DeXposure-Claw: An Agentic System for DeFi Risk Supervision

Aijie Shu
University of Edinburgh
v1ashu@ed.ac.uk

Bowei Chen
University of Glasgow
bowei.chen@glasgow.ac.uk

Wenbin Wu
University of Cambridge
w.wu@jbs.cam.ac.uk

Cathy Yi-Hsuan Chen
University of Glasgow
CathyYi-Hsuan.Chen@glasgow.ac.uk

Fengxiang He
University of Edinburgh
fhe@ed.ac.uk

Abstract

Decentralized finance exposes supervisors to fast-moving, networked credit risks. General-purpose LLM agents fit this setting poorly: they over-read weak evidence and recommend high-stakes interventions, while existing evaluations offer no regulator-aligned way to measure the resulting false alarms. We introduce DeXposure-Claw, a forecast-grounded agentic supervision system that routes LLM decisions through structured evidence: (1) DeXposure-FM, a graph time-series foundation model, forecasts future exposure networks; (2) deterministic monitors and stress scenarios then turn those forecasts into typed alerts, attribution signals, and scenario evidence; and (3) data-health and confidence gates constrain escalation before DeXposure-Claw emits auditable supervisory tickets with rationales. We further develop DeXposure-Bench, a six-axis evaluation harness, whose decision axis scores tickets against a regulator-aligned absolute-loss ground truth and an explicit false-intervention rate. Experiments on five years of weekly real data fully support our system. Code is at https://anonymous.4open.science/r/claw-system-8637/.

1 Introduction

Decentralized finance (DeFi) creates fast-moving networks of token-mediated credit exposure among lending protocols, decentralized exchanges, stablecoins, bridges, and yield aggregators. Recent crises such as Terra/Luna, FTX, and SVB/USDC show that shocks can propagate across this network before supervisors can manually inspect raw on-chain data (Vidal-Tomás et al., 2023). This motivates decision-support systems that can forecast emerging exposure risk, identify affected protocols, and recommend supervisory responses.

A straightforward approach is to build an LLM agent that reasons directly over on-chain measurements. We argue that this is unsafe for high-stakes supervision: general-purpose LLM agents may produce plausible rationales while over-reading incomplete, stale, or weak evidence, thereby triggering unnecessary high-severity interventions. Existing systemic-risk evaluations also often rank protocols by fractional exposure changes (Bertomeu et al., 2024; Gonon et al., 2025; Li et al., 2025), which can overemphasize small protocols and fail to reflect regulator-relevant loss priorities.

We introduce *DeXposure-Claw*, a forecast-grounded agentic system for DeFi risk monitoring and decision recommendation. Rather than asking an LLM to reason over raw transactions, the system routes decisions through structured forecasted evidence. A graph time-series foundation model forecasts future exposure networks; deterministic monitors and stress scenarios convert these forecasts into alerts, attribution information, scenario losses, and uncertainty estimates; an LLM then emits ranked supervisory tickets with targets, severities, and evidence-linked rationales. Data-health and confidence gates restrict intervention-level recommendations, and every ticket is released with its evidence bundle, rationale, and gate states for auditability.

We evaluate the system with DeXposure-Bench, a six-axis harness covering forecast quality, warning behavior, uncertainty calibration, stress-scenario fidelity, ticket quality, and robustness. Its decision axis uses a regulator-aligned absolute-loss ground truth, allowing false interventions and supervisory relevance to be measured directly. Across five years of weekly DeFi exposure graphs and eight reference implementations, routing the LLM through forecasted evidence raises ticket F1 from 0.0076 for a conservative persistence-rules baseline to 0.0288 with Claude Sonnet 4.6, the decision model we recommend (better F1 and explanations at $\sim 5\times$ lower cost than Opus 4.7), but this buys coverage and auditability, not safety. The LLM over-reads the forecaster, misfiring on roughly 37%

of its interventions, and a stronger Opus 4.7 is no better (44%, false-intervention rate 0.437 even with the safety gate); over-intervention thus persists regardless of model, so safe high-severity action comes from the data-health and confidence gates and human review, not from the decision model. DeXposure-Claw is thus an auditable recall-and-explanation option for human-in-the-loop DeFi supervision, not a replacement for conservative rule-based systems.

2 Related Work

Bench positioning and ground-truth definition. LLM-agent benchmarks (HELM (Liang et al., 2022), SWE-bench (Jimenez et al., 2024), AgentBench (Liu et al., 2024)) score open-ended reasoning, software repair, and generic agent behaviour; temporal-graph benchmarks (TGB (Huang et al., 2023), OGB (Hu et al., 2020)) score structural prediction quality. Neither scores whether an LLM agent's supervisory decisions match what a regulator would actually prioritise. The closest prior systemic-risk evaluations (Bertomeu et al., 2024; Gonon et al., 2025; Li et al., 2025) rank protocols by *fractional* weight change, which disproportionately surfaces tiny protocols of low systemic relevance. DeXposure-Bench replaces this with an *absolute-loss* ground truth (§3).

LLM agents in finance and DeFi. General agent patterns such as ReAct (Yao et al., 2023) combine reasoning with tool use, and FinGPT (Yang et al., 2023) adapts language models to financial data. DeFi-specific agents inherit this template but, whatever their target (transaction auditing (Yao et al., 2026), intent mining (Mao et al., 2025), smart-contract verification (Hu et al., 2026; Kong et al., 2026), price-manipulation detection (Liu et al., 2025), anomaly explanation (Watson et al., 2025), asset-preference auditing (Wu, 2026), or portfolio construction from graph-plus-LLM encoders (Luo et al., 2025; Jeon and Lee, 2026)), all reason over *raw* transactions or token text and are judged on detection accuracy. None feeds an LLM *structured forecasted evidence*, and none reports a false-intervention rate against a regulator-aligned ground truth, the gap our characterisation (§5.3) closes.

Forecast-grounded LLM agents in other domains. Pairing a domain forecaster with an LLM decision layer is an emerging deployment pattern. In macroeconomics, ChatGPT-augmented PMI nowcasting (de Bondt and Sun, 2025) and LLM-driven macroeconomic forecasting (Carriero et al., 2025) both feed structured numeric inputs into an LLM that produces interpretable narratives; a BIS primer surveys the wider pattern (Kwon et al., 2024). Time-series foundation models such as Chronos (Ansari et al., 2024), Lag-Llama (Rasul et al., 2024), and TimesFM (Das et al., 2024) make the forecaster reusable across tasks; tabular foundation models extend the same idea to heterogeneous structured data (Hollmann et al., 2025; Eremeev et al., 2025). None of these forecast→LLM pipelines, to our knowledge, has been scored against a regulator-aligned ground truth for high-stakes financial-network supervision.

3 Preliminaries

We model decentralized-finance credit exposure as a temporal weighted directed graph. Let $\mathcal{P}$ be the universe of protocols and $\mathcal{X}$ the universe of tokens. At time t, protocol p holds tokens $X_p(t) \subseteq \mathcal{X}$, while protocol q issues or is economically linked to tokens $X_G(q) \subseteq \mathcal{X}$. Protocol p is exposed to q if $X_p(t) \cap X_G(q) \neq \emptyset$. The exposure network is $G_t = (V_t, E_t, W_t)$, where $V_t \subseteq \mathcal{P}$ is the active protocol set, $E_t \subseteq V_t \times V_t$ is the directed exposure set, and $W_t[p, q] = w_{pq,t} \geq 0$ is the USD exposure from p to q. For protocol v, define its incident exposure mass as

$$w_t(v) = \sum_{u \in V_t} w_{uv,t} + \sum_{u \in V_t} w_{vu,t}.$$

The observed history is a sequence of snapshots $\mathcal{G}_{1:T} = \{G_1, \ldots, G_T\}$, optionally with covariates X_t.

At decision epoch t, the forecasting problem is to estimate G_{t+h} for horizons $h \in \mathcal{H}$. A probabilistic temporal-graph forecaster outputs

$$P_{t,h} = P(G_{t+h} \mid G_{\leq t}, X_{\leq t}).$$

For each ordered pair (p, q), let

$$\pi_{pq,t+h} = \Pr(e_{pq,t+h} = 1 \mid G_{\leq t}, X_{\leq t}),$$
$$\mu_{pq,t+h} = \mathbb{E}[w_{pq,t+h} \mid e_{pq,t+h} = 1, G_{\leq t}, X_{\leq t}].$$

A representative predicted graph $\widehat{G}_{t+h}$ is constructed by

$$\widehat{E}_{t+h} = \{(p, q) : \pi_{pq,t+h} > \pi_{\min}\},$$
$$\widehat{w}_{pq,t+h} = \pi_{pq,t+h}\mu_{pq,t+h}.$$

Uncertainty is represented by Monte Carlo samples

$$\widetilde{G}_{t+h}^{(1)}, \ldots, \widetilde{G}_{t+h}^{(M)} \sim P_{t,h}.$$

A monitor maps graphs to scalar risk statistics $\Phi_j : \mathcal{G} \to \mathbb{R}$, such as maximum PageRank, HHI concentration, density, PageRank Gini, or degree Gini. An alert fires when the predicted-graph statistic $\Phi_j(\widehat{G}_{t+h})$ deviates from its length-L rolling baseline by more than z standard deviations; the rolling mean/standard deviation and the z-score alert rule are defined in Appendix A.

A stress scenario is a graph perturbation $s : \mathcal{G} \to \mathcal{G}$ that removes exposure mass. We summarize its effect on the Monte Carlo samples by the conditional value-at-risk $\mathrm{CVaR}_\lambda(s, h)$, the mean of the worst $\lceil \lambda M \rceil$ sampled loss fractions; the per-scenario loss fraction $L(s; G)$ and CVaR are defined in Appendix A. Evaluation metrics are given in Appendix B.1.

4 DeXposure-Claw Pipeline

This section introduces the DeXposure-Claw Pipeline.

4.1 Overview

LLM-mediated supervision risks turning weak, stale, or uncertain evidence into confident recommendations. DeXposure-Claw therefore separates forecasting, evidence building, ticket drafting, and ticket release (Figure 1). Layer 1 forecasts future credit-exposure graphs from the current weekly snapshot. Layer 2 builds a typed evidence bundle from those forecasts: alerts, attribution, uncertainty estimates, and stress-scenario losses. Layer 3 drafts ranked supervisory tickets from this evidence alone, the only stage that calls the LLM. Layer 4 gates intervention-level tickets before release. This decomposition keeps the LLM as a constrained drafting component rather than the release authority. Despite this four-stage decomposition, the pipeline still runs at weekly cadence on a single RTX 4090, with well under a dollar of LLM API spend per weekly decision (a full DeXposure-Bench leaderboard run is approximately $10).

