applied sciences

MDPI

Article

Reconstruction of Criminal Event Timelines from Property Crime Investigation Documents Based on a Hybrid LLM-Rule Pipeline

Gwang-Jae Won, Ro-Seop Park, June Tschoy and Yong-Tae Lee *

Department of Forensic Information Science and Technology, Hallym University, Chuncheon 24252, Republic of Korea; wgj@hallym.ac.kr (G.-J.W.); rspark@hallym.ac.kr (R.-S.P.)
* Correspondence: ytlee@hallym.ac.kr; Tel.: +82-33-248-1984

Abstract

Fact-finding in criminal proceedings reconstructs past events into a coherent narrative based on fragmented evidence and testimony, yet remains vulnerable to cognitive confirmation bias. While automating timeline construction using large language models (LLMs) offers decision support, confidentiality-critical local models are prone to temporal hallucinations, inverting causal orders or generating groundless timestamps. To mitigate these vulnerabilities, this study proposes Deep-ARC (Analysis, Reconstruction, and Contradiction detection for criminal timeline construction), a hybrid LLM-rule pipeline that algorithmically regulates model reasoning through legal domain schemas and deterministic rule logic. Deep-ARC injects statutory crime element schemas to categorize event phases and enforces phase-aware boundary clamping grounded on explicit timestamps, systematically resolving sequential contradictions during missing time inference. In proof-of-concept experiments focusing on corporate property crimes (breach of trust and embezzlement) with Gemma-4-31B, Deep-ARC achieved a Micro PMR (exact sequence match ratio) of 0.343 and a Kendall's Tau (τ) of 0.861, representing an approximate fourfold improvement over the unconstrained baseline and demonstrating higher sequencing consistency than large-scale commercial LLMs. These results indicate that coupling domain-specific structural constraints with deterministic algorithms offers an effective and robust strategy for providing an objective chronological sequence to assist fact-finding without relying exclusively on parameter scaling.

Keywords: legal event extraction; temporal reasoning; timeline sequencing; large language model; rule engine; boundary clamping; criminal investigation; confirmation bias

Check for updates

Academic Editor: Piotr A. Kowalski

Received: 23 September 2026
Revised: 22 October 2026
Accepted: 23 October 2026
Published: 25 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

In criminal proceedings, fact-finding to establish the substantive truth is the process of reconstructing past events into a narrative based on fragmented evidence and statements [1,2]. In particular, an investigative environment heavily reliant on testimonial rather than physical evidence increases cognitive load. This condition makes investigators susceptible to confirmation bias, leading them to selectively accept information that aligns with initial hypotheses or preexisting beliefs [3–5].

To address cognitive distortion issues arising during criminal investigations and the fact-finding process, the necessity for analytical systems that externalize the adjudicator's reasoning process was proposed early on [6]. Recently, research has progressed toward automated timeline construction and contradiction detection based on generative artificial

intelligence (AI) to objectively review cases and extract key issues from criminal investigation documents [7,8].

With the recent advancement of large language models (LLMs), natural language processing-based information extraction technologies are being introduced into the legal and investigative domains. However, because criminal investigation documents contain sensitive personal information and undisclosed investigative secrets, transmitting data to commercial LLMs based on external cloud APIs is restricted. Therefore, on-premises local LLMs that can operate within the internal network environment of investigative agencies must be utilized. Yet, this deployment poses several persistent challenges. First, the open-ended generation characteristics of general-purpose LLMs induce "legal hallucinations," where legally meaningless procedural facts are extracted as criminal acts or non-existent facts are fabricated [9–11]. Second, since criminal investigation documents mostly contain implicit timestamps and missing temporal information, these models demonstrate vulnerabilities in logically reasoning about complex temporal sequences among multiple events [12,13]. That is, existing prompt engineering or simple retrieval-augmented generation (RAG) methods alone are insufficient to control the temporal contradictions generated by LLMs.

To overcome these limitations, this study proposes Deep-ARC (Analysis, Reconstruction, and Contradiction detection for criminal timeline construction), a rule-constrained pipeline that controls LLM reasoning by combining legal domain knowledge with a rule engine. During the event extraction process, the system injects a crime element schema to ensure legal significance. Additionally, it controls the LLM to perform missing time inference exclusively within the temporal boundaries defined by preceding and following explicit timestamps. Finally, by correcting temporal contradictions via phase-aware boundary clamping through the rule engine, it constructs a timeline that maintains causal-temporal consistency.

The main contributions of this paper are threefold:

1. We propose a crime element schema that structurally prevents sequential contradictions across legal phase inversions and causal sequence errors inherent in general-purpose LLMs, thereby improving temporal sequencing consistency.
2. We develop an algorithmic phase-aware boundary clamping mechanism anchored on explicit timestamps, ensuring robust sequencing performance regardless of temporal clue density.
3. We empirically validate the practical utility of our pipeline through experiments on an on-premises local LLM (Gemma-4-31B), confirming that upstream schema classification accuracy serves as the primary factor governing downstream performance.

2. Related Work

2.1. Event Extraction with LLMs in Legal Domain

With the advancement of natural language processing (NLP) technologies, research on event extraction to structure unstructured texts in the legal domain has been continuously conducted. In early studies, Lagos et al. [14] established a legal ontology framework to structure events and attempted forward reasoning, laying the groundwork for automated legal timeline construction. However, that research focused solely on unidirectional logical connections among explicitly revealed information in documents, without addressing retroactive consistency correction, such as recovering missing information to fill temporal gaps between events or re-verifying and correcting errors in inferred information.

Recently, with the development of large language models (LLMs), efforts to utilize artificial intelligence to increase the efficiency of information extraction have become prominent. Li et al. [15] proposed a schema induction method that incrementally expands hierarchical and temporal events using common sense knowledge, while Maoro et al. [16] designed a se-

mantic triple-based information extraction pipeline and a human-in-the-loop control model to ensure transparency and post-audit verifiability across criminal justice. Furthermore, He et al. [17] introduced a multi-round question-answering framework into a machine reading comprehension (MRC) model to effectively capture complex hierarchical dependencies among event elements. However, these studies primarily focus on commonsense event schemas or entity and relation extraction in general domains, presenting limitations in extracting events that involve domain-specific terminology or legal causation, such as in the legal field.

Yue et al. [18] introduced a zero-shot event extraction framework utilizing LLMs in the legal domain. Similarly, Ren [19] utilized a BERT-RCNN fusion model with hierarchical attention mechanisms for multi-label classification of judgment elements from legal instruments. However, these legal element extraction approaches primarily focus on static, sentence-level classification of case facts; they do not model temporal dependencies between extracted elements or reconstruct the chronological sequence of events within the investigative narrative.

In parallel, recent studies have attempted to enhance extraction performance by combining external knowledge or structural constraints into prompts. Du and Cardie [20] formulated event argument extraction as a question-answering task by transforming schema role descriptions into natural question templates to enhance semantic transfer. Similarly, Han et al. [21] incorporated probabilistic domain knowledge and structural constraints into a neural framework via Lagrangian relaxation to jointly optimize event type and temporal relation extraction. Furthermore, Mandravickaite et al. [22] introduced knowledge graph grounding (KG-grounding) and LLM ensembles.

However, these recent methodologies still exhibit notable technical limitations. Specifically, question-answering formulations often extract argument spans in isolation without modeling complex temporal dependencies across multiple events [20]. Furthermore, while distributional constraints enhance local extraction, enforcing them over noisy extractions risks error propagation and remains confined to adjacent sentences, without addressing cross-document temporal sequences [21]. Moreover, simple ensemble consensus policies may conversely omit contextually important causal information [22].

Because legal document analysis demands extreme precision—where subtle omissions or hallucinations directly undermine judicial integrity—relying solely on open-ended generation or generic filtering is insufficient. Instead, a structured approach that enforces statutory crime element schemas as top-down constraints is essential.

2.2. Temporal Reasoning and Event Sequencing

To arrange events extracted from unstructured text into a chronological timeline, the normalization of explicit time information and the inference of relationships among implicit events must be performed simultaneously. In early stages, research was conducted to extract explicit temporal expressions and convert them into standardized formats using rule-based time normalization tools such as the TimeML [23] specification and HeidelTime [24]. Subsequently, to infer logical sequential relationships between events, Leeuwenberg et al. [25] introduced Point Algebra and constraint propagation algorithms to integrate temporal clues appearing in the text into a single consistent chronological graph. However, because this research was designed on the premise of removing conflicting information or assuming a single narrative structure to maintain logical consistency, limitations exist in applying it to practical environments where conflicting statements coexist (e.g., criminal investigation documents).

Maintaining global temporal consistency in event ordering has fundamentally evolved from interval-based constraint satisfaction frameworks rooted in Allen's Interval Alge-

bra [26]. Recently, such interval logics have been extended to multi-document engineering applications; for instance, Zeng et al. [27] incorporated subsets of Allen's interval relations into spatiotemporal lattices to evaluate cross-document event consistency. In computational linguistics, relational temporal constraints have predominantly been formalized via Integer Linear Programming (ILP) to resolve transitivity contradictions across local pairwise predictions [28].

However, applying existing constraint-based frameworks to criminal investigation documents entails certain operational constraints. Global ILP formulations rely on combinatorial solvers and dense $O(N^2)$ pairwise relation evaluations, which introduce substantial computational overhead in resource-constrained, on-premise local LLM environments (air-gapped). Furthermore, lattice-based spatiotemporal linkages face challenges in addressing domain-specific behavioral characteristics, such as preparatory actions occurring well in advance of execution or underspecified temporal expressions.

To address these considerations, Deep-ARC's Phase-aware Boundary Clamping provides a deterministic alternative to generic mathematical programming solvers. By mapping events onto statutory Legal Phases (Pre-crime -> Crime Execution -> Post-crime), the proposed framework establishes upper and lower temporal bounds directly on event intervals. This domain-grounded formulation structurally prevents sequential contradictions through hierarchical multi-key sorting in O(NlogN) time, resolving missing timestamps without requiring external optimization libraries.

Recently, attempts to infer chronological structures and causal relationships between events utilizing large language models (LLMs) have actively progressed. Liao et al. [29] and Ong et al. [30] proposed methodologies that restore the causal flow (timeline) of past events utilizing relation-aware structures and temporal logic rules, thereby securing the temporal consistency of data and generating predictions and responses aligning with the context. However, existing methods typically target highly structured temporal knowledge graphs (e.g., GenTKG [29]) or sequentially accumulated dialogue histories (e.g., THEANINE [30]). Consequently, they exhibit fundamental limitations when transferred to unstructured legal narratives, where the actual occurrence order of events (factual time) and the descriptive order of statements (procedural time) are complexly intertwined within a single document.

To overcome these limitations, recent studies attempting to capture multi-granularity time or graph-based global interactions have emerged.

Chen et al. [31] attempted to learn sequence relationships by aggregating embeddings of different time units using sinusoidal positional encoding and multi-granularity temporal aggregation. Furthermore, to verify and optimize the validity of causal relationships, Mandravickaite et al. [22] compared the causality among extracted events against a global news database (GDELT).

Nevertheless, these studies also demonstrated limitations in explicit logical reasoning penetrating the entirety of complex, nonlinear narratives. Such multi-granularity aggregation relies heavily on implicit embeddings over structured knowledge graphs, thereby failing to resolve complex temporal constraints or enforce explicit chronological sequences across unstructured documents [31]. Moreover, verification through external databases was limited to merely identifying the presence of simple interactions (directional proxies), failing to form a macroscopic timeline through inter-document graph fusion [22].

Concurrently, in the field of information ordering, neural network-based structural studies have attempted to capture the global dependencies of the entire sequence. Gong et al. [32] introduced a Pointer Network to propose a model that infers the optimal order of a randomly shuffled set of sentences end-to-end. Cui et al. [33] combined a self-attention mechanism to implement ATTOrderNet, which learns high-dimensional context regardless of input order (order-invariant). However, these models primarily rely on the abstract coherence of text

or specific lexical patterns (e.g., 'first', 'second'), and lack separate structural mechanisms reflecting explicit temporal encoding or legal causation essential for timeline construction, thus exhibiting distinct limitations in inferring the complex sequential relationships within criminal investigation documents.

Documents in the legal domain entail significant difficulty due to their unique grammatical complexities. In Evaluations on LEXTIME [34], a recently proposed legal reasoning benchmark, modern LLMs perform relatively well when aligning explicit dates. However, they struggle to deduce complex sequential relationships among multiple events. This vulnerability stems from domain-specific linguistic hurdles, such as passive voice, double negatives, subordinate clause embeddings, and intricate paraphrasing.

To mitigate these limitations, Won et al. [8] proposed an Element Fact extraction pipeline that structures the unstructured narratives of legal documents through phase-based classification and strict admissibility criteria.

That study demonstrated that a hybrid context configuration, which provides both raw text and refined facts, functions as an attentional signal amplifier for LLMs. By simultaneously suppressing false positive entailments and false negative contradictions, this approach significantly improved relative temporal reasoning accuracy.

However, because the evaluation structure of that study is restricted to binary entailment classification of contradictions between two events (Yes/No QA) on the LEXTIME benchmark, clear technical limitations exist in directly applying it to the tasks of multi-event timeline sequencing and full timeline reconstruction penetrating the entire complex, nonlinear narrative.

In summary, completely reconstructing criminal investigation records where factual event timelines and procedural statement timelines are intertwined requires moving beyond sole reliance on the unconstrained, open-ended generation of general-purpose LLMs. Therefore, this study expands the previous binary entailment classification research [8] to global timeline construction, proposing a hybrid LLM-rule pipeline (Deep-ARC) that integrates the global context comprehension capability of an LLM with the deterministic sorting of a Python algorithm. The proposed algorithmic pre-sorting and phase-aware boundary clamping mechanisms utilize explicit timestamps as anchors to structurally prevent legal phase inversions and causal sequence errors induced by the model, thereby ensuring a timeline that consistently preserves statutory temporal order.

