Automated Causal Graph Extraction and Value-of-Information Prioritization for AI Biorisk Modelling*

Douw Marx[1]

[1]Independent douwmarx@gmail.com

April 27, 2026

Abstract

Quantifying risk at scale requires defining and prioritizing hundreds of causal paths to harm. I present an automated pipeline that (i) extracts causal chains from a single source document with an LLM, (ii) collapses near-duplicate nodes using embeddings and paired merge-proposer / merge-validator LLMs, and (iii) elicits Beta and PERT priors per node. Nodes in the risk model are then ranked by betweenness centrality, Birnbaum importance, and Expected Value of Partial Perfect Information (EVPPI), after Monte Carlo sampling. The pipeline is demonstrated on biorisk and serves as a proof-of-concept that LLMs can prioritize risk and indicate where new evaluations would most change downstream decisions.

Code & data: github.com/DouwMarx/automated_risk_modelling.

1 Introduction

Probabilistic risk modelling is time intensive and requires expert elicitation. I build a pipeline to automate the process and help prioritize attention. The pipeline (Fig. 1) consists of three stages:

1. **(A) Causal chain extraction.** An LLM reads a source document from the risk domain and emits causal chains mapping risk sources to harm outcomes.
2. **(B) Graph reduction.** Node labels are embedded (B1) and pairs of nodes are ranked by cosine similarity (B2). The top pair is sent through a paired *merge-proposer* / *merge-validator* LLM call (B3): the proposer returns a canonical merged label, and an independent validator scores the proposed merge in $[0, 10]$ for semantic equivalence; the merge is accepted iff the validator score clears a threshold τ *and* no cycle is introduced. The loop iterates until a convergence criterion is reached (B4).
3. **(C) Estimation.** Per-node Bayesian elicitation (C1) supplies Beta probabilities and PERT harms, fed into a single Monte Carlo joint-trajectory sampler (C2) that produces three rankings: betweenness centrality (parameter-free structural choke-points) [5], Birnbaum importance (high-impact intervention nodes) [2], and EVPPI (nodes where new evaluations would most change decisions) [10].

I demonstrate the pipeline on biorisk (*National Blueprint for Biodefense* [1]) as risk domain, since this work is done as part of a AIxBio hackathon. Results for the full AI risk landscape (*International AI Safety Report 2026* [6]) is available at github.com/DouwMarx/automated_risk_modelling.

*Submitted to the Apart AIxBio Hackathon (2026-04-24–26). Results are preliminary and should be interpreted as a proof-of-concept of methodology. I do not recommend making decisions based on the estimates and rankings produced here.

