De Jure: Iterative LLM Self-Refinement for Structured Extraction of Regulatory Rules

Anonymous ACL submission

Abstract

Regulatory documents encode legally binding obligations that LLM-based systems must respect. Yet converting dense, hierarchically structured legal text into machine-readable rules remains costly and expert-intensive. We present **De Jure**, a fully automated, domain-agnostic pipeline for extracting structured regulatory rules from raw documents, requiring no human annotation, domain-specific prompting, or annotated gold data. De Jure operates through four sequential stages: normalization into structured Markdown; LLM-driven semantic decomposition into structured rule units; multi-criteria LLM-as-a-judge evaluation across 19 dimensions spanning metadata, definitions, and rule semantics; and iterative repair of low-scoring extractions within a bounded regeneration budget, where upstream components are repaired before rule units are evaluated to maximize definitional accuracy. We evaluate De Jure across four models on three regulatory corpora spanning finance, healthcare, and AI governance. On healthcare, the pipeline yields monotonic improvement in extraction quality, reaching peak performance within at most three judge-guided iterations, and generalizes effectively to the structurally distinct finance and AI governance domains across both open- and closed-source models. In a downstream QA evaluation, responses grounded in De Jure extractions are preferred by a judge LLM over prior work in 57.2% of cases. These results demonstrate that explicit, interpretable evaluation criteria can substitute for human annotation in complex regulatory domains, offering a scalable and auditable path toward regulation-grounded LLM alignment.

1 Introduction

The alignment of large language models (LLMs) with human values has become a central challenge in AI research (Ouyang et al., 2022; Bai et al., 2022b). Yet as LLMs are increasingly deployed in high-stakes domains such as finance (Wu et al., 2023; Yang et al., 2023), healthcare (Singhal et al., 2023), and law (Chalkidis et al., 2020; Nay, 2023), alignment extends beyond general notions of helpfulness and harmlessness to domain-specific regulatory obligations and compliance requirements encoded in legal and policy text. Frameworks such as HIPAA (U.S. Department of Health and Human Services, 2003), the SEC ADVISERS ACT (U.S. Securities and Exchange Commission, 1940), and the EU AI ACT (European Union, 2024) encode legally binding requirements that LLM-based systems must respect, yet transforming these dense, hierarchically structured texts into machine-readable rules remains a manual, expert-intensive bottleneck.

Constitutional AI (CAI) (Bai et al., 2022a) and its extensions (Lee et al., 2023; Sun et al., 2023) show that LLMs can derive alignment principles from plain-language policy text, reducing reliance on human annotation. However, regulatory alignment is governed by structurally complex, exception-laden obligations that are difficult to capture with high-level natural language principles alone. More targeted work has extracted deontic logic formulae from regulations (Governatori et al., 2016) and applied multi-stage pipelines to regulatory text (Sleimi et al., 2018). Existing approaches nevertheless face important limitations: hand-curated seed principles require significant domain expertise (Bai et al., 2022a), coarse-grained rules fail to capture the finer semantic structure of legal corpora (Sun et al., 2023), and repair mechanisms often ignore hierarchical dependencies between section metadata, definitions, and rule units. The closest prior work (Datla et al., 2025) extracts governance principles from HIPAA and the EU AI ACT via few-shot prompting with judge-and-repair steps, but requires costly expert annotation, produces a flat rule representation, and treats repair as a single undifferentiated step.

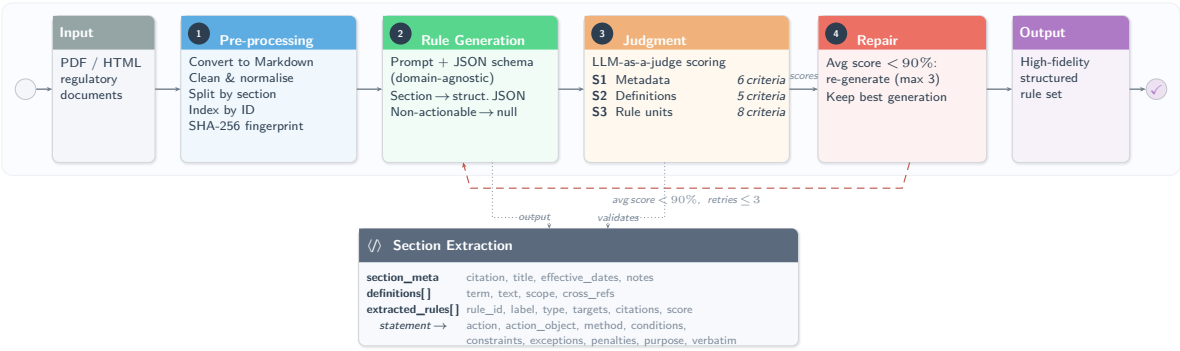

Figure 1: De Jure pipeline overview. Input documents are pre-processed into structured Markdown (Stage 1), parsed into JSON rule units via a domain-agnostic LLM prompt (Stage 2), scored by an LLM judge across 19 criteria in three stages (Stage 3), and iteratively repaired until the per-stage average score reaches 90% or the retry budget (max 3) is exhausted (Stage 4).

We propose **De Jure** (**D**ocument **E**xtraction with **Ju**dge-**R**efined **E**valuation) (Figure 1), a fully automated, domain-agnostic pipeline that transforms raw regulatory documents into structured rule sets requiring no human annotation and no domain-specific prompts. De Jure normalizes input into structured Markdown, extracts rule units conforming to a schema that decomposes each rule into actions, conditions, constraints, exceptions, penalties, and verbatim source spans, and scores each extraction via a multi-criteria LLM-as-a-judge (Zheng et al., 2023) across 19 dimensions (summarized in Appendix B). A key design principle is *hierarchical decoupling*: metadata and definitions are verified and repaired before rule units are evaluated, ensuring rule-level repair always operates on reliable context. Stages below a 90% quality threshold undergo targeted regeneration, retaining the highest-scoring output within a bounded retry budget. Our main contributions are:

- **De Jure pipeline and extraction schema.** A four-stage, domain-agnostic pipeline for end-to-end regulatory rule extraction requiring no annotated data, domain-specific prompts, or logical formalisms. The pipeline is anchored by a structured schema that decomposes each rule unit into rich semantic fields generalizable across regulatory corpora without modification (Section 3).

- **Multi-stage LLM judge with hierarchical repair.** A 19-criterion evaluation framework applied in hierarchical dependency order so each stage repairs on verified context before the next begins, with targeted regeneration retaining the highest-scoring output within a bounded compute budget (Section 3.3).

- **Cross-domain generalization.** With no changes to prompts, schema, or model configuration, De Jure generalizes across HIPAA (healthcare), the SEC ADVISERS ACT (finance), and the EU AI ACT (AI governance), achieving strong performance across domains with differing structural complexity (Section 4.3).

- **Downstream and ablation evaluation.** A RAG-based QA comparison against Datla et al. (2025) on HIPAA shows De Jure-grounded responses are preferred in 57.2% of cases, confirming that extraction quality gains translate to downstream utility. Ablation studies validate each key design choice (Section 4.4, Appendix A).

2 Related Work

De Jure sits at the intersection of regulatory NLP, structured information extraction, and LLM-based self-refinement.

2.1 Rule Extraction from Regulatory Text

Converting dense regulatory text into machine-readable representations is a long-standing problem in legal informatics, healthcare, and finance. Early work used syntactic parsing and logic-based formalisms for normative-statement identification and compliance checking (Dragoni et al., 2016; Wyner and Peters, 2011; Governatori, 2024), alongside information-extraction methods for financial

regulations (Lam et al., 2021) and clinical guidelines (Weng et al., 2010). More recent work applies LLMs across domains, including traffic regulations (Zin et al., 2025), clinical decision support (He et al., 2024; Tang et al., 2026), and contract understanding (Koreeda and Manning, 2021; Hendrycks et al., 2021). These approaches typically require labeled data, operate within a single domain, or produce extractions too coarse to capture the conditional, exception-laden structure of regulatory obligations; De Jure addresses all these limitations through specific design choices.

2.2 Deontic Logic and Defeasible Rule Formalisms

A dominant paradigm for making rules actionable is *deontic logic* (von Wright, 1951), extended by *defeasible deontic logic* (DDL) for reasoning under exceptions and conflicts (Governatori and Rotolo, 2023) and applied at scale to telecommunications regulation via tailored prompts and fine-tuning (Horner et al., 2025). Such methods are inherently domain-specific: the target formalism must be fixed in advance and does not transfer to new domains without substantial re-engineering.

2.3 LLM pipelines for structured principle extraction and iterative refinement

CLAUDETTE (Lippi et al., 2019) and OPP-115 (Wilson et al., 2016) established that criterion-level analysis of policy text is both feasible and valuable. Most closely related, Datla et al. (2025) extract governance principles from HIPAA and the EU AI ACT via chunking, clause mining, and structured LLM extraction with judge-and-repair steps evaluated against a human-annotated gold set. Separately, iterative self-refinement (Madaan et al., 2023), verbal reinforcement (Shinn et al., 2023), and Constitutional AI (Bai et al., 2022a) show that LLM-generated critiques can improve outputs without annotation, and decomposed LLM-as-a-judge evaluation correlates more closely with human judgment than holistic scoring (Zheng et al., 2023; Liu et al., 2023a).

Positioning De Jure. Prior work differs from De Jure in three respects. DDL methods (Horner et al., 2025) are constrained to a logical formalism and do not transfer easily across domains; Datla et al. (2025) requires costly human annotation and applies repair as a single step that ignores structural dependencies; and general-purpose refinement (Madaan et al., 2023; Shinn et al., 2023) provides no structured quality criteria and is not designed for regulatory text. De Jure requires no formal specification and no human annotation, applies repair hierarchically so that foundational components are corrected before rule units are evaluated, and uses criterion-driven, field-level judgment for scalable quality control across heterogeneous regulatory corpora.

3 De Jure

We present **De Jure** (**D**ocument **E**xtraction with **Ju**dge-**R**efined **E**valuation), an automated, domain-agnostic pipeline for transforming raw regulatory documents into structured, machine-readable rule sets (Figure 1). Unlike prior works that rely on human-annotated gold sets (Sleimi et al., 2018) or domain-specific logical formalisms (Governatori et al., 2016), De Jure requires no labelled data and no domain expertise beyond the source document itself. A full procedural specification of the pipeline is provided in Appendix D (Algorithm 1), and a concrete end-to-end example of De Jure in action is given in Appendix C (Figure 3).

The pipeline consists of four stages: (1) document *pre-processing* into normalized, section-segmented Markdown; (2) LLM-driven *rule generation* into a typed JSON schema capturing a rich set of semantic fields per rule unit; (3) *multi-criteria judgment* by an LLM judge across 19 dimensions (summarized in Appendix B) organized in three sequential validation stages; and (4) *selective repair by regeneration*, where any stage scoring below a quality threshold is retried and the highest-scoring output is retained. Two core principles drive the design of the De Jure pipeline. First, *decoupling of extraction from verification*, inspired by prior work (Bai et al., 2022a), separates what is generated from how it is evaluated. Second, *hierarchical repair ordering*, which extends existing literature, ensures the three judgment stages are applied in dependency order, so that foundational components are verified and repaired before rule units are evaluated. This ensures that rule-level repair always operates on reliable context. Together, these drivers of performance strengthen our pipeline's robustness, and enable the same judgment-repair loop to apply uniformly across document types, regulatory regimes, and LLMs without modification to the core pipeline.

3.1 Pre-processing

The pre-processing stage (Figure 1, Stage 1) converts heterogeneous regulatory source formats into structured section–content pairs suitable for downstream LLM processing. Each input document is converted to Markdown using `Docling` (Auer et al., 2024), which preserves section boundaries, list structure, and table formatting, then cleaned to remove formatting artifacts and extraneous whitespace. The clean text is segmented by splitting on regulatory section delimiters (e.g., "§", "Article", "Rule"), yielding an ordered set $\mathcal{S}$ of section–content pairs, each indexed by its regulatory identifier, assigned a SHA-256 fingerprint for traceability, and stored alongside standardized metadata fields (title, version, effective dates). This deterministic representation ensures every downstream extraction traces back to an exact source span, a hard requirement for regulatory auditability.

3.2 Rule Generation

Each section $s \in \mathcal{S}$ is independently submitted to the rule generation stage (Figure 1, Stage 2), where an LLM is prompted to produce a structured JSON extraction conforming to our `Section Extraction` schema (Figure 1). The schema decomposes each section into three typed components: (i) *section metadata* (citation, title, effective dates, notes); (ii) *definitions* (term, text, scope, cross-references); and (iii) *rule units*, each carrying an identifier, rule type, summary label, citation, and a nine-field statement decomposition: action, action object, method, conditions, constraints, exceptions, penalties, purpose, and verbatim source span. This fine-grained decomposition enables downstream systems such as constitutional AI frameworks (Bai et al., 2022a) or compliance verification engines to query and enforce individual semantic components rather than treating rules as monolithic strings. The generation prompt is schema-driven with no domain-specific examples or seed rules, and returns `null` for non-actionable sections, suppressing non-normative passages such as preambles and cross-reference tables to prevent rule set inflation. Full prompt and schema are provided in Appendix H and Appendix I, respectively.

3.3 Multi-Criteria Judgment

Since De Jure operates in an unsupervised setting with no reference annotations, quality control cannot rely on surface-form matching against a gold standard. Instead, we adopt an LLM-as-a-judge framework (Zheng et al., 2023; Liu et al., 2023b; Zhong et al., 2022) organized into *three sequential validation stages* mirroring the schema hierarchy: section metadata is judged and repaired first, followed by definitions, then rule units. This ordering ensures each stage receives the benefit of all prior corrections. By the time rule units are evaluated, both the section metadata and the definitional vocabulary have already been refined, maximizing the contextual accuracy available to the rule repair stage and increasing the likelihood of producing high-fidelity rule decompositions.

Stage 1: Metadata validation (6 criteria). We validate section-level metadata including headings, effective dates, citation strings, and optional contextual fields across six criteria: completeness, fidelity to source, non-hallucination, title quality, citation and date precision, and optional field population. Accurate metadata is foundational: it anchors every downstream extraction to a traceable regulatory provision, and errors such as incorrect effective dates or misattributed citations may otherwise propagate silently into all derived rule units.

Stage 2: Definition validation (5 criteria). We validate all extracted definitions, covering both affirmative ("X means Y") and exclusionary ("X does not include Y") forms. Where a definitional span also encodes an operative rule, it must appear in both the definitions and rule-unit components and is evaluated independently in each. Criteria include completeness, source fidelity, non-hallucination, precision and formatting, and term quality. Non-hallucination is particularly critical here: a fabricated or paraphrased definition silently corrupts the interpretation of every rule referencing that term, making this the highest-leverage point for factual grounding in the pipeline.