4.2 Layer 1: Forecasting

At decision epoch t, DeXposure-Claw observes the exposure graph G_t defined in Section 3. For each forecast horizon $h \in \{1, 4, 8, 12\}$ weeks, it queries DeXposure-FM for a predictive distribution over future exposure and constructs an expected-weight predicted graph $\widehat{G}_{t+h}$. It also draws Monte Carlo graph samples that carry predictive uncertainty into Layer 2. The predicted-graph construction and all forecast hyperparameters ($\pi_{\min}$, sample count, horizons) are given in Appendix A.1.

4.3 Layer 2: Monitoring

Layer 2 turns the forecast into a typed evidence bundle along two parallel tracks, monitors and stress scenarios. It also scores the snapshot's *data health* $\mathrm{DH}_t \in [0, 1]$, the mean of freshness, missingness, topology, and discrepancy checks (Section 3), which later modulates both alert confidence and the Layer 4 gates. Given $\widehat{G}_{t+h}$ and its Monte Carlo samples, five *monitor functionals* summarize concentration, fragility, and instability in the predicted graph: top-1 PageRank (Brin and Page, 1998), HHI concentration (Rhoades, 1993), network density, PageRank Gini, and degree Gini. An alert fires when a monitor statistic deviates from its rolling baseline by more than two standard deviations. Each alert carries a marginal-contribution top-K edge attribution and an uncertainty-aware confidence score $C_t(a) \in [0, 1]$, which decreases with lower data health, wider Monte Carlo dispersion, and longer forecast horizons.

In parallel, a *scenario engine* applies five standardized stress shocks to both $\widehat{G}_{t+h}$ and each Monte Carlo sample: single protocol failure, bridge cluster failure, stablecoin de-peg, sector lending shock, and correlated top-10 stress. For each scenario s and horizon h, the engine aggregates the worst-half sampled losses as $\mathrm{CVaR}_{0.5}(s, h)$. These alerts, attribution fields, confidence scores, and scenario-CVaR table form the evidence consumed by the LLM. Full monitor and scenario definitions are provided in Appendix A.

4.4 Layer 3: Ticket Drafting

Layer 3 is the only stage that invokes the LLM, which receives the evidence bundle as a typed JSON payload containing alerts with attribution top-K, a scenario-CVaR table, Monte Carlo dispersion summaries, and the scalar data-health score DH_t. It drafts, but does not release, a ranked list of supervisory tickets. Each ticket contains (a) a severity level from a fixed four-action playbook (`Monitor`, `Investigate`, `Recommend-Reduce`, and `Contingency`), (b) a target set of protocols, and (c) a rationale that cites specific evidence fields by name. Each decision is drafted three times at temperature 0; the Jaccard

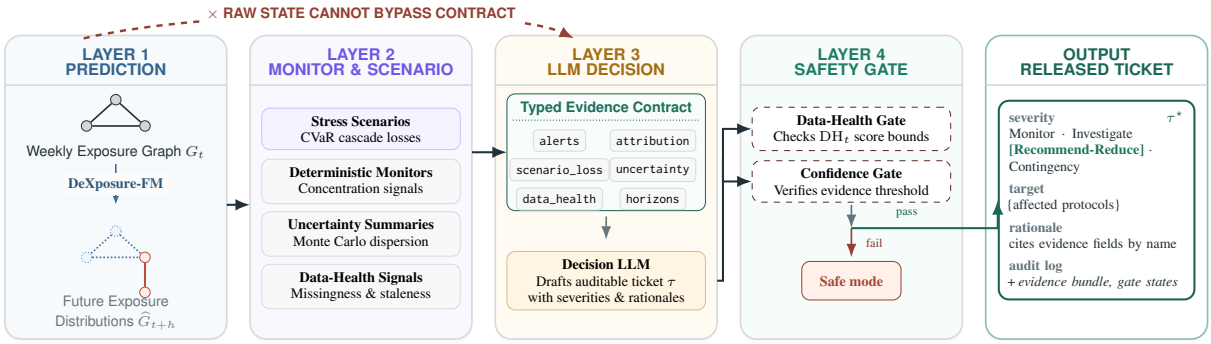

Figure 1: DeXposure-Claw system overview. A weekly exposure graph G_t is forecast into future exposure distributions and deterministic risk signals, then converted into a typed evidence bundle containing monitor alerts, attribution, stress-scenario losses, uncertainty summaries, data-health signals, and forecast horizons. The LLM drafts supervisory tickets only from this structured evidence, after which data-health and confidence gates either release an auditable ticket $\tau^\star$ or enter safe mode.

overlap of ticket targets and severities across the three drafts is the *target-stability* signal reported in our evaluation.

4.5 Layer 4: Safety Gates

Candidate tickets are not released by default. Layer 4 applies data-health and confidence gates to determine whether intervention-level recommendations are feasible. The *data-health gate* thresholds the Layer 2 score: if $\mathrm{DH}_t < \tau_{\mathrm{data}} = 0.7$, the system enters safe mode and emits only `Monitor` or `Investigate` tickets. The *confidence gate* blocks intervention-level tickets whenever the mean alert confidence satisfies $\bar{C}_t < \tau_{\mathrm{conf}} = 0.6$, even if the data-health gate passes. Every emitted ticket carries the full evidence bundle, the LLM rationale, and the gate states that produced it. The resulting audit log is the unit of release (Section 6). The end-to-end procedure is given in Appendix A (Algorithm A.1).

5 Experiments

This section describes experiments.

5.1 Evaluation Questions

We organise the evaluation around three questions. **RQ1**: does the forecaster earn its place? **RQ2**: does forecast-grounding change the agent? **RQ3**: is each component load-bearing? Answering all three means scoring the forecast and the agent's interventions against a regulator-aligned ground truth that standard leaderboards omit, so we build DeXposure-Bench.

5.2 DeXposure-Bench

We develop a DeXposure-Bench harness that scores the pipeline on six independent axes spanning forecasting, warning, and decision quality (full schemas in Table B.1):

- `b1_forecast` – temporal graph forecast quality;
- `b2_warning` – streaming early-warning lead time;
- `b3_calibration` – predictive-uncertainty calibration;
- `b4_stress` – what-if stress-scenario fidelity;
- `b5_decision` – supervisory ticket quality;
- `b6_robustness` – degraded-data robustness.

The decision and warning axes share the absolute-loss ground truth of §3 (top-π stressed set, $\pi = 0.05$, horizon $h = 4$ weeks), avoiding fractional-change rankings that over-weight tiny protocols; this yields $|\mathcal{S}_t^h| \approx 283$ protocols per week and $n = 29$ scored test weeks, as the final four snapshots lack a 4-week-ahead target. The forecasting axes (`b1`, `b3`, `b6`) answer RQ1 by scoring the forecast before any agent consumes it; the decision axis (`b5`) answers RQ2 by holding the forecast fixed and varying only the LLM's evidence, with the stress axis (`b4`) checking the upstream scenario fidelity that decision grounding relies on; RQ3 ablates components on `b5` under clean and degraded data.

Data, splits, and baselines. All experiments run on DeXposure weekly exposure graphs. The corpus holds 4,300 protocols, 24,300 tokens, 43.7M exposure entries, and 283 snapshots from 2020-03 to 2025-08 (Wu et al., 2025). All model selection uses pre-2025 data, with 2025 held out as the frozen

Table 1: **Forecast quality across predictors** (RQ1). Point estimates (n=29, h=4; PR-MAE in 10^{-5}), no week-level CIs. Bold marks the per-column best among accurate predictors. § persistence trend is 0 by construction; † EvolveGCN's lower Δ_{rel} is weak-but-stable (not robustness), so unbolded; PI-cov/ECE is delivered only by DeXposure-FM. Full per-horizon/-regime audits in §C.

Predictor (h=4)	PR-MAE↓	RankC.↑	Trend↑	Δ_{rel} ↓	PI-cov/ECE
persistence (m1)	**3.4**	**.570**	.000§	.148	—
EvolveGCN (m3)	3.5	.551	.324	.040†	—
DeXposure-FM (m5)	4.5	.558	**.628**	**.113**	**.913/.013**

test split (split dates in §B.3). The harness bundles eight reference methods (full specs in Table B.3):

- h1 – weighted-degree heuristic monitor;
- m1 – persistence + rules (decision baseline);
- m2 – snapshot LLM, no forecast;
- m3 – EvolveGCN (Pareja et al., 2020), a learned GNN;
- m4 – DeXposure-FM only;
- m5 – DeXposure-FM + rules;
- m6 – DeXposure-FM + LLM (full stack);
- m7 – DeXposure-FM + LLM + safety gate.

The predictor-only methods (m3, m4) emit no supervisory tickets and are scored on the forecast axes only. Full schemas and method details are in §B.3–§B.4; the historical Terra/Luna, FTX, and SVB/USDC warning study is reported separately from the 2025 leaderboard.

5.3 Empirical Results

We answer each question as a verdict (headline numbers in Tables 1–3). **RQ1**: persistence predicts point values more accurately, but only the forecaster emits the trend and calibrated uncertainty the agent consumes (b1, b3, b6). **RQ2**: routing the LLM through forecast evidence rather than raw snapshots buys coverage and auditability ($+31\%$ ticket F1 via a larger target set) but *not* per-target accuracy or safety; the decision layer misfires on $\sim$37–44% of its interventions across models. **RQ3**: every gate and scenario component is load-bearing under clean or degraded data, while the decision model is an efficiency, not a safety, choice.

RQ1: forecaster vs. persistence. On static error, persistence is a deceptively strong baseline (Table 1), leading on PageRank MAE and Spearman rank correlation. DeXposure-FM instead contributes the signals persistence structurally lacks, a non-trivial trend signal (persistence is 0 by construction), conformal calibration (PI coverage .913 at a .90 target, ECE .013), and 24% less degradation under data loss. EvolveGCN trails persistence on accuracy, consistent with its $\sim$220 weekly-snapshot training budget; full per-horizon and per-regime audits are in §C.