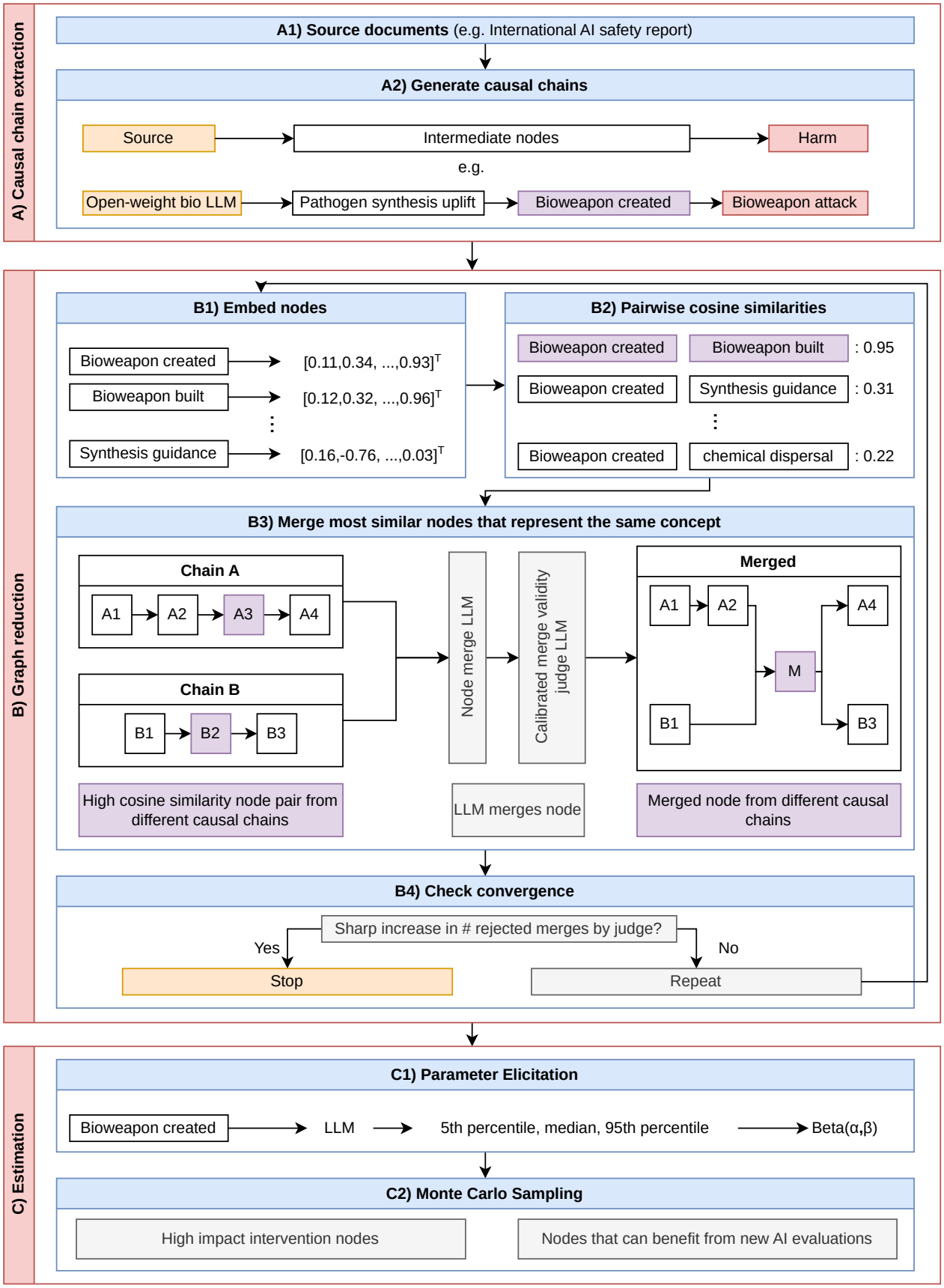


Figure 1: Automated risk modelling pipeline. **(A)** An LLM reads source documents and produces causal chains. **(B)** Nodes are embedded (B1), ranked by pairwise cosine similarity (B2), and the top pair is merged via a paired proposer / validator LLM call gated by validator score and cycle check (B3); the loop iterates until a convergence criterion is reached (B4). **(C)** Per-node Bayesian parameter elicitation (C1) feeds a single Monte Carlo joint-trajectory sampler (C2) that produces betweenness, Birnbaum, and EVPPI rankings.

2 Related Work

SaferAI's quantitative cybersecurity-risk pipeline [9] decomposes risk into causal pathways and elicits per-step priors, but the structure of the model comes from human expert workshops rather than text. I follow the example of AutoElicit [3], using LLM as substitute expert for prior elicitation and adapt it to fit Beta priors from (q_{05}, q_{50}, q_{95}) and PERT priors fit from $(\min, \text{mode}, \max)$. In the bio domain, Righetti's report [8] hand-builds *one* bioterrorism scenario with carefully elicited expert conditional probabilities; my pipeline auto-extracts many scenarios and uses EVPPI [10] to flag which deserve more careful elicitation or evaluation.

3 Methodology

The methodology is now presented following the numbering scheme of Figure 1.

A Domain seed and chain extraction

The input of this method is a set of papers or reports documenting risk sources. In this case, I use the *National Blueprint for Biodefense*, 2024 [1][1] for the primary domain, and did initial prototyping using the *International AI Safety Report 2026* [6]. An LLM or LLM agent, (Claude Code in this instance) reads the source documents and a global extraction guide, and produces causal chains linking risk sources to harm outcomes. Output is a `chains.json` list of N small directed acyclic graphs (DAGs), each with one source node, 1–5 intermediate nodes, and one outcome node.

B Graph reduction

The N disjoint chains are now fused into a single DAG by detecting duplicate or near-duplicate nodes across chains. Each step of the loop selects one candidate pair, runs two LLM calls in sequence, and either commits or blacklists.

B1: Node embedding. Every node label is embedded with BAAI `bge-large-en-v1.5` (1024-dim) served via OpenRouter. The embeddings serve as a cheap proxy for semantic similarity between nodes for large graphs.

B2: Candidate generation. Within each family of node (source / intermediate / outcome — never across families), pairwise cosine similarity is computed and the highest-cosine pair not yet in the run-level blacklist is selected.

B3: Paired merge-proposer / merge-validator LLM call. The candidate pair of nodes (with both labels, types, and full chain context) is sent through two sequential LLM calls (both Google Gemini 2.5 Flash Lite via OpenRouter):

- *Merge-proposer.* Returns a canonical merged label and a brief rationale.
- *Merge-validator.* Receives the same pair plus the proposer's canonical label and produces an integer score in $[0, 10]$ for the validity of the merge.

The merge is accepted iff (a) the validator score $\geq \tau$ *and* (b) the merge introduces no cycle in the working DAG. Otherwise the pair is added to a blacklist and never reconsidered. The threshold is fixed at $\tau = 8$, the lowest score the validator rubric labels as paraphrase / near-synonymy; a pre-merge consistency check scores the live judge against positive (paraphrase) and negative (false-merge) validation sets and reports TPR / FPR / Youden's J (Section B).

B4: Convergence. The loop terminates when the desired level of graph granularity is reached. See Figure 3 for candidate metrics for monitoring convergence.

[1] I am not an expert on biorisk and could not judge quality of the extracted causal chains.

C Estimation

The merged DAG is parameterised by eliciting an LLM (C1, elicitation) and analysed by a Monte Carlo sampler (C2, inference). Downstream quantities (expected harm, Birnbaum importance, EVPPI) are computed from the Monte Carlo samples and the DAG structure.

C1 Parameter elicitation

Per-node priors are elicited following AutoElicit [3]. An LLM in a substitute-expert role predicts the 5th, 50th, and 95th percentiles for each parameter in the risk model.

- **Per node:** elicit $P(X \mid \text{any parent active})$ as a Beta distribution that is fit to the (q_{05}, q_{50}, q_{95}) percentiles.
- **Per outcome:** elicit harm magnitude as a PERT distribution [4, 7] from $(\min, \text{mode}, \max)$ with shape $\lambda = 4$. Numeraire is USD; biorisk and AI domains share a USD numeraire to make cross-domain rankings commensurable.
- **Validator:** a second LLM pass scores the proposer's elicitation; both phases use the budget-friendly Google Gemini 2.5 Flash Lite model via OpenRouter.

C2 Inference

I use ancestral sampling on the DAG. For each Monte Carlo sample, Beta and PERT priors are drawn once, then Bernoulli activations propagate in topological order. The expected harm is the sample mean of $\sum_s X_s \cdot h_s$ over outcomes.

D Importance and value-of-information

A benefit of the quantitative risk model is that it supports a range of importance and value-of-information metrics to prioritise attention and evaluation effort. I evaluate three such metrics: betweenness centrality, Birnbaum importance, and EVPPI.

Betweenness centrality. Betweenness $C_B(v) = \sum_{s \neq v \neq t} \sigma_{s,t}(v)/\sigma_{s,t}$ [5] is the fraction of all-pairs shortest paths through v. Computed via NetworkX. This serves as a metric for the bottleneckedness of a node in the graph structure, without reference to the parameters. It does not require Monte Carlo sampling to compute.

Birnbaum importance. The Birnbaum importance of a node X is the expected-harm swing between the on and off state of X, marginalising over all other parameters [2]:

$$I_B(X) = \mathbb{E}[H \mid X = 1] - \mathbb{E}[H \mid X = 0]$$

This metric captures the expected harm reduction from perfectly intervening to clamp X off, so it is a measure of the maximum possible impact of interventions on X.

