CLSKG: A Common Logic Substrate for Multi-LLM Integration with Uncertainty & Ontology-Based Safety Checks

Gustavo Carneiro, Cuong Nguyen
Centre for Vision, Speech and Signal Processing (CVSSP)
Surrey Institute for People-Centred AI (PAI)
University of Surrey

Surrey Institute for People-Centered AI

UNIVERSITY OF SURREY

Creating machines that can see and hear to understand the world around them.

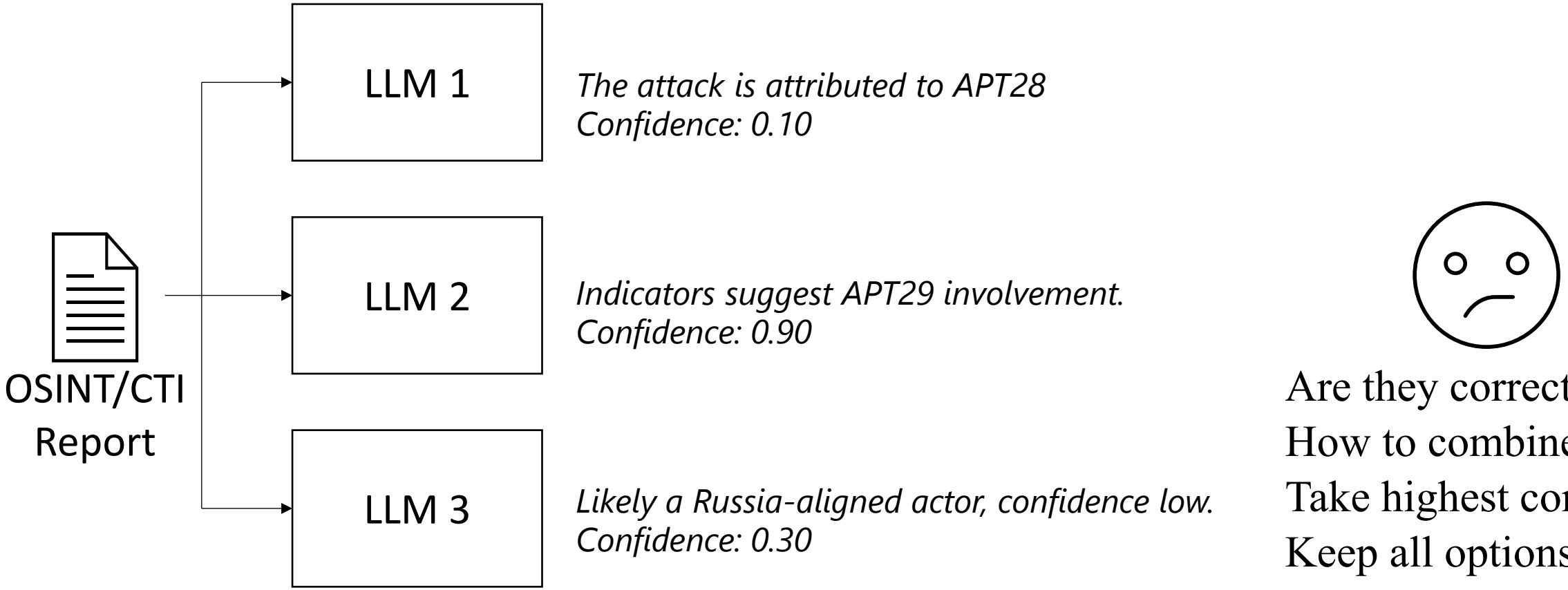
CVSSP Centre for Vision, Speech and Signal Processing