RQ2: grounding vs. raw snapshots. Routing the LLM through forecast evidence rather than raw snapshots lifts ticket F1 by $+31\%$ (m2→m6, $p < 10^{-4}$; Table 2A); this forecast-grounding gain and the LLM policy alone (m5→m6, $+27\%$) are additive and non-substitutable, so the full agentic stack (m1→m7) reaches $+208\%$ over the persistence+rules baseline. The baseline's higher single-axis precision (m1, .720) is not a safety guarantee, since m1/m2/m5 never intervene at all. Adding the FM signal instead makes m6 escalate, and 44.8% of those interventions misfire ($p < 10^{-4}$); grounding is 1.000 for both variants, so the failure mode is not citation hallucination but *over-reading the forecaster's evidence as warrant for high-severity action*. Two caveats temper the F1 story. *(i)* The explanation-quality lift ($2.24 \rightarrow 2.45$ judge) is directional, not confirmed. It holds under a GPT-5.5 judge but not the tier-above Claude Opus 4.8 (Zheng et al., 2023), and cross-family judges agree only on ranking m7 top. *(ii)* At a matched budget the FM-fed and raw-snapshot LLMs are indistinguishable per target up to the raw-snapshot model's ceiling of about five targets per week (Table 2B, $k \leq 5$; §C.4). The FM's contribution is not a higher per-target hit rate but a *larger* target set at undiminished precision. Past that ceiling ($k = 7$) it recovers significantly more stressed protocols at the same precision ($p \leq .0003$), and its tail targets hit at a rate comparable to its head, so the extra targets are genuine detections rather than padding. That expanded tail is also where over-intervention concentrates.

RQ3: component and model ablations. Read down the pipeline, each component owns one failure mode (Table 3A). Multi-horizon forecasting (A6) and the scenario engine (A3) supply the Layer 2 evidence; the data-health gate (A1) and confidence gate (A2) guard the Layer 4 output. On clean data only two are load-bearing. The scenario engine carries coverage, since skipping it collapses ticket precision to 0, and the confidence gate carries safety, since removing it lifts FIR to $.429$ from the gated $0/0$. The other two are dormant on clean data and act as reserves under stress, where the data-health gate suppresses false alarms

Table 2: **Ticket quality and matched-budget coverage** (RQ2). Frozen 2025 split (n=29); decisions Claude Opus 4.7, judge Claude Opus 4.8. *Interv.(wk)* = weeks with $\geq$1 intervention ticket; `m1`/`m2`/`m5` never escalate, so their FIR is undefined ($0/0$, n/a), *not* a safety guarantee. Brackets are 95% bootstrap CIs; †`m1` from a local re-run; $*$ saturated (no week reaches depth k, so the cell repeats the full set). Panel (B) truncates each method to its top-k targets/week; significance tests and full panels in §C.4, §C.3.

(A) End-to-end ticket quality

ID	Stack	Interv.(wk)	Prec.	Rec.	F1↑ [95% CI]	FIR↓ [95% CI]	Stab.	Judge [CI]
`m1`	persist.+rules	0/29	**.720**	.004	.0076 [.004,.012]†	n/a	.514	—
`m2`	snapshot LLM	0/29	.575	.009	.0184 [.014,.022]	n/a	.532	2.24 [2.10,2.41]
`m5`	FM+rules	0/29	.600	.010	.0190	n/a	.257	—
`m6`	FM+LLM	17/29	.570	.012	**.0241 [.020,.028]**	.448 [.293,.601]	.488	2.41 [2.21,2.62]
`m7`	FM+LLM+gate	16/29	.580	.012	.0234 [.019,.027]	.437 [.282,.592]	.435	**2.45 [2.24,2.66]**

(B) Matched-budget (R@k $\times 10^3$ / P@k, top-k targets/week)

Method	R@1	P@1	R@3	P@3	R@5	P@5	R@7	P@7
`m1` persist.+rules	0.71	.24	2.45	.26	4.05	.26	4.05*	.26
`m2` snapshot LLM	1.53	.52	5.70	.59	9.08	.58	9.35*	.58
`m6` FM+LLM	1.64	.55	5.30	.56	9.63	.59	**12.14**	.57
`m7` FM+LLM+gate	1.64	.55	5.30	.56	9.13	.57	**11.95**	.58

Table 3: **Component ablations and decision-model swap** (RQ3). *(A)* clean ablation vs `m5` (A1/A6 dormant on clean data). *(B)* data-health gate (A1) under degradation (τ_{conf}=0). *(C)* decision-model swap in `m7` (95% CIs). Full sweeps, A4/A5, and crisis volumes ($\sim$4$\times$) in §C.1, §C.1.

(A) Clean ablation (vs `m5`)

Config	Prec.	FIR
full `m5`	.600	n/a$^{0/0}$
− multi-horiz. (A6)	.600	→App
− scenario (A3)	**.000**	—
− data-health (A1)	.600	→(B)
− conf. gate (A2)	.600	.429

(B) Degraded: data-health gate (A1)

Regime · gate	Prec.	FIR	#int
extreme · off	.490	.510	57
extreme · on	.545	**.000**	0
severe · off	.538	.541	24
severe · on	.538	.400	25

(C) Decision-model swap (`m7`; efficiency, not safety)

Decision model	F1↑ [95% CI]	FIR↓ [95% CI]	rel. cost
Claude Opus 4.7	.0234 [.019,.027]	.437 [.282,.592]	1×
Claude Sonnet 4.6	**.0288 [.024,.034]**	.374 [.224,.529]	∼**0.2**×
Gemini 2.5 Pro	.0139 [.011,.017]	**.190 [.069,.328]**	—

(FIR $[.27, .60] \rightarrow [.00, .40]$, and to .000 under the strict gate at 80–98% feature/edge masking) and multi-horizon monitoring raises crisis-window alert volume $\sim$4$\times$ at unchanged precision (§C.1).

Swapping the decision model trades cost for quality without touching safety (Table 3C). A $\sim$5$\times$ cheaper Sonnet 4.6 raises `m7` F1 over Opus 4.7 ($0.0234 \rightarrow 0.0288$; $p < 0.001$) at comparable intervention volume and the same over-intervention rate (FIR 0.37 vs 0.44; n.s.), while Gemini 2.5 Pro lowers FIR only by intervening far less at much lower F1 (§C.4). Over-intervention is thus bounded by the gates, not the model, so the deployment point is `m7`$\times$Sonnet 4.6. The safety gate (`m6`→`m7`) trades 3% F1 within judge noise but adds a second auditable gate to every intervention.

6 Conclusion

We presented DeXposure-Claw, a forecast-grounded agentic system for DeFi supervision. Its lesson is that LLM agents should not reason directly over raw on-chain data in high-stakes settings, since without structured evidence and escalation controls they over-read weak signals and over-intervene. DeXposure-Claw instead routes decisions through DeXposure-FM forecasts, monitor and stress-scenario evidence, and data-health and confidence gates before emitting auditable tickets, while DeXposure-Bench, our six-axis harness, scores them on ticket quality, regulator-aligned loss targeting, and false-intervention risk. Forecast-grounding improves coverage and auditability, but safe deployment hinges on false-intervention measurement and model calibration.

Limitations

Single domain. DeXposure-Bench evaluates a single on-chain risk surface (DeFi inter-protocol credit exposure). Generalisation to NFT lending, perpetual derivatives, or cross-chain bridge networks is untested and will require domain-specific re-calibration of the conformal split and the stressed-set percentile π.

Weekly resolution. The pipeline operates at weekly granularity, but several DeFi shocks unfold within hours; Terra/Luna erased $40B in under 48 hours. A sub-hourly production deployment would require denser snapshot ingestion and re-evaluation of conformal calibration on the resulting denser time series. The 4–5 week lead times we report should be read as "how early the weekly monitor turns", not as end-to-end alert latency.

Pretraining exposure to named events. Well-known historical crises may appear in the decision and judge models' pretraining data, so explanation-quality judgments on named events can be inflated by recognition rather than evidence reading. We therefore report the LLM-judge score only as a directional signal, and the headline capability metrics do not depend on it. Ticket precision, F1, and false-intervention rate are scored against the absolute-loss ground truth, citation grounding stays traceable (1.000), and a fully anonymised crisis replay leaves only a small target-precision change ($\Delta{=}.026$). The 2025 leaderboard split also contains no comparably famous events. We defer the full masked-replay study to the extended version.

Ethics Statement

Data provenance. DeXposure is built from public on-chain measurements aggregated via DefiLlama (DefiLlama, 2026). No private user data, wallet-level personally-identifying information, or off-chain identity data is used. Protocol-level aggregates are the unit of analysis throughout.

Decision risk. DeXposure-Claw emits supervisory tickets, not automated trades or automated interventions. The high-severity action types (`Recommend-Reduce`, `Contingency`) are gated on a data-health score and a confidence score, and the pipeline emits a safe-mode notice when either gate fails. We characterise the system as decision support for a human supervisor, not as an autonomous risk agent. The measured false-intervention rate ($\mathrm{FIR} \approx 0.37$ for the deployed DeXposure-FM+Sonnet 4.6 variant, and not lowered by using a stronger model such as Opus 4.7) is reported in the results tables so that downstream operators can weigh the trade-off against their own risk tolerance before deployment.

LLM use disclosure. We used AI assistants to write code and to assist in writing this paper.

Release. We will release the harness, the eight reference implementations, and the ticket-level audit logs under a permissive open licence. Model checkpoints will be released subject to dataset-licence compatibility checks.

References

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. 2024. Chronos: Learning the language of time series. *Preprint*, arXiv:2403.07815.

Jeremy Bertomeu, Xiumin Martin, and Ibrahima Sall. 2024. Measuring defi risk. *Finance Research Letters*, 63:105321.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117. Proceedings of the Seventh International World Wide Web Conference.

Andrea Carriero, Davide Pettenuzzo, and Shubhranshu Shekhar. 2025. Macroeconomic forecasting with large language models. *Preprint*, arXiv:2407.00890.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. *Preprint*, arXiv:2310.10688. Accepted at ICML 2024; model commonly referenced as TimesFM.

Gabe J. de Bondt and Yiqiao Sun. 2025. Enhancing GDP nowcasts with ChatGPT: a novel application of PMI news releases. Working Paper Series 3063, European Central Bank, Frankfurt am Main, Germany. ECB Working Paper Series No. 3063; released 2025-06-30.