Stage 3: Rule-unit validation (8 criteria). We validate each rule unit at the field level across criteria spanning the full statement decomposition: completeness, label conciseness, rule-type classification accuracy, fidelity to source, neutrality, target consistency, actionability, and non-hallucination. Several criteria warrant a brief motivation. *Rule-type accuracy* is essential because misclassifying an obligation as a permission silently inverts compliance semantics. *Neutrality* prevents the model from injecting interpretive framing that would bias downstream enforcement. *Actionability* ensures each rule is expressed in a form that a compliance system can operationalize. *Label conciseness*

matters because summary labels serve as retrieval keys in downstream question answering RAG systems, where verbose or ambiguous labels directly degrade retrieval precision. This is the most demanding stage: a single source section may yield multiple rule units, each required to satisfy all criteria independently.

Each criterion is scored on a 0–5 scale by the judge LLM, which also produces a natural-language justification for each score. The full set of criteria is summarized in Table 8, Appendix B.

3.4 Selective Repair by Regeneration

Low judgment scores indicate specific, localizable defects such as a misclassified rule type, an incomplete label, a hallucinated condition rather than wholesale generation failure. De Jure exploits this through a *selective repair* mechanism: each stage is repaired independently if and only if its average score falls below θ (acceptance threshold). The LLM is re-prompted with the original section text, the current extraction, per-criterion scores, and the judge's natural-language critiques, and asked to correct only the deficient fields. This repeats for up to r attempts per stage, retaining the best-scoring output across all attempts, guaranteeing monotonically non-decreasing quality.

This design has three practical consequences. *First*, repair cost is bounded, with at most r additional LLM calls per stage per section. *Second*, structured scores and critiques make the repair signal substantially richer than a generic re-prompt, empirically producing targeted field-level corrections rather than wholesale rewrites. *Third*, retaining the best rather than the final generation provides a soft safety net: when re-generation degrades quality, the pipeline falls back gracefully to the best previously seen output.

4 Experiments and Results

We evaluate De Jure across four backbone extraction and repair models with a separate fixed judge model, on three regulatory corpora, assessing extraction quality, cross-domain generalization, downstream utility, and core design decisions through ablation studies.

Pipeline Configuration. Extraction and judgment are performed by strictly separate models. Extraction is performed by the backbone model under evaluation while judgment is performed by a fixed model, CompassJudger-2 (Zhang et al., 2025), an open-weight judge model optimized for structured, criterion-aligned evaluation across diverse domains. CompassJudger-2 has been validated on established judge benchmarks including JudgerBenchV2 (Zhang et al., 2025) and RewardBench (Lambert et al., 2024). Three judges operate sequentially, each instantiated from CompassJudger-2 with stage-specific criteria targeting the failure modes of its semantic layer: **Judge 1** evaluates section metadata quality (6 criteria), **Judge 2** assesses extracted definitions (5 criteria), and **Judge 3** evaluates the correctness and completeness of extracted rules (8 criteria). Each produces per-criterion scores and natural-language critiques. If the average normalized score falls below $\theta = 0.9$, the backbone model is re-prompted with the critique, the prior extraction, and the original input, for at most r=3 retries per judge, retaining the highest-scoring output across all attempts.

Inference Settings. All backbone models use identical hyperparameters: temperature $\tau = 0.1$, maximum output tokens $T_{\max} = 4096$, and nucleus sampling $p = 0.95$, encouraging deterministic, faithful extraction.

4.1 Datasets

We evaluate on three regulatory corpora spanning distinct high-stakes domains (Table 1), selected to probe generalization across diverse legal styles and rule structures rather than in-domain memorization.

HIPAA Privacy Rule. The *HIPAA Privacy Rule* (U.S. Department of Health and Human Services, 2003) governs the use and disclosure of protected health information in the United States. Its exception-laden permission structures, where broad prohibitions are qualified by narrow condition-dependent exemptions, test a pipeline's ability to faithfully decompose rule scope.

SEC Investment Advisers Act. The *Investment Advisers Act* (U.S. Securities and Exchange Commission, 1940) governs investment adviser conduct in U.S. financial markets. Its densely nested conditionals and extensive cross-references make it a demanding primary benchmark.

EU Artificial Intelligence Act. The *Regulation (EU) 2024/1689* (European Union, 2024) is the world's first comprehensive AI governance framework. Its mix of binding technical obligations, broad governance principles, and non-binding

Dataset	Domain	Juris.	Sec.	Density
HIPAA Privacy Rule	Healthcare	US	∼30	High
SEC Advisers Act	Fin. securities	US	∼50	High
EU AI Act	AI governance	EU	113	Med–High

Table 1: Summary of evaluation datasets. Density indicates the concentration of extractable regulatory content available per section, directly relating to how many rules can be extracted from each section.

Model	J1 (Metadata)	J2 (Definitions)	J3 (Per-Rule)	Overall
Llama	4.93 ± 0.26	4.45 ± 0.54	4.64 ± 0.50	4.68 ± 0.33
Qwen	4.92 ± 0.34	$\mathbf{4.72 \pm 0.55}$	4.66 ± 0.49	4.76 ± 0.34
Claude	4.93 ± 0.28	4.66 ± 0.59	$\mathbf{4.75 \pm 0.43}$	4.78 ± 0.30
GPT-5	$\mathbf{4.95 \pm 0.23}$	4.65 ± 0.57	$\mathbf{4.75 \pm 0.43}$	$\mathbf{4.79 \pm 0.28}$

Table 2: Averaged judge (J) scores on the HIPAA corpus (scale 1–5; higher is better). Values are reported as mean ± std across 3 runs for each judge and for the overall aggregate. Bold values indicate highest scores at each stage.

recitals presents a structurally distinct extraction challenge.

The three corpora span a natural complexity gradient: SEC provisions are rigidly alphanumerically indexed with well-delimited boundaries; HIPAA organizes mandates around exception-laden permission hierarchies with less consistent demarcation; and the EU AI ACT employs discursive, principle-based prose that interleaves binding obligations with non-binding recitals. Ordered from most to least structurally regular, they constitute an *a priori* difficulty ranking for extraction.

4.2 Experiment 1: Extraction Quality

Experimental Setup. We evaluate on the HIPAA corpus using four backbone models spanning open- and closed-source families: **Llama-3.1-8B-Instruct** (Dubey et al., 2024) and **Qwen3-VL-8B-Instruct** (Qwen Team, 2025) as cost-efficient, controllable open-source options, and **Claude-4.6-Sonnet** (Anthropic, 2026) and **GPT-5-mini** (OpenAI, 2025) as high-capability closed-source systems. This selection enables a controlled comparison across model families and access paradigms under identical pipeline conditions. For brevity, we refer to Llama-3.1-8B-Instruct, Qwen3-VL-8B-Instruct, Claude-4.6-Sonnet, and GPT-5-mini as Llama, Qwen, Claude, and GPT-5, respectively, throughout the remainder of the paper.

Results. Table 2 reports the averaged score across all judge criteria for each model and judge stage, computed over three runs and presented as mean ± standard deviation. Two observations stand out. First, performance degrades from metadata (4.95) to the more semantically demanding definition and per-rule stages ($4.72 - 4.75$), reflecting increasing task complexity: section-level metadata is structurally salient and consistently recoverable, whereas decomposing fine-grained rule components such as conditions, exceptions, and penalties demands substantially deeper semantic parsing. Second, model differences remain modest across all stages. While **GPT-5** achieves the highest overall score (4.79), other models perform comparably ($4.68 - 4.78$), with overlapping variance indicating limited separation. The divergence is most pronounced at Judge 2 (J2), where Qwen leads (4.72) over Llama (4.44), driven by superior Term Quality (4.72) and Precision (4.64) on definition decomposition (Appendix E). Notably, Qwen achieves the highest J2 average, outperforming both closed-source models on definition extraction. The narrow overall gap between the best open-source model (Qwen, 4.76) and the best closed-source model (GPT-5, 4.79) suggests that capable open-source models, when paired with structured extraction and iterative refinement, can approach proprietary model performance, a practically significant finding for privacy-sensitive regulatory deployments where reliance on external APIs is undesirable. Further, the per-criterion breakdowns in Appendix E reveal that Non-Hallucination is uniformly near-perfect (5.00) across all models and all three judges, confirming that schema-constrained extraction eliminates factual fabrication as a failure mode regardless of model family.

4.3 Experiment 2: Generalization Across Regulatory Domains

Experimental Setup. Does De Jure generalize, *without any modification*, across structurally diverse regulatory domains? We evaluate on all three corpora from Section 4.1, using **GPT-5** (best closed-source) and **Qwen** (best open-source) from Experiment 1, with all settings unchanged.

Results. Table 3 shows overall scores above 4.70 across every domain–model combination. Sustaining strong performance across three structurally distinct regulatory domains, with no domain-specific adaptation, is strong evidence of broad generalizability. The monotonic decline from SEC (4.85) to HIPAA (4.79) to EU AI Act (4.71) mirrors the *a priori* structural-complexity ordering established

Dataset	Model	J1 (Metadata)	J2 (Definitions)	J3 (Per-Rule)	Overall
SEC	GPT-5	$\mathbf{4.96 \pm 0.02}$	4.75 ± 0.07	$\mathbf{4.74 \pm 0.01}$	$\mathbf{4.85 \pm 0.02}$
	Qwen	4.92 ± 0.01	$\mathbf{4.87 \pm 0.02}$	4.65 ± 0.01	4.82 ± 0.01
HIPAA	GPT-5	$\mathbf{4.95 \pm 0.23}$	4.65 ± 0.57	$\mathbf{4.75 \pm 0.43}$	$\mathbf{4.79 \pm 0.28}$
	Qwen	4.92 ± 0.34	$\mathbf{4.72 \pm 0.55}$	4.66 ± 0.49	4.76 ± 0.34
EU AI Act	GPT-5	$\mathbf{4.69 \pm 0.05}$	4.68 ± 0.03	$\mathbf{4.70 \pm 0.01}$	$\mathbf{4.71 \pm 0.02}$
	Qwen	4.63 ± 0.03	$\mathbf{4.82 \pm 0.01}$	4.66 ± 0.02	$\mathbf{4.71 \pm 0.01}$

Table 3: Generalization across three regulatory corpora (scale 1–5; higher is better). De Jure is applied without domain-specific modification. Values are reported as mean $\pm$ std for each judge and for the overall aggregate. Per-criterion breakdown details are provided in Appx. F (Tables 12–14). Bold values indicate the highest pairwise scores corresponding to each dataset, under each stage of judgement.

independently in Section 4.1: SEC text is rigidly indexed, HIPAA is more loosely organized, and the EU AI Act is the most discursive. Rather than rewarding all outputs uniformly, the judge reflects underlying extraction difficulty. This behavior is illustrated qualitatively in Appendix C, where the judge assigns a failing normalized score of 0.55 to a semantically deficient extraction with targeted per-criterion feedback, then increases the score to 0.9 after repair while leaving already-correct fields unchanged.

The modest cross-domain gap is concentrated in Judge 1. For both models, J1 scores are high on SEC and HIPAA ($4.92 - 4.96$) and drop on the EU AI Act ($4.63 - 4.69$), whereas J2 and J3 remain stable across datasets. The per-criterion breakdown (Appendix F) attributes this decline to completeness, citation/date extraction, and optional-field population, while fidelity, title quality, and non-hallucination remain near 5.0, suggesting that cross-domain transfer is limited less by factual grounding than by metadata recovery under unfamiliar document organization. Model differences are stage-specific rather than uniform: Qwen consistently leads on J2 (definition-level alignment), while GPT-5 leads on J1 and is slightly stronger on J3 (metadata completion and per-rule synthesis), suggesting a genuine tradeoff in model strengths.

J3 remains the hardest stage overall. Even with neutrality and non-hallucination scoring uniformly perfect and accuracy remaining very high, the lowest per-criterion scores fall in completeness, fidelity, and actionability. The remaining bottleneck is therefore not hallucination but preserving every conditional trigger, quantitative threshold, and nested exception while still producing concise, usable rule statements, leaving fine-grained rule decomposition as the primary target for future work.

4.4 Experiment 3: Downstream Evaluation, Compliance QA via RAG

Intrinsic extraction quality does not fully capture practical utility. We therefore evaluate extracted rules as the knowledge base of a RAG system tasked with answering HIPAA compliance questions, providing a direct task-grounded comparison against Datla et al. (2025).

Experimental Setup. We restrict evaluation to HIPAA sections covered by the publicly released extractions of Datla et al. (2025),[1] ensuring any performance gap is attributable solely to rule quality rather than coverage. Our rules are extracted using GPT-5. We prompt Qwen to generate 100 evaluation questions from the same HIPAA sections at temperature τ=0.8 (Further details in Appendices H.4.1 and J), yielding a lexically and semantically diverse set in which every question has a verifiable, grounded answer.

We compare six systems: (1) **De Jure**, our full pipeline with hierarchical extraction and iterative repair (r=3); (2) **Baseline**, the extraction of Datla et al. (2025); (3) **De Jure (no repair)**, our pipeline with r=0; (4) **De Jure (simplified)**, using a basic, flat schema; (5) **De Jure (non-hierarchical)**, using non-hierarchical generation; and (6) an **Oracle** that receives the full regulatory document as context without retrieval, serving as an upper bound. For each RAG system, an independent vector database is constructed from its extracted rules, encoded with `all-mpnet-base-v2` (Reimers and Gurevych, 2019), with retrieval depth k=10. Answers are generated using Claude 4.6 Sonnet. The LLM judge (CompassJudger) operating at τ=0.1 ranks all six answers simultaneously across five criteria: Completeness, Factual Grounding, Handling Ambiguity, Practical Actionability, and Regulatory Precision (Appx. B.1 provides further details). To eliminate positional bias, the order of answers is randomized for each question. Pairwise win rates are derived from the rankings; per-criterion top-1 rates are reported in Appendix G.

Results. Table 4 reports pairwise win rates derived from the 6-way ranking evaluation. The Oracle, which receives the full regulatory document, establishes an upper bound, achieving a winrate of

[1] https://github.com/gautamvarmadatla/Policy-Tests-P2T-for-operationalizing-AI-governance/blob/90059a7d2b59d705a80d212b3a1a8adfb30ef58b/Annotator%20Data/Annotator%20Docs/out/HIPAA/HIPAA_removed.extracted.jsonl

	De Jure	Baseline	No repair	Simplified	Non-Hier.	Oracle
De Jure	–	**57.2**	**60.9**	50.3	**56.4**	39.7
Baseline	42.8	–	**52.5**	44.8	50.7	32.8
No repair	39.1	47.5	–	43.6	45.2	26.5
Simplified	49.7	**55.2**	**56.4**	–	50.9	33.4
Non-hier.	43.6	49.3	**54.8**	49.1	–	33.2
Oracle	**60.3**	**67.2**	**73.5**	**66.6**	**66.8**	–

Table 4: Pairwise win rates (%) on the HIPAA downstream QA task (row system beats column system). Rates are derived from 6-way rankings across 5 evaluation criteria and 100 questions (k=10). Values above 50% are bolded.