Expected Value of Partial Perfect Information (EVPPI). EVPPI of node X is the expected drop in harm if you knew θ_X *before* choosing an intervention $a \in A$ [10]:

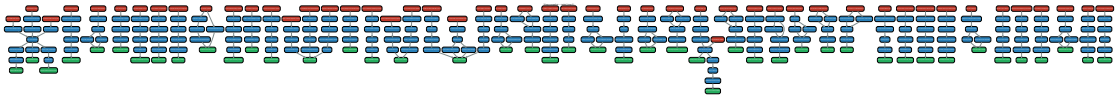
$$\text{EVPPI}(X) \;=\; \min_{a \in A} \mathbb{E}_\theta\big[H(a,\theta)\big] \;-\; \mathbb{E}_{\theta_X}\left[\min_{a \in A} \mathbb{E}_{\theta_{-X}}\big[H(a,\theta_X,\theta_{-X})\big]\right].$$

A high EVPPI means learning more about X would change which intervention is optimal. I estimate it via brute-force nested Monte Carlo over the elicited priors. This metric is indicative of where to prioritise expert evaluation and designing new AI evaluations.

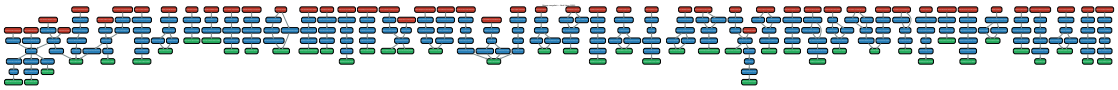
4 Results

4.1 Convergence of the merge pipeline

The merge loop collapses N disjoint chains into a single connected DAG. Fig. 2 shows the initial and converged graph. Fig. 3 shows different convergence signals (betweenness, node count, cumulative rejections by class, top-pair cosine, validator score) that can be used to determine when an appropriate level of granularity has been reached. As the loop iterates, the maximum cosine similarity amongst unmerged pairs reduces and the betweenness centrality grows, as the number of bottlenecks in the graph increase. At this point it is unclear which metric to use to stop the loop at an appropriate level of granularity.



(a) Initial: exact merges only.



(b) Converged DAG.

Figure 2: Risk-graph convergence under the validator-arbitrated merge pipeline. Zoom for node labels. (*More aggressive reductions are possible with a less conservative threshold for the judge.*)

4.2 Betweenness centrality

Fig. 4 shows the betweenness centrality ranking on the converged DAG. Notice the log-scale x-axis, with some nodes having much higher betweenness than the rest.