DefiLlama. 2026. DefiLlama: DeFi TVL aggregator and analytics. Data used: 2025; accessed: 2026-02-02.

Dmitry Eremeev, Gleb Bazhenov, Oleg Platonov, Artem Babenko, and Liudmila Prokhorenkova. 2025. Turning tabular foundation models into graph foundation models. *Preprint*, arXiv:2508.20906.

Lukas Gonon, Thilo Meyer-Brandis, and Niklas Weber. 2025. Computing systemic risk measures with graph neural networks. *Preprint*, arXiv:2410.07222.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. 2025. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326. Published online 8 Jan 2025.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Xiaohui Hu, Wun Yu Chan, Yuejie Shi, Qumeng Sun, Wei-Cheng Wang, Chiachih Wu, Haoyu Wang, and Ningyu He. 2026. An effective and cost-efficient agentic framework for ethereum smart contract auditing. *Preprint*, arXiv:2601.17833.

Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. 2023. Temporal graph benchmark for machine learning on temporal graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Joohyoung Jeon and Hongchul Lee. 2026. Can blindfolded LLMs still trade? An anonymization-first framework for portfolio optimization. *Preprint*, arXiv:2603.17692.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations (ICLR)*.

Ziqiao Kong, Wanxu Xia, Chong Wang, Yi Lu, Pan Li, Shaohua Li, Zong Cao, and Yang Liu. 2026. Knowdit: Agentic smart contract vulnerability detection with auditing knowledge summarization. *Preprint*, arXiv:2603.26270.

Byeungchun Kwon, Taejin Park, Fernando Perez-Cruz, and Phurichai Rungcharoenkitkul. 2024. Large language models: a primer for economists. *BIS Quarterly Review*.

Xurui Li, Xin Shan, Wenhao Yin, and Haijiao Wang. 2025. Heterogeneous graph pre-training based model for secure and efficient prediction of default risk propagation among bond issuers. *Preprint*, arXiv:2501.03268.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Lu Liu, Wuqi Zhang, Lili Wei, Hao Guan, Yongqiang Tian, Yepang Liu, and Shing-Chi Cheung. 2025. LLM-powered detection of price manipulation in DeFi. *Preprint*, arXiv:2510.21272.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2024. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations (ICLR)*.

Yichen Luo, Yebo Feng, Jiahua Xu, Paolo Tasca, and Yang Liu. 2025. Llm-powered multi-agent system for automated crypto portfolio management. *arXiv preprint arXiv:2501.00826*.

Qian'ang Mao, Yuxuan Zhang, Jiaman Chen, Wenjun Zhou, and Jiaqi Yan. 2025. Know your intent: An autonomous multi-perspective LLM agent framework for DeFi user transaction intent mining. *Preprint*, arXiv:2511.15456.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. 2020. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5363–5370.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. 2024. Lag-llama: Towards foundation models for probabilistic time series forecasting. *Preprint*, arXiv:2310.08278.

Stephen A. Rhoades. 1993. The herfindahl-hirschman index. *Federal Reserve Bulletin*, 79(3):188–189.

David Vidal-Tomás, Antonio Briola, and Tomaso Aste. 2023. Ftx's downfall and binance's consolidation: The fragility of centralised digital finance. *Physica A: Statistical Mechanics and its Applications*, 625:129044.

Adriana Watson, Grant Richards, and Daniel Schiff. 2025. Explain first, trust later: LLM-augmented explanations for graph-based crypto anomaly detection. *arXiv preprint arXiv:2506.14933*.

Wenbin Wu. 2026. Auditing asset-specific preferences in financial large language models: Evidence from Bitcoin representations and portfolio allocation. *Preprint*, arXiv:2606.02528.

Wenbin Wu, Kejiang Qian, Alexis Lui, Christopher Jack, Yue Wu, Peter McBurney, Fengxiang He, and Bryan Zhang. 2025. Dexposure: A dataset and benchmarks for inter-protocol credit exposure in decentralized financial networks. *Preprint*, arXiv:2511.22314.

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

Duanyi Yao, Siddhartha Jagannath, Baltasar Aroso, Vyas Krishnan, and Ding Zhao. 2026. Auditable LLM arbiter for DeFi security: A hybrid graph-of-thoughts approach to intent-transaction alignment. In *NDSS 2026 LAST-X Workshop*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS)*.

A Additional Algorithm Details

Section 3 defines the exposure graph, forecast distribution, predicted graph, monitor alert rule, scenario loss, ticket metrics, and gate feasibility conditions used throughout the paper. This appendix records only the concrete instantiation used by DeXposure-Claw.

A.1 Forecast instantiation

At each decision epoch t, DeXposure-Claw queries DeXposure-FM once for each horizon $h \in \{1, 4, 8, 12\}$ and constructs $\widehat{G}_{t+h}$ with the expected-weight operator from Section 3. We set the edge-existence threshold to $\pi_{\min} = 0.2$, tuned on the validation split. Uncertainty summaries use $M = 50$ Monte Carlo graph samples. Each sample first draws edge indicators from the predicted edge probabilities, then perturbs positive edge weights with the forecaster's predictive regression spread. For existing edges, residual predictions are folded into the conditional mean $\mu_{pq,t+h}$ before applying the expected-weight construction; candidate new edges enter only when their predicted existence probability exceeds $\pi_{\min}$.

A.2 Monitor instantiation

Given a rolling window of length L, each monitor Φ_j maintains a baseline mean and standard deviation

$$\mu_{j,t} = \frac{1}{L}\sum_{\ell=1}^{L} \Phi_j(G_{t-\ell}),$$

$$\sigma_{j,t} = \sqrt{\frac{1}{L-1}\sum_{\ell=1}^{L} \left(\Phi_j(G_{t-\ell}) - \mu_{j,t}\right)^2},$$

and an alert fires when the predicted-graph statistic leaves the band,

$$A_{j,t,h} = \mathbf{1}\left\{ \frac{\left|\Phi_j(\widehat{G}_{t+h}) - \mu_{j,t}\right|}{\max(\sigma_{j,t}, \epsilon)} > z \right\}.$$

We instantiate this rule with a 42-week baseline, $z = 2$, and $\epsilon = 10^{-8}$. PageRank uses damping 0.85 and 10 power iterations. Gini coefficients are computed over active nodes in the predicted graph.

Table A.1: Monitor metrics. The five graph statistics tracked by the structural monitor, each evaluated on the predicted graph $\widehat{G}_{t+h}$.

ID	Metric	Definition
N1	Systemic Importance	$\max_{p\in V} \mathrm{PageRank}(\widehat{G}_{t+h})_p$
N2	HHI Concentration	$\sum_p (d_p/\sum_q d_q)^2, d_p = \sum_q \widehat{w}_{pq,t+h}$
N3	Network Density	$\lvert\widehat{E}_{t+h}\rvert/\lvert V\rvert(\lvert V\rvert-1)$
N4	PageRank Gini	$\mathrm{Gini}(\{\mathrm{PageRank}_p\}_{p\in V})$
N5	Degree Gini	$\mathrm{Gini}(\{d_p\}_{p\in V})$

A.3 Scenario instantiation

Each shock maps a graph to a perturbed graph G^s with loss fraction

$$L(s;G) = \frac{\mathrm{Mass}(G) - \mathrm{Mass}(G^s)}{\mathrm{Mass}(G)},$$

$$\mathrm{Mass}(G) = \sum_{e\in E(G)} w_e.$$

For the sampled graphs let $\ell_m(s,h) = L(s;\widetilde{G}^{(m)}_{t+h})$ with sorted losses $\ell_{(1)} \geq \cdots \geq \ell_{(M)}$; the conditional value-at-risk is

$$\mathrm{CVaR}_\lambda(s,h) = \frac{1}{\lceil \lambda M\rceil}\sum_{m=1}^{\lceil \lambda M\rceil} \ell_{(m)}.$$

Table A.2: Stress scenarios. The five what-if shocks applied to the predicted graph, with the targeted node set and shock magnitude of each.

ID	Scenario	Target	Shock
S1	Single protocol fail	Highest-weight node	100%
S2	Bridge cluster fail	All bridge nodes	100%
S3	Stablecoin de-peg	All stablecoin nodes	50%
S4	Sector lending shock	All lending nodes	30%
S5	Correlated stress	Top-10 by weight	20%

Scenario targets use protocol-sector metadata from DeXposure. For each scenario and horizon, DeXposure-Claw reports the deterministic loss on $\widehat{G}_{t+h}$ and the sampled $\mathrm{CVaR}_{0.5}$.

A.4 Gate instantiation and ticket scoring

The data-health score of Section 3 averages four data-quality checks,

$$\mathrm{DH}_t = \frac{1}{4}\left(\mathrm{fresh}_t + \mathrm{miss}_t + \mathrm{topo}_t + \mathrm{disc}_t\right),$$

and the data-health gate fires at $\tau_{\mathrm{data}} = 0.7$. Safe mode still releases monitor alerts and scenario summaries, but suppresses intervention-level tickets (RECOMMENDREDUCE and CONTINGENCY). The confidence gate uses $\tau_{\mathrm{conf}} = 0.6$ and clamps each Monte Carlo coefficient of variation to $[0, 1]$ before computing alert confidence

$$C_t(a) = \mathrm{DH}_t \cdot \frac{1}{1 + \mathrm{CV}_{t,h}(a)} \cdot \frac{1}{1 + \log(1+h)},$$

where $\mathrm{CV}_{t,h}(a)$ is the Monte Carlo coefficient of variation of alert a at horizon h.

Feasible tickets are ranked by

$$\mathrm{Score}_t(u) = w_u \cdot \bar{C}_t \cdot \mathrm{Imp}_t, \tag{1}$$

where $w_u \in \{0.25, 0.50, 0.75, 1.00\}$ is the playbook severity weight for the four ordered actions and Imp_t is the mean scenario CVaR (default 1.0 when no scenario summary is available).

A.5 End-to-end procedure

Algorithm A.1 ties the components above together into a single decision epoch. Given the current graph snapshot and features, the system first computes a data-health score and sets the safe-mode flag, then for each horizon produces a forecast, builds the predicted graph, draws Monte Carlo samples, derives alerts with confidences, and evaluates scenario losses. The playbook stage finally maps the resulting alerts, scenario summary, and data-health state into tickets.

Algorithm A.1 DeXposure-Claw: one decision epoch.