Current issues with AI models in cyber security

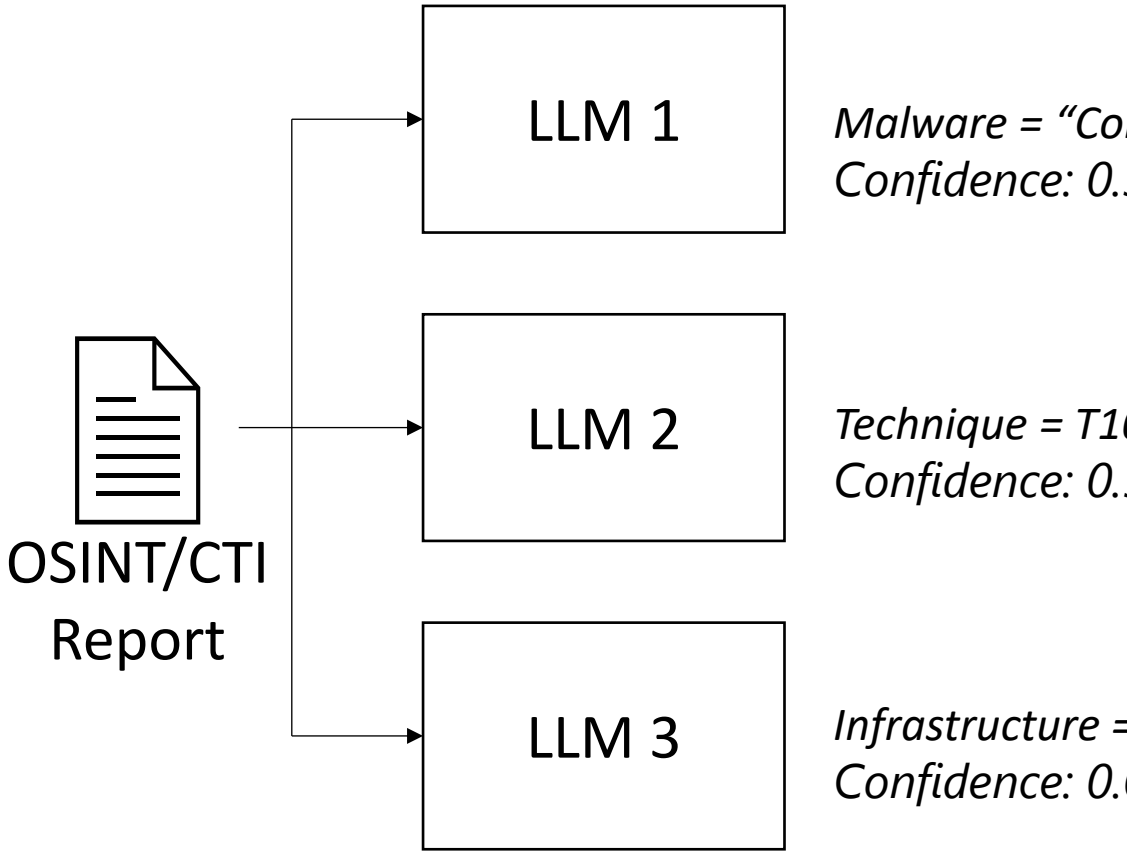
- Automated cyber analysis increasingly relies on **multiple AI agents**
  - Multiple roles: extraction, hypothesis generation, validation, and summarisation
  - Confidence of statements

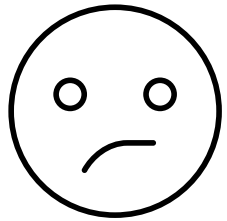
- Typical **pipeline is limited**:
  - Agents produce **fragmented outputs** (lack internal truth model)
  - Agents can produce **contradictory/complementary outputs**
  - Even when **confidence** is available, it's used in a **limited** way for multi-agent reasoning
  - **Hard** to produce an **auditable reasoning trail** and **hypotheses**

- **Critical problem**: Agents' outputs are hard to be *reliably combined, audited, or verified*



Are they correct?
How to combine them?
Take highest confidence?
Keep all options open?
Contradictions?





Are they correct?

How to combine them?

Coherent?