Unlike the localized pairwise evaluation in [8], Deep-ARC extends to global multi-event timeline reconstruction by incorporating missing time inference for unanchored events and enforcing deterministic boundary clamping to prevent temporal inversions.

3. Methodology

To overcome the stochasticity and legal hallucinations inherent in the open-ended generation of large language models (LLMs), this study proposes Deep-ARC. This architecture integrates the probabilistic reasoning capabilities of LLMs with the deterministic constraints of a rule engine.

3.1. Deep-ARC System Overview

Deep-ARC systematically ensures the logical consistency of judicial narratives through a four-stage approach. As illustrated in Figure 1, the overall architecture progressively constructs a consistent timeline through four stages: (1) schema-constrained extraction (Module 1), (2) initial sequence structuring via skeleton assembly (Module 2), (3) boundary-constrained missing time inference (Module 3), and (4) phase-aware boundary clamping and final sequencing (Module 4).

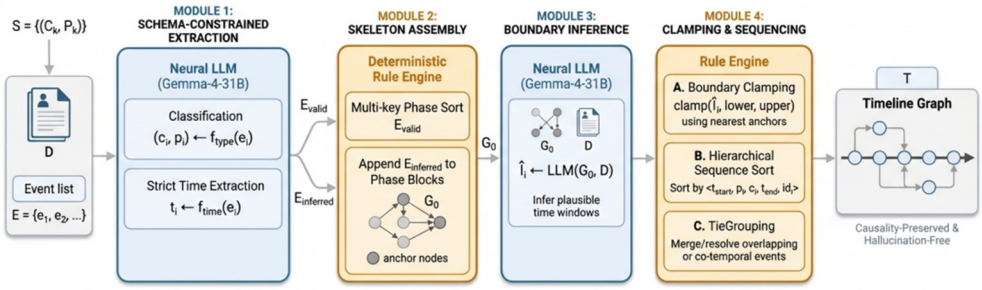


Figure 1. The overall architecture of the Deep-ARC pipeline. The system integrates Large Language Model (LLM)-based semantic reasoning (blue blocks) with a deterministic rule engine (yellow blocks). The pipeline strictly controls temporal hallucinations through schema-constrained extraction (Module 1) and explicit boundary clamping (Module 4) to ensure causal consistency.

Specifically, Module 1 executes schema-constrained extraction to yield categorized events alongside explicit timestamps. Module 2 performs initial sequence structuring via skeleton assembly, establishing temporal boundaries for missing time inference in Module 3. Finally, Module 4 enforces phase-aware boundary clamping and hierarchical sorting to generate the final chronological timeline.

The core tasks of this pipeline are the semantic classification and temporal sequencing mechanisms of events. To rigorously verify these mechanisms, this study excludes the boundary detection preprocessing step, which actively segments physical text boundaries from raw criminal investigation documents. In other words, the pipeline focuses on an inference control architecture that arranges pre-chunked target events into a global temporal sequence without sequential contradiction. This is based on the premise that legally significant factual relations have been refined and segmented in advance from a practical investigative perspective. By alternating Python-based structural controls with LLM-based semantic reasoning, this system ensures procedural chronological consistency while maintaining temporal consistency.

3.2. Schema-Constrained Classification (R1) and Explicit Time Extraction (R2)

The system receives a set of pre-chunked target events as input. Each event span corresponds to a core factual unit containing substantive factual relations, such as an expository statement in a complaint or an inquiry-response pair in an interrogation protocol. Legally meaningless spans, including procedural dialogues, have been removed during the preprocessing stage (for the definition of the crime element schema, refer to Tables A1 and A2 in Appendix A).

Let the input criminal investigation document be D and the set of target events be $E = \{e_1,e_2,\ldots,e_n\}$. The predefined crime element schema S is mathematically formulated as a set of tuples(C_k,P_k). Here, C_k represents a specific crime element type (e.g., B1_breach_of_trust_execution) and P_k denotes a hierarchical legal phase (e.g., preparation, execution, post-crime). The event classification function f_{type} maps each predefined event e_i to a strict schema element based on the document context:

$$c_i = f_{type}(e_i|D,S), \forall e_i \in E \quad (1)$$

The corresponding Legal Phase p_i is then deterministically assigned via a domain mapping function $p_i = \text{Phase}(c_i)$ governed by the statutory crime element schema.

Computationally, each statutory element is mapped to a hierarchical tuple (c_i,p_i), where specific crime element types c_i (such as Custody Authority $A1$, Pre-crime motive $A2$, or execution $B1$) operate under macro Legal Phase constraints p_i $Z \rightarrow A \rightarrow B \rightarrow C \rightarrow D$. Importantly, Deep-ARC models events as asserted within investigative transcripts

(representing deponents' allegations and testimonial claims under inquiry) rather than legally adjudicated facts, thereby functioning as an objective decision-support aid for fact-finding rather than an autonomous judicial determinant.

Simultaneously, the time extraction function f_{time} captures explicit timestamps. A strong constraint is imposed to prevent temporal hallucinations. That is, if an explicit date is not mentioned, the function must return a null value ($\emptyset$). Therefore, the explicit timestamp t_i is defined as follows:

$$t_i = f_{time}(e_i, c_i, D) \in T_{ISO} \cup \{\emptyset\} \quad (2)$$

Here, T_{ISO} denotes the domain of standardized ISO 8601 [35] timestamps. Through this process, the entire event set is strictly partitioned into an anchor subset with explicit timestamps $E_{valid} = e_i | t_i \neq \{\emptyset\}$ and an implicit subset without timestamps $E_{inferred} = e_i | t_i = \{\emptyset\}$. Events with relative temporal expressions (e.g., 'a few days later') are processed through the R4 missing time inference stage with phase-aware boundary clamping, while genuinely unknown times are preserved as null and ordered solely via the Legal Phase hierarchy.

3.3. Initial Sequence Structuring (R3-1) and Missing Time Inference (R4)

To reconstruct the chronological arrangement of events belonging to the implicit subset $E_{inferred}$, Deep-ARC utilizes an algorithm-based pre-sorting mechanism and document-grounded missing time inference.

Before LLM intervention, the Python rule engine establishes a structural framework to form the initial sequence skeleton (G_0), which implicitly forms a directed acyclic graph (DAG) through hierarchical sorting. The anchor events of E_{valid} undergo initial sequence structuring based on the legal phase p_i and the explicit timestamp t_i.

This process assembles the initial skeleton (G_0) of the entire timeline. Conversely, since the events of $E_{inferred}$ lack date information, they are appended chronologically based solely on the legal phase p_i.

This pre-sorted skeleton graph G_0 is provided to the LLM along with the entire document D. The system infers a valid time window $W_j = \left[t_j^{lower}, t_j^{upper}\right]$ for each event $e_j \in E_{inferred}$ that lacks explicit timestamps. The inferred boundaries are mathematically forced and adjusted (clamped) by the explicit timestamps of the logically preceding event $Pre(e_j)$ and succeeding $Post(e_j)$ within G_0:

$$W_j \subseteq \left[\max_{e_k \in Pre(e_j)} t_k, \min_{e_m \in Posi(e_j)} t_m\right] \quad (3)$$

This boundary constraint-based inference systematically ensures that the dates derived by the LLM remain strictly within the causal limits established by objective evidence. Furthermore, it converts the empty null values into explicit time ranges. In other words, when an event (e_j) containing a missing time inference is placed on the timeline, Equation (3) mathematically forces it to exist only within a specific interval. It sets the latest explicit timestamp among preceding events (max) as the lower bound and the earliest explicit timestamp among succeeding events (min) as the upper bound.

For the prompt architecture per module and the full text of detailed prompts, refer to Tables A3–A5 in Appendix B.

3.4. Rule-Based Phase-Aware Boundary Clamping and Final Timeline Sequencing (R3-2)

Once all events acquire accurate timestamps or inferred time ranges, the pipeline executes the final sequencing stage. To resolve residual logical tangles at the microscopic level that may be induced by the probabilistic generation of the LLM, a deterministic phase-aware boundary clamping algorithm is applied prior to final refinement. At this time, phase-aware boundary clamping targets only the R4 inferred times, leaving the R2 explicit timestamps unadjusted.

This algorithm merges E_{valid} and $E_{inferred}$ into a complete dataset E_{full} and enforces a multi-key hierarchical sorting criterion: (1) start time (t_{start}), (2) legal phase (p_i), (3) crime element type (c_i), (4) end time (t_{end}), and (5) event ID (id_i). This programmatic forced clamping automatically corrects sequential contradictions. Subsequently, the sequence undergoes final overlap resolution, which preserves all event IDs while merging co-temporal events to produce the final timeline (Algorithm 1).

In criminal investigation documents, temporal statements frequently appear as approximate expressions lacking specific days (e.g., 'around May 2023'). To reflect this domain characteristic and eliminate hallucinated false precision, monthly rounding is applied to inferred timestamps. The overlap resolution then employs a 3-rule co-temporal event merging strategy to consolidate concurrent events occurring within the same interval without imposing arbitrary sequential precedence.

Algorithm 1: Deep-ARC pipeline

Input: Document D, Event spans E, Schema S
Output: Ordered timeline T

//Stage 1: Schema-Constrained Classification (R1)
parallel for each $e_i \in E$ **do**
  $(c_i, p_i) \leftarrow f_{\text{type}}(e_i | D, S)$
//Stage 2: Explicit Time Extraction (R2)
parallel for each $e_i \in E$ **do**
  $t_i \leftarrow f_{\text{time}}(e_i, c_i, D)$
Split E **into:** $E_{valid} \leftarrow \{e_i | \, t_i \neq \emptyset\}$, $E_{inferred} \leftarrow \{e_i | t_i = \emptyset\}$
//Stage 3: Initial Skeleton Assembly (R3-1)
Sort E_{valid} by (t_i, p_i)
Slot $E_{inferred}$ into bounded Phase blocks based solely on p_i
$G_0 \leftarrow$ Assembled initial skeleton graph
//Stage 4: Missing Time Inference (R4)
for each $e_i \in E_{inferred}$ **do**
  $\hat{t}_i \leftarrow f_{LLM}(e_i \mid G_0, D)$
//Stage 5: Phase-Aware Boundary Clamping
$E_{phase_sorted} \leftarrow Sort(E_{valid} \cup E_{inferred})$ by $\langle p_i, t_{start}, c_i, id_i \rangle$
for each $e_i \in E_{inferred}$ **do**
  lower $\leftarrow$ nearest preceding R2 end timestamp in E_{phase_sorted}
  upper $\leftarrow$ nearest following R2 start timestamp in E_{phase_sorted}
  if *lower* $= \emptyset$ and *upper* $= \emptyset$ then continue
  if *lower* $>$ *upper* then *lower* $\leftarrow$ *upper*//resolve anchor inversion
  $\hat{t}_i.start \leftarrow clamp(\hat{t}_i.start, lower, upper)$
  $\hat{t}_i.end \leftarrow clamp(\hat{t}_i.end, lower, upper)$
  if $t_i.start > t_i.end$ then $t_i.end \leftarrow t_i.start$//collapse invalid interval
//Stage 6: Final Sequencing & Overlap Resolution (R3-2)
Sort all events E by $\langle t_{start}, p_i, c_i, t_{end}, id_i \rangle$
T $\leftarrow$ ResolveOverlap(T)//*Merge co-temporal and subsumed events*
Return T

Specifically, the lower bound is anchored to the latest end timestamp of preceding events, while the upper bound is anchored to the earliest start timestamp of succeeding events. For edge cases where temporal anchors are absent, the unconstrained LLM prediction is preserved; if an anchor inversion occurs (*lower* $>$ *upper*), the boundary is resolved to the upper anchor to prevent forward sequence disruption.

For example, if the explicit anchor for the Preparation phase is 1 March and the anchor for the Post-crime legal phase is 31 March, implicit events in the intervening Execution phase are strictly corrected into the window between 1 March and 31 March by Stage 5 (Bound-

ary Clamping) of Algorithm 1. This correction is applied regardless of any temporally ungrounded dates the LLM may produce.

When an upstream classification error occurs at Stage 1 (R1), the downstream module interaction depends directly on timestamp availability. If the misclassified event contains an explicit timestamp (R2), the hierarchical multi-key sorting in Stage 6 prioritizes the start timestamp (Key 1) over the legal phase hierarchy (Key 2), preserving chronological placement despite the phase misclassification. Conversely, if an unanchored event lacking explicit timestamps (R4) is misclassified, it is inevitably slotted into an incorrect phase block during skeleton assembly (Stage 3), causing its boundary clamping window (Stage 5) to be derived from the anchors of that erroneous phase. This leads to cascading sequence errors, a structural characteristic of unidirectional pipelines that is quantitatively evaluated via the oracle experiment in Section 4.7.

4. Experiments and Results

4.1. Dataset and Experimental Setup

4.1.1. Dataset Construction and Statistics

To verify the practical utility of the Deep-ARC, we constructed a synthetic criminal investigation dataset used in actual investigative environments. The dataset consists of a total of 10 cases (5 breach of trust, 5 embezzlement), comprising 34 documents including complaints, victim statement protocols, suspect interrogation protocols, and investigation reports. The document set contains a total of 312 target events. The number of events per single document is non-linearly distributed from a minimum of 4 to a maximum of 20 (Table 1). As a proof-of-concept (PoC) study, we specifically focus on property crimes (breach of trust and corporate embezzlement), where documentary evidence and temporal causalities are densely intertwined.