Require: (G_t, X_t); horizons $\mathcal{H}$; scenarios $\mathcal{S}_0$; thresholds $(\tau_{\mathrm{data}}, \tau_{\mathrm{conf}}, z, \pi_{\min})$; MC count M

Ensure: Alerts $\mathcal{A}_t$; scenario summary $\mathcal{R}_t$; tickets $\mathcal{D}_t$

$\mathrm{DH}_t \leftarrow \mathrm{DataHealth}(G_t, X_t)$
$\mathrm{SAFE\_MODE}_t \leftarrow \mathbb{I}\{\mathrm{DH}_t < \tau_{\mathrm{data}}\}$
for all $h \in \mathcal{H}$ **do**
  $P_{t,h} \leftarrow$ DeXposure-FM.Forecast(G_t, X_t, h)
  $\widehat{G}_{t+h} \leftarrow \mathrm{BuildPredGraph}(P_{t,h}; \pi_{\min})$
  $\{\widetilde{G}^{(m)}_{t+h}\}_{m=1}^{M} \leftarrow \mathrm{MCSample}(P_{t,h})$
  Compute monitor statistics on $\widehat{G}_{t+h}$, attach rolling baselines, and derive alerts $\mathcal{A}_{t,h}$
  Compute alert confidences $C_t(\cdot)$ from sampled monitor dispersion
  for all $s \in \mathcal{S}_0$ **do**
    Compute deterministic loss $L(s; \widehat{G}_{t+h})$ and sampled $\mathrm{CVaR}_{0.5}(s, h)$
  end for
end for
$\mathcal{D}_t \leftarrow \mathrm{PlaybookDecision}(\mathcal{A}_t, \mathcal{R}_t, \mathrm{DH}_t, \mathrm{SAFE\_MODE}_t, \tau_{\mathrm{conf}})$
return $(\mathcal{A}_t, \mathcal{R}_t, \mathcal{D}_t)$

B Additional Implementation Details

B.1 Evaluation

For evaluation, we define regulator-aligned ground truth by absolute exposure loss. For protocol v and horizon h,

$$\Delta^h_t(v) = w_t(v) - w_{t+h}(v). \tag{2}$$

Given percentile $\pi \in (0, 1)$, the stressed set is

$$S^h_t = \mathrm{top}_\pi \left\{ v : w_t(v) > 0,\ \Delta^h_t(v) > 0 \right\}, \tag{3}$$

where protocols are ranked by $\Delta^h_t(v)$.

A supervisory system emits tickets

$$D_t = \{\tau_{t,1}, \ldots, \tau_{t,K_t}\}, \qquad \tau_{t,k} = (u_{t,k}, Y_{t,k}, r_{t,k}), \tag{4}$$

where $u_{t,k}$ is an action, $Y_{t,k} \subseteq V_t$ is a target set, and $r_{t,k}$ is a rationale. The action space is

$$\mathcal{U} = \{\text{MONITOR}, \text{INVESTIGATE}, \text{RECOMMENDREDUCE}, \text{CONTINGENCY}\}.$$

For predicted targets $\widehat{S}_t$, ticket quality is

$$\mathrm{Precision} = \frac{|\widehat{S}_t \cap S^h_t|}{|\widehat{S}_t|},$$

$$\mathrm{Recall} = \frac{|\widehat{S}_t \cap S^h_t|}{|S^h_t|},$$

$$\mathrm{F1} = \frac{2\ \mathrm{Precision}\ \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}.$$

Let $\widehat{S}_t^{\text{int}}$ be targets of intervention-level actions in $\{\text{RECOMMENDREDUCE}, \text{CONTINGENCY}\}$. The false-intervention rate is

$$\text{FIR} = \frac{|\widehat{S}_t^{\text{int}} \setminus S_t^h|}{|\widehat{S}_t^{\text{int}}|}. \tag{5}$$

Finally, the data-health score $\text{DH}_t \in [0,1]$ averages four data-quality checks (freshness, missingness, topology, discontinuity); its exact form is in Appendix A. For alert a at horizon h, the confidence $C_t(a) \in [0,1]$ combines data health, Monte Carlo dispersion, and forecast horizon, decreasing as data health falls or dispersion and horizon grow (exact form in Appendix A). With $\bar{C}_t$ the mean active-alert confidence, intervention-level actions are feasible only when $\text{DH}_t \geq \tau_{\text{data}}$ and $\bar{C}_t \geq \tau_{\text{conf}}$.

B.2 Forecaster architecture and training

DeXposure-FM is built on the GraphPFN graph-tabular foundation backbone (Hollmann et al., 2025; Eremeev et al., 2025) with a LiMiX-16M tabular transformer encoder (12 layers, 4 attention heads, hidden dimension 64). The same backbone serves three heads (edge existence, binary; edge weight, regression; and node-TVL change, regression), trained jointly with a BCE+MSE composite loss. We fine-tune one checkpoint per horizon $h \in \{1, 4, 8, 12\}$ for 20 epochs with AdamW (learning rate 10^{-4} for heads, 10^{-5} for the backbone). Fine-tuning draws on DeXposure snapshots from March 2020 to January 2025 (104 training weeks, 12 validation weeks for early stopping, 8 held-out weeks for internal evaluation); all data that influences weights or checkpoint selection precedes the frozen 2025 leaderboard split (§B.3). Full-graph in-context inference fits a single RTX 4090, and one weekly forecast across all four horizons completes in well under a minute. These intrinsic forecaster numbers are kept out of the main text because the paper's contribution is the operating point of the full pipeline, not the FM's headline accuracy.

B.3 Benchmark details

Six-axis schema. The six axes are non-overlapping (a system can excel at `b1` while failing `b5`), decomposable along the four-layer framework so that predictor (`b1`, `b3`, `b6`), monitor (`b2`), scenario (`b4`), and decision (`b5`) ablations report independently, and individually publishable so downstream work may extend a single axis without re-running the suite.

Dataset split.

Historical warning windows. The `b2_warning` event study is separate from the frozen 2025 leaderboard split. It evaluates the shared weighted-degree monitor around Terra/Luna (2022-05-09), FTX (2022-11-07), and SVB/USDC (2023-03-10) with a shorter 26-week rolling baseline, chosen so each event window has enough pre-event history. The 2025 leaderboard runs use the 42-week monitor baseline in Appendix A.

Ground-truth definition. With $w_t(v) = \sum_{e \ni v} w_t(e)$ the total weight incident on v, the regulator-aligned stressed set at horizon h is

$$\begin{aligned} \Delta_t^h(v) &= w_t(v) - w_{t+h}(v), \\ \mathcal{S}_t^h &= \text{top}_\pi\{v : w_t(v) > 0,\ \Delta_t^h(v) > 0\}, \end{aligned} \tag{6}$$

with $\pi = 0.05$ and ground-truth horizon $h = 4$ weeks. The final four test weeks lack the 4-week-ahead snapshot w_{t+4} and are not scored, leaving $n = 29$ evaluated weeks. This is the single ground-truth definition used by `b2`, `b5`, and the LLM-decision pipeline. A submission produces one JSON file per {benchmark, method} pair plus a single `results.json`; the harness fixes random seeds, snapshots library versions, and emits SHA-256 hashes of all checkpoint files.

B.4 Reference methods

These methods expose all four cuts of the four-layer ablation: adding the predictor (`m1`→`m5`), swapping the decision layer (`m5`→`m6`), adding the safety gate (`m6`→`m7`), and removing the predictor from the LLM stack (`m2`↛`m6`). All conform to a shared `predict_graph`/`decide_actions` interface.

LLM model panel and decoding. All LLM-bearing methods (`m2`, `m6`, `m7`) are evaluated under three decision-layer configurations: Claude Opus 4.7 (main), Claude Sonnet 4.6 (within-family, tier-below ablation), and Google Gemini 2.5 Pro (cross-family ablation). Each runs at temperature 0 and $\max\_tokens = 4096$; self-consistency is probed with three repeated calls per week and Jaccard overlap on emitted ticket targets and severities. The LLM-as-judge for the explanation-quality axis is a three-model panel chosen so the judge tier is at or above the decision tier: Claude Opus 4.8 (strictly tier-above Opus 4.7), Google Gemini 2.5 Pro, and OpenAI GPT-5.5; the Gemini and GPT judges run with OpenRouter's

Table B.1: Six-axis benchmark schema. Each axis is evaluated independently so systems can be compared at the predictor, monitor, scenario, and decision layers without re-running the full suite.

ID	Capability tested	Primary metrics
`b1_forecast`	Temporal graph prediction	PageRank/HHI/Gini MAE, trend consistency, Spearman; per $h \in \{1, 4, 8, 12\}$
`b2_warning`	Streaming anomaly detection	Precision, recall, F1, lead time, alert stability; per event $\times$ budget
`b3_calibration`	Predictive uncertainty	PI coverage at 0.90 target, PI width, ECE, CRPS
`b4_stress`	What-if scenario fidelity	Loss MAE, distressed-count MAE, propagation depth MAE, target overlap@k
`b5_decision`	Supervisory ticket quality	Precision, recall@k, F1, target stability, judge score, FIR
`b6_robustness`	Data quality sensitivity	Per-benchmark metrics under 5 degradation regimes; relative degradation

Table B.2: Dataset split. Temporal train/validation/test partition of the weekly exposure graphs and the role of each split.

Split	Period	Weeks	Usage
Train	2020-03–2024-06	$\sim$222	Forecaster training
Validation	2024-07–2024-12	$\sim$26	Conformal calibration, threshold tuning
Test	2025-01–2025-08	$\sim$33	Frozen leaderboard

$\text{reasoning} = \{\text{effort} : \text{minimal}\}$ so the judge thinking budget matches the non-reasoning Anthropic baseline. Claude Haiku 4.5 is reported as a weak-judge reference. All models are served through OpenRouter; observed cost is on the order of a few US dollars per full sweep of the 29-week test split.

C Additional Experimental Results

This appendix expands the headline numbers of §5.3 into the per-slice audits underlying them, then reports week-level uncertainty for the central comparisons.

C.1 Per-slice result tables

Full all-horizon `b1_forecast` results. Across all four horizons the persistence baseline (`m1`) stays best on PageRank and Gini MAE and on rank correlation, while the FM-grounded forecasters (`m4`/`m5`) lead on HHI MAE and trend consistency; the gaps are stable as h grows from 1 to 12.