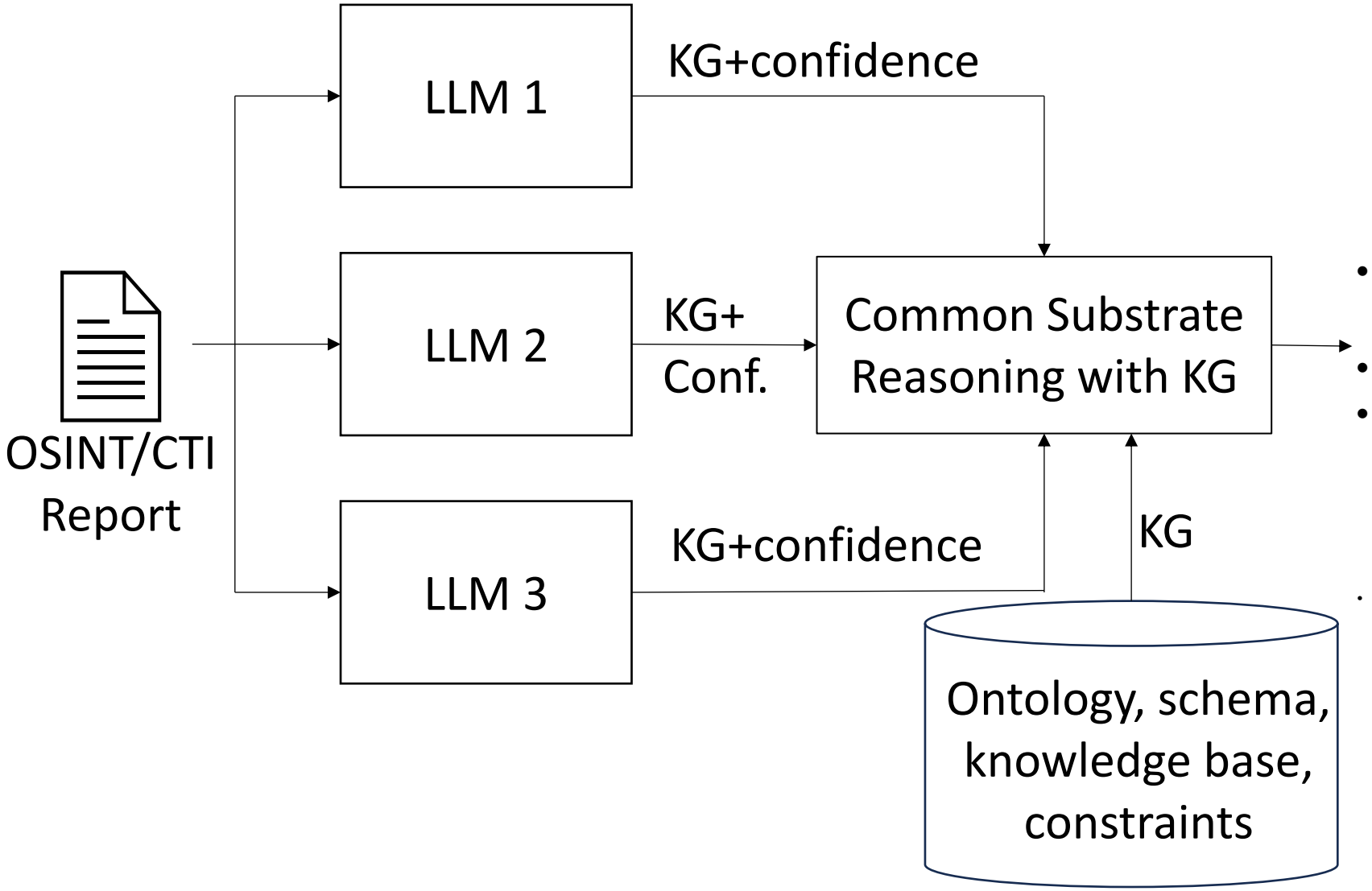
Contradictory?

Complementary?

Single campaign?

No common substrate for evidence, uncertainty, or safety to build an internal truth model

- Current automated **cyber analysis**:
  - Multiple **LLM orchestration** may **not** be **safe**: simple aggregation (e.g., thresholding, averaging, etc.)
  - Limited use of **knowledge graphs**: no confidence and provenance
  - **Missing a reasoning layer between LLMs and decisions**: schema (e.g., STIX), knowledge base (e.g., MITRE ATT&CK), ontology (e.g., TAC Ontology), constraints (e.g., SHACL), provenance (e.g., PROV-O), and confidence of statements
  - **Missing hypotheses formulation**: statements about campaigns that are not present in the input reports
- This **common substrate** would enable:
  - **Statement confidence:** evidence from LLMs, schema, knowledge base, ontology
  - **Detection of gaps, contradictions, and overlaps: formulate hypotheses**
  - **Safety verification** using **ontologies**