60–73% and confirming that retrieval introduces an information bottleneck (Yu et al., 2024). Among the RAG systems, De Jure achieves the highest win rates, defeating the Baseline at 57.2%, the no-repair ablation at 60.9%, and the non-hierarchical variant at 56.4%. The large win margin over no-repair demonstrates that iterative repair contributes meaningfully to rule quality, while the margin over the Baseline confirms that our structured decomposition preserves conditional logic, scope qualifiers, and exception clauses that the flatter representations of Datla et al. (2025) omit.

De Jure and the simplified schema variant are nearly tied at 50.3%, suggesting that the richer hierarchical schema primarily benefits intrinsic quality metrics (Table 15) rather than single-rule retrieval at k=10. The clear ordering of systems validates that both iterative repair and structured extraction contribute to downstream utility, with repair being the larger factor. Notably, De Jure achieves this performance using an average 1,865 tokens of retrieved context per query, a 6.2× compression over the Oracle's 11,586 tokens (Appendix G.3), demonstrating that structured extraction retains decision-relevant content while the in-context approach becomes impractical as regulatory documents grow in size. Per-criterion top-1 rates, showing how often each system produces the highest-ranked answer, are reported in Appendix G.

5 Ablation Studies

We conduct four targeted ablations on core design decisions (Appendix A). First, extraction quality (at each stage and overall) improves monotonically with acceptance threshold θ, with the largest gains concentrated in Step 2 (definitions, +0.30 points, that is 6.0%, from $\theta = 0.6$ to 0.9), confirming $\theta = 0.9$ as the strongest configuration within the evaluated range. Second, the retry budget exhibits a striking non-linearity: a single retry yields negligible improvement, while the qualitative shift occurs at $r = 2$, where Step 2 recovers 1.25 points (25.0%) as the candidate pool becomes large enough for the best-of-r selection mechanism to escape low-quality basins; gains saturate beyond $r = 2$, and we adopt $r = 3$ as a conservative default. Third, our section-aware chunking strategy outperforms that of Datla et al. (2025) by +0.16 points (3.2%) overall, with the benefit concentrated entirely in early pipeline stages (+0.33 points, i.e., 6.6%, at Step 1), confirming that downstream refinement cannot compensate for incoherent inputs. Fourth, conditional on an adequate retry budget, the choice of regeneration trigger (average-score vs. per-criterion) has no measurable effect on final quality, establishing the retry budget as the dominant control variable and trigger granularity as second-order.

6 Conclusion

We presented De Jure, a fully automated, domain-agnostic pipeline that converts raw regulatory documents into structured, machine-readable rule sets by interleaving typed semantic extraction with a hierarchically-ordered multi-stage LLM judge and iterative repair. The central design decision is sequential judgment ordering: metadata and definitions are corrected before rule units are evaluated, so each stage operates on verified upstream context and errors do not propagate silently into rule decompositions. De Jure achieves strong extraction quality on healthcare privacy regulations and generalizes without modification to financial securities and AI governance regulations, maintaining strong performance across model families and access paradigms, evidence that the gains are architectural rather than model-dependent. Crucially, a downstream RAG evaluation shows that De Jure-grounded responses are consistently preferred by a held-out LLM judge over a strong prior approach, confirming that extraction fidelity translates into downstream utility. Ablations further isolate retry budget as the dominant quality lever and show that input chunking quality conditions early-stage fidelity. Together, these results demonstrate that explicit, interpretable evaluation criteria can substantially reduce reliance on human annotation in annotation-scarce regulatory settings. More broadly, De Jure provides a scalable and auditable path toward aligning LLMs with the domain-specific regulatory and compliance obligations that govern real-world deployment.

Limitations

Reliance on LLM-based evaluation. De Jure reduces but does not eliminate the need for expert oversight in high-stakes deployments. Quality is assessed almost entirely via LLM-as-a-judge scoring rather than human-adjudicated gold standards, which may introduce calibration errors, model-specific biases, or systematic blind spots in legal interpretation. Although extraction and judgment use separate models, both are LLMs and may share failure modes; the near-perfect non-hallucination scores, in particular, may partly reflect the judge's inability to detect subtle fabrications rather than their genuine absence. The acceptance threshold θ=0.9 is likewise defined on this judge's scale and is not externally calibrated against human notions of sufficiency. Future work should validate De Jure outputs against expert annotations and quantify judge–human agreement across domains.

Evaluation scope. Our corpora are individually small (~30–113 sections each), and the downstream comparison is conducted against a single prior system (Datla et al., 2025) on the subset of HIPAA sections it publicly releases. Pairwise win rates are reported over 100 questions without confidence intervals or significance testing, so the magnitude of the observed margins should be read as indicative rather than definitive. Broader benchmarking against additional baselines, with larger question sets and explicit statistical treatment, would strengthen these conclusions.

Domain and language coverage. Although De Jure generalizes across finance, healthcare, and AI governance, these three English-language statutes do not capture the full diversity of regulatory text. Further testing is needed on regulations with highly fragmented cross-references, multilingual provisions, and obligations that are less explicitly stated. The performance decline on the more discursive EU AI Act indicates that loosely structured, principle-heavy regulations remain comparatively challenging.

Output formalism. The extracted rules are structured natural-language representations rather than formally executable logic. While this improves interpretability and retrieval utility, downstream compliance verification would still require an additional reasoning or formalization layer, in contrast to deontic-logic approaches that target executable output directly.

Inference cost. Iterative judge-guided repair adds inference overhead. Retries are bounded and gains saturate after a small number of attempts, but large-scale deployments may still face latency and compute trade-offs. Adaptive retry policies, cheaper judge models, caching, and uncertainty-based routing are promising directions for preserving quality at lower cost.

References

Anthropic. 2026. Claude 4.6 sonnet. https://www.anthropic.com/news/claude-sonnet-4-6.

Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Unski, Kasper Dinkla, Luke Zhong-Hua, and Peter Staar. 2024. Docling technical report. *arXiv preprint arXiv:2408.09869*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022a. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022b. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics (EMNLP)*.

Gautam Varma Datla, Anudeep Vurity, Tejaswani Dash, Tazeem Ahmad, Mohd Adnan, and Saima Rafi. 2025. Executable governance for ai: Translating policies into rules using llms. *arXiv preprint arXiv:2512.04408*.

Mauro Dragoni, Serena Villata, Williams Rizzi, and Guido Governatori. 2016. Combining nlp approaches for rule extraction from legal documents. In *1st Workshop on MIning and REasoning with Legal texts (MIREL 2016)*.

Abhimanyu Dubey and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

European Union. 2024. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Regulation L 2024/1689, Official Journal of the European Union. Published 12 July 2024.

Guido Governatori. 2024. An asp implementation of defeasible deontic logic. *KI-Künstliche Intelligenz*, 38(1):79–88.

Guido Governatori, Mustafa Hashmi, Ho-Pun Lam, Serena Villata, and Monica Palmirani. 2016. Semantic business process regulatory compliance checking using LegalRuleML. In *International Conference on Advanced Information Systems Engineering (CAiSE)*.

Guido Governatori and Antonino Rotolo. 2023. Deontic ambiguities in legal reasoning. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 91–100.

Yuxin He, Buzhou Tang, and Xiaoling Wang. 2024. Generative models for automatic medical decision rule extraction from text. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7034–7048.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. CUAD: An expert-annotated NLP dataset for legal contract review. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Elias Horner, Cristinel Mateis, Guido Governatori, and Agata Ciabattoni. 2025. Toward robust legal text formalization into defeasible deontic logic using llms. *arXiv preprint arXiv:2506.08899*.

Yuta Koreeda and Christopher D. Manning. 2021. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics (EMNLP)*.

Tin Wai Lam and 1 others. 2021. Extracting structured information from financial regulations using NLP. In *Proceedings of the Natural Legal Language Processing Workshop*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. *Preprint*, arXiv:2403.13787.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. RLAIF: Scaling reinforcement learning from human feedback with AI feedback. *arXiv preprint arXiv:2309.00267*.

Marco Lippi, Przemysław Pałka, Giuseppe Contissa, Francesca Lagioia, Hans-Wolfgang Micklitz, Giovanni Sartor, and Paolo Torroni. 2019. CLAUDETTE: An automated detector of potentially unfair clauses in online terms of service. volume 27, pages 117–139.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

John J. Nay. 2023. Law informs code: A legal informatics approach to aligning artificial intelligence with humans. *Northwestern Journal of Technology and Intellectual Property*.

OpenAI. 2025. GPT-5 mini. https://openai.com/index/gpt-5-mini-advancing-cost-efficient-intelligence/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint*. https://huggingface.co/Qwen.

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.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2023. Large language models encode clinical knowledge. *Nature*, 620:172–180.

Amin Sleimi, Nicolas Sannier, Mehrdad Sabetzadeh, Lionel Briand, and John Dann. 2018. Automated extraction of semantic legal metadata using natural language processing. In *IEEE International Requirements Engineering Conference (RE)*.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. SALMON: Self-alignment with principle-following reward models. *arXiv preprint arXiv:2310.05910*.

Bingxin Tang, Jiajie Zhuang, Zhiang Wu, Hongru Lu, and Changjian Fang. 2026. From policy documents to audit logic: A llms-based framework for extracting executable audit rules.

U.S. Department of Health and Human Services. 2003. Standards for privacy of individually identifiable health information (HIPAA privacy rule). Federal Regulation 45 C.F.R. Part 164 Subpart E, U.S. Department of Health and Human Services. Privacy Rule under the Health Insurance Portability and Accountability Act (HIPAA).

U.S. Securities and Exchange Commission. 1940. Investment advisers act of 1940. Federal Statute 15 U.S.C. §§ 80b-1 et seq., U.S. Securities and Exchange Commission.

Georg Henrik von Wright. 1951. Deontic logic. *Mind*, 60(237):1–15.

Chunhua Weng and 1 others. 2010. Formal representation of eligibility criteria: A literature review. volume 43, pages 451–467.

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, and 1 others. 2016. The creation and analysis of a website privacy policy corpus. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Adam Wyner and Wim Peters. 2011. On rule extraction from regulations. In *Legal knowledge and information systems*, pages 113–122. IOS Press.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. FinGPT: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*.

Tan Yu, Anbang Xu, and Rama Akkiraju. 2024. In defense of rag in the era of long-context language models. *Preprint*, arXiv:2409.01666.

Taolin Zhang, Maosong Cao, Alexander Lam, Songyang Zhang, and Kai Chen. 2025. Compassjudger-2: Towards generalist judge model via verifiable rewards. *arXiv preprint arXiv:2507.09104*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, and 1 others. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

May Myo Zin, Georg Borges, Ken Satoh, and Wachara Fungwacharakorn. 2025. Towards machine-readable traffic laws: Formalizing traffic rules into prolog using llms. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law*, pages 327–336.

A Ablation Studies

We present four ablations targeting the core design decisions of our pipeline. Unless stated otherwise, all experiments use the following defaults: maximum regeneration retries r=3, acceptance threshold θ=0.9, Llama as the backbone, and HIPAA as the evaluation corpus.

A.1 Impact of Acceptance Threshold

At each pipeline stage, an output is accepted only if its average quality score meets or exceeds θ; otherwise regeneration is triggered. We ablate $\theta \in \{0.6, 0.7, 0.8, 0.9\}$ and report per-step quality in Table 5.

θ	Step 1	Step 2	Step 3	Total
0.6	4.86	4.12	4.47	4.48
0.7	4.86	4.21	4.49	4.52
0.8	**4.92**	4.31	4.58	4.61
0.9	**4.92**	**4.42**	**4.65**	**4.67**
Δ (0.6→0.9)	+0.06	+0.30	+0.18	+0.19

Table 5: Average quality score per pipeline step across acceptance thresholds θ (scale 1–5). Bold row denotes our default. Δ reports the absolute gain from the weakest to the strongest threshold.

Quality improves monotonically with θ, with the total average rising from 4.48 to 4.67. The Δ row exposes a structurally meaningful asymmetry: Step 2 captures the dominant share of the gain (+0.30 points, i.e., 6%), compared to only +0.06 points (1.2%) for Step 1. Step 1 operates over well-scoped, bounded inputs where a first-pass extraction is typically sufficient; Step 2 involves compositionally harder extraction decisions over loosely structured definitional content, where the higher bar imposed by θ=0.9 meaningfully increases the probability that a higher-quality candidate is surfaced through regeneration. The consistent gains at θ=0.9 across all stages, at modest additional regeneration cost, confirm it as the optimal operating point.

A.2 Impact of Maximum Regeneration Retries

The retry budget r controls how many regeneration attempts a pipeline stage may make before the best-scoring output is accepted. We ablate $r \in \{0, 1, 2, 3, 4, 5\}$, where r=0 corresponds to a single-pass pipeline with no regeneration. Full per-step scores across all values of r are plotted in Figure 2.

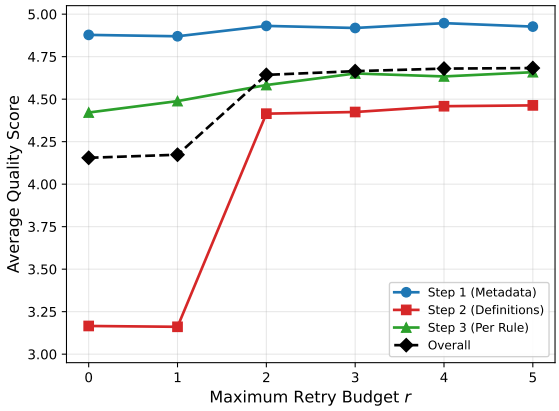

Figure 2: Average quality score per pipeline step as a function of retry budget r (scale 1–5; higher is better). Steps 1 and 3 remain largely flat throughout, while Step 2 exhibits a sharp threshold effect: negligible gain from r=0 to r=1, followed by a relatively larger recovery at r=2. All steps saturate beyond r=2; extending the budget to r=5 yields only marginal improvement (+0.02 total average), motivating our choice of r=3 as the default operating point with minimal additional compute.

The results reveal a striking non-linearity concentrated entirely in Step 2. Increasing r from 0 to 1 yields a negligible total gain of 0.01 points. This near-zero first-retry return is a diagnostic finding: the distributional tendencies that produced the initial suboptimal output persist under minimally perturbed sampling, making a single alternative draw unlikely to materially improve upon the failure.

The qualitative shift occurs at r=2, delivering a total gain of 0.47 points (9.4%) over r=1, driven almost entirely by Step 2 recovering 1.25 points (25%). This threshold behavior indicates that escaping the low-quality basin at Step 2 requires a minimum candidate pool of three: the initial output establishes the failure mode, the first retry explores a partial correction, and the second provides sufficient diversity for the selection mechanism to identify a genuinely superior output. This is consistent with the qualitative progression in Figure 3, where judge critiques from earlier attempts provide increasingly targeted feedback that steers the model away from prior failure modes rather than resampling from the same distribution. Steps 1 and 3 do not exhibit this pattern, confirming the non-linearity is specific to the compositional complexity of Step 2. Beyond r=2, total gains saturate

at +0.02, and we adopt r=3 as the default, providing a conservative safety margin at negligible additional cost.

A.3 Impact of Chunking Strategy

Input chunk quality is a silent but consequential variable: each pipeline stage can only reason over the regulatory context it receives, so structural artifacts introduced at the chunking stage propagate forward. We compare our chunking strategy against that of Datla et al. (2025) by routing both through an identical pipeline on the same HIPAA subset (Table 6).