Table 1. Dataset statistics summary of criminal investigation cases.

Statistic	Count/Value	Description
Total Cases	10	Unique criminal investigation cases
Total Documents	34	Investigation documents (e.g., Police Reports, Indictments)
Total Event Spans	312	Extracted target event phrases across all documents
Breach of Trust (배임)	5 Cases (111 Spans)	Cases primarily involving breach of fiduciary duty
Embezzlement (횡령)	5 Cases (201 Spans)	Cases primarily involving corporate embezzlement
Explicit Date Span	99 Spans	Events with directly extracted dates (text_explicit)
Grounded Date Span	56 Spans	Events anchored to dates via document context (document_grounded)
Implicit/Inferred Span	157 Spans	Events requiring pure semantic deduction (inferred)
Avg. Spans per Case	31.2	Average number of target events comprising a single case
Avg. Tokens per Doc	~6500 Tokens	Estimated average document length (approx. 5188 words)

This dataset is a synthetic criminal investigation dataset designed to prevent ethical issues such as sensitive privacy infringement and investigative secret leaks that may occur when directly utilizing original criminal investigation documents. The synthetic documents were structured and authored to model the narrative patterns and procedural workflows of authentic criminal investigations. An active criminal investigator reviewed a representative sample of three cases, confirming that the document layouts and evidentiary progressions closely reflected real-world records.

To establish an objective and unbiased evaluation baseline, the 312 gold-standard timelines were curated entirely through domain-expert human annotation (Table 2). Specifically, two trained legal domain annotators (both doctoral researchers specializing in forensic science and AI-based criminal investigation) independently conducted cross-validation on a representative subset of four cases (two breach of trust and two embezzlement cases); initial labeling discrepancies were systematically resolved through structured consensus discussions, establishing consistent annotation rubrics that were subsequently applied across the entire dataset (Table 2). Prior to consensus, the initial labeling demonstrated substantial inter-annotator agreement of 88.29% (Cohen's $\kappa = 0.863$) on Crime Element classification and Kendall's $\tau = 0.911$ (agreement: 90.47%) on event sequencing. The established rubric was subsequently applied to the remaining six cases by the primary annotator, with a secondary review pass to verify consistent application of the calibrated labeling criteria. When conflicting accounts were present within a document, crime element labels and temporal anchors were determined by jointly considering the investigator's structured question and the respondent's answer. Since the investigator's questions consistently target specific crime elements and focal events regardless of the respondent's subjective perspective, they served as a stable anchor for both classification and chronological sequencing, even when suspect and victim testimonies diverged.

Table 2. Golden data sample of extracted events from a synthetic investigation document. Each event is anchored to a specific legal phase and crime element schema, demonstrating the mapping between unstructured text and structured temporal anchors.

Span ID	Event Text (Anonymized)	Ground Truth Crime Schema	Temporal Anchor
T001	The respondent, [Person A], was appointed as a director in 2015 and has served as Vice Chairman since September 2021, playing a key role in the company's management and investment decisions.	A1_trust_position_formation (Status and duty of fiduciary)	1 September 2021
T003	In breach of this fiduciary trust, the respondent, [Person A], borrowed 150 million KRW from a bank starting in September 2021 to expand their personal business, [Company B].	A2_crime_motive (Pre-crime motive)	1 September 2021
T004	in June 2022, without adequate deliberation and consent from the board of directors, forced [Company C] to provide a joint guarantee for the loan.	B1_breach_of_trust_execution (Act constituting breach of trust)	1 June 2022
T008	This fact came to light in January 2024 when the bank demanded full repayment of the entire borrowed amount from [Company C].	C1_damage_discovery (Damage discovery/Awareness of crime)	1 January 2024

4.1.2. Experimental Setup and Implementation Details

To reflect the operational constraints of real-world criminal investigations, where strict confidentiality and personally identifiable information (PII) mandate an on-premises, air-gapped deployment, all experiments were designed and conducted in a local server environment with blocked external network access. Among accessible open-weights models, we selected Gemma-4-31B [36] as the backbone LLM. This model provides an optimal balance between parameter scale and multi-step reasoning performance, ensuring strict constraint adherence while remaining computationally viable for on-premises, air-gapped serving.

For technical implementation, all inferences were conducted with low-variance parameters (temperature = 0.1, top_p = 0.95, and seed = 42) to ensure experimental reproducibility, using Python 3.10+ and Pydantic for structured JSON parsing. Detailed model identifiers, including comparison baselines, and complete hyperparameters are summarized in Table A6.

To thoroughly evaluate pipeline robustness across varying temporal complexities, the evaluation cases were further stratified into subgroups based on the degree of temporal explicitness. Specifically, cases were categorized into High- and Low-Explicitness subsets using the median ratio of explicit timestamps.

All evaluation tables report the average of three independent runs (mean ± standard deviation). Only the oracle experiment used a single deterministic run.

4.2. Evaluation Metrics

In the field of unstructured text-based event extraction and temporal sequencing, standard evaluation metrics established in the NLP information and sentence ordering domain are applied. These metrics measure the relative sequential relationships and global consistency of the reconstructed sequence. This study systematizes four primary sequence metrics to multidimensionally verify the consistency between the gold standard timeline $T^* = (e_1^*, e_2^*, \ldots, e_n^*)$ and the predicted timeline $\widehat{T} = (\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n)$. It also utilizes macroscopic phase-level metrics and temporal extraction precision metrics to verify procedural chronological progression and temporal strictness specific to the criminal justice domain.

4.2.1. Primary Sequence Consistency Metrics

- Kendall's Tau (τ): Measures the global rank correlation between the gold standard order and the predicted order. Where n is the total number of events and K is the number of discordant/inverted pairs, it is defined as follows. τ ranges from -1 (completely reversed) to $+1$ (perfect match). It penalizes sequential contradictions in the global narrative structure more heavily than simple positional shifts, making it suitable for evaluating the overall consistency of the event flow [37]:

$$\tau = 1 - \frac{2 \cdot K}{\frac{n(n-1)}{2}}$$

- Pairwise Accuracy: Measures the ratio of correctly ordered relative temporal relationships among all possible $\binom{n}{2}$ event pairs (e_i, e_j). Here $\mathbb{I}(\cdot)$ is an indicator function. It directly quantifies the local relative ordering accuracy between any two random events, regardless of absolute positional shifts [32,33]:

$$\mathrm{Acc}_{\mathrm{pair}} = \frac{1}{\binom{n}{2}} \sum_{1 \le i < j \le n} \mathbb{I}\Big(\mathrm{rank}(\hat{e}_i) < \mathrm{rank}\Big(\hat{e}_j\Big) \iff \mathrm{rank}(e_i^*) < \mathrm{rank}\Big(e_j^*\Big)\Big)$$

- LCS Ratio (Longest Common Subsequence Ratio): Calculated by dividing the length of the longest event subsequence that maintains the relative temporal order between the gold standard timeline and the predicted timeline by the number of gold standard

events n. It measures the preservation rate of the main narrative storyline where the context of the chronological sequence is correctly maintained, even if non-contiguous [32]:

$$\text{LCS Ratio} = \frac{\left|\text{LCS}(\widehat{T}, T^*)\right|}{n}$$

- Micro PMR (Perfect Match Ratio): Measures the ratio of documents where the entire predicted timeline matches the gold standard timeline 100%, without allowing a single positional error or sequential contradiction.
  Here, M represents the total number of evaluated documents ($M = 34$) and $\mathbb{I}(\cdot)$ is an indicator function returning 1 if the condition is satisfied and 0 otherwise. $\widehat{T}_m$ and T^*_m denote the predicted and gold standard timelines for the m-th document, respectively. All reported PMR scores represent the mean averaged across three independent experimental runs. This is the strict metric for evaluating whether the system achieves exact sequence ordering and complete timeline integrity [32,33]:

$$\text{Micro PMR} = \frac{1}{M}\sum_{m=1}^{M}\mathbb{I}\left(\widehat{T}_m = T^*_m\right)$$

4.2.2. Phase-Level and Temporal Metrics

- Phase-Level Metrics: To evaluate the block alignment consistency at the legal phase level rather than at the microscopic event level, Phase Pairwise Accuracy and Phase PMR were measured.
  Reflecting investigative practice where legal causation hinges on procedural transitions (Pre-crime → Crime Execution → Post-crime), these metrics adapt standard Pairwise Accuracy and PMR to evaluate chronological consistency across statutory legal phases.
- Temporal Metrics: To evaluate the precision of inferred temporal anchors, three evaluation tiers were established: (1) L1 (Strict) enforces exact day-level matching (YYYY-MM-DD) for both boundaries; (2) L2 (Relaxed) evaluates monthly-level granularity (YYYY-MM); and (3) L3 (Broad) permits any temporal overlap between the predicted and ground truth intervals.

4.3. Experimental Conditions and Comparative Baselines

We established three progressive experimental conditions to evaluate the proposed Deep-ARC. This configuration isolates the individual contributions of its core components, specifically the crime element schema and the deterministic rule engine (Figure 2, Table 3):

- Unconstrained pipeline (Vanilla): A zero-shot environment where the LLM autonomously extracts and orders events based solely on text and general-purpose prompts, without the intervention of a domain schema or rule engine.
- Schema-guided pipeline (+ Schema): A method that injects the crime element schema into the prompt to assist the LLM in comprehending semantic context, while leaving the final ordering to the autonomous judgment of the LLM.
- Deep-ARC pipeline (+ Schema + Rule Engine): The proposed full pipeline, which integrates top-down schema injection with missing time inference and phase-aware boundary clamping to algorithmically mitigate sequential contradictions.

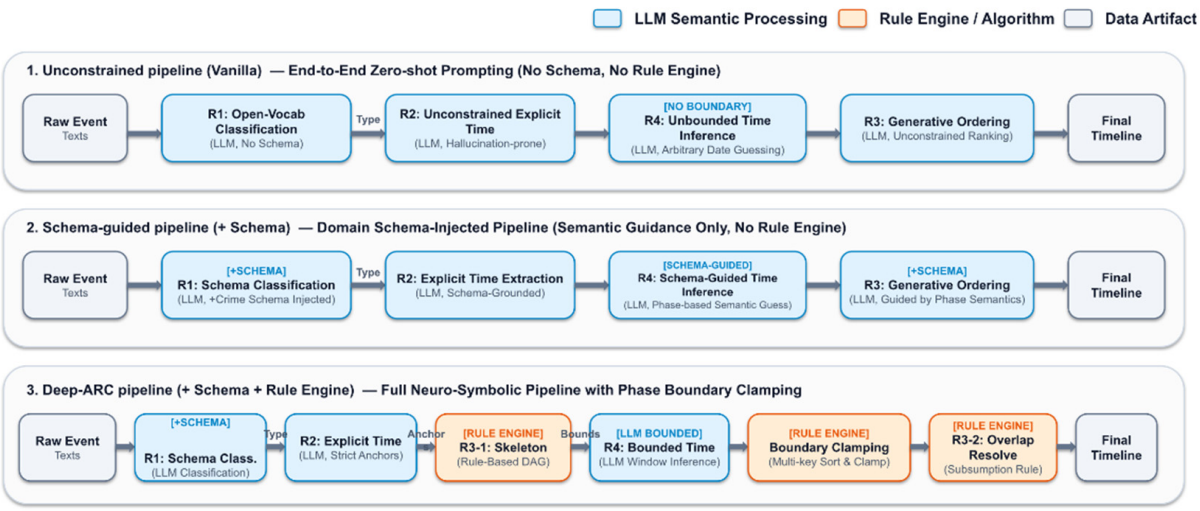


Figure 2. Comparison of Pipeline Architectures across Experimental Conditions.

Table 3. Comparison of pipeline experimental conditions (Unconstrained vs. Schema-guided vs. Deep-ARC).

Pipeline	LLM	Crime Schema	Rule Engine
Unconstrained (LLM)	✓	–	–
Schema-guided (LLM+ Schema)	✓	✓	–
Deep-ARC (LLM + Schema + Rule Engine)	✓	✓	✓

In particular, this experiment designed the terminal architecture of the pipeline (post-R4) asymmetrically to reflect the fundamental inference differences among the comparison groups. The Unconstrained and Schema-guided pipelines allow the LLM to macroscopically comprehend the overall document context. They autonomously derive the event sequence (R3) and missing time inference (R4) during the text generation process. That is, since the generation result of the LLM itself is considered the final timeline, no separate deterministic post-rearrangement intervenes.

Conversely, the proposed Deep-ARC model structurally controls the autonomy of the LLM through an algorithmic rule engine. Therefore, phase-aware boundary clamping based on explicit anchors intervenes for the time values inferred in the R4 stage. Consequently, a rule-based sequencing stage (R3-2) is essentially required to complete the pipeline. This stage ultimately refines and reassembles minor phase shifts or co-temporal events without sequential contradiction during the clamping process.

4.4. Experimental Results and Analysis

4.4.1. Main Results

As a result of measuring stepwise performance changes from the baseline (Unconstrained pipeline) to the proposed model (Deep-ARC) (Table 4), we confirmed that the model groups injected with the crime element schema (Schema-guided, Deep-ARC) demonstrated improvements in overall sequence ordering performance compared to the unconstrained pipeline. This indicates that the top-down injection of structured domain knowledge acts as a compass for the temporal reasoning of LLMs.

Table 4. Experimental results of timeline sequence ordering performance across pipelines using Gemma-4-31B (mean ± standard deviation).