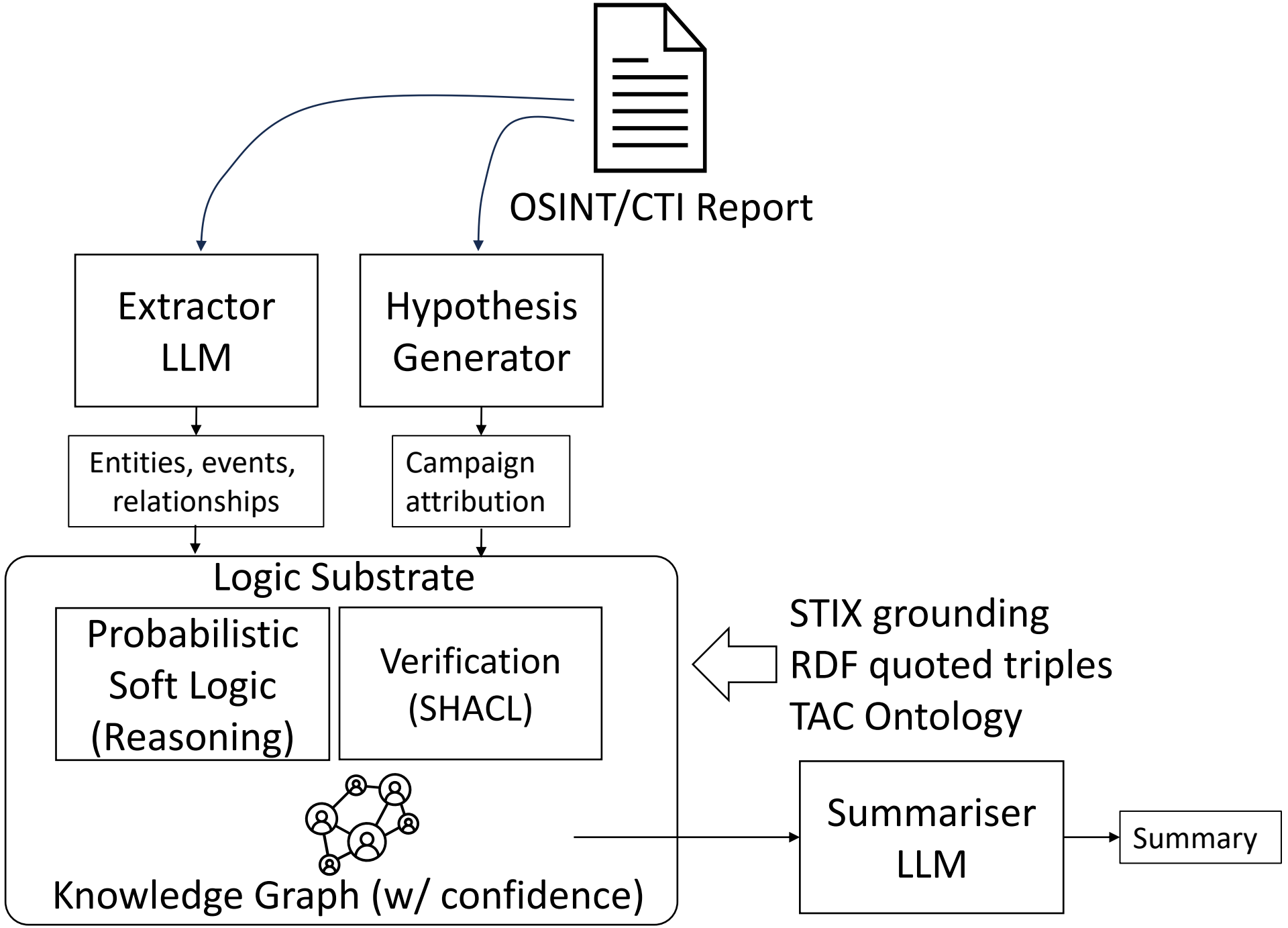


- High confidence the incident is Russia-aligned (ATP28 ATP29 are Russia-aligned).
- Low confidence on specific group.
- High likelihood that malware, technique, and infrastructure describe a single coherent campaign stage.

.