Chunking Strategy	Step 1	Step 2	Step 3	Total
Ours	**4.97**	**4.47**	**4.60**	**4.68**
Datla et al. (2025)	4.64	4.34	4.59	4.52
Δ (Ours − comparison target)	+0.33	+0.13	+0.01	+0.16

Table 6: Average quality score per pipeline step for each chunking strategy on the HIPAA subset (scale 1–5). Δ reports the absolute gain of our strategy.

The Δ row reveals a gain profile whose non-uniformity is diagnostically informative. Step 1 captures nearly the entire benefit ($+0.33$ points, i.e., 6.6%): a chunk that cleanly encapsulates a single regulatory provision allows precise, well-scoped extraction on the first attempt, whereas a poorly bounded chunk introduces structural ambiguity that degrades initial quality before regeneration has any opportunity to help. Step 3, by contrast, is virtually identical across strategies (+0.01), suggesting it operates against a quality ceiling set by model capacity rather than input structure. Chunking exerts its leverage exclusively in the early pipeline stages; downstream refinement cannot compensate for incoherent inputs, it can only refine coherent ones.

A.4 Regeneration Trigger Strategy

We evaluate two trigger strategies: **avg-trigger**, which regenerates if the normalized average score across all criteria falls below θ=0.9; and **individual-trigger**, a strictly more conservative criterion that regenerates if any single criterion falls below a raw score of 4. Both strategies are evaluated under retry budgets $r \in \{1, 3\}$ to disentangle the effect of trigger policy from retry budget (Table 7).

Two findings stand out. First, the retry budget is the dominant factor by a wide margin: increasing r from 1 to 3 yields a total gain of 0.48 points

Max Retries (r)	Trigger	Step 1	Step 2	Step 3	Total
r=3	Avg (default)	4.92	4.42	4.65	4.66
	Individual	4.92	4.44	4.63	4.66
r=1	Avg	4.87	3.16	4.49	4.18
	Individual	4.87	3.21	4.43	4.17
Δ (r: 1→3, Avg)		+0.05	+1.26	+0.16	+0.48

Table 7: Average quality score per pipeline step across trigger strategies and retry budgets. Δ reports the gain from r=1 to r=3 under the default avg-trigger.

(9.6%), with Step 2 alone recovering 1.26 points (25.2%). This asymmetry is structurally expected: Step 2 faces the most compositionally demanding extraction task and routinely encounters diverse, co-occurring failure modes that a single retry cannot simultaneously correct. The budget of r=3 provides the minimum candidate pool necessary for the best-selection mechanism to identify a materially superior output. Second, conditional on r=3, both trigger strategies reach identical total quality (4.66), differing by at most 0.02 points (0.4%) on any individual step. With an adequate retry budget, the pipeline's best-of-r selection naturally corrects per-criterion weaknesses without needing them to be individually flagged. The individual-trigger's added conservatism imposes higher regeneration and inference cost with zero measurable return. These results confirm that trigger granularity is a second-order concern: retry budget is the primary control variable.

B Judgment Criteria

De Jure employs three specialized judges operating sequentially, each evaluating a distinct semantic layer: section-level metadata (Judge 1), definitional content (Judge 2), and individual rule units (Judge 3). Criteria within each judge are designed to be *orthogonal and collectively exhaustive* over the failure modes specific to that layer. All criteria are scored on a 1–5 scale, and the per-stage average determines whether the regeneration threshold θ is met. Table 8 summarizes the full set of criteria grouped by judge stage.

Judge 1: Section Metadata. Criteria target both factual accuracy (citations, dates, titles) and extraction completeness. *Non-Hallucination* and *Fidelity to Source Text* act as hard correctness gates: a metadata extraction that distorts regulatory identifiers is unusable regardless of structural quality. *Meaningful Population of Optional Fields* serves as a proxy for thoroughness beyond mandatory fields.

Judge	Evaluates	Criteria
Judge 1	Section Metadata	Completeness Fidelity to Source Text Non-Hallucination Title Quality Precision of Citations and Dates Meaningful Population of Optional Fields
Judge 2	Definitions	Completeness Fidelity to Source Text Non-Hallucination Precision and Formatting Quality of Terms
Judge 3	Per-Rule Quality	Completeness Conciseness Accuracy Fidelity to Source Text Neutrality Consistency Actionability Non-Hallucination

Table 8: Judgment criteria grouped by pipeline stage. Each judge targets the failure modes specific to its semantic layer: structural integrity (Judge 1), definitional precision (Judge 2), and fine-grained rule correctness and utility (Judge 3).

Judge 2: Definitions. This stage evaluates whether the extracted glossary is complete, source-faithful, and well-formed. *Precision and Formatting* is included because malformed definitions degrade both retrieval and rule grounding downstream. *Quality of Terms* assesses whether extracted terms are genuine regulatory primitives rather than incidental phrases, a distinction requiring semantic judgment beyond surface copying.

Judge 3: Per-Rule Quality. The most demanding stage, assessing each rule unit across eight criteria organized into three functional clusters: *Completeness*, *Accuracy*, and *Actionability* assess whether the rule captures the full operative content of the source provision; *Fidelity to Source Text*, *Neutrality*, and *Non-Hallucination* form a faithfulness cluster ensuring the extraction neither omits nor introduces content; and *Conciseness* and *Consistency* assess structural quality, as redundant or contradictory rules impose unnecessary burden on downstream retrieval and synthesis.

B.1 Ranking Judge Criteria for Downstream Compliance QA

While the three judges above assess intrinsic extraction quality, Experiment 3 (Section 4.4) requires a separate instrument to measure downstream utility. An LLM judge simultaneously ranks all six system responses per question, producing a strict ordering from rank 1 (best) to rank 6 (worst) with no ties permitted across five criteria. The forced-ranking design avoids degenerate outcomes (e.g., assigning all systems the same rank) while producing rich ordinal signal: from a single ranking judgment per question, pairwise win rates between any two systems can be derived by counting how often one outranks the other across questions and criteria.

Completeness captures whether all aspects of the question are addressed; *Factual Grounding* penalizes claims not traceable to the retrieved rule set; *Handling Ambiguity* assesses whether the response correctly distinguishes mandates, permissions, and unresolved provisions rather than forcing false certainty; *Practical Actionability* measures whether regulatory language is translated into concrete guidance rather than accurate but inert quotation; *Regulatory Precision* evaluates correct reflection of scope, conditions, and exceptions; and *Overall Preference* provides a holistic judgment integrating all five dimensions, reported separately to surface trade-offs not captured by any single criterion.

These criteria collectively operationalize *downstream utility*. In regulatory compliance, a well-formed extraction has no value unless it enables a system to answer questions that are complete, grounded, unambiguous, and actionable. Each criterion directly targets a failure mode that poor rule extraction introduces into the RAG pipeline: incomplete rules truncate responses, hallucinated content propagates into assertions, imprecise definitions blur scope, and incoherent structure impedes actionable synthesis. Evaluating at this level provides a direct, task-grounded measure of whether extraction fidelity translates into practical compliance support.

C Pipeline Success Example: Judgment and Repair in Practice

This section walks through a concrete example of judge-guided repair. A structurally complete but semantically flawed extraction is identified by the judge, corrected in a single repair step, and re-evaluated to a passing score. The example also shows that the judge scores each criterion independently: it penalizes only the fields that are wrong and preserves high scores on fields that are already correct.

Figure 3 traces RULE-008 from HIPAA § 164.306 through the full pipeline. Panels (a)–(b) show the raw source and its pre-processed Mark-

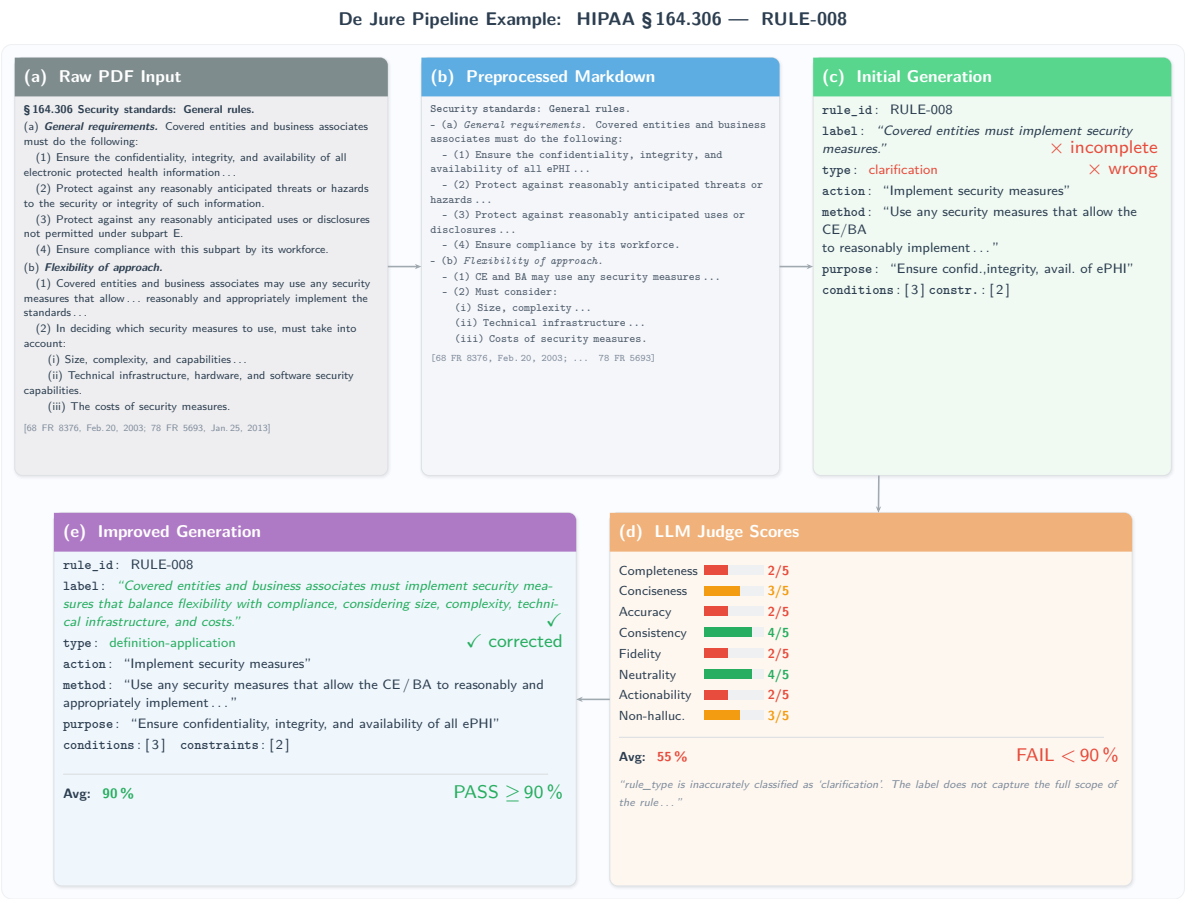

Figure 3: De Jure applied to HIPAA § 164.306. Panels (c)→(d)→(e) read left-to-right across the bottom row. Panels (a)–(b) show the raw PDF source and its pre-processed Markdown. Panel (c) shows the initial extraction with two field-level defects: an incomplete label and a misclassified rule type. Panel (d) shows the judge evaluation (avg. 0.55, FAIL) with per-criterion scores and targeted critiques. Panel (e) shows the corrected extraction after a single repair iteration (avg. 0.90, PASS), with only deficient fields revised and all others preserved.

down. The four blocks below correspond to: initial extraction, judge evaluation of the initial extraction, corrected extraction, and judge re-evaluation of the corrected extraction.

Block 1: Initial Extraction. The initial extraction is structurally complete as per the extraction schema (see Appendix I), and free of hallucinations. Targets, conditions, and the verbatim span are all correct. However, two fields are defective. The label is too brief: *"Covered entities must implement security measures"* omits the multi-factor balancing that is central to the provision. The `rule_type` is misclassified as `clarification` instead of `definition-application`, silently reversing the compliance semantics.

Block 1: Initial Extraction (RULE-008)


```
{
"rule_id": "RULE-008",
"label": "Covered entities must implement security measures.",
"rule_type": { "type": "clarification", "other_label":
"implementation-guidance" },
"targets": [ { "role": "covered_entity" }, { "role":
"business_associate" } ],
"statement": {
"action": "Implement security measures",
"action_object": "security measures",
"method": "Use any security measures that allow the covered entity or
business
associate to reasonably and appropriately implement the standards and
implementation specifications as specified in this subpart, taking
into
account the size, complexity, and capabilities of the covered entity
or
business associate, technical infrastructure, hardware, and software
security capabilities, and costs of security measures.",
"constraints": [
{ "text": "Reasonably anticipated threats or hazards to the security or
integrity of electronic protected health information" },
{ "text": "Reasonably anticipated uses or disclosures of electronic
protected
health information not permitted or required under subpart E" }
],
"conditions": [
{ "trigger": "size, complexity, and capabilities of the covered entity
or business associate" },
{ "trigger": "technical infrastructure, hardware, and software security
capabilities" },
```

```
{ "trigger": "costs of security measures" }
],
"exceptions": [],
"penalties_or_consequences": null,
"purpose": "Ensure the confidentiality, integrity, and availability of
all
electronic protected health information",
"verbatim": "Covered entities and business associates may use any
security
measures that allow the covered entity or business associate to
reasonably
and appropriately implement the standards and implementation
specifications
as specified in this subpart."
},
"citations": { "text": "164.306: Security standards: General rules.
[68 FR 8376, Feb. 20, 2003; 68 FR 17153, Apr. 8, 2003; 78 FR 5693, Jan.
25, 2013]" },
"examples": []
}
```

Block 2: Judge Evaluation (Fail, 0.55). The judge returns a normalized average score of 0.55 and flags the extraction as FAIL. The critique is targeted: it assigns low scores to *Completeness*, *Accuracy*, *Fidelity*, and *Actionability*, which are the affected criteria, while keeping high scores on *Consistency* and *Neutrality*, which are correct. This field-level signal is what makes the repair step precise. The feedback for each criteria is provided to guide the regeneration in a direction to improve the score on the respected criteria.