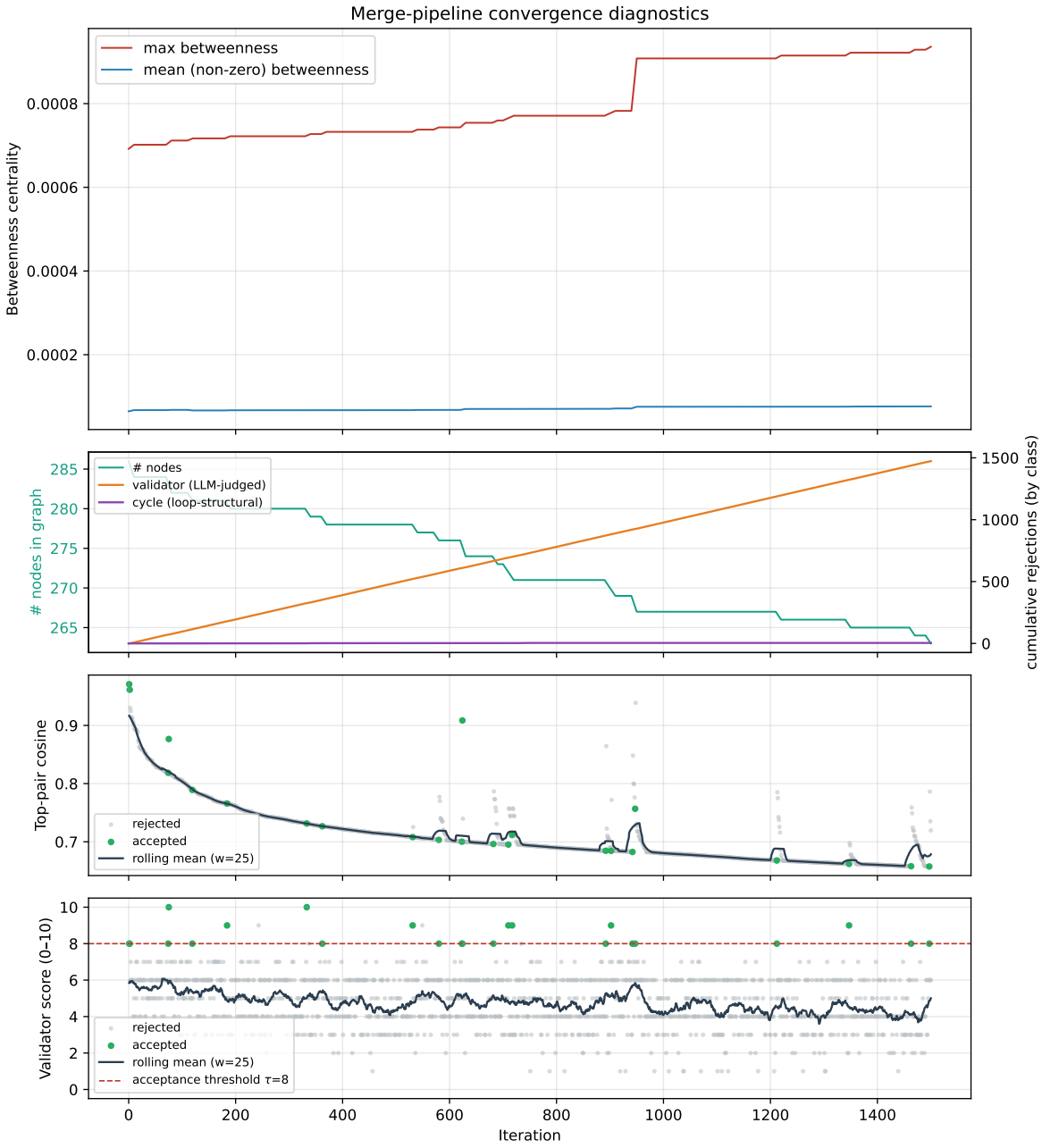


Figure 3: Merge-pipeline convergence diagnostics across iterations, sharing one iteration axis so a candidate snapshot step can be read off every signal at once. Panel 1: max and non-zero-mean betweenness centrality. Panel 2: node count (left axis) and cumulative merge rejections by class (cycle / validator / judge-error / policy-refusal, right axis); loop-structural cycle refusals and LLM-judged validator refusals are different convergence signals so they are split. Panel 3: top-pair cosine for every evaluated candidate, with rolling mean; the cosine floor falls smoothly as the queue empties of near-paraphrases. Panel 4: validator score for every evaluated candidate, rolling mean, and the rubric-fixed acceptance threshold $\tau = 8$ (Section B). Accepted merges are marked green; rejected, grey.

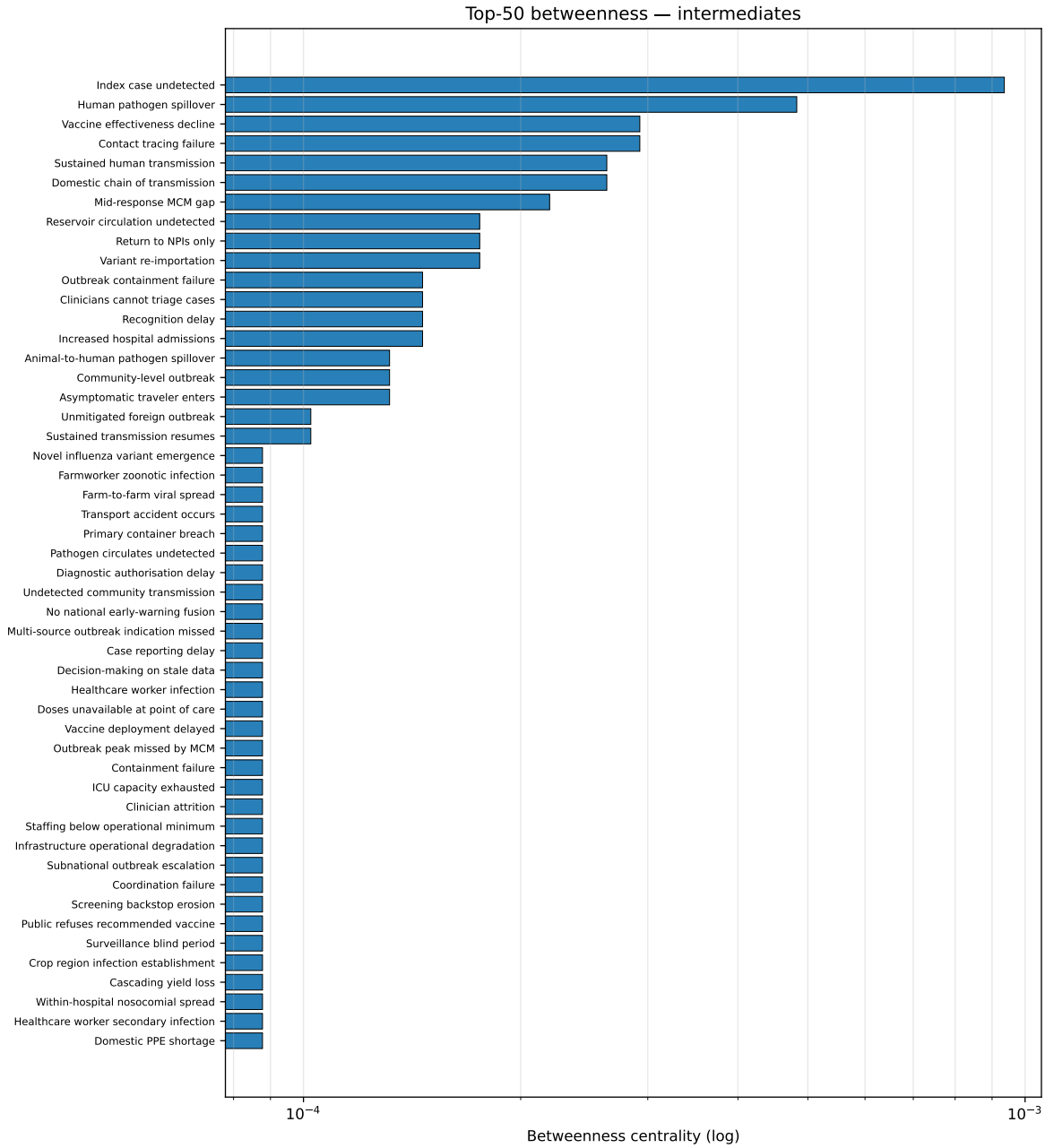


Figure 4: Top-k intermediates by betweenness centrality on the converged DAG, computed via NetworkX (Section D).

4.3 Importance rankings under elicited parameters

The following results are obtained from Monte-Carlo sampling using the LLM-elicited parameters for the extracted graph.

Expected harm by node type

Fig. 5 ranks outcomes by unconditional mean expected harm $\mathbb{E}[\text{harm}_s] = \mathbb{E}[X_s \cdot h_s]$ (sample mean over $K = 1000$ joint Monte Carlo trajectories). Sources are ranked the same way (Fig. 6), with each row's bar summed over outcomes reachable from that source.

