities: establishment, tasks and powers.
Ch. 7 (60–76)	Cooperation and consistency mechanism; one-stop-shop; European Data Protection Board (EDPB) opinions and binding decisions.
Ch. 8 (77–84)	Remedies, liability and penalties: complaints, judicial remedies, compensation, administrative fines.
Ch. 9 (85–91)	Specific processing situations: research and statistics, archiving in the public interest, employment, expression and information, national identifiers.
Ch. 10 (92–93)	Delegated and implementing acts.
Ch. 11 (94–99)	Final provisions: relationship with prior law, entry into force and application.

Abbrev.: SCCs = Standard Contractual Clauses; BCRs = Binding Corporate Rules; DPIA = Data Protection Impact Assessment; DPO = Data Protection Officer; EDPB = European Data Protection Board.

B Supplementary Experimental Information

B.1 Implement Detail

All experiments were run on a server equipped with NVIDIA RTX Blackwell generation GPUs, running Ubuntu 22.04 and Python 3.11.9. The underlying LLMs tested include OpenAI's GPT-4o, GPT-4.1, GPT-5, and other publicly available models such as Llama-3-8B-Instruct. For all LLM inferences, we used deterministic decoding with `temperature=0.0` and the `text-embedding-3-large` model for embeddings, and enforced JSON object output where available. The `max_output_token` limit was set to at least 80% of each model's maximum capacity to prevent premature truncation. In cases where a response was truncated by this limit, the partial generation was forcibly parsed into a JSON object for analysis.

B.2 Ablation study results

Table 7: Ablation study results. Performance delta (ΔF1) is in percentage points (pp) against the full model.

Setting	Precision	Recall	F1	ΔF1 (pp)
S0 (Full Model)	46.1	69.4	**55.4**	–
S1 (w/o PG)	45.3	59.5	51.4	-4.0
S2 (w/o CG)	71.4	33.1	45.2	-10.2
S3 (w/o Anchoring)	51.7	43.5	47.3	-8.1
S4 (w/o Ref. Trav.)	41.0	51.8	45.8	-9.6

Note: 'w/o' denotes a component was removed. PG: Policy Graph; CG: Context Graph; Ref. Trav.: Reference traversal.

C Algorithms and Representations

C.1 Policy Graph Construction

Algorithm 1 BuildPolicyGraph

Input: Policy corpus Doc
Output: Policy graph $G_P = (V, E)$
Algorithm:
1: $\text{doc\_json} \leftarrow \textsc{JsonParser}(Doc)$ ▷ Input schema convert
2: $V, E \leftarrow \varnothing$
3: $root \leftarrow \textsc{AddNode}(V, \text{doc\_json}, \text{doc\_json.title})$
4: $preorder \leftarrow [document, chapter, article, point]$
5: **for** each node n in doc_json.preorder **do** ▷ structure pass
6: $\quad k \leftarrow n.\text{type}$
7: $\quad id \leftarrow \textsc{AddNode}(V, k, n.\text{title}|\text{text})$
8: $\quad \textsc{AddEdge}(E, \textsc{contain}, \textsc{parent}(n), id)$
9: $\quad$ **if** $\textsc{IsPremise}(n.\text{title})$ **then**
10: $\quad\quad \textsc{Mark}(id, \text{premise})$
11: $\quad$ **else**
12: $\quad\quad \textsc{Mark}(id, \text{compliance\_unit})$
13: $\quad$ **end if**
14: **end for**
15: $\text{Items} \leftarrow \{ p \mid \textsc{Role}(p) = \text{compliance\_unit} \}$ ▷ collect clauses that are not premises for CU extraction
16: **for** each batch B in $\textsc{Batch}(\text{Items})$ **do**
17: $\quad O \leftarrow \text{LLM.Call}(\text{"cu.extract"})$
18: $\quad \textsc{MatchAndLinkCU}(O, V, E)$ ▷ map (p, CU_list) pairs to CU nodes and link $p \xrightarrow{\text{DERIVES}} cu$
19: **end for**
20: **for** each batch Q in $\textsc{Batch}(\textsc{CUNodes}(V))$ **do**
21: $\quad R \leftarrow \text{LLM.Call}(\text{"cu.reference"})$
22: $\quad \textsc{AttachReferences}(Q, R)$ ▷ add ref field to each CU
23: **end for**
24: **return** (V, E)