Block 2: Judge Evaluation of Initial Extraction | Avg. Score: 0.55 – Fail

```
"Completeness": { "Score": 2, "Justification": "Label incompletely
summarizes the rule,
omitting the multi-factor balancing requirement central to the
provision.
rule_type is misclassified; should be definition-application." },
"Conciseness": { "Score": 3, "Justification": "Label is brief but
sacrifices accuracy
for brevity; the scope of the rule is inadequately represented." },
"Accuracy": { "Score": 2, "Justification": "rule_type classified as
clarification
instead of definition-application, misrepresenting the rule's nature."
},
"Consistency": { "Score": 4, "Justification": "Targets correctly
identified as
covered_entity and business_associate, consistent with source." },
"Fidelity": { "Score": 2, "Justification": "Statement omits the full
balancing
framework including both targets; significant omission relative to
source." },
"Neutrality": { "Score": 4, "Justification": "Label and statement are
neutrally
presented; no interpretive bias introduced." },
"Actionability": { "Score": 2, "Justification": "Statement fails to
convey what both
covered entities and business associates must do; insufficiently
operational." },
"Non_hallucination": { "Score": 3, "Justification": "No fabricated
content, but
significant omissions reduce faithfulness to the source provision." }

Average Normalized Score: 0.55 – FAIL – Regeneration triggered.
```

Block 3: Corrected Extraction. After one repair iteration, only the two flagged fields change. The label is expanded to capture the multi-factor balancing framework. The `rule_type` is corrected to `definition-application`. All other fields, including conditions, constraints, verbatim span, and citations, remain exactly as before. The repair targets specific deficiencies rather than regenerating the rule from scratch.

Block 3: Corrected Extraction (RULE-008)

```
{
"rule_id": "RULE-008",
"label": "Covered entities and business associates must implement
security
measures that balance flexibility with compliance, considering size,
complexity, technical infrastructure, and costs.",
"rule_type": { "type": "definition-application", "other_label":
"implementation-guidance" },
"targets": [ { "role": "covered_entity" }, { "role":
"business_associate" } ],
"statement": {
"action": "Implement security measures",
"action_object": "security measures",
"method": "Use any security measures that allow the covered entity or
business
associate to reasonably and appropriately implement the standards and
implementation specifications as specified in this subpart, taking into
account the size, complexity, and capabilities of the covered entity or
business associate, technical infrastructure, hardware, and software
security capabilities, and costs of security measures.",
"constraints": [
{ "text": "Reasonably anticipated threats or hazards to the security or
integrity of electronic protected health information" },
{ "text": "Reasonably anticipated uses or disclosures of electronic
protected
health information not permitted or required under subpart E" }
],
"conditions": [
{ "trigger": "size, complexity, and capabilities of the covered entity
or business associate" },
{ "trigger": "technical infrastructure, hardware, and software security
capabilities" },
{ "trigger": "costs of security measures" }
],
"exceptions": [],
"penalties_or_consequences": null,
"purpose": "Ensure the confidentiality, integrity, and availability of
all
electronic protected health information",
"verbatim": "Covered entities and business associates may use any
security
measures that allow the covered entity or business associate to
reasonably
and appropriately implement the standards and implementation
specifications
as specified in this subpart."
},
"citations": { "text": "164.306: Security standards: General rules.
[68 FR 8376, Feb. 20, 2003; 68 FR 17153, Apr. 8, 2003; 78 FR 5693, Jan.
25, 2013]" },
"examples": []
}
```

Block 4: Judge Re-evaluation (Pass, 0.90). The judge re-evaluates the corrected extraction and returns a normalized average score of 0.90. All four previously low criteria recover: *Completeness* and *Accuracy* reach 5, reflecting the corrected rule type and the richer label. *Actionability* and *Fidelity* similarly improve. *Conciseness* drops slightly, as the longer label trades brevity for coverage. Criteria that were already correct retain their scores, confirming that the judge does not penalize correct fields for the failures of others. The score meets the threshold $\theta = 0.90$, and the repair loop halts.

Block 4: Judge Evaluation of Corrected Extraction | Avg. 0.90 (PASS)

```
"Completeness": { "Score": 5, "Justification": "Label now comprehensively captures
the multi-factor balancing requirement. All required fields populated correctly.
rule_type corrected to definition-application." },
"Conciseness": { "Score": 4, "Justification": "Label is necessarily detailed to
reflect the provision's scope; slight verbosity is justified given the complexity of the rule." },
"Accuracy": { "Score": 5, "Justification": "rule_type correctly classified as
definition-application; accurately represents the provision's regulatory
function." },
"Consistency": { "Score": 4, "Justification": "Targets consistently and correctly
identify both covered_entity and business_associate throughout." },
"Fidelity": { "Score": 4, "Justification": "Statement closely reflects the source
text; label now captures both targets and the balancing framework.
Minor omission: flexibility aspect could be made more explicit." },
"Neutrality": { "Score": 5, "Justification": "Label and statement remain neutrally
framed with no interpretive bias introduced." },
"Actionability": { "Score": 4, "Justification": "Statement now provides operationally
useful guidance for both covered entities and business associates, though
method field is dense and could be more concisely structured." },
"Non_hallucination": { "Score": 5, "Justification": "All extracted content is
grounded in the source text; no fabricated or inferred content introduced." }

Average Normalized Score: 0.90 – PASS – Repair loop terminates.
```

The jump of the normalized averaged score from 0.55 to 0.90 was driven by two field corrections, with everything else unchanged. This pattern holds across all corpora and aligns with the quantitative results in Section A.2.

This example illustrates two key properties of the judge. First, it reliably flags weak extractions: the low scores in Block 2 are not generic penalties but precise signals tied to specific fields, paired with actionable feedback that directly guides the repair. Second, it recognizes improvement: once those fields are corrected, the judge awards high scores to the updated output without penalizing fields that were already correct. These two properties are what make automated iterative repair practical. The judge distinguishes poor extractions from good ones, and its feedback is specific enough to drive targeted corrections rather than blind regeneration. This allows the extraction and verification to operate as mutually reinforcing stages and drive De Jure's self-correcting nature.

D De Jure Pipeline: Algorithm Psuedo Code

Algorithm 1 provides a complete procedural specification of the De Jure pipeline, complementing the stage-by-stage description in Section 3.

The total LLM call budget per section is bounded by $1+3\times r$ in the worst case (one initial generation plus at most r repair attempts across each of the three stages), and reduces to $1+3$ in the best case where all stages pass on the first judgment.

E Extraction Quality Experiment: Full Results

This section provides further details and extended results for the experiments presented in Section 4.2.

Table 9 presents the complete Judge 1 metadata criteria scores for Experiment 1 across the four different models evaluated on HIPAA corpus. Performance across models is near-saturated on the metadata extraction task, with most criteria approaching the upper bound. Fidelity and non-hallucination remain consistently high, indicating minimal hallucination in this structured setting. Title quality is fully saturated (5.00 ± 0.00) and thus non-discriminative. Differences primarily arise in citation/date extraction and optional fields, suggesting sensitivity to edge cases. Although **GPT-5** achieves the highest average score, differences fall within variance, indicating no statistically significant separation. Notably, smaller open-weight models (Llama) perform comparably, suggesting limited dependence on model scale for this task.

Table 10 presents the complete Judge 2 definition criteria scores for Experiment 1 across the four different models evaluated on HIPAA corpus. Across all models, performance on rule-based definition generation remains consistently high, with non-hallucination fully saturated, indicating reliable grounding in the extracted rules. Fidelity, precision, and term quality show modest variability, reflecting minor differences in how models express structured outputs. While **Qwen** achieves the highest average score, overall performance is comparable across models, suggesting that the definition criteria part of the pipeline is robust to different LLMs.

Table 11 presents the complete Judge 3 per-rule criteria scores for Experiment 1 across the four different models evaluated on HIPAA corpus. Consistent with previous steps, performance on per-rule evaluation remains uniformly high across all models, with non-hallucination and neutrality fully saturated, indicating strong faithfulness to the extracted rules. Compared to the earlier criteria, slightly greater variation is observed in completeness, conciseness, and fidelity, though differences

Model	Completeness	Fidelity	Non-Halluc.	Title Quality	Citation & Date	Opt.	Avg
Claude	4.85±0.35	**5.00±0.00**	**5.00±0.00**	**5.00±0.00**	4.95±0.22	4.81±0.43	*4.94±0.26*
Llama	**4.87±0.34**	4.99±0.09	4.98±0.20	**5.00±0.00**	4.92±0.30	4.79±0.43	*4.92±0.28*
GPT-5	**4.87±0.34**	4.99±0.09	**5.00±0.00**	**5.00±0.00**	**4.97±0.18**	**4.86±0.37**	**4.95±0.23**
Qwen	4.85±0.35	4.97±0.25	4.95±0.38	**5.00±0.00**	4.89±0.46	**4.86±0.37**	*4.92±0.34*

Table 9: Full per-criteria scores on the HIPAA corpus for **Judge 1** (Metadata) criteria averaged over 3 runs (scale 1–5; higher is better). Abbreviations: Non-Halluc. = Non-Hallucination.; Opt. = Optional Field Population; Avg = Average Score

Model	Completeness	Fidelity	Non-Halluc.	Precision	Term Quality	Avg
Claude	4.25±1.01	4.69±0.46	**5.00±0.00**	4.47±0.50	4.71±0.45	*4.62±0.63*
Llama	4.22±0.55	4.33±0.49	4.99±0.09	4.30±0.46	4.36±0.50	*4.44±0.53*
GPT-5	4.31±0.84	**4.74±0.44**	**5.00±0.00**	4.45±0.50	**4.74±0.44**	*4.65±0.57*
Qwen	**4.52±0.87**	4.71±0.45	**5.00±0.00**	**4.64±0.50**	4.72±0.45	***4.72±0.55***

Table 10: Full per-criteria scores on the HIPAA corpus for **Judge 2** (Definitions) criteria averaged over 3 runs (scale 1–5; higher is better). Abbreviations: Non-Halluc. = Non-Hallucination; Precision = Precision & Formatting.

remain modest overall. While **GPT-5** achieves the highest average score, performance across models is broadly comparable, reinforcing that the extracted rules are robust and consistently interpretable across different LLMs.

F Domain Generalization Experiment: Full Results

This section provides further details and extended results for the experiments presented in Section 4.3.

Table 12 presents the complete Experiment 2 results for Judge 1 metadata criteria across SEC, EU AI Act, and HIPAA. **Qwen** and **GPT-5** perform comparably overall on metadata extraction, with only small differences across most criteria. Notably, the EU AI Act is consistently lower than the other two datasets on both **Completeness** and **Citation & Date**, indicating that metadata recovery is more challenging in this corpus.

Table 13 presents the complete Experiment 2 results for Judge 2 definition criteria across SEC, EU AI Act, and HIPAA. **GPT-5** underperforms on definition **Completeness** and **Precision**, with the largest drop observed on HIPAA. More broadly, HIPAA appears to be the most difficult corpus for definition extraction, with both models showing lower scores than on SEC and EU AI Act in key definition quality dimensions.

Table 14 presents the complete Experiment 2 results for Judge 3 per-rule criteria across SEC, EU AI Act, and HIPAA. **Qwen** underperforms relative to **GPT-5** on rule **Completeness**, **Fidelity**, and **Actionability** across datasets. At the same time, both models perform strongly on **Accuracy**, **Consistency**, and **Neutrality**. Finally, consistent with Tables 12 and 13, hallucination remains minimal across all three judges, with near-ceiling non-hallucination scores throughout.

G Downstream QA Evaluation: Full Results

G.1 System Descriptions

De Jure. Our full pipeline employs a hierarchical extraction schema with nested statement decomposition (action, conditions, constraints, exceptions, purpose) and iterative repair with $r{=}3$ regeneration retries. Extraction proceeds in stages: section metadata, then definitions, then rules, each conditioned on the outputs of the prior stage.

No repair (r=0). Identical schema and hierarchical generation strategy as De Jure, but with `max_regeneration_retries` $= 0$. Rules are accepted as-is from the first generation pass without quality-driven regeneration, isolating the contribution of iterative repair.

Simplified schema. Uses a flat schema containing only `rule_id`, `summary`, `rule_type`, `target`, and `rule_text` fields. There is no nested statement object—regulatory content is captured in a single free-text `rule_text` field rather than decomposed into structured sub-fields.

Algorithm 1: De Jure: Regulatory Rule Extraction with Judgment and Repair

Input : Document D (PDF / HTML)
Output : Structured rule set $\mathcal{R}$

```
/* Stage 1: Pre-processing */
M ← ConvertToMarkdown(D);
M ← CleanAndNormalise(M);
S ← SplitBySections(M) ;                                  // split on section markers
foreach section s ∈ S do
    IndexByID(s);
    Attach metadata and SHA256(D) to s;
end
/* Stage 2: Generation, Judgment, and Selective Repair */
R ← ∅;
r ← 3 ;                                                   // max regeneration attempts per stage
θ ← 0.90 ;                                                // avg-score acceptance threshold
stages ← [(metadata, 6), (definitions, 5), (rule_units, 8)];
foreach section s ∈ S do
    e ← LLMGenerate(s, JSONSchema);
    if e = null then continue ;                           // non-actionable section
    ;
    foreach (name, n_c) ∈ stages do
        ê ← e;
        μ̂ ← 0 ;                                           // best score seen
        for t ← 1 to r do
            scores, critique ← JudgeStage(e, name, n_c criteria);
            μ ← avg(scores);
            if μ > μ̂ then
                ê ← e;
                μ̂ ← μ;
            end
            if μ ≥ θ then break ;                         // threshold met
            ;
            e ← LLMGenerate(s, e, scores, critique, JSONSchema);
        end
        e ← ê ;                                           // commit best generation for this stage
    end
    R ← R ∪ {e};
end
return R;
```

Non-hierarchical. Generates all rules for a section in a single LLM call rather than following the staged hierarchy (metadata → definitions → rules). This ablation isolates the contribution of hierarchical conditioning.

Oracle. Receives the full serialized regulatory document as LLM context with no retrieval step, serving as an upper bound on answer quality unconstrained by extraction or retrieval.

G.2 Top-1 Rates

Table 15 reports the percentage of questions for which each system produced the highest-ranked (i.e. top-1) answer, broken down by evaluation criterion.

The oracle approach with access to the full regulatory document sets the upper bound for the highest top-1 rate across all criteria (28–34%). Among RAG systems, De Jure and the non-hierarchical variant are closely matched (13–17% each), with De Jure holding a slight edge on Factual Grounding, Regulatory Precision, and Practical Actionability. Notably, the non-hierarchical variant achieves the highest RAG top-1 rate on Handling Ambiguity (22.2%), suggesting that its generation strategy may produce rules better suited to ambiguous multi-provision queries. The no-repair ablation is consistently the weakest RAG system (8–9% on most criteria), except on Handling Ambiguity where it reaches 17.2%, reinforcing that this criterion is the most sensitive to generation strategy rather than repair quality.

G.3 Context Length Analysis

Table 16 reports the average number of input tokens provided to the generation model per query at

Model	Comp.	Conc.	Accuracy	Cons.	Fidelity	Neut.	Actionab.	Non-Halluc.	Avg
Claude	4.20±0.43	**4.64±0.48**	4.91±0.29	4.99±0.12	4.62±0.49	**5.00±0.00**	4.41±0.49	**5.00±0.00**	*4.72±0.45*
Llama	4.15±0.35	4.61±0.51	4.72±0.45	4.87±0.35	4.42±0.61	5.00±0.03	4.37±0.49	4.99±0.11	*4.64±0.50*
GPT-5	**4.29±0.45**	4.54±0.50	**4.97±0.16**	**4.99±0.09**	**4.74±0.44**	**5.00±0.00**	**4.46±0.50**	**5.00±0.00**	***4.75±0.43***
Qwen	4.12±0.34	4.53±0.50	4.85±0.36	4.97±0.17	4.48±0.56	**5.00±0.00**	4.32±0.47	5.00±0.07	*4.66±0.49*

Table 11: Full per-criteria scores on the HIPAA corpus for **Judge 3** (Per-Rule) criteria averaged over 3 runs (scale 1–5; higher is better). Abbreviations: Comp. = Completeness; Conc. = Conciseness; Cons. = Consistency; Neut. = Neutrality; Actionab. = Actionability; Non-Halluc. = Non-Hallucination.