Pipeline	Kendall's Tau	Pairwise Acc	LCS Ratio	Micro PMR
Unconstrained (Baseline)	0.721 ± 0.012	0.785 ± 0.007	0.822 ± 0.005	0.088 ± 0.000
Schema-guided	0.834 ± 0.004	0.865 ± 0.005	0.896 ± 0.002	0.235 ± 0.030
Schema-guided + Rule Sort (w/o Clamping)	0.843 ± 0.005	0.882 ± 0.002	0.907 ± 0.003	0.333 ± 0.017
Deep-ARC	**0.861** ± 0.004 *	**0.885** ± 0.001 **†	**0.916** ± 0.001 *	**0.343** ± 0.017 †

Note: Values are the mean of three independent experimental runs on 34 criminal documents. * $p < 0.05$, ** $p < 0.01$ compared to Unconstrained baseline; † $p < 0.05$ compared to Schema-guided pipeline. Bold values indicate the best performance in each column.

Specifically, Deep-ARC exhibited notable improvement in micro-level sequencing of individual events, which is the ultimate goal of the system. In Micro PMR, a strict metric that allows no errors within the entire timeline, Deep-ARC (0.343) achieved approximately a fourfold improvement compared to the Unconstrained baseline (0.088). Furthermore, in Kendall's Tau (τ) metric representing overall rank correlation, Deep-ARC (0.861) surpassed the Schema-guided pipeline (0.834), achieving the highest performance.

To evaluate the statistical significance of these observed improvements across the 34 evaluated documents ($df = 33$), paired two-tailed t-tests and Wilcoxon signed-rank tests were conducted. The results confirmed that Deep-ARC achieved statistically significant improvements over the unconstrained baseline in Pairwise Accuracy ($t(33) = 3.006$, $p = 0.005$, $d = 0.516$), LCS Ratio ($t(33) = 2.675$, $p = 0.012$, $d = 0.459$), and Kendall's Tau (Wilcoxon $p = 0.039$, applied due to the bounded nature of the rank correlation coefficient). Furthermore, comparing the full rule-constrained pipeline against the Schema-guided baseline, Deep-ARC demonstrated statistically significant gains in both Pairwise Accuracy ($t_{33} = 2.043$, $p = 0.049$) and Micro PMR ($t_{33} = 2.069$, $p = 0.047$, $d = 0.355$), confirming that the deterministic rule engine (combining hierarchical sorting and boundary clamping) provides a quantifiable enhancement to narrative consistency. To account for within-case document dependence, supplementary case-level analysis ($N = 10$) confirmed that the improvement in Pairwise Accuracy remained statistically significant for Schema-guided ($p = 0.007$), Schema-guided + Rule Sort (w/o Clamping) ($p = 0.045$), and Deep-ARC ($p = 0.049$) over the baseline (Table A7 in Appendix B).

Given that the permutation space of event sequences scales factorially ($N!$) with document length and that Micro PMR assigns a score of zero to any sequence with a single displacement, a ratio of 0.343 reflects a measurable improvement over the baseline (0.088). This indicates that the rule-constrained pipeline mitigates sequential contradictions more effectively than autonomous generation (Figure 3).

To decouple the independent contribution of Phase-Aware Boundary Clamping (Stage 5) from deterministic rule sorting (Stage 6), we conducted an ablation analysis by evaluating a pipeline variant without temporal interval clamping (Schema-guided + Rule Sort (w/o Clamping) in Table 4). Incorporating rule-based sorting and co-temporal merging alone increases Micro PMR from 0.235 to 0.333 and Pairwise Accuracy from 0.865 to 0.882, indicating that the hierarchical key sorting rules account for the primary gain in exact sequence alignment. Subsequently, enforcing Phase-Aware Boundary Clamping (Deep-ARC Full) yields incremental improvements over rule sorting alone, elevating Micro PMR from 0.333 to 0.343 ($t_{33} = 1.000$, $p = 0.325$; Wilcoxon $p = 0.317$), Pairwise Accuracy from 0.882 to 0.885 ($t_{33} = 0.508$, $p = 0.615$), Kendall's Tau (τ) from 0.843 to 0.861 ($t_{33} = 1.355$, $p = 0.185$; Wilcoxon $p = 0.208$), and LCS Ratio from 0.907 to 0.916 ($t_{33} = 1.096$, $p = 0.281$). Because boundary clamping selectively intervenes only on unanchored (R4) events whose

inferred intervals violate adjacent explicit (R2) anchors (modifying sequences in 8 of the 34 documents while leaving 26 documents unchanged), its corpus-wide statistical margin over rule sorting alone does not reach formal significance ($p > 0.05$). Nevertheless, these results indicate that while deterministic sorting accounts for the primary statistical gain in sequence alignment, boundary clamping serves as a targeted safeguard against residual cross-phase temporal inversions.

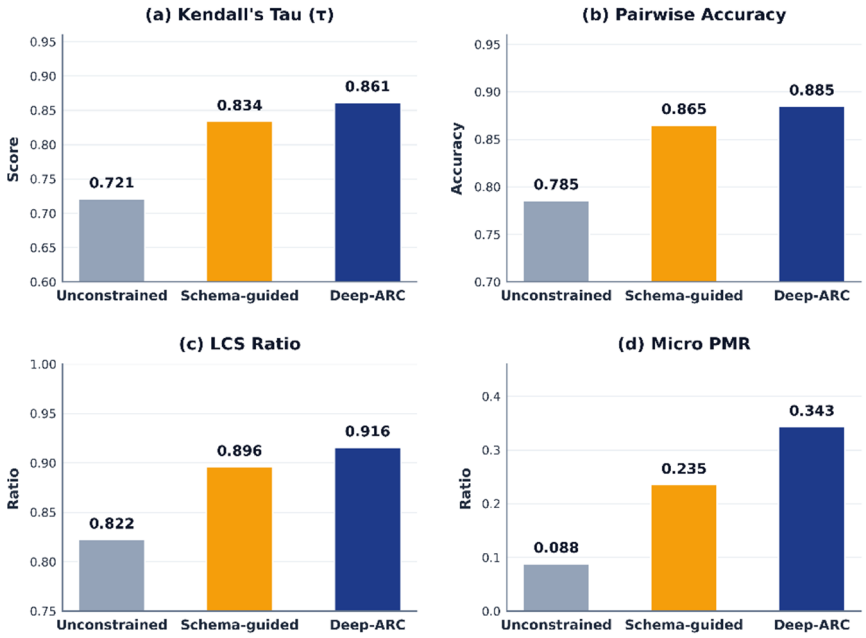


Figure 3. Comparison of sequence ordering performance across different pipeline architectures. The proposed Deep-ARC demonstrates significant improvements in all four primary metrics: (**a**) Kendall's Tau (τ), (**b**) Pairwise Accuracy, (**c**) LCS Ratio, and (**d**) Micro PMR.

Interestingly, in the Phase Pairwise and Phase PMR metrics evaluating the arrangement of macroscopic legal phases, the Schema-guided pipeline without the rule engine recorded slightly higher scores than Deep-ARC (Phase PMR: Schema-guided 0.794 vs. Deep-ARC 0.745). This phenomenon suggests an inherent trade-off within the Schema-guided pipeline. The Schema-guided pipeline, injected only with the schema, is proficient in flexibly inferring the macroscopic semantic flow (macro-narrative) connecting Preparation → Execution → post-crime. In particular, atypical retrograde cases (e.g., C → B → C pattern where post-discovery concealment or pretext fabrication intervenes following an initial audit detection) account for approximately 5.9% (2 out of 34 evaluated investigation documents). In these cases, open-ended generation without strict rule constraints coincidentally aligned with the gold standard sequence, partially driving up the Phase PMR metric. However, it repeatedly failed to precisely sequence how individual events are intertwined within those macroscopic phases, resulting in a Micro PMR stalled at 0.235.

Conversely, Deep-ARC enforces a phase-aware boundary clamping rule that combines explicit timestamps and algorithms. This deterministic control process foregoes flexibly grouping the minority of atypical sequences (5.9%). Instead, it forces a rearrangement of the dominant, normal timeline comprising 94.1% of the entire narrative without allowing a single sequential contradiction. In other words, prioritizing objective temporal evidence (facts) causes mechanical friction that partially segments macroscopic phase blocks, leading to a slight decrease in phase metrics. However, in investigative practice, the core factor in proving alibis and causal relationships is not aggregated phase classification but the precise temporal consistency of individual event units (Micro Metrics). For instance, establishing evidentiary validity in indictment specification requires verifying the chronological se-

quence between constituent crime elements, such as whether fund disbursement preceded or followed board authorization, where even a single event displacement can invalidate the prosecutorial narrative.

Therefore, the slight sacrifice in macroscopic flexibility is an inevitable structural cost entailed to preserve statutory procedural order. By effectively regulating microscopic precision of the majority of events (94.1%) and elevating Micro PMR, we empirically demonstrate that Deep-ARC provides a viable and tailored framework for the judicial and investigative domain (Table 5).

Table 5. Comparison of chronological sequencing performance at the macro-legal phase level.

Pipeline	Phase Pairwise	Phase PMR
Unconstrained (Base)	0.868	0.353
Schema-guided	**0.975**	**0.794**
Deep-ARC (Full)	0.960	0.745

Note: Bold values indicate the best performance in each column.

4.4.2. Temporal Extraction Performance

The ability to capture the occurrence time of events is a core factor determining timeline quality. Initially, in the explicit time extraction (R2) stage, the model groups (Schema-guided, Deep-ARC) applying schema-based constraints exhibited improved extraction capabilities compared to the unconstrained pipeline (from 0.902 to 0.949 based on L3). The identical R2 metrics for Schema-guided and Deep-ARC pipelines in Table 6 stem from both pipelines sharing the same schema injection module up to the event classification (R1) and explicit time extraction (R2) stages. The differences between Schema-guided and Deep-ARC emerge from the sequencing (R3) and missing time inference (R4) stages.

Table 6. Performance evaluation of explicit timestamp extraction (R2) and document-grounded missing time inference (R4).

Pipeline	R2 Explicit L1	R2 Explicit L2	R2 Explicit L3	R4 Doc-Grounded L1	R4 Doc-Grounded L2	R4 Doc-Grounded L3
Unconstrained	0.623	0.848	0.902	**0.292**	0.357	0.845
Schema-guided	**0.657**	**0.916**	**0.949**	0.256	**0.381**	0.875
Deep-ARC	**0.657**	**0.916**	**0.949**	0.214	0.363	**0.911**

Note: L1 = Strict exact match; L2 = Relaxed monthly-level match (YYYY-MM); L3 = Intersecting time boundaries allowed. Bold values indicate the best performance in each column.

The most interesting finding appears in the results of the R4 stage, which performs missing time inference on the pre-constructed skeleton. As shown in Table 6, the unconstrained pipeline recorded the highest value of 0.292 under the strict exact match criterion (L1), whereas the proposed model (Deep-ARC) decreased to 0.214. However, within the overlap-allowed range (L3) that does not induce sequential contradiction, Deep-ARC achieved the highest score of 0.911.

This decrease in L1 alongside the increase in L3 indicates that phase-aware boundary clamping effectively constrains speculative timestamp inferences within empirically grounded intervals. The unconstrained baseline strongly exhibits temporal hallucinations, arbitrarily determining and generating specific days when information is scarce. When this coincidentally hits the target, the L1 metric rises. However, this overconfidence of the LLM causes severe sequential contradictions in relationships with other events.

Conversely, Deep-ARC forces event sequencing based on objective explicit timestamps (R2) and phases through algorithms (Stages 5 and 6 of Algorithm 1) to prevent the LLM from fabricating random dates. It normalizes inferred times on a monthly basis, replacing them with safe time windows. Consequently, while the L1 accuracy for hitting pinpoint targets structurally decreases, the L3 overlap rate, measuring whether the inferred interval intersects with the ground-truth event occurrence range, increases to 91.1%. In criminal investigation analysis, providing bounded time intervals grounded on explicit timestamps is preferable to estimating unverified exact dates. Therefore, this result indicates that Deep-ARC moderates ungrounded temporal speculation in LLMs by anchoring inferred intervals to explicit textual evidence. From a practical perspective in chronological reconstruction, providing an evidence-bounded interval (L3: 0.911) is preferable to generating speculative pinpoint dates, as it reduces arbitrary date fabrication when structuring chronological records.

To assess whether the high L3 score (0.911) stems from broad interval predictions, we evaluated the temporal width and anchor precision of R4 intervals. The median predicted interval width is 341 days across both the baseline (340 days) and Deep-ARC (341 days), reflecting duration-type facts inherent to property crime investigations (such as corporate tenure and fiduciary status) rather than model-induced expansion. Furthermore, Deep-ARC achieves an exact match rate of 0.72 (0.82 within ±31 days) for predicted start dates against the gold standard, and maintains an L3 of 0.889 on intervals under one year, confirming that the improvement reflects accurate temporal positioning rather than artificial interval broadening.

4.5. Subgroup Analysis by Difficulty

To verify the robustness of the models according to the clarity of temporal clues within the documents, we conducted a subgroup analysis based on temporal explicitness (Explicit-High/Low) (Table 7). Specifically, Figure 4 presents Kendall's Tau, LCS Ratio, Pairwise Accuracy, and Phase Pairwise Accuracy to illustrate both micro-level event ordering and macro-level legal phase consistency across difficulty subgroups.

Table 7. Subgroup analysis of temporal sequencing performance based on task difficulty. Documents are divided into Explicit-High (rich temporal clues) and Explicit-Low (sparse temporal clues) subgroups. The unconstrained pipeline exhibits a performance drop in Explicit-High conditions due to information overload, whereas Deep-ARC maintains robust performance across both subgroups.