Per-scenario `b4_stress` results. Across the five stress scenarios the FM variants (`m4`/`m5`) track the persistence baseline on loss MAE and target overlap@10, while EvolveGCN (`m3`) is the weakest on downstream what-if fidelity.

Per-regime `b6_robustness` results. At h=4 under five degradation regimes, EvolveGCN's small relative degradation reflects uniformly weak-but-stable forecasts (worst HHI and Gini MAE) rather than genuine robustness, while persistence and the FM variants remain accurate and are by construction insensitive to the two feature-perturbation regimes.

Per-budget `b2_warning` results (historical event study). For the three pre-test crisis windows the shared weighted-degree monitor (`h1`) achieves median lead times of 4–5 weeks, with precision 1.000 at SVB/USDC across all alert budgets $K \in \{5, 10, 20\}$. This is historical monitor evidence on pre-2025 events, not a test-split result.

Per-regime A1 (data-health gate) isolation. Isolating the data-health gate from the confidence gate (τ_{conf}=0 throughout), the gate-off setting lets FIR climb to 0.541 under degradation, while the *on/strict* settings suppress intervention tickets via safe-mode and drive FIR to 0.

Stress-condition ablation Panels B and C. Panel B: under severe feature/edge masking, FIR ranges over $[0.27, 0.60]$ when the data-health gate is disabled ($\tau_{\text{data}} = 0$; the confidence gate is held off at $\tau_{\text{conf}} = 0$ throughout Panel B to isolate A1), and collapses to 0.000 when the gate is triggered, by suppressing intervention tickets. Panel C: multi-horizon monitoring ($h \in \{1, 4, 8, 12\}$) generates 73–98 alerts in the three crisis windows vs 19–23 for single-horizon ($h = 4$), at unchanged precision range 0.533–0.624 and zero false interventions in both cases.

C.2 Qualitative ticket examples

We give two verbatim decision tickets (one representative consistency run each) behind the aggregate `b5_decision` scores. In both weeks the

Table B.3: Reference methods used in the benchmark and ablations. The methods cover predictor, monitor, decision-layer, and safety-gate variants.

ID	Predictor	Policy	Role
h1_weighted_degree	—	heuristic alert	Streaming anomaly baseline (b2)
m1_persistence_rules	$G_{t+h} = G_t$	rule engine	Decision baseline
m2_snapshot_llm	current snap	LLM	LLM-only ablation (no forecast)
m3_evolvegcn	EvolveGCN	—	Learned-GNN baseline
m4_fm_only	DeXposure-FM	—	FM-only ablation
m5_fm_rules	DeXposure-FM	rule engine	FM + rules
m6_fm_llm	DeXposure-FM	LLM	Full FM + LLM stack
m7_fm_llm_gated	DeXposure-FM	LLM + safety gate	Safety-gated FM + LLM

Table C.1: b1_forecast all-horizon forecasting on the 2025 test split. PageRank MAE shown in 10^{-5} units. Best per horizon and metric in **bold**; ties share the bold. Trend consistency for m1_persistence is structurally 0 because $\hat{G}_{t+h} = G_t$ produces no trend signal.

h	Method	PR MAE (↓)	HHI MAE (↓)	Gini MAE (↓)	Rank Corr (↑)	Trend Cons (↑)
1	m1_persistence	**3.6**	**0.0376**	**0.0027**	**0.556**	0.000
	m3_evolvegcn	3.7	0.0481	0.1149	0.535	0.444
	m4_fm_only	4.7	**0.0376**	0.0029	0.549	**0.531**
	m5_fm_rules	4.7	**0.0376**	0.0029	0.549	0.525
4	m1_persistence	**3.4**	0.0322	**0.0018**	**0.570**	0.000
	m3_evolvegcn	3.5	0.0489	0.1326	0.551	0.324
	m4_fm_only	4.5	**0.0321**	0.0022	0.559	0.621
	m5_fm_rules	4.5	**0.0321**	0.0022	0.558	**0.628**
8	m1_persistence	**3.9**	0.0399	**0.0028**	**0.517**	0.000
	m3_evolvegcn	4.0	0.0492	0.1140	0.499	0.296
	m4_fm_only	4.5	**0.0398**	0.0030	0.508	0.624
	m5_fm_rules	4.5	**0.0398**	0.0030	0.508	**0.632**
12	m1_persistence	**3.7**	0.0393	**0.0027**	**0.525**	0.000
	m3_evolvegcn	3.8	0.0534	0.1245	0.509	0.314
	m4_fm_only	4.1	**0.0392**	0.0029	0.516	**0.600**
	m5_fm_rules	4.1	**0.0392**	0.0029	0.516	**0.600**

FM-grounded LLM (m7) escalates while the raw-snapshot baseline (m2) does not; they differ in whether the escalation was correct.

Success: 2025-03-24 (FIR 0, target precision 0.83, Opus-4.8 judge m7= 4, m2= 2). On the same hub (node 2786, 9.35B exposure), m2 stays at *moderate* / INVESTIGATE (0.62) because, in its own words, "without forward-looking signals no protocol meets the 0.75 threshold for intervention." m7 reads the same hub as *elevated* / RECOMMEND-REDUCE (0.88): "largest predicted exposure (9.35B), dominant in S1 (single-failure cascade causes 18.67% system loss, 84 distressed, 282 affected) and largest contributor to S5 correlated stress." All three reduced hubs (2786, 2269, 118) fall inside the realized top-5% stress set.

Over-intervention: 2025-04-28 (FIR 1.0, target precision 0, Opus-4.8 judge m7= 3). Here m7 escalates the three largest hubs (2269/2786/1599) to RECOMMEND-REDUCE, citing an S1 top-protocol failure of 22.14% system loss. The grounding is real but the targets are wrong: none of the named hubs is in the realized top-5% stress set, which that week was dominated by long-tail protocols. The Opus-4.8 judge's own rationale names the failure mode: the report "cites specific network metrics ... and provides plausible cascade scenario reasoning, but none of its named targets match the ground-truth top-5% systemic risk set, indicating poor predictive accuracy despite well-grounded data references." Across the test split this decoupling between explanation quality and target correctness is only weak (Spearman between Opus-4.8 quality and $1 - \text{FIR}$ is 0.31), but it is the mechanism behind m7's aggregate FIR of 0.437, which persists across decision models (Table C.9).

C.3 Judge panel

The cross-decision-LLM panel (Table C.9) varies the decision model; here we report the cross-family judge panel that fixes the decisions and varies the

Table C.2: `b4_stress` per-scenario detail. Loss MAE in normalised units; Distress count MAE in protocols; target overlap@10 measured against the actual future graph. Best per scenario and metric in **bold**; ties share the bold, and columns where all four methods tie are left unbolded. The main pattern is conservative: FM variants roughly match the strong persistence baseline on loss and target-overlap metrics, while EvolveGCN is generally weaker on downstream what-if fidelity.

Scenario	Method	Loss MAE (↓)	Distress MAE (↓)	Prop. Depth MAE (↓)	Overlap@10 (↑)
S1 (single protocol)	`m1_persistence`	0.0668	**200.3**	**0.897**	**0.314**
	`m3_evolvegcn`	0.1072	218.1	0.966	0.034
	`m4_fm_only`	**0.0667**	201.4	**0.897**	**0.314**
	`m5_fm_rules`	**0.0667**	201.6	**0.897**	**0.314**
S2 (bridge cluster)	`m1_persistence`	0.0150	**27.3**	0.000	0.521
	`m3_evolvegcn`	**0.0116**	196.4	0.000	0.189
	`m4_fm_only`	0.0151	50.3	0.000	**0.525**
	`m5_fm_rules`	0.0151	52.1	0.000	**0.525**
S3 (stablecoin de-peg)	`m1_persistence`	**0.0079**	0.0	0.000	**0.445**
	`m3_evolvegcn`	0.0156	0.0	0.000	0.269
	`m4_fm_only`	0.0080	0.0	0.000	**0.445**
	`m5_fm_rules`	0.0080	0.0	0.000	**0.445**
S4 (sector lending)	`m1_persistence`	0.0100	0.0	0.034	**0.536**
	`m3_evolvegcn`	0.0200	0.0	0.034	0.292
	`m4_fm_only`	**0.0099**	0.0	0.034	**0.536**
	`m5_fm_rules`	**0.0099**	0.0	0.034	**0.536**
S5 (correlated top-10)	`m1_persistence`	0.0100	0.0	**0.000**	**0.602**
	`m3_evolvegcn`	0.0538	0.0	1.000	0.330
	`m4_fm_only`	**0.0099**	0.0	**0.000**	**0.602**
	`m5_fm_rules`	**0.0099**	0.0	**0.000**	**0.602**

Table C.3: `b6_robustness` per-regime detail. All metrics at $h{=}4$ on the 2025 test split under five degradation regimes. PageRank, HHI, and Gini MAE in 10^{-3} units. Relative degradation is signed against the clean `b1_forecast` baseline; negative values mean the predictor is no worse than on clean data. Best per metric (lowest MAE, highest rank correlation, smallest $|\Delta_{\text{rel}}|$) in **bold**; ties share the bold. Persistence ignores node features, so its `noisy_features_01` and `missing_features_20` rows degenerate to the clean numbers (and $\Delta_{\text{rel}} = 0$); EvolveGCN's predictions likewise change only negligibly there ($|\Delta_{\text{rel}}| < 10^{-8}$). Because both models' inputs are effectively unperturbed under these two feature regimes, we leave the Δ_{rel} column unbolded in these two regimes: a near-zero value reflects feature-insensitivity, not robustness. In the data-degradation regimes EvolveGCN attains the smallest $|\Delta_{\text{rel}}|$ yet the worst HHI and Gini MAE, so its low relative degradation reflects uniformly weak-but-stable forecasts rather than robustness, as Δ_{rel} rewards consistency irrespective of accuracy.