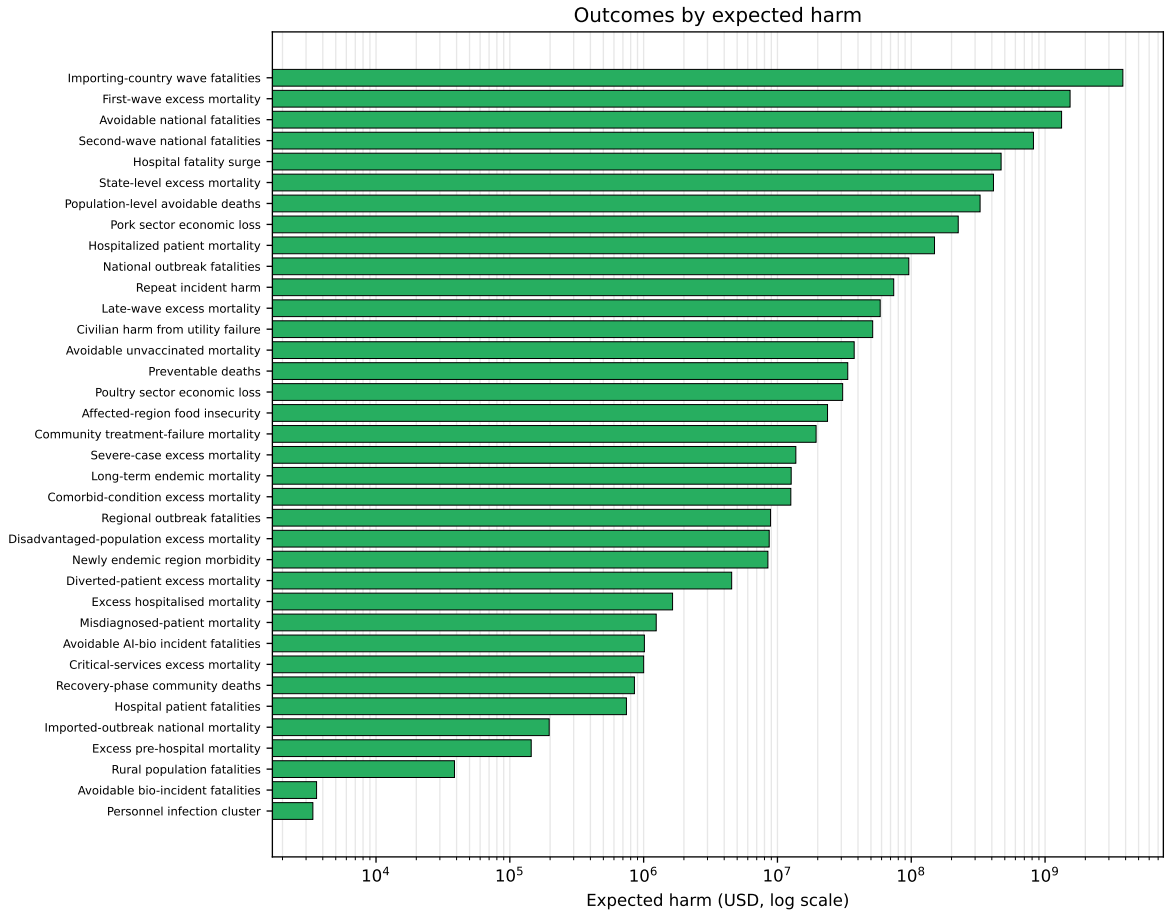


Figure 5: Top outcomes ranked by mean expected harm applying Monte Carlo sampling to the extracted risk model. Log x-axis, USD.