Listing 1: Sample compliance unit node from the GDPR Policy Graph

```
{
```

```
"id": "DOC:GDPR/CHAPTER:IV/SECTION:4/ARTICLE:37/POINT:1/CU
    :397313605152",
"kind": "compliance_unit",
"label": "{\"subject\": \"controller and processor\", \"condition
    \": {\"any\": [ ... ]}",
"attrs": {
  "subject": "controller and processor",
  "condition": {
    "any": [
      "processing is carried out by a public authority or body,
    except for courts acting in their judicial capacity",
      "core activities consist of processing operations requiring
    regular and systematic monitoring of data subjects on a large
    scale",
      "core activities consist of processing on a large scale of
    special categories of data (Art. 9) and personal data relating
     to criminal convictions and offences (Art. 10)"
      ]
  },
  "constraint": ["shall designate a data protection officer"],
  "context": null,
  "char_span": {
    "subject": [4, 25],
    "condition": [78, 478],
    "constraint": [26, 70],
    "context": null
  },
  "references": ["A9", "A10"]
},
"type": "actor_cu"
}
```

C.2 Context Graph Construction

Algorithm 2 BuildContextGraph

Input: Context CTX, Policy graph G_P
Output: Context graph G_C
Algorithm:

1: $G_C \leftarrow [\,]$
2: $ER \leftarrow$ LLM.Call("ctx.extract") ▷ ER-triple from CTX
3: $H \leftarrow$ LLM.Call("ctx.hypernym", ER.entity, G_P.premise) ▷ map mentions → policy hypernyms using G_P
4: INJECTHYPERNYMS(ER, H) ▷ attach best hypernym per entity
5: $G_C \leftarrow$ BUILDGRAPH(ER)
6: **return** G_C

Listing 2: Context Graph Sample

```
{
  "entities": [
    {
      "id": "e1",
      "name": "IT operations manager",
      "type": "actor",
      "hypernym": "controller"
    },
    {
      "id": "e5",
      "name": "patient discharge date",
      "type": "data_item",
      "hypernym": "data concerning health"
    },
    ...
  ],

  "relations": [
    {"subj": "e2", "pred": "located_in", "obj": "e3"},
    {"subj": "e4", "pred": "contains",   "obj": "e5"},
    ...
  ]
}
```

```
}
```

C.3 Compliance Gate

Algorithm 3 ComplianceGate

Input: Policy graph G_P, Context graph G_C
Output: Decisions D
Algorithm:

1: $A \leftarrow$ EXTRACTANCHORS(C_G) ▷ units of evaluation from G_C
2: **for** each $a \in A$ **do**
3:     $P \leftarrow$ PRESELECT(G_P, a) ▷ subject-only similarity
4:     $R \leftarrow$ RERANK(P, a) ▷ cross-encoder reranking
5:     $Items \leftarrow$ COMPILEPLANS(R) ▷ compile CU→plan
6:     $J \leftarrow$ LLM.Call("judge") ▷ verdicts for ($a, Items$)
7:     $S \leftarrow \{$ (base $= j$.cu_id, refs $=$ CLOSURE(G_P, base)) $\mid$ $j \in J, j$.verdict $=$ NON_COMPLIANT $\}$ ▷ bidirectional REFERS/DERIVES, unlimited hops
8:     **if** $S \neq \varnothing$ **then**
9:        $O \leftarrow$ LLM.Call("judge.refs") ▷ override
10:        $J \leftarrow$ APPLYOVERRIDES(J, O) ▷ replace verdicts
11:     **end if**
12:     ACCUMULATE(D, J) ▷ store per-CU decisions with scores/why/evidence
13: **end for**
14: $D \leftarrow$ AGGREGATEBYARTICLE(D) ▷ prefer NON_COMPLIANT, else highest score
15: **return** D

D GCS-300 Benchmark Construction and Samples

This subsection describes the construction of the GDPR case-based benchmark and how it is communicated. In compliance with research ethics and source-specific licenses/reuse conditions, we cannot release the full benchmark. Instead, to ensure transparency and enable reproducibility, we first disclose the end-to-end pipeline—collection, normalization, and labeling—in detail. The benchmark further undergoes synthetic rewriting and de-identification to meet research-ethics requirements. All prose and labels are grounded in *first-party legal materials* (e.g., judicial/administrative decisions) *(Labels are grounded in first-party legal materials).*

We rely on *first-party* sources (DPA/court decisions and official notices/press) as the basis for labels, while *second-party portals* (e.g., GDPRhub, Enforcement Tracker) are used solely as discovery indexes. We do not quote their prose; labeling decisions are made from first-party documents.

The construction pipeline is as follows: (1) public-web collection with robots/TOS compliance; (2) normalization and de-duplication; (3) **LLM pre-digest**: we use *GPT-5 Thinking* to condense key facts and candidate GDPR articles into a compact paragraph; (4) **human-in-the-loop review** that focuses on detecting any *omissions of decisive grounds* and amends the summary where necessary; (5) de-identification and synthetic rewriting (removal/replacement of real names and entities); (6) labeling (`violation`, `violation_types`,

articles, lawful_basis, risk_level) with light cross-checks; and (7) documentation (dataset-card style summary).

Regarding representativeness and bias, we took two concrete measures. First, we **maximized the coverage of violated articles** so that a broad range of GDPR provisions appears in the distribution. Second, we **flattened the sampling across time** to reduce temporal skew (e.g., bursts by year or quarter). Any residual limitations (e.g., jurisdictional or sectoral skew) are noted in the dataset card.