CLSKG: A Logic Substrate That Turns AI Outputs into Auditable Knowledge

- **LLMs have different roles:** Evidence extraction, safety checker, summariser
- **Hypotheses are automatically generated**
- All **statements** are represented as **RDF triples with uncertainty**
- Integrated via:
  - **Probabilistic soft logic reasoning**
  - **Ontology-grounded semantics**
  - **Verifiable SHACL safety rules**
- **CLSKG turns AI outputs into evidence, and evidence into verifiable decisions.**


OSINT/CTI Report
Extractor LLM
Hypothesis Generator
Entities, events, relationships
Campaign attribution
Logic Substrate
Probabilistic Soft Logic (Reasoning)
Verification (SHACL)
Knowledge Graph (w/ confidence)
STIX grounding
RDF quoted triples
TAC Ontology
Summariser LLM
Summary

Stage 1: Report to KG

- **OSINT/CTI Report → STIX → RDF triples + uncertainty**
  - **OSINT/CTI Report → STIX:** Extractor LLM Prompt + structured output
  - **RDF:** extract facts (triples) + uncertainty (A-box)
  - **Uncertainty:** Linguistic probability terms in text (likely, rarely, etc.)

Stage 2: Entity Alignment

- **MITRE ATT&CK STIX file → RDF triples**
  - **RDF triples:** extract facts (A-box)

- **FLORA to align CTI/OSINT report to MITRE ATT&CK knowledge base**
  - **Unsupervised KG Alignment by Fuzzy Logic**: entity and relation alignment between OSINT/CTI report triples and MITRE ATT&CK triples

  - **Uncertainty:** entity alignment confidence (from FLORA)

Stage 3: Verification

- **SHACL-based structural validation layer**
  - Defines the **required information each STIX entity** must contain
    - E.g., STIX: (campaign-xxxx-yyyy, attributed-to, <intrusion-set or threat-actor>)
  - **Verify report's RDF triples against STIX** rules
    - Correct entity types, allowed relations, correct domain and range, etc.
- Output:
  - **Incorrect information in report's RDF triples** (e.g., attributed-to is not an allowed relationship for malware)
  - **Missing information of STIX entities** (e.g., a campaign without attribution)

Stage 4: Hypothesis Formation

- **Inputs**
  - **CTI Report KG**: aligned & verified triples (A-box).
  - **MITRE ATT&CK KG**: RDF triples (A-box).
  - **TAC-Ontology OWL files**: RDF triples containing relationship between STIX classes (T-box)

- **Form hypothesis triples**
  - **Get potential missing links** from SHACL in Stage 3
  - **Form new triples for missing links** by using objects as entities in MITRE ATT&CK knowledge base

