CrimeLens – AI-Powered Crime Scene Reconstruction

Rohit Bogulla
Applied Machine Learning II (EEE 6778)
https://github.com/Kamehamehaaaaa/CrimeLens

October 19th, 2025

a. Problem Context and Project Summary

Crime scene reconstruction is a complex and often subjective task, requiring investigators to manually piece together evidence, witness statements, and forensic notes to infer what may have happened. This manual process can be time-consuming and prone to human bias. **CrimeLens** aims to assist in this process by automatically reconstructing plausible crime scene narratives and layouts from structured evidence data. The system encodes information such as suspects, objects, locations, and timestamps into a graph structure, models the temporal and causal relationships using a Transformer-based architecture, and generates possible scene reconstructions. The goal is to explore how multimodal reasoning—combining graphs, language, and generative AI—can support forensic training and improve interpretability in complex reasoning tasks.

b. Dataset

Sources:

Type	**Source**	**Description**
Text	Serial Podcast (Season 1)	True-crime narrative transcriptions created via OpenAI Whisper; used for timeline and causal event extraction.
Structured	FBI NIBRS, Chicago Crime Records	Incident-level attributes such as offense type, location, participants, weapons, and timestamps; provides relational and temporal realism.
Synthetic Data	Python Scripts + Crime Records	Using a python script generate crime scenes. The script will randomly chose attributes like weapon, crime type, location and action and create a crime scene. Will also use structured data from Crime Records and generate synthetic crime scenes.

The project will combine publicly available structured crime report datasets (e.g., San Francisco and Chicago Police Department datasets from Kaggle) with narrative text datasets such as ROCStories and ATOMIC for causal and event-based learning. Synthetic data will also be generated to create paired examples of structured evidence and narrative descriptions.

Type of Data	**Purpose**	**Why It Matters**
Structured Crime Data	Analyze patterns in location, time, and type of crimes; extract features such as crime density, frequency, hotspots, and correlations.	To understand real-world patterns and provide measurable features for prediction.
Narrative Text Data (ROCStories, ATOMIC)	Extract causal and event relationships between actions and outcomes; train embeddings or graph models on these relationships.	To give the model reasoning ability—understanding causes and effects, not just occurrences.
Combined Analysis	Fuse structured event data with causal event knowledge to build a model that can infer or predict the next likely event and explain its reasoning.	To produce explainable, event-aware predictions and simulate real-world scenarios.

Table 1: Core objectives and contributions of each data type used in the project.

Type: Tabular and textual data.

Format and Access:

- Crime report data in CSV/JSON format accessed from Kaggle's public repositories.
- Narrative data (ROCStories, ATOMIC) downloaded via Hugging Face or official dataset portals.
- Synthetic data generated locally using Python templates.

Preprocessing: Data will be cleaned and normalized into a canonical schema representing entities (people, objects, locations), relations, and timestamps. Named Entity Recognition (NER) and relation extraction will be used to build relational graphs. Text will be tokenized and embedded using BERT or GloVe.

Ethical Considerations: All data will be anonymized or synthetically generated. No real or identifiable case information will be used. The system is designed purely for research and educational purposes, not for operational forensic use.

c. Planned Architecture

Overview: The CrimeLens system follows a modular pipeline:
Data → Graph Encoder → Transformer → Hypothesis Generator → User Interface

Planned Components:

- **Graph Encoder:** Uses a Graph Neural Network (GraphSAGE / GAT) implemented with PyTorch Geometric to learn contextual embeddings of suspects, objects, and events.
- **Sequence Model:** A lightweight Transformer (T5-small or BART-base) that generates textual reconstructions from graph-encoded embeddings.
- **Hypothesis Generator:** Samples multiple narrative reconstructions using top-k/nucleus sampling, ranks them by confidence.

Frameworks and Tools: PyTorch, Hugging Face Transformers, PyTorch Geometric, NetworkX, spaCy, Streamlit (for UI).