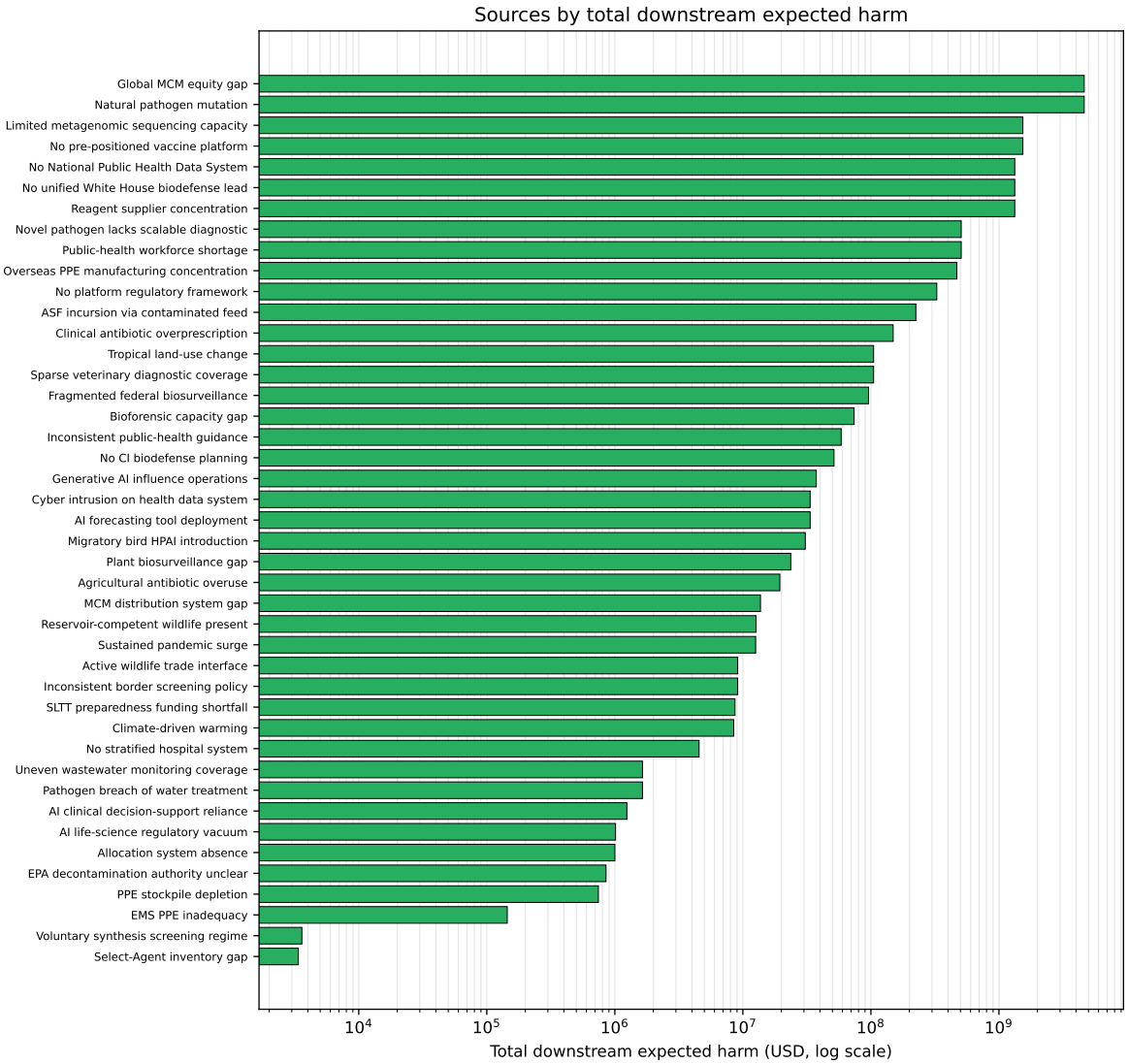
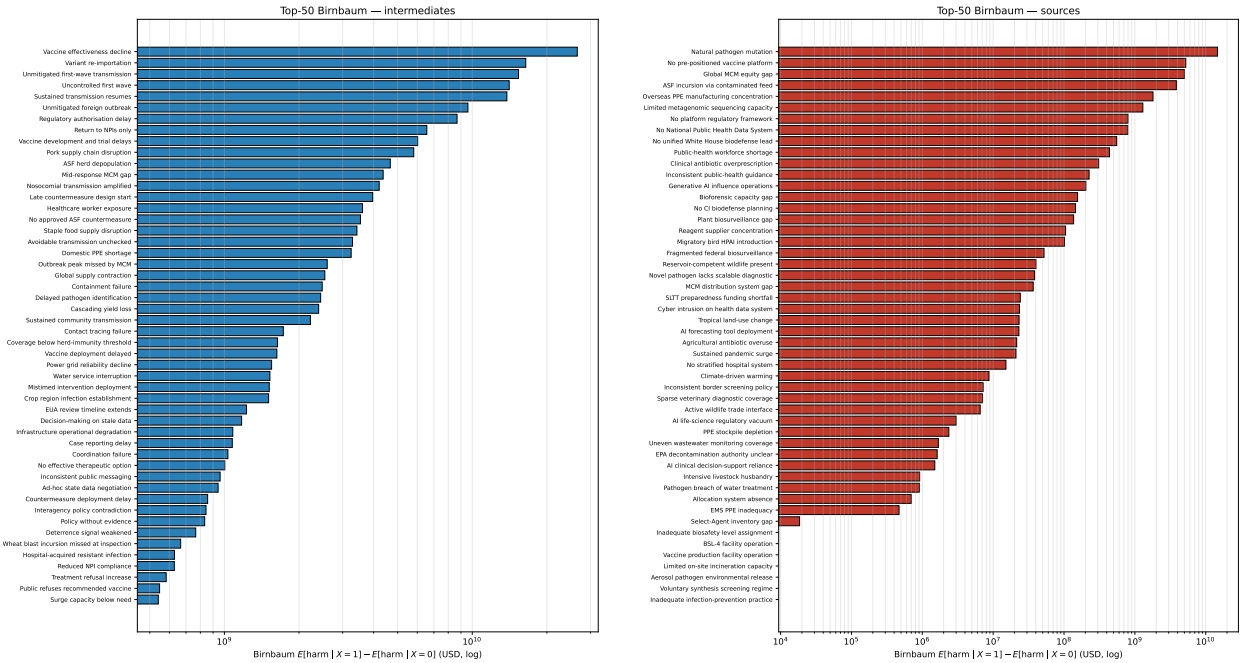


Figure 6: Top sources ranked by total downstream expected harm, summed over reachable outcomes.

Birnbaum importance

Fig. 7 reports the on/off swing per node type. This metric captures the expected harm reduction from perfectly intervening to clamp a node off, so it is a measure of the maximum possible impact of interventions on that node. Notice again, the log-scale x-axis, with the top few nodes having much higher Birnbaum importance than the rest.



(a) Intermediate nodes.

(b) Sources.

Figure 7: Top-k Birnbaum importance per node type. X-axis: log-scale expected-harm swing in the domain numeraire.

4.4 Expected Value of Partial Perfect Information (EVPPI)

Top-EVPPI nodes are the ones where additional evaluations, monitoring, or expert elicitation have the highest expected value of information. Fig. 8 shows the EVPPI ranking for intermediate nodes.

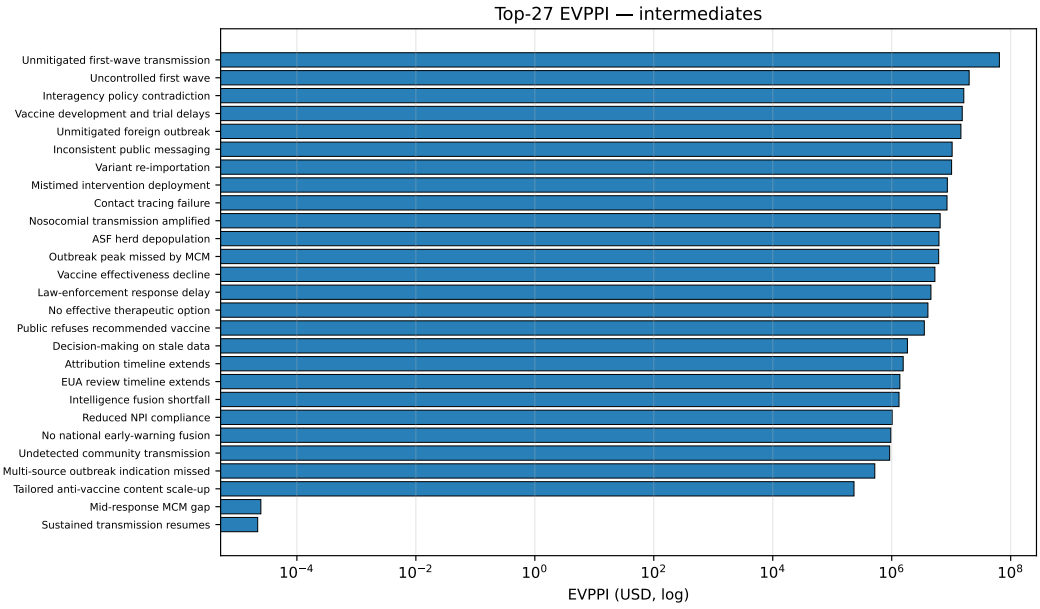


Figure 8: EVPPI ranking on intermediate nodes. Higher EVPPI means that learning the node's state (through for example an AI evaluation) would change the optimal intervention, and is therefore useful for decision-making that mitigates risk.