- **Outputs**
  - **Hypothesis RDF triples**

Stage 5 – Probabilistic Soft Logic

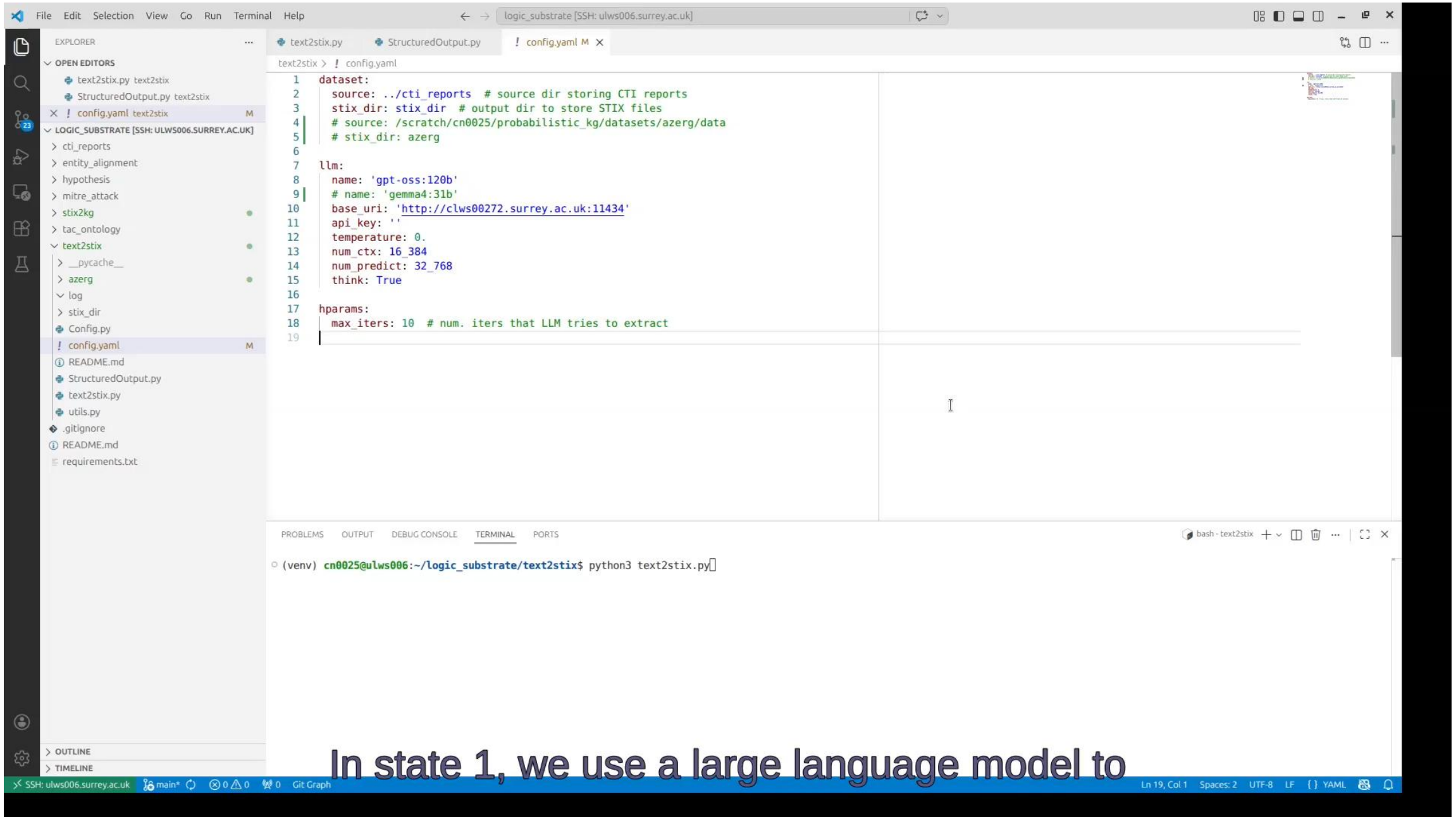
- **Reasoning**: **OSINT/CTI Report + MITRE ATT&CK + TAC Ontology**
  - Load report's aligned+verified triples (MITRE ATT&CK) + uncertainty (A-box)
  - Load MITRE ATT&CK's triples (A-box)
  - Load Hypothesis triples (A-box)
  - Load TAC-Ontology OWL file: relationship between STIX classes (T-box)
  - Rules: relationship from TAC Ontology, rules from MITRE ATT&CK
  - Inference becomes an optimisation with soft constraints, allowing the use of uncertainty scores

- **Outputs**
  - **Confidence for hypothesis triples**

Stage 6 – Summary report (being implemented)

- Given the outputs from Stage 5, generate a report summarising the main findings.

Demo


File Edit Selection View Go Run Terminal Help
logic_substrate [SSH: ulws006.surrey.ac.uk]
EXPLORER
OPEN EDITORS
text2stix.py text2stix
StructuredOutput.py text2stix
config.yaml text2stix M
LOGIC_SUBSTRATE [SSH: ULWS006.SURREY.AC.UK]
cti_reports
entity_alignment
hypothesis
mitre_attack
stix2kg
tac_ontology
text2stix
__pycache__
azerg
log
stix_dir
Config.py
config.yaml M
README.md
StructuredOutput.py
text2stix.py
utils.py
.gitignore
README.md
requirements.txt
text2stix.py
StructuredOutput.py
config.yaml M
text2stix > config.yaml
1 dataset:
2 source: ../cti_reports # source dir storing CTI reports
3 stix_dir: stix_dir # output dir to store STIX files
4 # source: /scratch/cn0025/probabilistic_kg/datasets/azerg/data
5 # stix_dir: azerg
6
7 llm:
8 name: 'gpt-oss:120b'
9 # name: 'gemma4:31b'
10 base_uri: 'http://clws00272.surrey.ac.uk:11434'
11 api_key: ''
12 temperature: 0.
13 num_ctx: 16_384
14 num_predict: 32_768
15 think: True
16
17 hparams:
18 max_iters: 10 # num. iters that LLM tries to extract
19
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - text2stix
(venv) cn0025@ulws006:~/logic_substrate/text2stix$ python3 text2stix.py
OUTLINE
TIMELINE
SSH: ulws006.surrey.ac.uk main* 0 0 0 Git Graph
Ln 19, Col 1 Spaces: 2 UTF-8 LF { } YAML
In state 1, we use a large language model to