Pipeline	Subgroup	Kendall's Tau (τ)	Pairwise Acc	LCS Ratio	Micro PMR
Unconstrained	Explicit-High	0.675	0.779	0.813	0.056
	Explicit-Low	0.773	0.793	0.834	0.125
Schema-guided	Explicit-High	0.885	0.883	0.913	0.278
	Explicit-Low	0.776	0.844	0.877	0.188
Deep-ARC	Explicit-High	0.885	0.891	0.935	0.407
	Explicit-Low	0.835	0.878	0.894	0.271

Note: Explicit-High and Explicit-Low subgroups were split by the median density of explicit timestamps.

The analysis results indicate that the unconstrained open-ended language model demonstrated notable performance divergence. Contrary to general expectations, the Unconstrained pipeline exhibited an inversion phenomenon where its performance dropped more significantly in the Explicit-High environment (Micro PMR 0.056), which is rich in

temporal clues, than in the Explicit-Low environment (Micro PMR 0.125), which has sparse temporal clues. This suggests that the general-purpose LLM experiences the fundamental limitation of information overload. It becomes overloaded by the abundant fragmented date information, resulting in sequential contradictions.

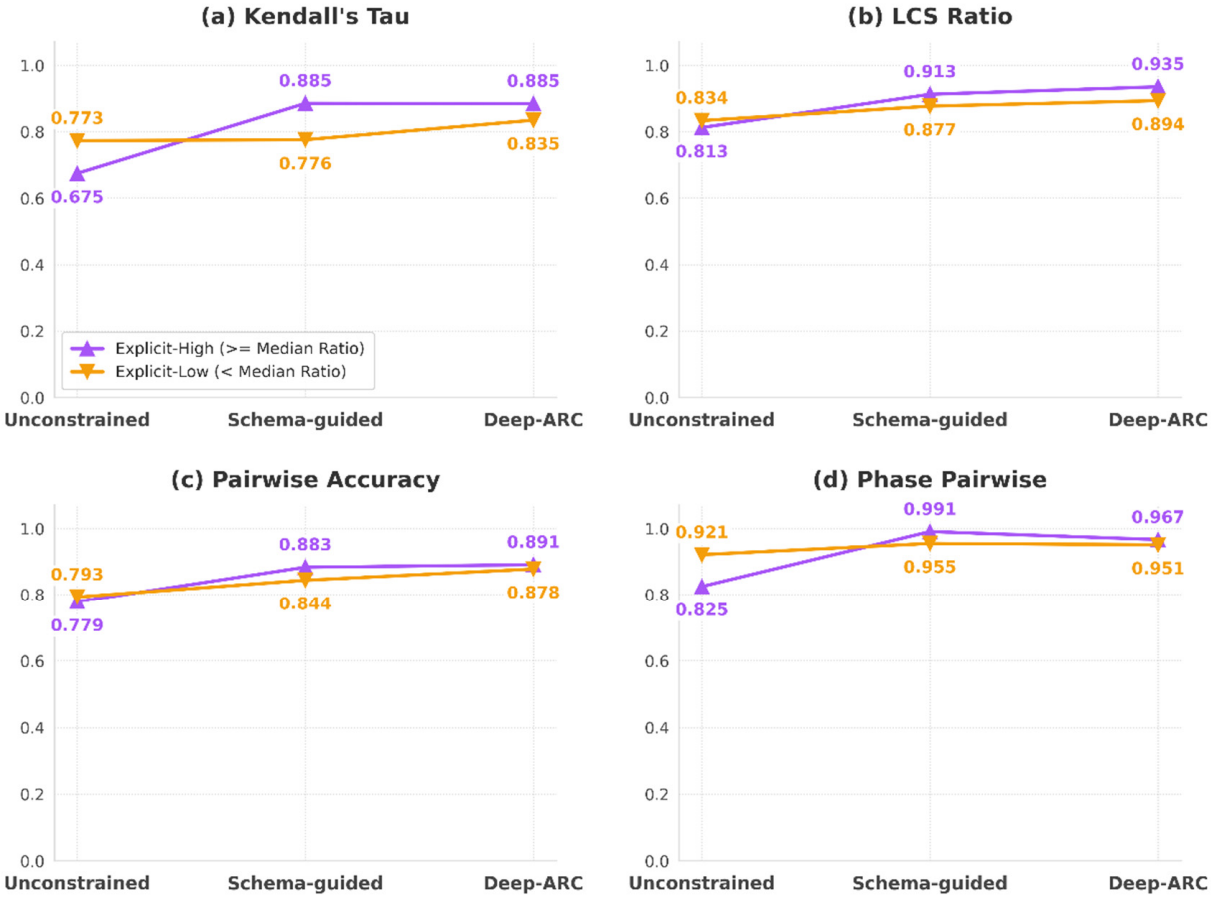


Figure 4. Performance comparison across difficulty levels based on explicit timestamp ratios.

Conversely, the Schema-guided and Deep-ARC pipelines, which include domain schema constraints, demonstrated a logical and predictable trajectory where performance improved as information increased, thereby validating their high system stability. In particular, Deep-ARC recorded a Micro PMR of 0.407 in the Explicit-High environment, where temporal clues were explicitly provided. This indicates that the rule engine acts as a strong anchor that remains stable even amidst an abundance of information.

In the Explicit-Low condition where temporal clues are sparse, Deep-ARC yielded a Micro PMR of 0.271 and a Kendall's Tau (τ) of 0.835. Although the absolute match ratio remains modest due to the inherent challenge of inferring unanchored events, it still remains higher than the unconstrained baseline (0.125). This suggests that the boundary clamping mechanism provides comparative stability even when explicit evidence is limited.

4.6. Cross-Model Benchmarking

To evaluate the architectural generalizability and model-agnostic efficacy of the proposed pipeline, we conducted extensive cross-model benchmarking. We evaluated three commercial large language models (GPT-5.6-Luna, Gemini-3.8-Flash, and Claude-Sonnet-4.6) operating in an external cloud API environment, alongside our on-premises local backbone (Gemma-4-31B). All commercial models were evaluated across all three pipeline configurations: Unconstrained (Vanilla), Schema-guided (+Schema), and Deep-ARC

(+Schema + Rule Engine) on the 34 criminal documents (Table 8). All commercial API calls were performed in September 2026 using fixed snapshot versions (gpt-5.6-luna, claude-sonnet-4-6, and gemini-3.8-flash) under identical prompt templates (Tables A3–A5 in Appendix B.1) and deterministic decoding parameters across three independent runs (temperature = 0.1, seed = 42; summarized in Table A6).

Table 8. Cross-model performance comparison across commercial LLMs and the local Deep-ARC pipeline on 34 criminal investigation documents.

Model	Pipeline	Kendall's Tau (τ)	Pairwise Acc	LCS Ratio	Micro PMR	Phase PMR
GPT-5.6-Luna	Unconstrained	0.750	0.788	0.826	0.059	0.549
	Schema-guided	0.828	0.842	0.889	0.137	0.784
	Deep-ARC	0.791	0.859	0.880	0.343	**0.775**
Gemini-3.8-Flash	Unconstrained	0.674	0.730	0.797	0.049	0.402
	Schema-guided	0.862	0.886	0.908	0.324	0.814
	Deep-ARC	0.826	0.876	0.894	**0.392**	0.745
Claude-Sonnet-4.6	Unconstrained	0.735	0.783	0.804	0.088	0.559
	Schema-guided	0.777	0.808	0.849	0.059	0.814
	Deep-ARC	0.775	0.856	0.867	0.294	0.745
Gemma-4-31B(Ours)	Deep-ARC	**0.861**	**0.885**	**0.916**	0.343	0.745

Note: Values represent the mean across three independent runs. Best-performing values per column are highlighted in bold.

As presented in Table 8 and illustrated in Figure 5, the benchmarking results reveal several key insights regarding parameter scaling versus structural constraints.

First, when operating under the unconstrained paradigm (Unconstrained), all commercial models struggled with micro-level timeline sequencing. While commercial models exhibited reasonable macroscopic narrative comprehension (Phase PMR ranging from 0.402 to 0.559), their exact event sequencing accuracy remained consistently low (Micro PMR $\leq$ 0.088, with Gemini-3.8-Flash recording just 0.049). This demonstrates that parameter scaling alone cannot resolve microscopic temporal hallucinations and causal inversions in investigation documents.

Second, integrating the Crime Element Schema and the Rule Engine consistently improved performance across all architectures. Injecting the Crime Element Schema stabilized macro-level Legal Phase ordering, raising Phase PMR across all commercial models under the Schema-guided condition to 0.78–0.81. Furthermore, Phase-aware Boundary Clamping in Deep-ARC substantially enhanced Micro PMR: increasing from 0.059 to 0.343 in GPT-5.6-Luna, 0.049 to 0.392 in Gemini-3.8-Flash, and 0.088 to 0.294 in Claude-Sonnet-4.6. This uniform gain indicates that Deep-ARC effectively resolves sequential contradictions and achieves competitive performance against commercial LLMs under air-gapped deployment constraints.

Third, the On-premise Local LLM (Gemma-4-31B) with Deep-ARC recorded a Kendall's Tau (τ) of 0.861, Pairwise Accuracy of 0.885, and Micro PMR of 0.343. These values are higher than those of the unconstrained commercial models, and comparable to GPT-5.6-Luna (0.343) and Claude-Sonnet-4.6 (0.294) under Deep-ARC, though lower in Micro PMR than Gemini-3.8-Flash (0.392). These results suggest that an on-premises model

combined with the Rule Engine can serve as a viable alternative for analyzing criminal investigation documents in confidentiality-critical environments.

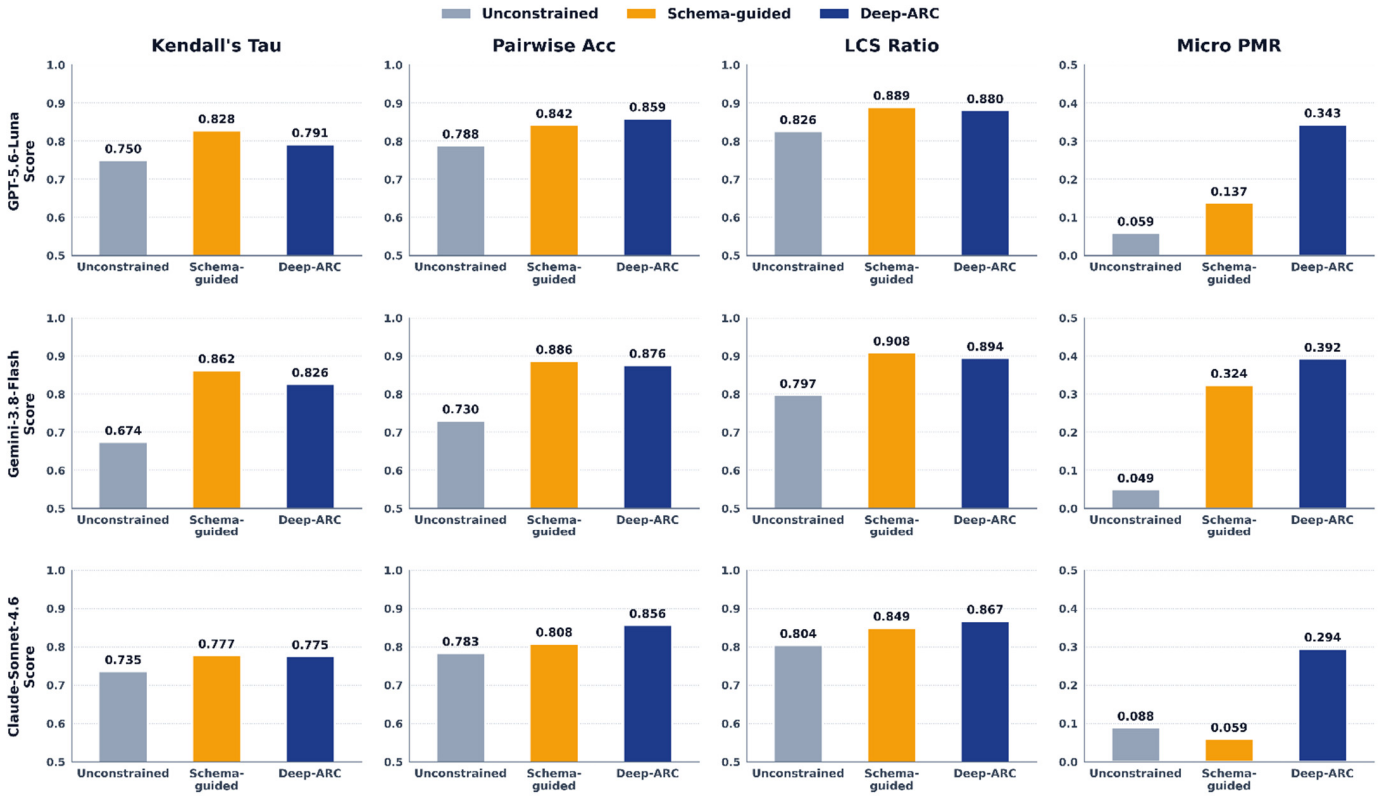


Figure 5. Comparison of the four primary sequence consistency metrics across commercial LLMs and the proposed local Deep-ARC pipeline under Unconstrained, Schema-guided, and Deep-ARC architectures: Kendall's Tau (τ), Pairwise Accuracy, LCS Ratio, and Micro PMR.

To investigate whether the difficulty-dependent fluctuations observed in Section 4.5 generalize across model scales, we evaluated all four architectures across the Explicit-High and Explicit-Low conditions (Figure 6). Full multi-metric subgroup results across all models are provided in Table A8 in Appendix B.