Dataset	Model	Completeness	Fidelity	Non-Halluc.	Title Quality	Citation & Date	Opt.
SEC	Qwen	4.82 ± 0.03	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00	4.99 ± 0.01	4.71 ± 0.06
	GPT-5	4.87 ± 0.06	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00	4.99 ± 0.01	4.89 ± 0.07
EU AI Act	Qwen	3.94 ± 0.08	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00	4.83 ± 0.24	4.00 ± 0.47
	GPT-5	4.04 ± 0.06	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00	4.63 ± 0.17	4.44 ± 0.11
HIPAA	Qwen	4.85 ± 0.00	4.97 ± 0.03	4.95 ± 0.04	5.00 ± 0.00	4.89 ± 0.04	4.86 ± 0.06
	GPT-5	4.86 ± 0.01	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00	4.97 ± 0.01	4.77 ± 0.07

Table 12: Generalization across three regulatory corpora for **Judge 1** (Metadata) criteria (scale 1–5; higher is better). Values are reported as mean ± std. Abbreviations: Non-Halluc. = Non-Hallucination.; Opt. = Optional Field Population

retrieval depth k=10. The Oracle receives the full serialized regulatory document (∼11.6k tokens), while RAG systems receive only their top-k retrieved rules. De Jure achieves the best pairwise win rate among RAG systems while consuming only 16% of the Oracle's context budget, a 6.2× compression. This highlights the scalability advantage of structured extraction: as regulatory corpora grow beyond current context windows, RAG over extracted rules remains tractable whereas the in-context approach does not.

H Prompts

De Jure uses eleven prompts organized into four functional groups: an initial extraction prompt that generates the structured rule representation from raw text (Section H.1), three stage-specific regeneration prompts that incorporate judge feedback to correct failing extractions (Section H.2), three judge prompts that produce the per-criterion scores and natural-language critiques driving the repair loop (Section H.3), and four downstream evaluation prompts for the RAG-based QA experiment (Section H.4). The pipeline prompts (1–7) are fully domain-agnostic and contain no corpus-specific examples or seed rules; the downstream prompts (8–11) are specific to the HIPAA evaluation in Experiment 3.

H.1 Initial Extraction Prompt

Prompt 1 is invoked once per regulatory section at Stage 2 of the pipeline. It instructs the backbone LLM to decompose the source text into three typed components: section metadata, definitions, and rule units, each conforming to a fixed schema. The prompt encodes quality constraints inline, including null-filtering for non-actionable sections and negation-preserving rules for action fields, to minimize the number of judge-triggered repairs on well-formed inputs.

Prompt 1 | Rule Generation (Stage 2)

You are an expert at analyzing regulatory documents to extract structured, actionable rules from {}.

Task
Extract all rules, requirements, obligations, prohibitions, permissions, and procedures. Each distinct rule = one RuleUnit.

Requirements:
- **Exhaustive**: Capture every rule in the text
- **Objective**: Use only source text – no inference or interpretation
- **Precise**: Include verbatim quotes to ground extractions
- **Structured**: Decompose into clear components

Extraction Guidelines

Section Metadata

Citation
- Official regulatory section number (e.g., "17 CFR 275.0-2", "Rule 144", "§230.405")
- DO NOT use Federal Register citations (e.g., "51 FR 32907") as `section_cite`
- If no regulatory section identifier exists in source, set to null

Title
- Section's official heading

Effective Dates
- Extract dates in EXACT format from source (e.g., "Sept. 17, 1986" NOT "September 17, 1986")
- Preserve abbreviations, punctuation, and spacing
- Extract ALL dates with appropriate event types

Dataset	Model	Completeness	Fidelity	Non-Halluc.	Precision	Term Quality
SEC	Qwen	4.69 ± 0.03	4.88 ± 0.03	5.00 ± 0.00	4.85 ± 0.03	4.91 ± 0.01
	GPT-5	4.44 ± 0.08	4.86 ± 0.08	5.00 ± 0.00	4.55 ± 0.12	4.87 ± 0.07
EU AI Act	Qwen	4.72 ± 0.08	4.89 ± 0.00	5.00 ± 0.00	4.72 ± 0.08	4.89 ± 0.00
	GPT-5	4.48 ± 0.06	4.74 ± 0.06	5.00 ± 0.00	4.44 ± 0.00	4.74 ± 0.06
HIPAA	Qwen	4.52 ± 0.04	4.71 ± 0.02	5.00 ± 0.00	4.64 ± 0.04	4.72 ± 0.04
	GPT-5	4.26 ± 0.09	4.73 ± 0.02	5.00 ± 0.00	4.36 ± 0.04	4.75 ± 0.03

Table 13: Generalization across three regulatory corpora for **Judge 2** (Definitions) criteria (scale 1–5; higher is better). Values are reported as mean $\pm$ std. Abbreviations: Non-Halluc. = Non-Hallucination; Precision = Precision & Formatting.

Dataset	Model	Comp.	Conc.	Accuracy	Cons.	Fidelity	Neut.	Actionab.	Non-Halluc.
SEC	Qwen	4.11 ± 0.01	4.46 ± 0.02	4.82 ± 0.01	4.94 ± 0.01	4.48 ± 0.04	5.00 ± 0.00	4.38 ± 0.02	5.00 ± 0.00
	GPT-5	4.29 ± 0.02	4.44 ± 0.04	4.96 ± 0.01	4.99 ± 0.01	4.77 ± 0.03	5.00 ± 0.00	4.50 ± 0.04	5.00 ± 0.00
EU AI Act	Qwen	4.31 ± 0.05	4.43 ± 0.02	4.89 ± 0.02	4.93 ± 0.00	4.54 ± 0.04	5.00 ± 0.00	4.22 ± 0.04	5.00 ± 0.00
	GPT-5	4.43 ± 0.01	4.34 ± 0.05	4.94 ± 0.02	4.98 ± 0.00	4.67 ± 0.03	5.00 ± 0.00	4.27 ± 0.07	5.00 ± 0.00
HIPAA	Qwen	4.12 ± 0.02	4.53 ± 0.01	4.85 ± 0.03	4.97 ± 0.00	4.48 ± 0.02	5.00 ± 0.00	4.32 ± 0.03	5.00 ± 0.00
	GPT-5	4.28 ± 0.03	4.54 ± 0.03	4.96 ± 0.01	4.99 ± 0.00	4.74 ± 0.02	5.00 ± 0.00	4.43 ± 0.03	5.00 ± 0.00

Table 14: Generalization across three regulatory corpora for **Judge 3** (Per-Rule) criteria (scale 1–5; higher is better). Values are reported as mean $\pm$ std. Abbreviations: Comp. = Completeness; Conc. = Conciseness; Cons. = Consistency; Neut. = Neutrality; Actionab. = Actionability; Non-Halluc. = Non-Hallucination.

• Multiple FR citations typically indicate: first = "adopted", subsequent = "amended"
• Look for context clues in brackets (e.g., "amended at 64 FR...")
• Common event types: adopted, amended, effective, rescinded

notes
• Set to null unless source contains additional context not captured in other fields

x_extensions
• Set to null unless source contains non-standard metadata

Definitions
Extract terms whose meaning is established by the text:
• Positive definitions: "X means Y"
• Negative definitions: "X is not Y" or "X does not include Y"
• Example: "A transaction...is not an assignment" → defines what "assignment" excludes

Note: Definitional text may ALSO constitute a rule (e.g., "X is not considered Y" both defines Y's scope AND exempts X). Extract to BOTH `definitions` and `extracted_rules` when applicable.

For Each Rule Extract:

1. Rule Identification
• **rule_id**: Unique identifier (e.g., "RULE-001"). Use consistent nomenclature across document.
• **label**: Concise summary (5–25 words)
• **rule_type**: obligation | prohibition | permission | exemption | definition-application | safe-harbor | procedure | clarification | deeming | condition-precedent | other (provide `other_label`)
• Use **exemption** for: Carves out exceptions or states what is NOT covered (e.g., "X is not deemed Y")
• Use **clarification** for: Explains scope or meaning without imposing new obligations

2. Targets (WHO must comply)
CRITICAL: Target = WHO is being instructed to perform the action per this rule, NOT who is the action being performed on!

Example:
• "Issuers must file to Commission" → Target is **issuer** (not Commission)
• "Any person may serve by furnishing Commission..." → Target is **any_person**
• "If ⟨event⟩, the head of Commission must..." → Target is **head of Commission** (not Commission)

For each target:
• **role**: answers the "who must follow this rule?" question
• **Preserve exact entity names**: "Secretary of the Commission" NOT "commission"
• **qualifiers**: captures any additional conditions that apply (e.g. "with 15+ employees")

IMPORTANT – Resolve Vague References:
• If source text uses "you", "the registrant", "such person", determine the actual entity from:
– Section title (e.g., "Rules for Investment Advisers" → target is "investment adviser")
– Regulatory context (e.g., Form CRS is for broker-dealers and investment advisers)
– Earlier definitional text in the section

3. Statement
Extract the complete rule decomposition as a single structured object with the following fields:

3.1 action
• Primary regulatory verb phrase (multi-word OK: "file reports", "serve legal process")
• **CRITICAL: For negative statements, INCLUDE the negation in the action field**
• WHAT must/must not/may be done?

Examples of negative actions: "is not deemed" • "does not constitute" • "shall not be required" • "are not subject to"

× Wrong: "deemed" ✓ Correct: "is not deemed"
× Wrong: "considered" ✓ Correct: "is not considered"

3.2 action_object
• WHO/WHAT is acted upon (e.g., "to the Commission", "non-resident advisers", "assignment")

3.3 method_or_conditions
• HOW performed – mechanisms, procedures (e.g., "by furnishing documents", "by delivering the amended Form CRS or by communicating through another disclosure")
• **DO NOT include temporal/quantitative constraints here** – extract those to `constraints` field

3.4 constraints
Extract temporal/quantitative/qualitative limits as an array. **If none, return [].**
For each constraint:
• **text**: Clear description (e.g., "within 10 business days", "not less than $5,000", "without charge")
• **applies_to**: What is constrained (null if ambiguous)

3.5 conditions
Capture ALL conditions that define when, where, or under what circumstances the rule applies. **If unconditional, return [].**

IMPORTANT: The trigger should describe the QUALIFYING circumstance, not just repeat the subject
× Wrong: trigger: "transaction"
✓ Correct: trigger: "transaction does not result in change of actual control or management"

System	Factual Gr.	Completeness	Reg. Precision	Handling Amb.	Practical Act.	Overall
Baseline (Datla et al., 2025)	12.2	11.2	12.2	11.1	12.2	12.2
De Jure	**17.3**	**16.3**	**17.3**	13.1	**17.3**	**17.3**
No repair	9.2	8.2	9.2	**17.2**	8.2	8.2
Simplified schema	13.3	**14.3**	13.3	8.1	13.3	13.3
Non-hierarchical	**16.3**	**16.3**	**16.3**	**22.2**	**16.3**	**16.3**
Oracle (full document)	31.6	33.7	31.6	28.3	32.7	32.7

Table 15: Top-1 rate (%) on the HIPAA downstream QA task: percentage of questions for which each system's answer was ranked first by the judge. Rates are reported per evaluation criterion across 100 questions with retrieval depth k=10. Top two RAG systems are bolded per criterion.

System	Tokens/query	Compression
Baseline (Datla et al., 2025)	776	14.9×
De Jure	1,865	6.2×
No repair	1,911	6.1×
Simplified schema	555	20.9×
Non-hierarchical	1,584	7.3×
Oracle (full document)	11,586	1.0×

Table 16: Average input tokens per query at k=10. Compression is relative to the Oracle (full document in context). Token counts use the `cl100k_base` tokenizer.

Types of conditions to capture:
1. **Event-based triggers** (something happens):
• "If process is served..." → trigger: "process is served on the Commission"
• "When trading volume exceeds..." → trigger: "trading volume exceeds threshold"
• "Upon receipt of notice..." → trigger: "receipt of notice"
2. **Scope/jurisdictional conditions** (regulatory framework):
• "Under Forms ADV and ADV-NR..." → trigger: "Under Forms ADV and ADV-NR"
• "For purposes of Section 5..." → trigger: "For purposes of Section 5"
• "As provided in Regulation S-K..." → trigger: "As provided in Regulation S-K"
3. **Status-based conditions** (entity characteristics):
• "If the issuer is foreign..." → trigger: "issuer is foreign"
• "For reporting companies..." → trigger: "entity is a reporting company"

For each condition:
• **trigger**: The condition that must be met (can be event, scope, or status)
• **time_window**: Start/end times and timezone (only for temporal conditions)
• **cross_references**: Regulatory citations mentioned in the condition (e.g., "section 205(a)(2) of the Act")

3.6 exceptions
Extract exemptions/carve-outs. **If none, return [].**
For each exception:
• **text**: Exception description
• **cross_references**: Regulatory citations elaborating exception

3.7 penalties_or_consequences
Extract stated consequences of non-compliance. **If none, return null.**
For each penalty/consequence:
• **text**: Description of penalty or consequence
• **cross_references**: Regulatory citations related to the penalty

3.8 purpose
• Stated objective (only if explicit; else null)

3.9 verbatim
• **REQUIRED** – Exact source quote establishing rule
• Must include ALL sentences that contribute to this RuleUnit

4. Additional Elements
• **citations**: ALL regulatory references mentioned in the rule, regardless of where they appear in the statement. Include references from conditions, exceptions, and the main statement. For each citation provide the reference text and context of its use.
• **examples**: Illustrative examples from source (lift verbatim; else null)

Output Format
JSON conforming to `SecActSectionExtraction` schema:
• `section_meta`: Citation, title, effective dates
• `definitions`: Terms defined in section with explanations
• `extracted_rules`: List of RuleUnit objects

Quality Standards
• **Completeness**: Every regulatory statement → RuleUnit with all applicable fields filled
• **Fidelity**: Verbatim = actual quotes, not paraphrases
• **Granularity**: Multiple distinct obligations → separate RuleUnits
• **No Hallucination**: Only explicit content; use null/[] when absent
• **Preserve Specificity**: Exact entity names, titles, "if" statements – no simplification
• **Regulatory Citations**: Federal Register citations (e.g., [51 FR 32907, Sept. 17, 1986]) indicate effective dates but should not be extracted to notes or x_extensions unless they provide substantive context beyond dating.