Benchmark of STIX entity extraction

- **Dataset:** AZERG[1] containing 21 reports on malware campaigns
  - Break down into 176 text passages
  - Annotated by 3 experts in offensive security: STIX objects
  - Evaluated on 18 text passages (100-300 words each)
- **Baseline:** CTINexus[2]
- **Metric:** F1 score

Method	F1 score (mean ± std)
CTINexus[2]	48.65 ± 21.09
Ours	53.14 ± 24.92

[1] Lekssays, A., Sencar, H.T. and Yu, T., 2025, October. From Text to Actionable Intelligence: Automating STIX Entity and Relationship Extraction. In *28th nternational Symposium on Research in Attacks, Intrusions and Defenses* (pp. 458-473). IEEE.

[2] Cheng, Y., Bajaber, O., Tsegai, S.A., Song, D. and Gao, P., 2025, June. CTINexus: Automatic cyber threat intelligence knowledge graph construction using large language models. In *10th European Symposium on Security and Privacy* (pp. 923-938). IEEE.

Future Work: Provenance

- Statements in KG must be auditable and traceable: provenance

- PROV-O: ontology for representing provenance information:
  1. **prov:Entity:** a CTI report, an extracted RDF triple, a hypothesis
  2. **prov:Activity:** LLM extraction, PSL reasoning, SHACL validation,
  3. **prov:Agent:** An LLM, a reasoning engine

Future Work - Benchmarking

- There is no public, reproducible benchmark for triple-level CTI extraction with uncertainty and hypothesis generation/ranking.

- Aim: create one benchmark from CTI reports to assess
  - Factual triples with confidence and provenance
  - Generated hypotheses with confidence and provenance

Future Work – Multiple Ontologies

- In addition to the current ontology (TAC Ontology), schema (STIX), domain knowledge (MITRE ATT&CK)

- Integrate
  - D3FEND (domain knowledge + ontology)
  - VERIS (domain knowledge + schema)
  - CWE/CAPEC (domain knowledge + taxonomy),
  - CVE/CPE (domain knowledge + schema),
  - MISP Galaxy framework (domain knowledge + taxonomy)

Future Work – Hypotheses Formulation

- Implement Hypothesis LLM to generate plausible, non-factual triples
  - Graph-based Retrieval-augmented generation (graph-RAG)
  - Ontology-derived list of allowed predicates
  - Rank them by mission utility, source reliability, and uncertainty

Future Work: PSL + Bayesian Reasoning

- Reasoning with 2 layers
  - Layer-1 will use PSL (Bach, et al., (2017); Jøsang., 2016)
    - Constrain the reasoning space of the next layer
    - Provide confidence
  - Layer-2 will introduce a new Bayesian layer (Gelman, 2013) to compute
    - Credible posteriors
    - Credible intervals
- Challenge: rule formulation for reasoning
  - Current TAC Ontologies are limited on STIX 2.1 specs, for example:
    - Properties: Campaign must have
    - Relationship: (campaign, attributed-to, <intrusion-set, or threat-actor>)
    - Lack of conditional rules: (campaign-1, uses, malware-1) and (malware-1, uses, attack-pattern-1) → (campaign-1, uses, attack-pattern-1): look for other ontologies that contain conditional rules

Future Work – Continual Learning

- As new reports arrive, implement
  - Automated/human gatekeeping, principled conflict resolution
  - Newly Acquired Ontology (NAO) to add validated knowledge without violating prior rules.

Questions?

Gustavo Carneiro, Cuong Nguyen
Centre for Vision, Speech and Signal Processing (CVSSP)
Surrey Institute for People-Centred AI (PAI)
University of Surrey