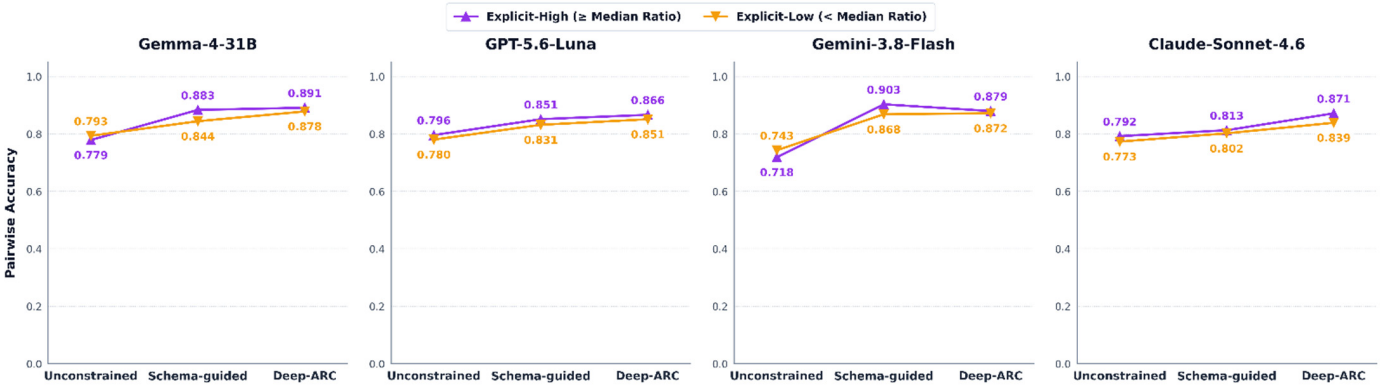


Figure 6. Subgroup difficulty analysis of Pairwise Accuracy across the on-premise local model (Gemma-4-31B) and commercial LLMs (GPT-5.6-Luna, Gemini-3.8-Flash, Claude-Sonnet-4.6) under Explicit-High and Explicit-Low conditions.

As illustrated in Figure 6, Pairwise Accuracy under the unconstrained baseline exhibits noticeable instability depending on temporal clue density, with Gemma and Gemini showing counterintuitive performance inversions where Explicit-High scores fall below Explicit-Low.

When integrated with Deep-ARC, however, this disparity is largely eliminated: both Explicit-High (purple) and Explicit-Low (orange) trajectories converge closely across all four architectures, stabilizing within a narrow, robust range of 0.839–0.891 (0.891 vs. 0.878 in Gemma, 0.866 vs. 0.851 in GPT, 0.879 vs. 0.872 in Gemini, and 0.871 vs. 0.839 in Claude).

These consistent trends across the four distinct model families confirm that the Rule Engine provides an effective inductive bias that substantially narrows the performance gap between on-premises local and commercial architectures for criminal investigation documents.

4.7. Cascading Effect of Event Classification (R1) Performance on Temporal Sequencing (Oracle Experiment)

To measure the cascading error that initial stage errors inflict upon subsequent temporal sequencing within the event extraction pipeline, we comparatively analyzed performance between an oracle experiment (R1-Gold) and an autonomous prediction environment (R1-Predicted) (Figure 7). In the R1-Gold setup, 100% ground-truth answers were injected into the R1 (crime element schema classification) stage.

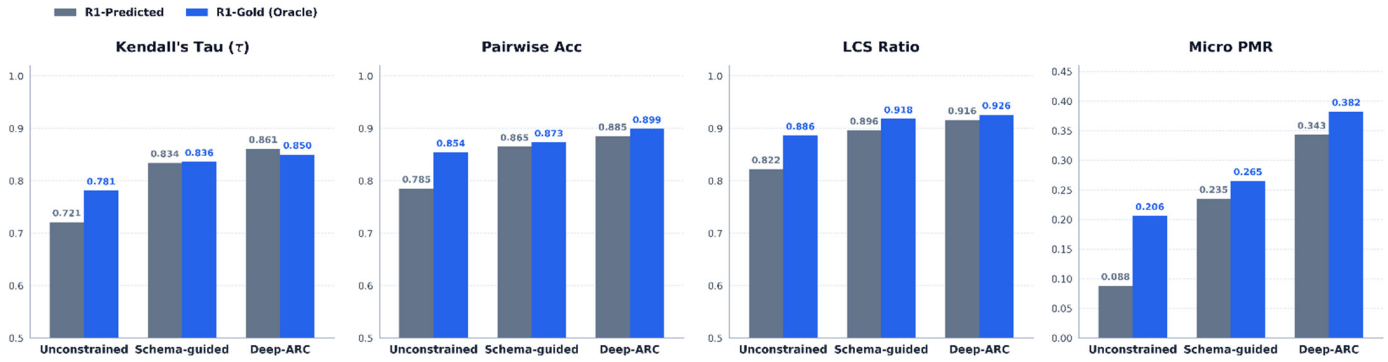


Figure 7. Impact of upstream event classification (R1) accuracy on downstream temporal ordering performance. Bars illustrate the performance across four metrics under realistic predicted inputs (R1-Predicted, gray) versus ground-truth inputs (R1-Gold Oracle, blue) for Unconstrained, Schema-guided, and Deep-ARC pipelines.

The analysis results showed that when the accuracy of the initial R1 classification was provided via gold labels, subsequent sequencing metrics uniformly increased across all model groups. The most distinct improvement occurred in the unconstrained baseline, where Micro PMR more than doubled in the R1-Gold environment, increasing from 0.088 to 0.206. This indicates that the open-ended LLM experiences severe hallucinations in the initial task of semantically mapping crime elements and legal phases, rather than in the temporal reasoning itself. Consequently, this initial error serves as the primary cause inducing the degradation of subsequent sequencing.

The proposed model, Deep-ARC, also expanded its performance upper bound when provided with the R1-Gold environment, recording a Pairwise Accuracy of 0.899 and an LCS Ratio of 0.926. Its Micro PMR, the strictest metric, increased to 0.382. This indicates that the internal missing time inference (R4) and deterministic phase-aware boundary clamping algorithms operate soundly as a temporal sorting mechanism with structural consistency.

However, in the case of Deep-ARC, a slight decrease in Kendall's Tau (τ) from 0.861 to 0.850 was observed upon R1-Gold injection. This suggests that even when perfect legal

phase answers are injected, errors or contradictions in the explicit timestamps (R2) embedded within the documents can cause mechanical friction during the forced clamping process of the rule engine, potentially resulting in minor shifts in the macroscopic topology (specifically observed in two documents; see Appendix C and Table A9 for a detailed qualitative case study).

Consequently, this oracle experiment empirically demonstrates that the temporal sequencing capability of the current Deep-ARC pipeline is highly dependent on the accuracy of the R1 classification stage, rather than algorithmic limitations. In other words, it indicates that the core bottleneck for maximizing final system performance lies in securing the stability of the initial event classification model.

5. Discussion

5.1. Practical Implications for Criminal Fact-Finding

The empirical findings of the proposed Deep-ARC pipeline present a viable pathway for algorithmically assisting cognitive fact-finding in criminal proceedings. During narrative reconstruction from fragmented evidence, investigators are often susceptible to confirmation bias, selectively adopting details that corroborate initial hypotheses or standard crime scripts [4,5].

Our results indicate that phase-aware boundary clamping, anchored on explicit timestamps, operates as a stabilizing mechanism against such cognitive distortions. Rather than delegating non-linear and fragmented temporal clues entirely to the probabilistic interpretation of LLMs, programmatic constraints ground event ordering on verifiable facts. This approach yields a consistent timeline that minimizes subjective distortion. Consequently, the proposed pipeline demonstrates potential not merely as a text summarization utility, but as an objective decision-support system assisting investigators in evaluating complex chronological evidence objectively.

5.2. Theoretical Implications for Legal AI

The subgroup evaluations (Table 7) and cross-model benchmarking (Table 8) highlight fundamental constraints inherent in unconstrained large language models within complex legal domains.

First, the unconstrained baseline demonstrated an inversion phenomenon in Explicit-High conditions, where dense temporal clues degraded sequencing accuracy. This behavior aligns with findings by Liu et al. [38], who noted that scattered temporal anchors across extended contexts can disperse model attention and induce hallucinations. This pattern suggests that information overload poses an inherent bottleneck for unconstrained generation.

Second, while injecting crime schemas enabled the Schema-guided pipeline to approximate broad legal phases (Phase PMR = 0.794), its micro-level event precision remained limited (Micro PMR = 0.235). As noted by Kambhampati [39], while LLMs excel at approximate semantic retrieval across broad knowledge spaces, they exhibit structural limitations in rigorous, principle-based symbolic reasoning and planning.

Therefore, scaling model parameters alone does not eliminate temporal inversions across legal phases. Rather, combining the contextual comprehension of LLMs with rule-based constraints provides a practical and effective approach to ensuring timeline consistency.

5.3. Bottleneck Analysis via Oracle Experiments

To structurally isolate error sources within the Deep-ARC pipeline, we conducted an oracle experiment (R1-Gold) by providing ground-truth schema labels to the classification stage (Figure 7).

The oracle results show that under optimal R1 conditions, downstream sequencing metrics improved substantially across all configurations, with Deep-ARC achieving a Micro PMR of 0.382 and Kendall's Tau (τ) of 0.850. These gains indicate that the downstream missing time inference (R4) and boundary clamping algorithms operate with sound structural consistency. Conversely, the substantial degradation observed under predicted inputs confirms that timeline discrepancies largely originate from upstream classification errors that cascade into downstream sorting, rather than from algorithmic limitations in the rule engine. Thus, securing classification stability in the initial extraction stage represents the primary operational bottleneck for overall pipeline optimization.

This minor variation in Kendall's Tau (τ) (0.861 to 0.850) reflects a recognized trade-off documented by Han et al. [21], where strict constraint enforcement over noisy intermediate extractions occasionally introduces localized sequence errors despite gains in global consistency (as empirically demonstrated in the Case 002 study in Appendix C).

Future work will incorporate confidence-weighted soft constraints to absorb upstream timestamp noise while maintaining statutory Legal Phase boundaries.

5.4. Limitations and Future Work

While this study validates the utility of Deep-ARC on criminal investigation documents, several technical and domain-specific limitations warrant consideration.

First, the pipeline exhibits a structural limitation as it operates on the premise of receiving a pre-chunked target events dataset rather than processing raw text in its entirety. In practical investigative environments, an extraction function where the model autonomously identifies legally significant event spans within vast and fragmented raw texts is necessary. The current research design focuses on the classification and sequencing of pre-segmented text while omitting this boundary detection process. Future research should integrate a complete end-to-end extraction module in the initial pipeline stage to actively detect and segment event boundaries directly from original documents. To this end, the Crime Element Schema currently used for event classification can additionally serve as a segmentation guide for splitting compound statements into atomic event spans, enabling schema-driven boundary detection that simultaneously segments and classifies events in a single pass.

Second, as confirmed in the preceding oracle experiment, the cascading error problem originating from the initial classification (R1) stage in the unidirectional sequential pipeline structure must be addressed. The current zero-shot prompting approach presents difficulties in securing classification stability in situations where complex precedents and unstructured narratives are mixed. It possesses a structural vulnerability where initial errors cumulatively transfer to downstream sequencing stages. Consequently, future work plans to adopt bidirectional feedback loops. This involves fine-tuning a secure on-premises local LLM with a corpus specific to criminal investigation documents to enhance schema mapping robustness. It also includes introducing joint extraction techniques to simultaneously optimize action classification and temporal anchoring. Additionally, the system will return constraint violation signals from the downstream rule engine to the upstream LLM module to induce re-inference.

Third, while our ablation analysis (Table 4) revealed that deterministic rule sorting drives the primary gain in exact sequence matching (0.235 to 0.333), the additional margin provided by Phase-Aware Boundary Clamping (0.333 to 0.343 in Micro PMR and 0.843 to 0.861 in Kendall's Tau (τ)), though not reaching corpus-wide statistical significance, serves as a targeted safeguard against cross-phase chronological inversions in the 8 documents where R4 inferences cross R2 anchors.

Fourth, there are quantitative limitations and generalizability issues associated with the synthetic criminal investigation dataset utilized in this experiment (totaling 10 cases and

312 events). Because this study aimed to verify the pipeline mechanism in a zero-shot inference environment rather than updating model weights through fine-tuning, a large-scale corpus was not strictly required. This study confirms the feasibility of a hybrid LLM-rule pipeline for mitigating temporal hallucinations. However, as a focused proof-of-concept (PoC) study, the current small-scale data and crime element schemas centered on property crimes (breach of trust and embezzlement) are limited in covering all crime types and exceptional situations occurring in investigative practice. The synthetic records may not fully capture the irregular noise of authentic case files despite investigator review. Future research plans to expand the schema scope to include fraud and violent crimes and construct large-scale investigative data, further enhancing the generalizability and robustness of the proposed system. In such domains, violent offenses such as homicide and assault would demand finer temporal precision due to their rapid progression, while multi-victim fraud cases would feature identical Legal Phases executed repeatedly across individual victims.

Fifth, as reported in Section 4.4.1, 5.9% of evaluated criminal investigation documents exhibited atypical retrograde patterns where the canonical Legal Phase progression did not hold. While the rule engine mitigates this through its multi-key sorting (overriding the Legal Phase hierarchy when explicit timestamps contradict it), deterministic constraints may still introduce mechanical friction or sequential contradictions in genuinely non-standard cases. Replacing hard Legal Phase constraints with soft probabilistic priors that adapt to case-specific narrative structures is identified as an important direction for future work.