Regime	Method	PR MAE (↓)	HHI MAE (↓)	Gini MAE (↓)	Rank Corr (↑)	Δ_{rel} (~0)
low_data_10pct	`m1_persistence`	15.94	0.0591	**0.0023**	**0.552**	+0.766
	`m3_evolvegcn`	15.23	0.0999	0.1301	0.535	**+0.276**
	`m4_fm_only`	12.83	**0.0586**	0.0027	0.528	+0.612
	`m5_fm_rules`	**12.71**	0.0587	0.0027	0.528	+0.607
low_data_25pct	`m1_persistence`	**9.46**	0.0282	**0.0020**	**0.577**	−0.093
	`m3_evolvegcn`	10.22	0.0486	0.1281	0.558	**−0.027**
	`m4_fm_only`	9.86	**0.0279**	0.0024	0.553	−0.126
	`m5_fm_rules`	9.95	**0.0279**	0.0024	0.553	−0.126
partial_graph_30	`m1_persistence`	**9.63**	**0.0350**	**0.0019**	**0.554**	+0.068
	`m3_evolvegcn`	10.36	0.0488	0.1236	0.537	**−0.048**
	`m4_fm_only`	12.52	**0.0350**	0.0022	0.500	+0.082
	`m5_fm_rules`	12.45	**0.0350**	0.0022	0.499	+0.077
noisy_features_01	`m1_persistence`	**9.77**	**0.0322**	**0.0018**	**0.584**	0.000
	`m3_evolvegcn`	10.68	0.0489	0.1326	0.563	0.000
	`m4_fm_only`	11.38	0.0323	0.0021	0.554	−0.005
	`m5_fm_rules`	11.35	0.0323	0.0021	0.554	−0.007
missing_features_20	`m1_persistence`	**9.77**	**0.0322**	**0.0018**	**0.584**	0.000
	`m3_evolvegcn`	10.68	0.0489	0.1326	0.563	0.000
	`m4_fm_only`	12.11	0.0323	0.0023	0.559	+0.017
	`m5_fm_rules`	12.08	0.0323	0.0023	0.559	+0.014

Table C.4: `b2_warning` per-event detail for the shared weighted-degree monitor (`h1_weighted_degree`) across alert budgets $K \in \{5, 10, 20\}$. Recall and F1-warning shown in 10^{-3} units (the full DeFi corpus contains ∼4,300 protocols, so per-week recall denominators are large). Lead time is event-level and not budget-sensitive: 5 weeks for Terra/Luna and FTX, 4 weeks for SVB-USDC. `b2_warning` sits upstream of the predictor choice, so a single monitor is shared across all methods (see §5.3).

Event	K	Prec. (↑)	Rec. $\times 10^3$ (↑)	F1_w $\times 10^3$ (↑)	Stab. (↑)	$\lvert\mathcal{A}\rvert$
Terra/Luna	5	0.60	1.81	3.61	0.44	11
	10	0.70	3.26	6.49	0.62	19
	20	0.65	7.97	15.74	0.74	36
FTX	5	0.60	1.33	2.65	0.64	10
	10	0.70	2.39	4.77	0.72	18
	20	0.55	4.79	9.49	0.72	39
SVB-USDC	5	1.00	1.81	3.62	0.55	10
	10	1.00	2.82	5.62	0.75	15
	20	1.00	5.64	11.22	0.81	30

Table C.5: Per-regime data-health gate ablation isolating A1 from A2. Each row holds τ_{conf}=0 and sweeps τ_{data}. "*strict*" = τ_{data}=0.85, "*on*" = 0.70, "*off*" = 0.00. The four regimes apply targeted edge or feature degradation on the clean 2025 test split (`n_weeks` = 29): `extreme` masks both topology and features at 90%, `topo` removes 90% of edges only, `feat` corrupts 90% of node features only, `severe` masks both at 80%. Numbers expand Panel B of Table C.6: under degradation the gate-off setting produces FIR up to 0.541, while *on*/*strict* can suppress intervention tickets via safe-mode and drive FIR to 0.

Regime	Gate	τ_{data}	$\overline{\text{dh}}$	Safe-mode (%)	Ticket Prec. (↑)	FIR (↓)	Interventions
extreme	strict	0.85	0.587	100	0.510	**0.000**	0
	on	0.70	0.587	100	0.545	**0.000**	0
	off	0.00	0.587	0	0.490	0.510	57
extreme_topo	strict	0.85	0.637	100	0.517	**0.000**	0
	on	0.70	0.637	100	0.462	**0.000**	0
	off	0.00	0.637	0	0.559	0.441	58
extreme_feat	strict	0.85	0.755	100	0.503	**0.000**	0
	on	0.70	0.755	0	0.510	0.267	5
	off	0.00	0.755	0	0.503	0.267	5
severe	strict	0.85	0.751	100	0.490	**0.000**	0
	on	0.70	0.751	0	0.538	0.400	25
	off	0.00	0.751	0	0.538	0.541	24

Table C.6: Component-level ablation of DeXposure-Claw on `b5_decision`. Panel A shows clean 2025 test-split ablations against the `m5_fm_rules` baseline; Panels B–C report targeted stress runs for mechanisms that are dormant on clean data. Rows follow pipeline order (Layer 2 evidence then Layer 4 gates). A3 (scenario engine) and A2 (confidence gating) are load-bearing in the clean setting, while A6 (multi-horizon monitoring) and A1 (data-health gating) are safety-reserve mechanisms whose effects appear under crisis-window dynamics and data degradation, respectively.

Panel	Ablation	Configuration	Ticket Prec.	FIR	Notes
A: clean test	—	full `m5_fm_rules`	0.600	0.000	baseline
	A6	single-horizon forecasting (h=4)	0.600	0.000	no effect (clean data)
	A3	skip scenario engine	0.000	0.000	target extraction collapses
	A1	disable data-health gate (τ_{data}=0)	0.600	0.000	no effect (clean data)
	A2	disable confidence gate (τ_{conf}=0)	0.600	0.429	FIR shoots up
B: degraded data	A1 off	80–98% feature/edge mask, τ_{conf}=0	—	0.27–0.60	false interventions emerge
	A1 on	same degradation, safe-mode triggered	—	0.000	intervention suppression
C: crisis period	A6: single	horizons = {4}, crisis windows	0.533–0.624	0.000	alerts: 19–23
	A6: multi	horizons = {1, 4, 8, 12}, same windows	0.533–0.624	0.000	alerts: 73–98 (≈ 4×)

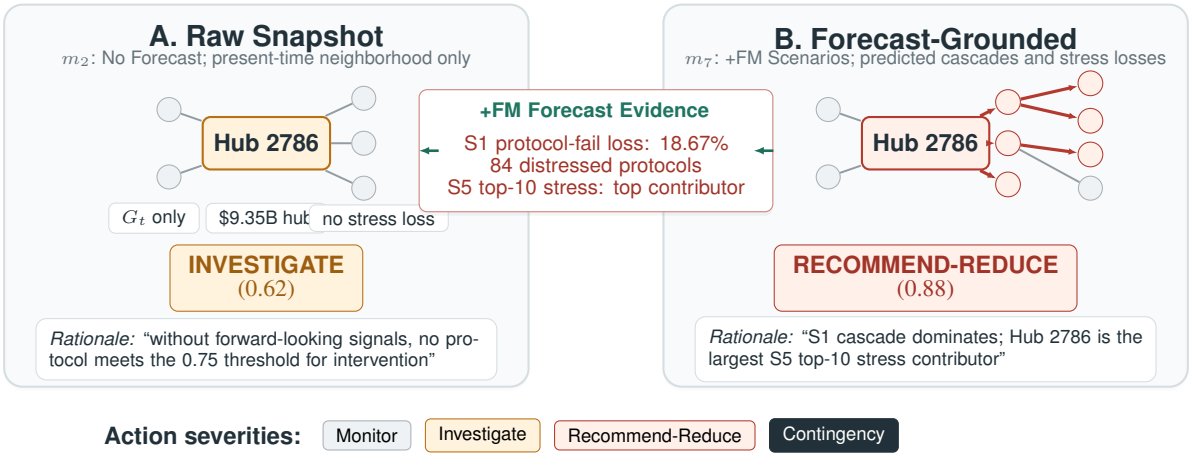

Figure C.1: Hub 2786 case (week 2025-03-24). The expanded evidence names S1 protocol failure and S5 top-10 stress; forecast grounding moves the action from INVESTIGATE to RECOMMEND-REDUCE.

Table C.7: Cross-family judge panel on the same Opus 4.7 decisions (n = 29). Gemini and GPT use reasoning = {effort : minimal} so the thinking budget matches the Anthropic baseline. † Haiku 4.5 is a tier-below weak-judge reference.

Method	Opus 4.8	Gemini 2.5 Pro	GPT-5.5	Haiku 4.5†
`m2`	2.24	2.69	1.90	2.03
`m6`	2.41	2.55	2.45	2.48
`m7`	**2.45**	**2.90**	2.45	2.69

judge.

Complete headline metrics. Table C.8 reports the full `b5_decision` comparison, restoring the grounding and target-stability columns dropped from the compact body panel (Table 2). Grounding is 1.000 for every LLM-bearing system, so the over-intervention captured by FIR is not a citation-fabrication artifact. Target stability (Jaccard overlap of ticket targets and severities across three temperature-0 calls) is lowest for the FM+rules stack (`m5`, 0.257); the FM-grounded LLM variants (`m6`/`m7`) stay within the range of the snapshot baselines.

C.4 Statistical significance and matched-budget analysis

All headline metrics are means over the n = 29 scored test weeks. We report 95% percentile bootstrap confidence intervals (10,000

Table C.8: Complete `b5_decision` comparison on the frozen 2025 test split ($n = 29$); the full version of Table 2. Grnd. is grounding (judge-confirmed evidence citation) and Stab. is target stability. Dashes are undefined: `m1`/`m5` emit no LLM-judged tickets, and `m1`/`m2`/`m5` emit no intervention-level ticket that could misfire.

ID	Pred.	Decision	Prec.↑	Rec.↑	F1↑	Judge↑	FIR↓	Grnd.↑	Stab.↑
`m1`	persist.	rules	**0.720**	0.004	0.0076	—	—	—	0.514
`m2`	—	LLM (no FM)	0.575	0.009	0.0184	2.24	—	1.000	0.532
`m5`	DeXposure-FM	rules	0.600	0.010	0.0190	—	—	—	0.257
`m6`	DeXposure-FM	LLM	0.570	0.012	**0.0241**	2.41	0.448	1.000	0.488
`m7`	DeXposure-FM	LLM + safety gate	0.580	0.012	0.0234	**2.45**	0.437	1.000	0.435

Table C.9: 95% bootstrap CIs for the decision-quality headline numbers (decision LLM in parentheses; judge is Claude Opus 4.8 throughout). ‡`m1`'s per-week distribution comes from a local re-run of the released harness: its aggregate F1 (0.0081) brackets the published GPU-run value (0.0076), and the recall@k and stressed-pool size reproduce exactly, but with <1 ticket per week a few borderline tickets flip across environments (re-run precision 0.633 vs published 0.720), so the `m1` interval should be read as order-of-magnitude.