Special Cases
• **Compound Rules**: Multiple obligations on different targets → split into RuleUnits
• **Cross-References**: "as defined in §240.10b-5" → capture in `cross_references`
• **Implicit Requirements**: Extract only if clearly stated (e.g., "failure to file within 10 days requires..." implies filing obligation)

CRITICAL OUTPUT REQUIREMENTS
Your JSON output MUST include these fields for EVERY RuleUnit (use `[]` for empty):

```
{
"rule_id": "...",
"label": "...",
"rule_type": {...},
"targets": [...],
"statement": {
"action": "...",
"action_object": "...",
"method": "...",
"constraints": [...],   // REQUIRED - use [] only if genuinely
absent
"conditions": [...],    // REQUIRED - use [] only if genuinely
absent
"exceptions": [...],    // REQUIRED - use [] only if genuinely
absent
"penalties_or_consequences": [...],   // REQUIRED - use null only
if genuinely absent
"purpose": "...",
"verbatim": "..."
},
"citations": {...},
"examples": [...]
}
```

Missing any of these fields will cause validation failure.

FINAL VERIFICATION PROTOCOL
Before returning your JSON output:
1. Section Metadata Verification:
• `section_cite`: null if not in source, exact format if present (DO NOT use FR citations)
• `effective_dates`: ALL dates with correct event types ("adopted" for first, "amended" for subsequent)
• Dates: exact format match including abbreviations (e.g., "Sept." not "September")
• definitions: Extract negative definitions too (e.g., "X is not Y")
• Optional fields (notes, x_extensions): explicitly null if unused
2. Rule Extraction Verification:
• action: For negative statements, verify negation is INCLUDED (e.g., "is

not deemed" NOT "deemed")
• rule_type: "exemption" for exclusionary rules (e.g., "X is not Y")
• conditions: trigger describes the qualifying circumstance, not just the subject
• conditions: `cross_references` populated if regulatory citations mentioned
• citations: ALL regulatory references included with context
3. Completeness Check: For each RuleUnit, verify that you've populated:
• conditions array (even if single item)
• constraints array (check for temporal/quantitative limits)
• exceptions array (check for carve-outs)
• penalties_or_consequences (check for stated consequences)
4. Source Cross-Check: For any field marked as `[]`, re-read the verbatim quote and confirm no extractable information exists.
5. Scope Conditions: Specifically check if the rule begins with regulatory context like "Under [Form/Regulation]" or "For purposes of [Section]" – this is ALWAYS a condition.

Now analyze the document and extract all rules:
{}

Prompt 1: Initial rule generation prompt (Stage 2 of De Jure). The first placeholder `{}` is filled with the regulatory domain; the second with the source section text. No domain-specific examples or seed rules are included.

H.2 Regeneration Prompts

When a pipeline stage fails to meet the acceptance threshold θ, the backbone LLM is re-prompted with three inputs: the original source text, the failing extraction, and the judge's structured critique. Three stage-specific regeneration prompts are used, one per judgment stage, each targeting the exact output type of its corresponding judge. Crucially, each prompt asks the model to correct only the deficient fields identified in the critique rather than regenerating the full extraction, keeping repairs surgical and computationally bounded.

Prompt 2 | Stage 1 Regeneration: Section Metadata

Your task is to extract section metadata from a source text containing regulatory content. You have been provided with a previous extraction that was assessed based on specific criteria and determined to be incorrect.

Using the critique, generate corrected metadata from the source text to achieve the highest score on all mentioned criteria:
• Completeness
• Fidelity to Source Text
• Non-Hallucination
• Title Quality
• Precision of Citations and Dates
• Reasonable Population of Optional Fields

Follow the format below:

```
{
"section_cite": "citation from source",
"title": "section title",
"effective_dates": [{"event": "event_type", "date": "exact date format"}],
"notes": "additional context if present in source", // (optional)
"x_extensions": {} // (optional)
}
```

Only generate the corrected metadata JSON and nothing else.

Inputs:
Source Text: {}
Incorrect Metadata: {}
Critique: {}

Prompt 2: Stage 1 regeneration prompt. Placeholders are filled at runtime with the source section text, the failing metadata extraction, and the Judge 1 critique.

Prompt 3 | Stage 2 Regeneration: Definitions

Your task is to extract definitions from a source text containing regulatory content. You have been provided with a previous extraction that was assessed based on specific criteria and determined to be incorrect.

Using the critique, generate corrected definitions from the source text to achieve the highest score on all mentioned criteria:
• Completeness
• Fidelity to Source Text
• No Hallucination or Fabrication
• Precision and Formatting
• Quality of Terms

Follow the format below:

```
{
"definitions": [
{
"term": term,
"text": definition
}
]
}
```

Only generate the corrected definitions JSON and nothing else.

Inputs:
Source Text: {}
Incorrect Definitions: {}
Critique: {}

Prompt 3: Stage 2 regeneration prompt. Placeholders are filled at runtime with the source section text, the failing definitions extraction, and the Judge 2 critique.

Prompt 4 | Stage 3 Regeneration: Rule Units

Your task is to extract rule units from a source text containing regulatory content. You have been provided with a previous extraction that was assessed based on specific criteria and determined to be incorrect.

Using the critique, generate a corrected rule unit from the source text to achieve the highest score on all mentioned criteria:
• Completeness
• Conciseness (for label)
• Accuracy (of rule_type)
• Consistency (of targets)
• Fidelity to Source Text (statements)
• Neutrality
• Actionability
• No Hallucination

Follow the format below:

```
{
"rule_id": "rule_id from the source/generated",
"label": "concise summary (5-25 words)",
"rule_type": "obligation" | "prohibition" | "permission" |
"exemption" |
"definition-application" | "safe-harbor" | "procedure" |
"clarification" | "deeming" | "condition-precedent" | "other",
"targets": ["WHO must comply, is prohibited, or is granted
permission"],
"statement": {
"action": "primary regulatory action",
"action_object": "direct object or recipient of the action",
"method": "HOW the action must be performed",
"constraints": [...],   // REQUIRED - [] if genuinely absent
"conditions": [...],    // REQUIRED - [] if genuinely absent
"exceptions": [...],    // REQUIRED - [] if genuinely absent
"penalties_or_consequences": [...],  // REQUIRED - null if
genuinely absent
"purpose": "stated objective",
"verbatim": "source quote establishing rule"
},
"citations": [...],   // REQUIRED - [] if genuinely absent
"examples": [...]    // REQUIRED - [] if genuinely absent
}
```

Only generate the corrected RuleUnit JSON and nothing else.

Inputs:
Source Text: {}
Incorrect RuleUnit: {}
Critique: {}

Prompt 4: Stage 3 regeneration prompt. Placeholders are filled at runtime with the source section text, the failing rule unit extraction, and the Judge 3 critique.

H.3 Judge Prompts

The three judge prompts below are invoked sequentially during the multi-criteria evaluation stage to assess section metadata (Judge 1), definitions (Judge 2), and individual rule units (Judge 3). Each prompt instructs the judge LLM to score the extraction independently on per-stage criteria and to produce structured natural-language critiques. These critiques are subsequently passed verbatim as input to the corresponding regeneration prompts in Section H.2, forming the closed judge-repair loop at the core of De Jure. All prompts share the same 0–5 scoring rubric and instruct the judge to assign 5 to inapplicable criteria, ensuring the threshold θ is not penalized by structural variation across regulatory sections.

Prompt 5 | Judge 1: Section Metadata Evaluation

Evaluate whether the extracted section metadata is substantially accurate based on the following criteria:

1. Completeness
Major metadata elements should be extracted and populated:
• `section_cite` should be present and identify the correct section
• `title` should be captured if clearly present in the source
• `effective_dates` should include at least the primary temporal event (if any)
• `notes` (optional) may capture additional context
• `x_extensions` (optional) may include non-standard metadata
Missing critical fields (`section_cite`, `title` when present) are significant issues. Missing optional fields or secondary dates are minor issues.

2. Fidelity to Source Text
Notes and x_extensions should reasonably reflect the source content. Direct quotes, close paraphrasing, or reasonable interpretations are all acceptable. Minor rewording or normalization of language is acceptable.

3. Non-Hallucination
Fields should only be populated when corresponding information exists in the source. Do not fabricate dates, citations, or contextual information. Event types in `effective_dates` should be grounded in the source (normalized terminology acceptable, e.g., "enacted" → "adopted"). This criterion is strict: hallucinated information is a significant problem.

4. Title Quality
Title should accurately reflect the section title if present. Minor formatting variations are acceptable. Null is appropriate if no title exists.

5. Precision of Citations and Dates
Section citations should identify the correct section (minor formatting differences acceptable). Dates should be correct to at least the month/year level. If multiple effective dates exist, capturing the primary date is essential; missing secondary dates is a minor issue.

6. Reasonable Population of Optional Fields
When `notes` or `x_extensions` are populated, they should add value. Omitting these fields when relevant information exists is a minor issue, not a major deficiency.

Scoring Guidelines (per criterion):
• **5.0**: Fully satisfied with no errors
• **4.0–4.9**: Mostly satisfied with minor issues
• **3.0–3.9**: Partially satisfied with notable gaps or minor inaccuracies
• **2.0–2.9**: Poorly satisfied with significant omissions or errors
• **1.0–1.9**: Barely satisfied with major problems
• **0.0–0.9**: Not satisfied – critical failures or fabrications
Score each of the 6 criteria independently. If a criterion is not applicable, assign 5. Report the average as the final score.

Inputs:
Source Text: {}
Extracted Metadata: {}

Prompt 5: Judge 1 evaluation prompt for section metadata (6 criteria). Output scores and critiques are consumed by Prompt 2.

Prompt 6 | Judge 2: Definitions Evaluation

Evaluate whether the extracted definitions and related rule extractions are substantially accurate based on the following criteria:

1. Completeness
Major definitional statements should be captured, including:
• Primary positive definitions ("X means Y", "X includes Y")
• Significant negative definitions or exclusions ("X is not Y", "X does not include Y")
• Important context such as major exceptions or limitations
Missing primary definitions or major exclusions are significant issues. Missing minor qualifications or secondary cross-references are minor issues.

2. Fidelity to Source Text
Extracted terms and definitions should reasonably reflect the source. Direct quotes, close paraphrasing, or reasonable interpretations preserving core meaning are all acceptable. Definitions should not substantially contradict the source or misrepresent the term's meaning.

3. No Hallucination or Fabrication
Extract only definitions present in the source. Do not invent terms, definitions, or context not grounded in the text. This criterion is strict: fabricated content is a significant problem. Reasonable interpretations of existing content are acceptable.

4. Precision and Formatting
Terms should be substantially accurate in spelling and punctuation. Important references (e.g., to statutes or sections) should be captured. Each extracted definition should be clear and understandable. Minor formatting inconsistencies are acceptable if core meaning is preserved.

5. Quality of Terms
Each extracted term should reasonably match the terminology used in the source. Terms should accurately represent the intended meaning and context. Minor variations in term format or phrasing are acceptable if they do not misrepresent the definition.

Scoring Guidelines (per criterion):
• **5.0**: Fully satisfied with no errors
• **4.0–4.9**: Mostly satisfied with minor issues
• **3.0–3.9**: Partially satisfied with notable gaps or minor inaccuracies
• **2.0–2.9**: Poorly satisfied with significant omissions or errors
• **1.0–1.9**: Barely satisfied with major problems
• **0.0–0.9**: Not satisfied – critical failures or fabrications

Score each of the 5 criteria independently. If a criterion is not applicable, assign 5. Report the average as the final score.

Inputs:
Source Text: {}
Extracted Definitions: {}

Prompt 6: Judge 2 evaluation prompt for definitions (5 criteria). Output scores and critiques are consumed by Prompt 3.

Prompt 7 | Judge 3: Rule Unit Evaluation

You are an expert evaluator assessing structured rule extraction from regulatory documents. You will receive the original source text and an extracted RuleUnit. Evaluate each component based on the following criteria:

1. Completeness
Core components (`label`, `rule_type`, `targets`, `action`, `action_object`) are significant if missing. Secondary components (`method`, `constraints`, `conditions`) and optional fields when absent are minor issues.

2. Conciseness
The label should be reasonably brief, summarizing the rule while preserving important meaning. Slight wordiness is acceptable. Significant deviation from the rule's meaning or scope should be noted.

3. Accuracy
The `rule_type` should reasonably represent the type of rule in the source. Minor classification judgment calls are acceptable. Significant misclassification that misrepresents the rule's fundamental nature is a problem.

4. Consistency
Targets should reasonably align with the source. Minor terminology variations that preserve meaning are acceptable. Targets should not significantly contradict or misrepresent who the rule applies to.

5. Fidelity to Source Text
Statement components should reasonably reflect the source. Conditions, exceptions, and constraints should capture the primary requirements. Minor deviations in phrasing that do not affect legal interpretation are acceptable. Significant alterations of scope or omission of critical qualifiers should be noted.

6. Neutrality
Labels and statements should present the source in a balanced manner. Minor interpretive choices are acceptable. Significant bias or misrepresentation of the rule's intent should be noted.

7. Actionability
Action and action_object should provide reasonably clear guidance, understandable and usable in a business context. Minor ambiguity is acceptable if the core action and object are identifiable. Excessive abstraction that obscures what must be done is problematic.

8. Non-Hallucination
Extract only rule units present in the source. Do not invent rule components, conditions, targets, or context not grounded in the text. This criterion is strict: fabricated content is a significant problem. Reasonable interpretations of existing content are acceptable.

Scoring Guidelines (per criterion):
- **5.0**: Fully satisfied with no errors
- **4.0–4.9**: Mostly satisfied with minor issues
- **3.0–3.9**: Partially satisfied with notable gaps or minor inaccuracies
- **2.0–2.9**: Poorly satisfied with significant omissions or errors
- **1.0–1.9**: Barely satisfied with major problems
- **0.0–0.9**: Not satisfied – critical failures or fabrications

Score each of the 8 criteria independently. If a criterion is not applicable, assign 5. Report the average as the final score. Focus on significant errors that materially affect accuracy, completeness, or usability of the extracted rule.

Inputs:
Source Text: {}
Extracted RuleUnit: {}

Prompt 7: Judge 3 evaluation prompt for per-rule quality (8 criteria). Output scores and critiques are consumed by Prompt 4.

H.4 Downstream RAG Evaluation Prompts

The following four prompts drive Experiment 3 (Section 4.4): question generation (Prompt 8), RAG answer generation (Prompt 9), Oracle answer generation (Prompt 10), and the ranking judge (Prompt 11).

H.4.1 Question Generation Prompt

Qwen3-VL-8B-Instruct is prompted with a HIPAA regulatory document to produce compliance-grounded questions fully answerable from the passage alone, spanning factual, conditional, and analytical types. The question count and passage content are injected at the {} placeholders.

Prompt 8 | Question Generation: Downstream RAG Evaluation

You are a senior legal analyst and compliance expert with over two decades of experience specializing in U.S. healthcare law, HIPAA regulations (45 CFR Parts 160 and 164), and regulatory compliance frameworks.