Finally, while this study focused on narrative analysis and timeline reconstruction, future work will expand the contradiction detection logic of our previous study into a global timeline structure. This direction will integrate an automated contradiction detection module into the pipeline to identify subtle discrepancies and temporal conflicts among individuals (e.g., suspects, victims, and witnesses).

Specifically, this involves exception handling for edge cases where sequential contradictions exist among explicitly extracted timestamps (R2) due to a deponent's error or a typographical error within the criminal investigation documents themselves. For instance, an error occurs when a Pre-crime event is recorded with a later date than an Execution event. If such a contradiction is detected, the system will temporarily relax the hard clamping and assign a contradiction flag to the respective span. This approach will be developed to prevent mechanical failure while simultaneously inducing cross-verification by human investigators.

6. Conclusions

This study proposed Deep-ARC, a rule-constrained large language model pipeline designed to reconstruct objective event timelines from unstructured criminal investigation documents. To address temporal hallucinations and causal sequence inversions induced by unconstrained LLM generation, the system integrates top-down statutory crime element schemas with deterministic, phase-aware boundary clamping algorithms within a hybrid LLM-rule architecture.

Evaluated on an on-premise local model (Gemma-4-31B) reflecting air-gapped security environments, Deep-ARC achieved a notable performance improvement over the unconstrained baseline, yielding a Micro PMR of 0.343 and a Kendall's Tau (τ) of 0.861, while maintaining higher sequencing consistency than large-scale commercial models operating under zero-shot prompting across pre-segmented event spans within the synthetic investigation dataset.

Consequently, this research demonstrates that combining domain-specific structured knowledge with deterministic rule algorithms provides a practical and effective strategy for mitigating temporal hallucinations, rather than relying solely on expanding model

parameters. Furthermore, the proposed pipeline presents potential practical utility as a decision-support tool to assist investigative evidence analysis.

To establish clear research priorities, future work is structured into two phases. In the short term, we prioritize developing an active and automated event boundary extraction module to process unsegmented investigation documents directly, thereby mitigating the dependency on pre-chunked spans. In the medium to long term, we will expand the Crime Element Schema beyond property crimes to heterogeneous categories (e.g., fraud and violent offenses) and integrate multi-agent contradiction detection to build an end-to-end decision-support system for judicial practice.

Author Contributions: Conceptualization, G.-J.W., R.-S.P. and Y.-T.L.; methodology, G.-J.W. and R.-S.P.; software, G.-J.W.; validation, G.-J.W., R.-S.P. and J.T.; formal analysis, G.-J.W. and J.T.; investigation, G.-J.W.; resources, R.-S.P.; data curation, G.-J.W. and R.-S.P.; writing—original draft preparation, G.-J.W.; writing—review and editing, R.-S.P., J.T. and Y.-T.L.; visualization, G.-J.W.; supervision, R.-S.P. and Y.-T.L.; project administration, R.-S.P.; funding acquisition, R.-S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Commercialization Promotion Agency for R&D Outcomes (COMPA) funded by the Ministry of Science and ICT (MSIT) (No. RS-2025-02318092).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Although the dataset comprises synthetic cases, it is not publicly available due to ethical considerations regarding the replication of criminal techniques and the confidentiality of investigative procedures consulted with law enforcement authorities. The pipeline source code is available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the active criminal investigator for providing expert consultation and domain validation on the synthetic case documents. Additionally, generative AI and large language models were utilized solely as assistive tools for syntactic verification, linguistic cross-examination, and language proofreading during the manuscript preparation process, under strict human oversight and final validation by the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Legal Phase and Element Schema

Table A1. Legal Phase definition and Description.

Phase Code	Description
background (Z)	General background context, corporate governance, organizational history, or preliminary factual relations not constituting specific criminal elements.
pre_crime (A)	Formation of fiduciary status, custody authority, or preceding criminal motive.
during_crime (B)	Substantive acts of embezzlement, breach of duty, fund transfer, concealment, and damage occurrence.
post_crime (C)	Discovery of damage, evidence destruction, flight, or restitution attempts following crime completion.
legal_process (D)	Formal investigative and judicial procedures (investigation initiation, interrogation, indictment, trial).

Table A2. Crime Element Schema.

Code	Embezzlement Slot	Breach of Trust Slot
Background Phase (Z)		
Z1		
Pre-crime Phase (A)	Background Information	Background Information
A1	Custody Authority	Fiduciary Duty
A2	Pre-crime Motive	Pre-crime Motive
Crime Execution Phase (B)		
B1	Embezzlement Execution	Breach of Trust Execution
B2	Property Transfer	Benefit Acquisition
B3	Concealment	Concealment
B4	Consumption/Disposal	Benefit Realization
B5	Refusal to Return	(Not applicable)
B6	Damage Occurrence	Principal Damage Occurrence
B7	Concurrent Motive	Concurrent Motive
Post-crime Phase (C)		
C1	Damage Discovery	Damage Discovery
C2	Evidence Destruction/Flight	Evidence Destruction/Flight
C3	Damage Recovery Attempt	Damage Recovery Attempt

Appendix B. Task Prompts and Implementation Specifications

Appendix B.1. Task Prompts by Experimental Condition

Table A3. Unconstrained pipeline Prompts (Baseline: No Schema, Full LLM Generation).

[R1: Event Type Classification (No-Schema)]
Analyze the [Target Event Span] and extract the type of crime or event (Event Type) it describes.
No predefined choices (schemas) are provided. Freely name and return a concise noun phrase representing the core act (e.g., Fund Transfer, Document Forgery, Contract Execution).
Output only the single noun phrase without markdown, symbols, or supplementary explanations.

[Full Document Text]
{text}

[Target Event Span]
{target_event_text}

[R2: Explicit Time Extraction]
Extract the explicitly mentioned occurrence time from the [Target Event Span].
The event represented by this passage is [{event_type}].

[Extraction Principles]
1. If multiple times are mentioned in the sentence, select the time when the [{event_type}] event actually occurred.
2. If a period is mentioned (e.g., from ~ to ~), extract both the start and end dates (e.g., YYYY-MM-DD/YYYY-MM-DD or YYYY-MM-DD to YYYY-MM-DD).
3. Extract only explicit times; do not arbitrarily speculate or infer times not present in the passage. If no explicit time exists, return null.

Table A3. *Cont.*

Result JSON Format:
{{
"temporal_expression": "original text expression",
"temporal_normalized": "normalized timestamp (single date: YYYY-MM-DD, period: YYYY-MM-DD/YYYY-MM-DD)"
}}

[Target Event Span]
{target_event_text}

[R3: LLM Autonomous Sequencing]
You are a criminal investigative analyst. Below is the list of [Extracted Event Information] from this document.
Based on the event contents (Event Type) and times (Time), order the events chronologically according to actual occurrence and logical causation.
Even events without explicit timestamps must be assigned an appropriate position within the overall timeline.

[Sequence Array Output Rules]
1. The prediction result JSON must include the "predicted_sequence" field.
2. Format specification: 2D array (List of Lists) structure: [["ID1"], ["ID2", "ID3"], ...]
3. Ordering and overlap logic:
- Arrange events in sequence according to their logical and chronological order of occurrence.
- Group events that occurred concurrently or whose order cannot be definitively distinguished into a single sub-array (e.g., [["T001"], ["T002", "T003"]]).
4. [Critical Constraints]
- Use 100% of the extracted event IDs without omission; do not arbitrarily generate fabricated IDs.

[Extracted Event Information List]
{extracted_events_list}

[R4: Time Inference (Missing Time Deduction)]
You are an investigative analyst inferring missing timestamps based on a timeline.
The target event [{event_type}] currently lacks an explicit date but occupies a specific position within the completed [Overall Timeline Skeleton].
Infer and return the plausible "Time Window" during which this target event occurred, based on the occurrence times of preceding and succeeding events and the overall context of the criminal investigation document.

[Time Inference Rules]
1. Refer to the [Overall Timeline Skeleton] and strictly consider the occurrence times of preceding and succeeding events (upper and lower bounds of the Time Window).
2. Follow the ISO 8601 format (YYYY-MM-DD) for normalized time; if it is an inferred range, represent the interval using a slash (/) like "2023-03-01/2023-05-31".

Result JSON Format:
{{
"temporal_expression": "summary of inference rationale",
"temporal_normalized": "normalized time interval (e.g., 2023-03-01/2023-05-31)"
}}

Table A3. *Cont.*

[Overall Timeline Skeleton (including Phase Structure)] {timeline_graph} [Full Document Text] {text} [Target Event Span] {target_event_text}

Table A4. Schema-guided pipeline Prompts (Experimental Group 1) (Note: The Schema-guided pipeline uses the same R2 and R4 prompts as the unconstrained baseline. Below are the unique R1 and R3 prompts injecting Schema).

[R1: Event Type Classification (Schema Injected)] Referring to the [Full Document Text] and [Target Event Span], classify which statutory category from the provided [Crime Element Schema] this passage corresponds to. Select exactly one most appropriate code defined in the schema (e.g., A1_보관자지위및권한, B1_임무위배행위). The code format must strictly follow "CategoryCode_KoreanName" (e.g., B6_피해발생). Do not output multiple codes or separate them with commas (,). Output only plain text without markdown, symbols, or explanations. If unclassifiable, return null. [Full Document Text] {text} [Target Event Span] {target_event_text} [Crime Element Schema Definition] {schema_str}   [R3: LLM Autonomous Sequencing (Schema Injected)] You are a criminal investigative analyst. Below is the list of [Extracted Event Information] from this document. Based on the definitions in the provided [Crime Element Schema] and the logical progression of the crime, order the events chronologically according to actual occurrence and logical causation. [Sequence Array Output Rules] 1. The prediction result JSON must include the "predicted_sequence" field. 2. Format specification: 2D array (List of Lists) structure: [["ID1"], ["ID2", "ID3"], ...] 3. Ordering and overlap logic: - Arrange events in sequence according to their logical and chronological order of occurrence. - Group events that occurred concurrently or whose order cannot be definitively distinguished into a single sub-array. 4. [Critical Constraints] - Use 100% of the extracted event IDs without omission; do not arbitrarily generate fabricated IDs. [Extracted Event Information List] {extracted_events_list} [Crime Element Schema Definition] {schema_str}

Table A5. Deep-ARC Model Prompts (Experimental Group 2) (Note: Deep-ARC does NOT use an LLM prompt for R3. It uses a Python symbolic rule engine (Algorithm 1) for sequencing and clamping. It shares the R2 and R4 prompts with others).

[R1: Event Type Classification (Schema Injected)]
Referring to the [Full Document Text] and [Target Event Span], classify which statutory category from the provided [Crime Element Schema] this passage corresponds to. Select exactly one most appropriate code defined in the schema (e.g., A1_보관자지위및권한, B1_임무위배행위). The code format must strictly follow "CategoryCode_KoreanName" (e.g., B6_피해발생). Do not output multiple codes or separate them with commas (,). Output only plain text without markdown, symbols, or explanations. If unclassifiable, return null.

[Full Document Text]
{text}

[Target Event Span]
{target_event_text}

[Crime Element Schema Definition]
{schema_str}

[R4: Time Inference (Phase Block Hint Based)]
You are an investigative analyst inferring missing timestamps based on a timeline. The target event [{event_type}] currently lacks an explicit date but occupies a specific position within the completed [Overall Timeline Skeleton]. Infer and return the plausible "Time Window" during which this target event occurred, based on the occurrence times of preceding and succeeding events and the overall context of the criminal investigation document.

[Time Inference Rules]
1. Refer to the [Overall Timeline Skeleton] and strictly consider the occurrence times of preceding and succeeding events (upper and lower bounds of the Time Window).
2. Follow the ISO 8601 format (YYYY-MM-DD) for normalized time; if it is an inferred range, represent the interval using a slash (/) like "2023-03-01/2023-05-31".

Result JSON Format:
{{
"temporal_expression": "summary of inference rationale",
"temporal_normalized": "normalized time interval (e.g., 2023-03-01/2023-05-31)"
}}

[Overall Timeline Skeleton (including Phase Structure)]
{timeline_graph}

[Full Document Text]
{text}

[Target Event Span]
{target_event_text}

Appendix B.2. Implementation Specifications and Hyperparameters

Table A6. Technical implementation specifications and LLM hyperparameters.

Category	Specification/Value	Description
Model Identifiers	google/gemma-4-31B-it	On-premises local model
	gpt-5.6-luna	Commercial API (OpenAI)
	claude-sonnet-4-6	Commercial API (Anthropic)
	gemini-3.8-flash	Commercial API (Google)
Backbone Model Details(Gemma-4-31B)	Gemma-4-31B: 30.7B parameters, Apache 2.0 license, 256K context	Release: April 2026 Knowledge cutoff: January 2025
Inference Environment	On-premises institutional API (OpenAI-compatible endpoint)	Quantization and runtime framework not disclosed by provider
LLM Hyperparameters	temperature = 0.1, max_tokens = 8000, top_p = 0.95, seed = 42	Strictly deterministic settings for reproducibility
Python Version	Python 3.10+	Core runtime environment
Key Libraries	Pydantic == 2.12.5, OpenAI == 2.29.0	API client interface and strict JSON schema validation

Appendix B.3. Extended Subgroup Difficulty Analysis

Table A7. Case-level aggregated sequence ordering performance across pipelines using Gemma-4-31B (mean ± standard deviation).

Pipeline	Kendall's Tau	Pairwise Acc	LCS Ratio	Micro PMR
Unconstrained (Baseline)	0.696 ± 0.013	0.773 ± 0.006	0.817 ± 0.004	0.075 ± 0.000
Schema-guided	**0.809 ± 0.002**	0.852 ± 0.002 **	0.887 ± 0.002	0.192 ± 0.025
Schema-guided + Rule Sort (w/o Clamping)	0.768 ± 0.005	0.845 ± 0.002 *	0.884 ± 0.003	0.253 ± 0.019
Deep-ARC	0.789 ± 0.003	0.849 ± 0.001 *	**0.894** ± 0.001	**0.261** ± 0.017

Note: Values are the mean of three independent experimental runs on 10 cases. * $p < 0.05$, ** $p < 0.01$ compared to Unconstrained baseline.