The label set is defined concisely as follows. `violation`: scenario-level binary judgment. `violation_types`: concise categories (e.g., `transparency_information`, `international_transfers`). `articles`: GDPR provisions directly linked to the case (e.g., Art. 9, Arts. 44–49). `lawful_basis`: legal bases for processing (e.g., consent, legitimate_interests). `risk_level`: overall risk (e.g., low/medium/high). *Each label is assigned by mapping verifiable facts to articles evidenced in first-party materials.*

From an ethics/legal perspective, the public sample includes **no personal data and no real organisation names**. To reduce re-identification risk, we minimize rare attribute combinations; details that cannot be shared are not included in the sample. Short quotations are used only when necessary, with attribution.

The record below is a *synthetic, de-identified* example that illustrates the schema and labeling principles. While the labels are grounded in first-party materials, the distributed text is adapted and condensed to meet research-ethics requirements.

Listing 3: Synthetic GDPR case context (example record; minimized and masked)

```
{
  "id": "ex001",
  "text": "I'm the IT operations manager at a private hospital in
    city_A. We plan to export from the EHR a weekly file
    containing: patient discharge date, ICD-10 diagnosis codes,
    lab result flags (e.g., HbA1c>7), year of birth, sex, and 5-
    digit postcode, plus a stable pseudonymous patient ID. The
    file will be ingested into our customer data platform to build
    lookalike audiences and to retarget discharged patients on a
    major social platform via advertising integrations. Our
    admission form currently has a single bundled consent ('we may
    use your data for service improvement and offers'); we have
    not collected explicit, separate consent for using health data
    for marketing. Marketing proposes to rely on legitimate
    interests and to continue sending events to US-based ad
    vendors. We have not completed an updated SCC/TIA package for
    these transfers.",
  "facts": {
    "purpose": ["marketing","retargeting"],
    "lawful_basis": ["legitimate_interests"],
    "data_categories": ["health_data","identifiers","contact"],
    "special_categories": ["health"],
    "data_subjects": ["patients"],
    "recipients": ["advertising_vendor","social_media_platform"],
    "international_transfers": ["US"],
    "retention": "365d",
    "role": "controller"
  },
  "jurisdiction": ["EU","<MASK_COUNTRY>"],
  "sector": "healthcare",
  "language": "en",
  "labels": {
    "violation": true,
    "violation_types": [
      "special_category_processing",
```

```
      "purpose_limitation",
      "international_transfers",
      "consent_invalid",
      "transparency_information"
    ],
    "articles": [
      "Art.9(1)",
      "Art.5(1)(b)",
      "Arts.44-49",
      "Art.7",
      "Art.4(11)",
      "Arts.12-14"
    ],
    "risk_level": "high"
  }
}
```

E Computation of Metrics

We report **micro-F1**, **macro-F1**, **micro-F2**, **macro-F2** (with $\beta{=}2$), and **MCC** by directly linking predictions to the dataset labels. Gold labels per scenario come from D `violation.articles`; we frame evaluation as *article-level multi-label classification* (set match between predicted and gold articles for each scenario).

Table 8: Formulas used in this paper (P, R: precision/recall; $\beta{=}2$). MCC is computed once on the flattened article-by-scenario matrix.

Metric	Formula
micro-F1	$F_{1,\mu} = \dfrac{2P_\mu R_\mu}{P_\mu + R_\mu}$
micro-F2	$F_{2,\mu} = \dfrac{(1+\beta^2)P_\mu R_\mu}{\beta^2 P_\mu + R_\mu},\ \beta{=}2$
macro-F1	$F_{1,\text{macro}} = \dfrac{1}{\|\mathcal{A}\|}\sum_{a\in\mathcal{A}} F_{1,a}$
macro-F2	$F_{2,\text{macro}} = \dfrac{1}{\|\mathcal{A}\|}\sum_{a\in\mathcal{A}} F_{2,a}$
MCC†	$\dfrac{TP\cdot TN - FP\cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

†Computed once on the binary article-by-scenario matrix (after article-level linking).

Interpretation (what the scores mean).
- **micro-F1**: How precisely and completely the system *predicts* frequent GDPR articles in practice.
- **macro-F1**: Whether the system also handles *rare (long-tail)* articles rather than only common ones.
- **F2** ($\beta{=}2$): Higher micro-/macro-F2 means the system is tuned to *avoid missing severe violations* (recall priority), accepting some extra false positives.
- **MCC**: Overall *balanced performance* on violation and non-violation labels under label imbalance—i.e., strong correlation with ground truth without positive/negative skew.

Note on scale mismatch. Real cases may report points/paragraphs, whereas we score at the article level; in practice, mismatches that still map to the same parent article are rare. Any residual nuance is further checked in the qualitative *LLM Rater*.