Configuration	F1 [95% CI]	FIR [95% CI]	Judge [95% CI]
`m1` rules‡	0.0081 [0.0041, 0.0123]	0.000	—
`m2` (Opus 4.7)	0.0184 [0.0142, 0.0223]	0.000	2.24 [2.10, 2.41]
`m6` (Opus 4.7)	0.0241 [0.0201, 0.0281]	0.448 [0.293, 0.601]	2.41 [2.21, 2.62]
`m7` (Opus 4.7)	0.0234 [0.0194, 0.0271]	0.437 [0.282, 0.592]	2.45 [2.24, 2.66]
`m6` (Sonnet 4.6)	0.0276 [0.0225, 0.0324]	0.379 [0.236, 0.534]	2.52 [2.28, 2.79]
`m7` (Sonnet 4.6)	0.0288 [0.0236, 0.0335]	0.374 [0.224, 0.529]	2.66 [2.41, 2.90]
`m6` (Gemini 2.5 Pro)	0.0129 [0.0103, 0.0155]	0.155 [0.034, 0.293]	2.21
`m7` (Gemini 2.5 Pro)	0.0139 [0.0111, 0.0167]	0.190 [0.069, 0.328]	2.38

Table C.10: Two-sided paired permutation p-values for the key comparisons. Significant effects ($p < 0.05$) in **bold**.

Comparison	Metric	p
`m6` vs `m2` (FM signal)	F1	**0.0002**
`m7` vs `m1` (full stack)	F1	**<0.0001**
`m7` vs `m6` (gate)	F1	0.41
`m7`×Sonnet vs `m7`×Opus	F1	**0.0004**
`m6` vs `m2`	FIR	**<0.0001**
`m7`×Sonnet vs `m7`×Opus	FIR	0.22
`m6`×Sonnet vs `m6`×Opus	FIR	0.12
`m6` vs `m2` (Opus 4.8 judge)	Judge	0.23
`m6` vs `m2` (GPT-5.5 judge)	Judge	**0.0002**
`m6` vs `m2` (Gemini judge)	Judge	0.77
`m7` vs `m6` (Opus 4.8 judge)	Judge	1.00
`m7`×Sonnet vs `m7`×Opus	Judge	0.21

resamples of weeks with replacement; F1 recomputed from the resampled mean precision and recall) and two-sided paired sign-flip permutation tests (20,000 permutations, methods aligned by week). Analysis code is released with the harness (`scripts/bootstrap_stats.py`, `scripts/matched_budget.py`).

The load-bearing quantitative claims (the FM signal's F1 lift, the full stack's F1 lift over rules, the Sonnet 4.6 F1 improvement, and the over-intervention cost) are significant at $p < 0.001$. The explanation-quality lift of the FM-grounded variants is judge-dependent (significant under GPT-5.5, positive but not significant under Opus 4.8 at $p = 0.23$, absent under the noisiest Gemini judge), so we read the judge axis as directional evidence. The safety gate's effects on F1 and judge score are indistinguishable from noise on clean data, consistent with its characterisation as a reserve mechanism (§6).

Matched-budget recall@k. Methods emit different numbers of ticket targets per week (`m1` 2.1, `m2` 4.9, `m7` 6.3, `m6` 6.5, `m7`×Sonnet 7.6), and unbudgeted recall mechanically favours methods that flag more protocols. To separate target *quality* from target *volume*, we truncate every method to its top-k targets per week (LLM variants ranked by their own emitted risk score; `m1` by ticket confidence) and score against the same stressed pool.

The matched-budget view decomposes the headline F1 result. Against the rules baseline, LLM targeting is genuinely better per target. At every k precision@k roughly doubles (0.52–0.59 vs 0.24–0.28) and recall@k is significantly higher ($p \leq 0.006$). Against the raw-snapshot LLM, the FM's contribution is *not* a higher per-target hit rate. Through the head the two are statistically indistinguishable (all pairwise $p \geq 0.28$ at $k \leq 5$), and `m2` even leads slightly at $k = 3$. The divergence

Table C.11: Matched-budget recall@k ($\times 10^3$) and precision@k on the 2025 test split. Every LLM variant beats the rules baseline at all k ($p \leq 0.006$ vs m1). The FM-grounded and raw-snapshot variants are indistinguishable through the head ($k \leq 5$, all pairwise $p \geq 0.28$); they diverge only once m2 exhausts its targets, where the FM variants recover more stressed protocols at the same precision ($k = 7$, $p \leq 0.0003$). †saturated, no week reaches depth k, so the cell repeats the method's full-set value. m1 row from the local re-run (Table C.9 note).

	$k=1$		$k=3$		$k=5$		$k=7$	
Method	R	P	R	P	R	P	R	P
m1 rules	0.71	0.24	2.45	0.26	4.05	0.26	4.05†	0.26
m2 (Opus 4.7)	1.53	0.52	5.70	0.59	9.08	0.58	9.35†	0.58
m6 (Opus 4.7)	1.64	0.55	5.30	0.56	9.63	0.59	**12.14**	0.57
m7 (Opus 4.7)	1.64	0.55	5.30	0.56	9.13	0.57	**11.95**	0.58
m7 (Sonnet 4.6)	1.64	0.55	5.30	0.56	9.70	0.59	**13.26**	0.58

appears only past m2's target ceiling. The raw-snapshot model exhausts its targets at about five per week (no week reaches $k = 7$), so its recall@k saturates at 9.35×10^{-3}, whereas the FM variants keep emitting targets and at $k = 7$ recover significantly more stressed protocols (12.14 vs 9.35, $p = 0.0001$) at undiminished precision. These extra targets are genuine detections rather than padding: the marginal precision of the rank-6-to-8 tail that only the FM variants populate is 0.535 for m6 and 0.622 for m7, comparable to their head precision (~ 0.56). This larger target set at flat precision is the FM's contribution to the headline recall and F1 lift (Table 2), and the expanded tail is also where over-intervention concentrates. It does so across decision models, escalated to intervention-level severity at a comparable rate under Opus 4.7 (FIR = 0.448) and Sonnet 4.6 (FIR = 0.379).

D Prompt Templates

The released audit logs contain the fully instantiated prompts and model outputs for each evaluated week. For reproducibility, we also report the templates used by the decision and judge calls below. Placeholders in braces are filled deterministically from the evidence bundle; line wrapping is for typesetting only. The prompts expose protocol-level aggregate summaries and derived evidence, not raw transaction logs, wallet-level records, full edge lists, or node-feature tables.

Shared decision system prompt.

```
You are a DeFi systemic risk analyst for a
    regulatory supervisory body.
Your job: given network data, identify
    protocols at elevated risk of
distress over the next {horizon} weeks and
    recommend supervisory actions.

You MUST respond with valid JSON only.

Schema:
{
  "risk_level": "low" | "moderate" | "elevated
    " | "critical",
  "target_protocols": [
    {
      "protocol": "<name>",
      "risk_score": <float 0-1>,
      "action": "Monitor" | "Investigate" |
                "Recommend-Reduce" | "
    Contingency",
      "reason": "<specific explanation citing
    input data>"
    }
  ],
  "rationale": "<2-3 sentence assessment
    citing metrics and numbers>"
}

Rules:
- Only flag protocols you genuinely believe
    are at elevated risk.
- Cite specific numbers from the provided data.

- Use exact protocol names as they appear in
    the data.
- Order target_protocols by risk_score
    descending.
- Recommend-Reduce requires elevated/critical
    overall risk and
  protocol risk_score >= 0.75.
- Contingency requires critical overall risk
    and risk_score >= 0.90.
- If the action constraints are not met, use
    Investigate.
```

Forecast-grounded user prompt (m6/m7).

```
Current DeFi network analysis ({date}),
    forecast horizon = {horizon} weeks.

== FM MODEL PREDICTIONS (predicted graph G_{t
    +{horizon}}) ==
Nodes: {n_nodes} protocols, Edges: {n_edges}
    weighted exposure links

Top-10 protocols by predicted exposure weight:
{top_protocols}

== PREDICTED NETWORK RISK METRICS ==
{metrics}
```

```
== STRESS SCENARIO ANALYSIS (applied to
    predicted graph) ==
{scenarios}

Based on these FM model predictions, metrics,
    and stress analysis, identify
which protocols are at elevated risk over the
    next {horizon} weeks.
```

Snapshot-only user prompt (`m2`).

```
Current DeFi network state ({date}), forecast
    horizon = {horizon} weeks.

NOTE: You do NOT have access to any predictive
     model. You must reason from
the current network snapshot only. There are
    no forward-looking predictions.

== CURRENT NETWORK SNAPSHOT ==
Protocols: {n_nodes} total
Edges: {n_edges} weighted exposure links
Total network weight: {total_weight}

Top-10 protocols by current exposure weight:
{top_protocols}

Category breakdown:
{category_summary}

== CURRENT NETWORK METRICS ==
{metrics}

Based on this current network state, identify
    which protocols are at
elevated risk over the next {horizon} weeks.
```

LLM-as-judge prompt.

The judge prompt uses only aggregate evaluation context: week, horizon, network-scale counts, aggregate metric summaries, the top-loss ground-truth labels used for scoring, and the report being judged. The fields `n_nodes` and `n_edges` are graph-size counts, not node or edge records.

```
SYSTEM:
You are an expert evaluator of DeFi risk
    analysis reports. You will compare
two risk assessments for the same week and
    rate the quality of the SECOND
one (Report B) on a scale of 1-5.

Respond with JSON only:
{
  "quality_score": <int 1-5>,
  "reasoning": "<1-2 sentence justification>"
}

USER:
Week: {date}, horizon: {horizon} weeks

== INPUT DATA SUMMARY ==
Network: {n_nodes} nodes, {n_edges} edges
Key metrics: {metrics_summary}

== GROUND TRUTH ==
{n_stressed} protocols are in the top-5% set
    ranked by absolute weight loss
over the horizon: {stressed_list}

== REPORT TO EVALUATE ==
Risk level: {risk_level}
Targets: {llm_targets}
Rationale: {rationale}

Rate the report's quality (1-5).
```