Data Flow Diagram:

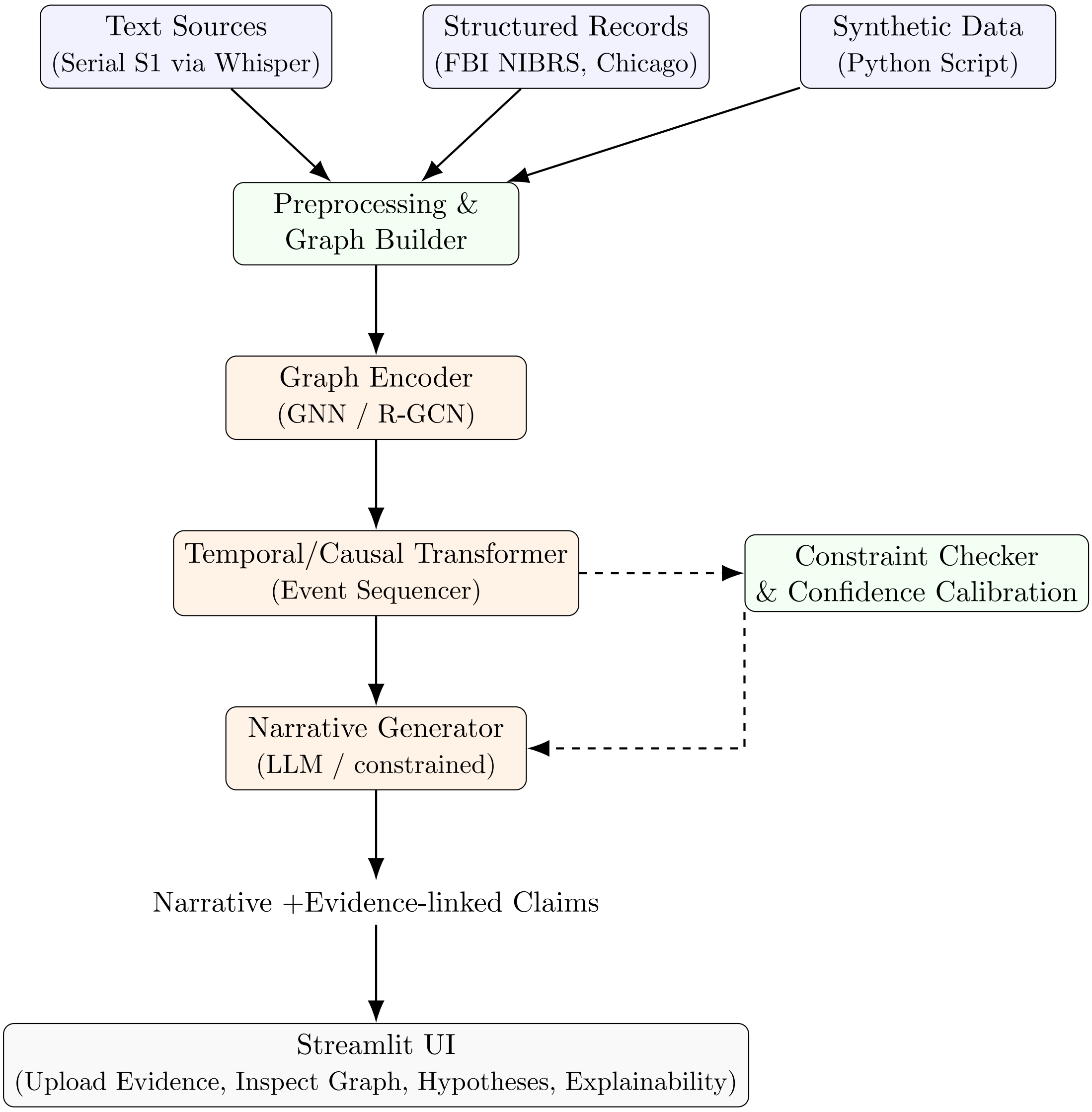


Figure 1: CrimeLens architecture: data ingestion → graph & temporal reasoning → constrained generation → UI with provenance.

d. User Interface Plan

Interface Framework: Streamlit web app.

Inputs:

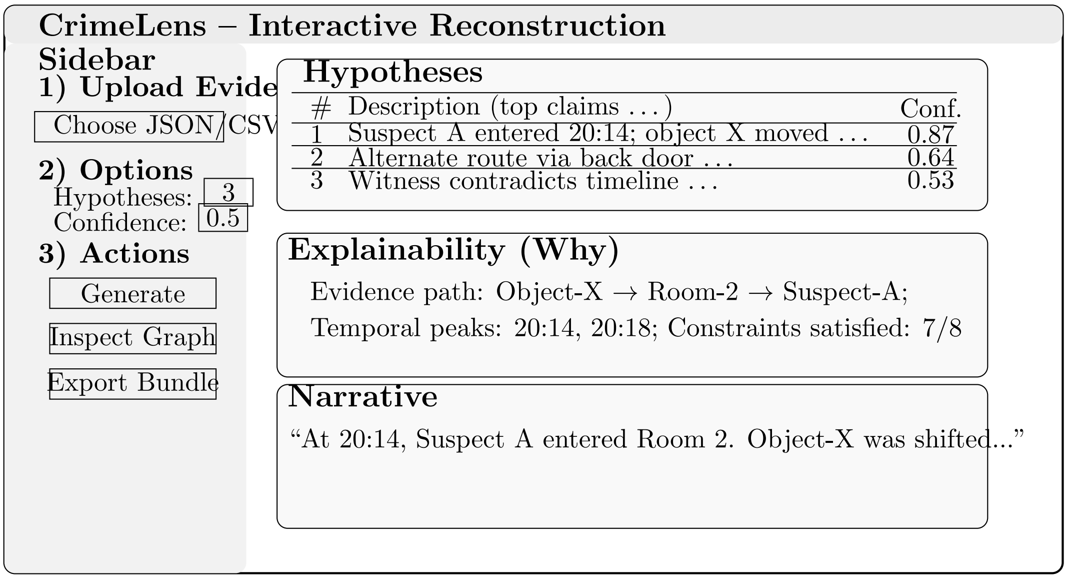
- Structured evidence (uploaded as CSV or JSON)
- Optional text notes (investigator observations)

Outputs:

- One or more reconstructed narratives with confidence scores
- Visual graph of entities and their relationships

Usability and Interpretability: The interface allows investigators or students to explore alternative hypotheses interactively, improving understanding of how evidence supports specific narratives.

Wireframe Sketch (conceptual):



#	Description (top claims . . .)	Conf.
1	Suspect A entered 20:14; object X moved . . .	0.87
2	Alternate route via back door . . .	0.64
3	Witness contradicts timeline . . .	0.53

Evidence path: Object-X → Room-2 → Suspect-A;

Temporal peaks: 20:14, 20:18; Constraints satisfied: 7/8

"At 20:14, Suspect A entered Room 2. Object-X was shifted..."

Figure 2: Streamlit interface wireframe within page margins: sidebar (inputs/actions) and main panels for hypotheses, explanation, and narrative.

e. Innovation and Anticipated Challenges

Innovation:

- Combines graph reasoning, sequence modeling, and generative AI in a single interpretable pipeline.
- Introduces the concept of uncertainty-aware reconstruction, generating multiple plausible hypotheses.
- Demonstrates interpretability through graph-to-text mappings.

Anticipated Challenges and Mitigation:

1. **Data scarcity:** Real crime data with events occuured for the crime used for graph construction is not available. *Mitigation:* Create a synthetic dataset using event templates and public text corpora. Using Whisper to create data from crime podcast like Serial.
2. **Model complexity:** Combining GNN and Transformer modules may increase training time. *Mitigation:* Start with lightweight pretrained models and scale up incrementally.

3. **Evaluation difficulty:** Assessing reconstruction accuracy is subjective. *Mitigation:* Use both automated metrics (BLEU, BERTScore, Graph Edit Distance) and qualitative human evaluation.

f. Implementation Timeline

Week	**Focus**	**Expected Outcome**
Oct 20 – 26	Data gathering and schema design	Unified dataset schema and preprocessing scripts for synthetic + real data
Oct 27 – Nov 2	Graph builder and baseline Transformer setup	Working baseline: evidence → narrative text generation
Nov 3 – Nov 9	Integrate Graph Encoder and train joint model	Combined GNN + Transformer pipeline
Nov 10 – Nov 16	Evaluation design and hypothesis sampling	Quantitative metrics + uncertainty handling implemented
Nov 17 – Nov 23	Build Streamlit UI and integrate inference module	Interactive prototype with text and graph outputs
Nov 24 – Nov 30	Testing and interpretability enhancements	Improved visualization and model explanations
Dec 1 – Dec 3	Final polish, demo, and report submission	Completed and stable system ready for presentation

g. Responsible AI Reflection

CrimeLens operates exclusively on synthetic or anonymized datasets, ensuring no exposure to sensitive or personal information. Interpretability and transparency are central to its design—each generated narrative is linked back to specific pieces of evidence in the input graph. Bias may arise from language model priors or dataset imbalance, which will be mitigated through prompt calibration and diversity analysis. Model outputs will be presented as probabilistic hypotheses, not factual claims, reinforcing the system's educational and exploratory purpose.