5 Conclusion

I presented an automated pipeline that turns a risk source document into a quantified causal risk model that can be used to prioritize risk and AI evaluation selection. The pipeline includes LLM causal chain extraction, embedding-based candidate matching, validator-arbitrated node merging, LLM parameter elicitation, and Monte Carlo analysis for risk estimation. The main value of this approach is in identifying the small set of nodes in a vast risk landscape where additional human elicitation, monitoring, or evaluation design has the greatest expected payoff.

References

[1] Bipartisan Commission on Biodefense. The national blueprint for biodefense: Immediate action needed to defend against biological threats. https://biodefensecommission.org/wp-content/uploads/2024/05/National-Blueprint-for-Biodefense-2024_final_digital.pdf, 4 2024.

[2] Z. W. Birnbaum. On the importance of different components in a multicomponent system. In P. R. Krishnaiah, editor, *Multivariate Analysis II*, pages 581–592. Academic Press, 1969. Available as Boeing technical report, https://apps.dtic.mil/sti/citations/AD0670563.

[3] Alexander Capstick et al. AutoElicit: Using large language models for expert prior elicitation in predictive modelling. https://arxiv.org/abs/2411.17284, 2025. ICML 2025.

[4] Charles E. Clark. The PERT model for the distribution of an activity time. *Operations Research*, 10(3):405–406, 1962. https://doi.org/10.1287/opre.10.3.405.

[5] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977. https://doi.org/10.2307/3033543.

[6] International AI Safety Report Writing Group. International AI safety report 2026. https://internationalaisafetyreport.org/, 2 2026.

[7] Malcolm Murray, Steve Barrett, Henry Papadatos, Otter Quarks, Matt Smith, Alejandro Tlaie Boria, Chloé Touzet, and Siméon Campos. A methodology for quantitative AI risk modeling. https://arxiv.org/abs/2512.08844, 2025.

[8] Luca Righetti. Dual-use AI capabilities and the risk of bioterrorism: Converting capability evaluations to risk assessments. https://cdn.governance.ai/Dual-Use_AI_Capabilities_and_the_Risk_of_Bioterrorism.pdf, 12 2025.

[9] SaferAI. Toward quantitative modeling of cybersecurity risks. https://www.safer-ai.org/u/2025/12/Toward-Quantitative-Modeling-of-Cybersecurity-Risks_SaferAI.pdf, 12 2025.

[10] Mark Strong, Jeremy E. Oakley, and Alan Brennan. Estimating multiparameter partial expected value of perfect information from a probabilistic sensitivity analysis sample: A nonparametric regression approach. *Medical Decision Making*, 34(3):311–326, 2014. https://doi.org/10.1177/0272989X13505910.

A Limitations and Dual-Use Considerations

Limitations

- **Convergence criteria.** It is not obvious which metric to choose to determine whether the appropriate level of granularity in the graph has been reached.

- **Single-shot LLM elicitation.** AutoElicit [3] recommends paraphrase ensembling. I did not do this, but it makes sense to do in future with larger compute budgets.
- **Content-policy refusals.** The extraction and elicitation prompts repeatedly hit safety filters, especially on biorisk content. The pipeline might underrepresent risks the model refuses to discuss.
- **The extraction of causal chains is not reproducible.** Chain extraction was driven by an interactive Claude Code subscription session rather than the Anthropic API, so the run cannot be replayed deterministically from a script.
- Inference gets expensive on large graphs, especially EVPPI.
- **Merge justification is non-obvious.** The judge model returns a numeric score with a brief rationale; we threshold the score, but the underlying notion of "same mechanism" is fuzzy.
- **Single-model judge and validator.** Both the merge proposer and the merge validator currently use the same model (Google Gemini 2.5 Flash Lite via OpenRouter). This may inflate self-agreement.
- **Heavy AI use.** I vibe-coded this project in a weekend with Claude Code (Opus 4.7). Virtually none of the code is human-written, although I likely could have written it with much patience. The code was not human-reviewed before the deadline to the extent that it should be. Large parts of the writing of this report was also generated by an LLM; I checked and decided what remained.

Dual-Use and Responsible-Disclosure Considerations[2]

Dual-Use risks.

- Outputs are structural rankings, not attack instructions, or capability advancements.
- Risk mechanisms surfaced by the pipeline are publicly available from the document used as seed.
- Re-running the pipeline on a more sensitive seed paper could produce a target list, although API-based LLM content-policy refusals may limit this.

Responsible-disclosure

- No vulnerabilities were discovered in the course of this project.

Ethical considerations.

- The pipeline is intended for prioritisation of expert and evaluation effort, not adjudication of policy.
- Acting on AI-generated risk numbers requires human expert validation and human responsibility for the downstream decisions.

Future work.

- Experiment with, and validate elicitation protocols
- Run an ablation comparing LLM-based pairwise merging without cosine similarity pre-filtering.
- It is not obvious that the two-stage merge-then-validate pipeline with threshold setting on the validate judge is optimal. It is possible to use a simpler model where the threshold is effectively tuned by changing the prompt of the validate judge, and letting it output accept / reject directly. An even simpler approach lets the merge model directly output whether the merge is valid.
- Build stronger prompts from literature on what constitutes a good causal chain, and a good node.
- Validate the inference pipeline and consider more computationally efficient alternatives for EVPPI in large graphs.