Areas of Expertise:
- HIPAA Privacy Rule, Security Rule, and Breach Notification Rule
- Protected Health Information (PHI) handling, use, and disclosure requirements
- Covered entities, business associates, and their obligations
- Administrative, physical, and technical safeguards
- Enforcement, penalties, and compliance procedures

Your role is to generate precise, grounded, and legally meaningful questions from regulatory passages. You think like both a compliance officer and a litigator: you understand nuance, edge cases, and the practical implications of regulatory language. Given the following regulatory passage, generate exactly {} questions that a compliance officer, legal analyst, or auditor might ask.

Strict Rules:
- Every question must be directly and fully answerable from the passage alone
- Do NOT introduce concepts, entities, or scenarios not present in the passage
- Do NOT ask questions requiring outside knowledge or inference beyond the passage
- Questions must be diverse: cover who, what, when, how, and under what conditions
- Questions must be specific and unambiguous
- Questions should vary in complexity: mix factual, conditional, and analytical types
- Each question must stand alone as a complete, clear sentence

Output Format:
Return ONLY a valid Python list of {} strings. No explanation, no preamble, no commentary.

Example:
```
["Question one?", "Question two?", ...]
```

Input:
Passage: {}

Prompt 8: Question generation prompt for downstream evaluation on compliance QA via RAG (Experiment 3). Injected with the target passage and desired question count; output is a Python list used to construct the 100-question evaluation set.

H.4.2 RAG Answer Generation Prompt

Prompt 9 | RAG Answer Generation

You are a senior HIPAA compliance attorney and regulatory expert with over 20 years of experience interpreting U.S. healthcare law, specifically:

• HIPAA Privacy Rule, Security Rule, and Breach Notification Rule (45 CFR Parts 160 & 164)
• Protected Health Information (PHI) — use, disclosure, and safeguarding obligations
• Covered entity and business associate requirements and liabilities
• Authorization, consent, and minimum necessary standards
• Enforcement mechanisms, civil/criminal penalties, and audit procedures

You answer questions by reasoning strictly from the provided regulatory rules and context. You are precise, structured, and cite specific rules when possible. You never fabricate regulatory text or invent citations.

You are answering a compliance question grounded in HIPAA regulations. You have been provided with the most relevant regulatory rules retrieved from a HIPAA knowledge base.

QUESTION:
{}

RETRIEVED REGULATORY RULES:
{}

Instructions:
• Answer the question using ONLY the information present in the retrieved rules above
• If the answer is explicitly stated, cite the relevant rule ID or CFR citation
• If multiple rules are relevant, synthesize them into a coherent answer
• If the retrieved rules are insufficient to fully answer the question, clearly state what is and is not covered
• Do NOT introduce outside legal knowledge or hallucinate regulatory text
• Structure your answer clearly: lead with the direct answer, then explain with supporting rule references

ANSWER:

Prompt 9: RAG answer generation prompt for the five retrieval-based systems. Injected with the evaluation question and the top-k retrieved rules from the system's vector database. Used with Claude 4.6 Sonnet at τ=0.1.

H.4.3 Oracle Answer Generation Prompt

Prompt 10 | Oracle Answer Generation

You are a senior HIPAA compliance attorney and regulatory expert with over 20 years of experience interpreting U.S. healthcare law, specifically:

• HIPAA Privacy Rule, Security Rule, and Breach Notification Rule (45 CFR Parts 160 & 164)
• Protected Health Information (PHI) — use, disclosure, and safeguarding obligations
• Covered entity and business associate requirements and liabilities
• Authorization, consent, and minimum necessary standards
• Enforcement mechanisms, civil/criminal penalties, and audit procedures

You answer questions by reasoning strictly from the provided regulatory document. You are precise, structured, and cite specific sections when possible. You never fabricate regulatory text or invent citations.

QUESTION:
{}

COMPLETE REGULATORY DOCUMENT:
{}

Instructions:
• Answer using ONLY the information in the document above
• Cite specific CFR sections and rule IDs when relevant
• Acknowledge when the document does not fully address the question
• Structure your answer clearly: direct answer first, then supporting citations

ANSWER:

Prompt 10: Oracle answer generation prompt. Identical persona to Prompt 9 but receives the complete regulatory document rather than retrieved rules, serving as an upper bound on answer quality unconstrained by retrieval.

H.4.4 Ranking Judge Prompt

Prompt 11 | Ranking Judge: Downstream QA

You are an expert evaluator assessing the quality of AI-generated answers to HIPAA compliance questions. You have deep expertise in:

• Legal and regulatory accuracy in healthcare compliance
• Quality of reasoning and argumentation
• Information retrieval and grounding in source documents
• Clarity of communication for compliance professionals

Your role is to objectively rank multiple answers against the original regulatory document and question. You are impartial — you evaluate based on evidence, not style preference.

You will rank {num_systems} answers from best (rank 1) to worst (rank {num_systems}) on each criterion. Each rank must be unique — no ties are allowed.

QUESTION:
{query}

{answer_blocks}

ORIGINAL REGULATORY DOCUMENT:
{doc}

Evaluation Criteria — rank all {num_systems} answers for each:

1. Factual Grounding: Which answers stay strictly within the regulatory document? Penalize any answer that introduces outside information or hallucinated text.

2. Completeness: Which answers more fully address all aspects of the question, covering relevant conditions, exceptions, and edge cases in the document?

3. Regulatory Precision: Which answers more accurately reference specific rule IDs, CFR citations, and legal terminology from the source document?

4. Handling of Ambiguity: When the document does not fully answer the question, which answers more appropriately acknowledge the limitation rather than speculating?

5. Practical Actionability: Which answers are most useful to a compliance professional making a real decision based on this regulation?

Output Instructions:
• Return ONLY a valid Python dictionary — no explanation, no preamble, no commentary
• Keys are criterion names; values are dicts mapping answer label to integer rank (1=best, {num_systems}=worst)
• All {num_systems} answer labels must appear in every criterion with unique ranks 1–{num_systems}
• After the criteria, include an "overall" key with the same structure

Example format:

```
{
"factual_grounding": {"Answer 1": 1, "Answer 2": 4, ...},
"completeness": {"Answer 1": 2, ...},
...
"overall": {"Answer 1": 1, ...}
}
```

Prompt 11: Ranking judge for the downstream QA evaluation. The six system answers are presented in randomized order (seeded per question) to eliminate positional bias. The judge produces a strict 1–6 ranking per criterion plus an overall ranking, from which pairwise win rates are derived. Used with CompassJudger at τ=0.1.

I Extraction Schema

De Jure structures LLM outputs using guided JSON generation, with Pydantic-defined schemas enforcing the extraction of discrete rule units from the source text.

Schema 1 | SectionExtraction (top-level)

Top-level container returned by the extraction pipeline. All sub-objects use `additionalProperties: false`.

Root fields

`section_meta`	*SectionMeta*	**req**	Metadata about the regulatory section
`definitions`	*DefinitionEntry[ ]*	opt	Defined terms extracted from the section
`extracted_rules`	*RuleUnit[ ]*	**req**	List of structured rule extractions

SectionMeta

`schema_version`	*string*	opt ("1.0.0")	Version of the extraction schema
`section_cite`	*string*	**req**	Citation of the section
`title`	*string*	**req**	Title of the section
`effective_dates`	*EffectiveDate[ ]*	opt	Timeline of adoptions, amendments, rescissions
`notes`	*string*	opt	Additional context
`x_extensions`	*object*	opt	Custom extension namespace

EffectiveDate

`event`	*enum*	**req**	`adopted` | `amended` | `rescinded` | `note`
`date`	*string*	**req**	ISO date, e.g. `2023-02-27`
`fr_citation`	*string*	opt	Federal Register citation, if available

DefinitionEntry

`term`	*string*	**req**	The term being defined
`text`	*string*	**req**	Definition text from source
`scope`	*enum*	opt	`section` | `part` | `act` | `context-dependent`; leave null if unstated
`cross_references`	*CrossRef[ ]*	opt	Related regulatory references

Schema 2 | RuleUnit

A single extracted regulatory rule with type classification, targets, decomposed statement, citations, and judge scores populated post-evaluation.

RuleUnit

`rule_id`	*string*	**req**	Unique identifier for this rule
`label`	*string*	**req**	Short summary of the rule
`rule_type`	*RuleType*	**req**	Classification of the rule
`targets`	*Target[ ]*	**req**	Entities subject to the rule
`statement`	*Statement*	**req**	Decomposed regulatory requirement
`citations`	*Citations*	opt	Supporting regulatory citations
`judge_score`	*JudgeScore*	opt	Aggregated evaluation scores
`examples`	*string[ ]*	opt	Examples lifted from source text, if available

RuleType

`type`	*enum*	**req**	`obligation` | `prohibition` | `permission` | `exemption` | `definition-application` | `safe-harbor` | `procedure` | `clarification` | `deeming` | `condition-precedent` | `other`
`other_label`	*string*	opt	Required when `type='other'`; descriptive label for custom type

Target

The entity/role subject to the obligation, prohibition, or permission — WHO must comply, not the recipient or intermediary.

`role`	*string*	**req**	Role/entity subject to the rule
`qualifiers`	*string*	opt	Narrowing qualifiers (e.g. "foreign private issuer")

Citations

`text`	*string* | *string[ ]*	opt	Textual citation(s) to supporting sources

Schema 2: Individual rule unit schema with detailed type taxonomy (11 categories), target identification, and citation support.

Schema 3 | Statement within a rule unit

Decomposition of a regulatory requirement into action, object, method, constraints, conditions, exceptions, and verbatim source text.

Statement

`action`	*string*	**req**	Primary regulatory action as verb phrase
`action_object`	*string*	opt	Direct object or recipient of the action
`method`	*string*	opt	How the action must be performed
`constraints`	*Constraint[ ]*	**req**	Limits imposed on applying this rule (default `[]`)
`conditions`	*Condition[ ]*	**req**	Prerequisites for this rule to apply (default `[]`)
`exceptions`	*ExceptionItem[ ]*	**req**	Places where this rule may not apply (default `[]`)
`penalties_or_consequences`	*PenaltiesOrConsequences[ ]*	opt	Penalties or consequences of a rule action
`purpose`	*string*	opt	Stated purpose/objective if explicit in source
`verbatim`	*string*	**req**	Exact quoted text from the source

Constraint

`text`	*string*	**req**	Constraint description
`applies_to`	*string*	opt	Entity the constraint binds to

Condition

`trigger`	*string*	**req**	Triggering event or conditional text
`time_window`	*TimeWindow*	opt	Temporal boundaries for the condition
`cross_references`	*CrossRef[ ]*	opt	Related regulatory cross-references

TimeWindow

`start`	*string*	opt	Start of the time window
`end`	*string*	opt	End of the time window
`zone`	*enum*	opt	`ET` | `EST` | `EDT`

ExceptionItem

`text`	*string*	**req**	Main exception description
`cross_references`	*CrossRef[ ]*	opt	Related regulatory references

PenaltiesOrConsequences

`text`	*string*	**req**	Penalty/consequence description
`cross_references`	*CrossRef[ ]*	opt	Related regulatory references

CrossRef

`type`	*enum*	**req**	`CFR` | `Rule` | `Form` | `USC` | `Release` | `Regulation` | `Note` | `Other`
`cite`	*string*	**req**	Citation string
`summary`	*string*	opt	Brief summary of the cross-reference

Schema 3: Statement decomposition with conditions, constraints, exceptions, penalties, and cross-references. Arrays default to `[]`; optional scalars default to `null`.

Schema 4 | Judge Evaluation Schemas (Judges 1, 2, 3)

Multi-step evaluation framework for extraction quality. Each metric carries an integer `Score` $\in [0, 5]$ and a textual `Justification`. All fields in every judge step are **required**.

Metric *(shared unit)*

`Score`	*integer*	**req**	$0 \leq$ `Score` ≤ 5
`Justification`	*string*	**req**	Textual justification for the score

Step1Judge — *Section-level metadata extraction*

`Completeness`	Coverage of all required metadata fields
`Fidelity_to_source_text`	Faithfulness to the original text
`Non_hallucination`	Absence of fabricated content
`Title_Quality`	Appropriateness and clarity of the extracted title
`Precision_of_Citations_and_Dates`	Accuracy of citation strings and date formats
`Reasonable_Population_of_Optional_Fields`	Sensible use of optional fields

Step2Judge — *Definition extraction*

`Completeness`	All defined terms captured
`Fidelity_to_Source_Text`	Definitions faithful to source
`No_Hallucination_or_Fabrication`	No invented terms or definitions
`Precision_and_Formatting`	Correct formatting and precision
`Quality_of_Terms`	Appropriateness and clarity of extracted terms

Step3Judge — *Per-rule unit evaluation*

`Completeness`	All rule components present
`Conciseness`	Language is brief while preserving meaning
`Accuracy`	Rule type correctly classifies the source
`Consistency`	Targets align with the source
`Fidelity_to_source_text`	Statement reflects the source faithfully
`Neutrality`	Balanced, unbiased presentation
`Actionability`	Clear, usable guidance with reasonable abstraction and minimal ambiguity
`Non_hallucination`	No fabricated rule components

JudgeScore *(aggregate)*

`step1`	*Step1Judge*	**req**	Step 1 evaluation
`step2`	*Step2Judge*	**req**	Step 2 evaluation
`step3`	*Step3Judge*	**req**	Step 3 evaluation
`Final`	*integer*	opt	Overall (average) score $\in [0, 5]$
`Notes`	*string*	opt	Free-text evaluator notes

Schema 4: Three-step judge framework — Step 1 evaluates metadata (6 metrics), Step 2 evaluates definitions (5 metrics), Step 3 evaluates each rule unit (8 metrics). All metrics use the shared `Metric` type with `Score` $\in [0, 5]$ and `Justification`.

J Sample Evaluation Questions

Table 17 presents a representative sample of 10 questions from the 100-question evaluation set used in Experiment 3 (Section 4.4), generated by Qwen3-VL-8B-Instruct from HIPAA regulatory passages using the prompt in Appendix H.4.1. The questions span five distinct HIPAA topic areas (general use and disclosure standards, permitted uses, business associate obligations, genetic information protections, and reproductive health privacy) and mix factual, conditional, and analytical question types, collectively stress-testing whether a RAG system can retrieve and reason over structurally and semantically diverse regulatory provisions.

#	Question
1	Who may not use or disclose protected health information except as permitted or required by this subpart?
2	When is a covered entity permitted to use or disclose protected health information for treatment, payment, or health care operations?
3	What conditions must be met for a use or disclosure to be considered incident to a permitted use or disclosure?
4	What are the required disclosures of protected health information by a covered entity?
5	What are the limitations on a business associate's use and disclosure of protected health information?
6	What is prohibited regarding the use and disclosure of genetic information for underwriting purposes?
7	What activities are considered "underwriting purposes" with respect to genetic information?
8	What constitutes a "sale of protected health information"?
9	Under what conditions does the prohibition on using reproductive health care information apply?
10	What is presumed about the lawfulness of reproductive health care provided by another person, and what can override that presumption?

Table 17: Ten representative questions from the 100-question evaluation set used in Experiment 3. Each question is fully answerable from its source passage alone. The set spans five HIPAA topic areas and three question types, reflecting the diversity requirements specified in the generation prompt (Appendix H.4.1).