Table A8. Complete subgroup difficulty analysis across four LLM architectures under Explicit-High and Explicit-Low conditions.

Model	Pipeline Architecture	Subgroup	Kendall's Tau (τ)	Pairwise Acc	LCS Ratio	Phase Pairwise
Gemma-4-31B (Local)	Unconstrained	Explicit-High	0.675	0.779	0.813	0.825
		Explicit-Low	0.773	0.793	0.834	0.921
	Schema-guided	Explicit-High	0.885	0.883	0.913	0.991
		Explicit-Low	0.776	0.844	0.877	0.955
	Deep-ARC (Full)	Explicit-High	0.885	0.891	0.935	0.967
		Explicit-Low	0.835	0.878	0.894	0.951

Table A8. *Cont.*

Model	Pipeline Architecture	Subgroup	Kendall's Tau (τ)	Pairwise Acc	LCS Ratio	Phase Pairwise
GPT-5.6-Luna (Commercial)	Unconstrained	Explicit-High	0.737	0.796	0.837	0.900
		Explicit-Low	0.764	0.780	0.813	0.939
	Schema-guided	Explicit-High	0.850	0.851	0.905	0.988
		Explicit-Low	0.803	0.831	0.871	0.958
	Deep-ARC (Full)	Explicit-High	0.807	0.866	0.888	0.962
		Explicit-Low	0.773	0.851	0.872	0.936
Gemini-3.8-Flash (Commercial)	Unconstrained	Explicit-High	0.678	0.718	0.803	0.892
		Explicit-Low	0.669	0.743	0.790	0.912
	Schema-guided	Explicit-High	0.903	0.903	0.931	0.989
		Explicit-Low	0.816	0.868	0.883	0.978
	Deep-ARC (Full)	Explicit-High	0.855	0.879	0.902	0.957
		Explicit-Low	0.794	0.872	0.884	0.945
Claude-Sonnet-4.6 (Commercial)	Unconstrained	Explicit-High	0.717	0.792	0.819	0.909
		Explicit-Low	0.756	0.773	0.786	0.935
	Schema-guided	Explicit-High	0.805	0.813	0.863	0.983
		Explicit-Low	0.746	0.802	0.833	0.968
	Deep-ARC (Full)	Explicit-High	0.813	0.871	0.887	0.940
		Explicit-Low	0.732	0.839	0.844	0.934

Note: Evaluation cases were stratified into Explicit-High and Explicit-Low subgroups based on the median density of explicit timestamps across the 34 evaluated criminal documents.

Appendix C. Case Study: Resolution Friction Under R1-Gold Oracle

In our ablation study, providing gold event types (R1-Gold Oracle) slightly decreased the overall Kendall's Tau (0.861 → 0.850). To investigate the underlying mechanism, we conducted an error analysis on a representative document (Breach of Trust Case 002, Interrogation Protocol 3) where this metric dropped from 0.712 (R1-Predicted) to 0.481 (R1-Gold).

Table A9. Conflict between broad explicit timestamps and Legal phase hierarchy in Case 002.

Event ID	Event Type (Gold Phase)	Extracted Date Span (R2)	Gold Rank	Oracle Predicted Rank
T009	Z1_Background Information (Phase Z)	2021-09-01/2024-01-31	5	1
T001	A1_Fiduciary Duty (Phase A)	2021-09/2024-01	1	2
T004	B1_Breach of Trust Execution (Phase B)	2021-09-01/2024-01-31	4	4

In this document, multiple events across different legal phases extracted identical, wide explicit date spans (2021-09-01/2024-01-31) from high-level corporate background statements. Under such temporal ties, the rule engine's multi-key sorting deterministically breaks ties using the Legal Phase hierarchy (Z → A → B → C → D). Consequently, event T009 (classified as Z1_Background Information) was rigidly pulled to the very start of

the sequence (Rank 1). However, in the ground-truth timeline, T009 represented a broad contextual background that co-existed with execution events and was ordered at Rank 5.

Notably, the Schema-guided pipeline produced an identical ordering for this document, placing T009 at Rank 1 as well. This confirms that the observed friction is not attributable to the rule engine alone, but rather originates from the upstream extraction of overly broad temporal spans (R2), which neutralize timestamp-based discrimination and force both LLM-based and rule-based ordering to fall back on phase hierarchy. This finding underscores that improving temporal extraction granularity in R2 is a key prerequisite for further sequencing gains.

References

1. Pennington, N.; Hastie, R. Explaining the Evidence: Tests of the Story Model for Juror Decision Making. *J. Pers. Soc. Psychol.* **1992**, *62*, 189–206. [CrossRef]
2. Bex, F.J. *Arguments, Stories and Criminal Evidence: A Formal Hybrid Theory*; Law and Philosophy Library; Springer: Dordrecht, The Netherlands, 2011; Volume 92, ISBN 978-94-007-0139-7.
3. Jang, M. Exploring the Quantity and Type of Evidence Collected during Criminal Investigations in South Korea. *Forensic Sci. Int. Synerg.* **2024**, *9*, 100544. [CrossRef] [PubMed]
4. Nickerson, R.S. Confirmation Bias: A Ubiquitous Phenomenon in Many Guises. *Rev. Gen. Psychol.* **1998**, *2*, 175–220. [CrossRef]
5. Findley, K.A.; Scott, M.S. The Multiple Dimensions of Tunnel Vision in Criminal Cases. *Wis. Law Rev.* **2006**, *2006*, 291–397.
6. Wagenaar, W.A.; van Koppen, P.J.; Crombag, H.F.M. *Anchored Narratives: The Psychology of Criminal Evidence*; Wheatsheaf, H., Ed.; St. Martin's Press: Hertfordshire, NY, USA, 1993; ISBN 978-0-7450-1458-6.
7. Won, G.-J.; Park, R.-S.; Park, S.-M. An AI-Based Hierarchical Timeline Generation and Verification System to Mitigate Confirmation Bias in Criminal Investigations. *J. Police Law* **2025**, *23*, 89–149. [CrossRef]
8. Won, G.-J.; Park, R.-S. Enhancing Zero-Shot Temporal Entailment in Legal Documents via Hybrid Element Fact Extraction. In Proceedings of the Workshop on Reasoning with Evidence in Law Enforcement and Forensics (RELAF 2026) Co-located with the 20th International Conference on Artificial Intelligence and Law (ICAIL 2026), Singapore, 8 June 2026; Volume 4253; Available online: https://CEUR-WS.org (accessed on 15 September 2026).
9. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* **2023**, *55*, 1–38. [CrossRef]
10. Dahl, M.; Magesh, V.; Suzgun, M.; Ho, D.E. Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models. *J. Leg. Anal.* **2024**, *16*, 64–93. [CrossRef]
11. Magesh, V.; Surani, F.; Dahl, M.; Suzgun, M.; Manning, C.D.; Ho, D.E. Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools. *arXiv* **2024**, arXiv:2405.20362.
12. Su, Z.; Li, J.; Zhang, J.; Zhu, T.; Qu, X.; Zhou, P.; Bowen, Y.; Cheng, Y.; Zhang, M. Living in the Moment: Can Large Language Models Grasp Co-Temporal Reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Bangkok, Thailand, 2024; pp. 13014–13033.
13. Chu, Z.; Chen, J.; Chen, Q.; Yu, W.; Wang, H.; Liu, M.; Qin, B. TimeBench: A Comprehensive Evaluation of Temporal Reasoning Abilities in Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Bangkok, Thailand, 2024; pp. 1204–1228.
14. Lagos, N.; Segond, F.; Castellani, S.; O'Neill, J. Event Extraction for Legal Case Building and Reasoning. In *Intelligent Information Processing V*; Shi, Z., Vadera, S., Aamodt, A., Leake, D., Eds.; IFIP Advances in Information and Communication Technology; Springer: Berlin/Heidelberg, Germany, 2010; Volume 340, pp. 92–101. ISBN 978-3-642-16326-5.
15. Li, S.; Zhao, R.; Li, M.; Ji, H.; Callison-Burch, C.; Han, J. Open-Domain Hierarchical Event Schema Induction by Incremental Prompting and Verification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Toronto, ON, Canada, 2023; pp. 5677–5697.
16. Maoro, F.; Geierhos, M. Contestable AI for Criminal Intelligence Analysis: Improving Decision-Making through Semantic Modeling and Human Oversight. *Front. Artif. Intell.* **2025**, *8*, 1602998. [CrossRef] [PubMed]
17. He, L.; Zhao, X.; Zhao, L.; Zhang, Q. An Event Extraction Approach Based on a Multi-Round Q&A Framework. *Appl. Sci.* **2023**, *13*, 6308. [CrossRef]
18. Yue, L.; Liu, Q.; Zhao, L.; Wang, L.; Gao, W.; An, Y. Event Grounded Criminal Court View Generation with Cooperative (Large) Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: Washington, DC, USA, 2024; pp. 2221–2230.
19. Ren, J. Extraction of Judgment Elements from Legal Instruments Using an Attention Mechanism-Based RCNN Fusion Model. *Inf. Dyn. Appl.* **2024**, *3*, 223–233. [CrossRef]

20. Du, X.; Cardie, C. Event Extraction by Answering (Almost) Natural Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 671–683. [CrossRef]
21. Han, R.; Zhou, Y.; Peng, N. Domain Knowledge Empowered Structured Neural Net for End-to-End Event Temporal Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 5717–5729.
22. Mandravickaitė, J.; Krilavičius, T. Harnessing LLM Ensembles for KG-Grounded Narrative Extraction: Disinformation vs. Trustworthy News. *Appl. Sci.* **2026**, *16*, 1962. [CrossRef]
23. Pustejovsky, J.; Castaño, J.M.; Ingria, R.; Saurí, R.; Gaizauskas, R.; Setzer, A.; Katz, G.; Radev, D. *TimeML: Robust Specification of Event and Temporal Expressions in Text.*; American Association for Artificial Intelligence (AAAI): Menlo Park, CA, USA, 2003; pp. 28–34.
24. Strötgen, J.; Gertz, M. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010), Uppsala, Sweden, 15–16 July 2010; pp. 321–324.
25. Leeuwenberg, A.; Moens, M.-F. A Survey on Temporal Reasoning for Temporal Information Extraction from Text (Extended Abstract). In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence; International Joint Conferences on Artificial Intelligence Organization: Yokohama, Japan, 2020; pp. 5085–5089.
26. Allen, J.F. Maintaining Knowledge about Temporal Intervals. *Commun. ACM* **1983**, *26*, 832–843. [CrossRef]
27. Zeng, W.; Tang, W.; Yuan, D.; Zhang, B.; Zeng, Y. Spatiotemporal Lattice-Constrained Event Linking and Automatic Labeling for Cross-Document Accident Reports. *Appl. Sci.* **2026**, *16*, 595. [CrossRef]
28. Ning, Q.; Feng, Z.; Wu, H.; Roth, D. Joint Reasoning for Temporal and Causal Relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 2278–2288.
29. Liao, R.; Jia, X.; Li, Y.; Ma, Y.; Tresp, V. GenTKG: Generative Forecasting on Temporal Knowledge Graph with Large Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*; Association for Computational Linguistics: Mexico City, Mexico, 2024; pp. 4303–4317.
30. Ong, K.T.; Kim, N.; Gwak, M.; Chae, H.; Kwon, T.; Jo, Y.; Hwang, S.; Lee, D.; Yeo, J. Towards Lifelong Dialogue Agents via Timeline-Based Memory Management. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*; Association for Computational Linguistics: Albuquerque, NM, USA, 2025; pp. 8631–8661.
31. Chen, Z.; Liao, J.; Zhao, X. Multi-Granularity Temporal Question Answering over Knowledge Graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Toronto, ON, Canada, 2023; pp. 11378–11392.
32. Gong, J.; Chen, X.; Qiu, X.; Huang, X. End-to-End Neural Sentence Ordering Using Pointer Network. *arXiv* **2016**, arXiv:1611.04953.
33. Cui, B.; Li, Y.; Chen, M.; Zhang, Z. Deep Attentive Sentence Ordering Network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 4340–4349.
34. Barale, C.; Barrett, L.; Bajaj, V.S.; Rovatsos, M. LexTime: A Benchmark for Temporal Ordering of Legal Events. *arXiv* **2025**, arXiv:2506.04041.
35. *ISO 8601-1:2019*; Date and Time—Representations for Information Interchange—Part 1: Basic Rules. International Organization for Standardization: Geneva, Switzerland, 2019.
36. Team, G.; Abd, S.E.; Aggarwal, V.; Algayres, R.; Andreev, A.; Bachem, O.; Ballantyne, I.; Brick, C.; Cărbune, V.; Casbon, M.; et al. Gemma 4 Technical Report. *arXiv* **2026**, arXiv:2607.02770.
37. Lapata, M. Automatic Evaluation of Information Ordering: Kendall's Tau. *Comput. Linguist.* **2006**, *32*, 471–484. [CrossRef]
38. Liu, N.F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; Liang, P. Lost in the Middle: How Language Models Use Long Contexts. *Trans. Assoc. Comput. Linguist.* **2024**, *12*, 157–173. [CrossRef]
39. Kambhampati, S. Can Large Language Models Reason and Plan? *Ann. N. Y. Acad. Sci.* **2024**, *1534*, 15–18. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.