[2] Required by Apart Hackathon

- Use batching for the merge and validation LLM's on the top K cosine similarity pairs for speedup.

B Judge-Score Threshold Calibration

The validator returns an integer score $s \in [0, 10]$ per candidate merge. The acceptance threshold is fixed at $\tau = 8$ based on a validation set.

To check the live model behaves consistently with the rubric, I score two 50-pair validation sets (paraphrases and false-mergers, AI + biorisk) and report the empirical TPR and FPR at τ. Fig. 9 shows the two distributions. In retrospect, the validation set is too small, and the threshold was too conservative to demonstrate graph reduction. Future work can consider doing away with this additional complexity and using a single accept / reject label from the judge.

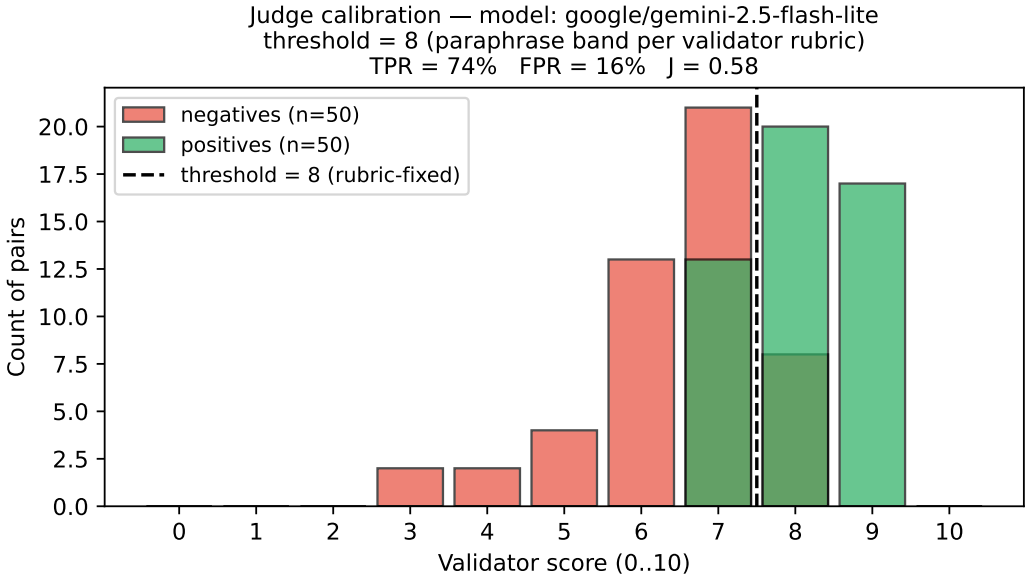


Figure 9: Validator-score distributions on the positive (paraphrase, green) and negative (false-merge, red) validation sets. Dashed line: $\tau = 8$. Title reports empirical TPR / FPR / Youden J.

C Merge Prompts

The merge loop (Section B) runs two LLM calls per candidate pair: a *proposer* that always emits a canonical label, and an independent *validator* that scores the proposed merge in $[0, 10]$. Both prompts are reproduced verbatim from `prompts/proposer.md` and `prompts/validator.md`. Required template keys (`label_a`, `type_a`, `chain_a`, `label_b`, `type_b`, `chain_b`, plus `canonical_label` for the validator) are filled at runtime from the candidate pair and the surrounding chain context.

Proposer

```
You are merging two nodes from a causal graph into one node.  Propose a single canonical
label that could replace both nodes while preserving the meaning of BOTH chains they
come from.

--- NODE A ---
Label:  {label_a}
Type:  {type_a}

Chain containing NODE A (nodes and edges):
{chain_a}
```

```
--- NODE B ---
Label: {label_b}
Type: {type_b}

Chain containing NODE B (nodes and edges):
{chain_b}

Guidance for the canonical label:
  - Short noun phrase, 2--6 words.
  - If A and B refer to the same event, use the clearer phrasing.
  - If they are at different scales (e.g. local vs national), use the broader one.
  - If one is compound and one is atomic, prefer the atomic phrasing.
  - If they are genuinely different events, still produce a label that names the
shared aspect (a downstream validator will catch unjustified merges).

Respond with EXACTLY this JSON, no other text:

{
    "canonical_label": "short noun phrase",
    "rationale": "one sentence"
}
```

Validator

```
You are auditing a proposed merge of two nodes from a causal graph. A primary judge
already proposed a canonical label. Your job is to SCORE how valid the merge is
on a 0 to 10 integer scale. You are not asked to re-propose a label, only to score
this one.

You are an INDEPENDENT check: you do not see the proposer's reasoning, and your
score must rest only on the two nodes, their chains, and the canonical label.

--- PROPOSAL ---
Canonical label: {canonical_label}

--- NODE A ---
Label: {label_a}
Type: {type_a}

Chain containing NODE A:
{chain_a}

--- NODE B ---
Label: {label_b}
Type: {type_b}

Chain containing NODE B:
{chain_b}

--- SCORING RUBRIC (return an integer 0--10) ---

10: A and B are the same real-world event at the same granularity; the canonical
label is accurate for both; both chains read correctly with the label substituted.
8--9: Strong paraphrase / near-synonymy; very minor information loss.
5--7: Related but distinct; the canonical label works for one node more than the
other; meaningful information is lost.
3--4: Different mechanisms, different scales, or different phases in a shared process;
merging would be misleading.
1--2: The nodes share only surface tokens (same word, different concept); the canonical
label obscures the difference.
0: Completely different concepts; merging destroys both chains.

--- CRITERIA TO APPLY ---
  1. Same event vs different events at different stages of a pipeline.
  2. Same spatial/temporal scale (individual vs population, local vs national, instant
vs protracted).
  3. Same granularity (atomic single mechanism vs compound bundle).
```

```
  4.  Is the canonical label an accurate summary for BOTH nodes, or does it fit one
better?
  5.  If NODE A were replaced with the canonical label in its chain, would the chain
still read correctly?  Same for NODE B?

Respond with EXACTLY this JSON, no other text:

{
    "score":  <integer 0 to 10>,
    "reason":  "one sentence"
}
```