TimeClaimBench: A Claim-Level Benchmark for Grounded LLM Explanations of Time Series

Anonymous ACL submission

Abstract

Large language models (LLMs) are increasingly used to explain numerical time series, but fluent explanations can misstate when a change occurs, overstate a trend, or hallucinate temporal patterns. Existing evaluations often score forecasts, answers, or whole explanations, leaving claim-level temporal grounding undermeasured. We introduce TIMECLAIMBENCH, a benchmark for evaluating whether individual claims in LLM-generated time-series explanations are supported by the observed signal. TIMECLAIMBENCH combines controlled synthetic time series with a Real-NAB localization diagnostic, parses explanations into atomic temporal claims, and verifies them against an extracted event graph. Under the fixed verifier, many free-form claims lack extracted-event support, and chain-of-thought increases claim volume without improving automatic support. In contrast, event-grounded prompting consistently improves automatic claim faithfulness, raising the supported-claim rate by 18.3 points and reducing verifier-unsupported claims by 27.9 points. Human audits further show that the task is annotatable and that the automatic verifier is human-calibrated rather than a substitute for human judgment. TIMECLAIMBENCH makes temporal grounding a first-class evaluation target for LLM-based time-series analysis. [1]

1 Introduction

Automatically describing time series in natural language is useful, but early time-series captioning work already showed a core risk: fluent descriptions can be factually incorrect with respect to the underlying signal (Jhamtani and Berg-Kirkpatrick, 2021). This risk becomes more consequential as large language models (LLMs) are increasingly used as natural-language interfaces for time-series analysis. Such interfaces make time-series analysis more accessible, but they also make it easier for users to trust fluent explanations that misstate when a change occurs, overstate the strength of a trend, hallucinate seasonality, or introduce an unsupported causal interpretation. Figure 1 illustrates why these failures need claim-level evaluation rather than only whole-explanation scoring.

Existing work has moved time-series language evaluation closer to natural-language explanations. Truth-conditional captioning and recent time-series captioning work study how to describe salient temporal patterns (Jhamtani and Berg-Kirkpatrick, 2021; Trabelsi et al., 2025); feature-understanding and exam-style benchmarks test whether models recognize trends, noise, anomalies, causality, or other time-series properties (Fons et al., 2024; Cai et al., 2024); BEDTime and CaTS-Bench evaluate time-series description and captioning abilities at broader scale (Sen et al., 2026; Zhou et al., 2026); and LLM-as-a-judge work studies reference-free scoring of candidate time-series explanations (Zheng et al., 2023; Sivalingam et al., 2026). Together, these efforts reveal an evaluation mismatch: users read multi-claim explanations, while most reported scores collapse each response into a single caption, answer, or judge score.

The gap is therefore not that time-series captioning, understanding, or explanation evaluation is unexplored. The central missing unit is **claim-level temporal grounding**. A single explanation can contain both a correct claim and a wrong one: for example, it may correctly identify a sharp rise, but place the subsequent decline in the wrong interval, overstate a weak trend, or add a seasonal pattern that is absent. Reference-based text metrics and explanation-level judgments can hide these mixed cases. NLP work on faithfulness, attribution, and fact verification provides useful precedents for checking whether generated statements are supported by evidence (Jacovi and Goldberg, 2020;

[1] https://anonymous.4open.science/r/TimeClaimBench-36EC

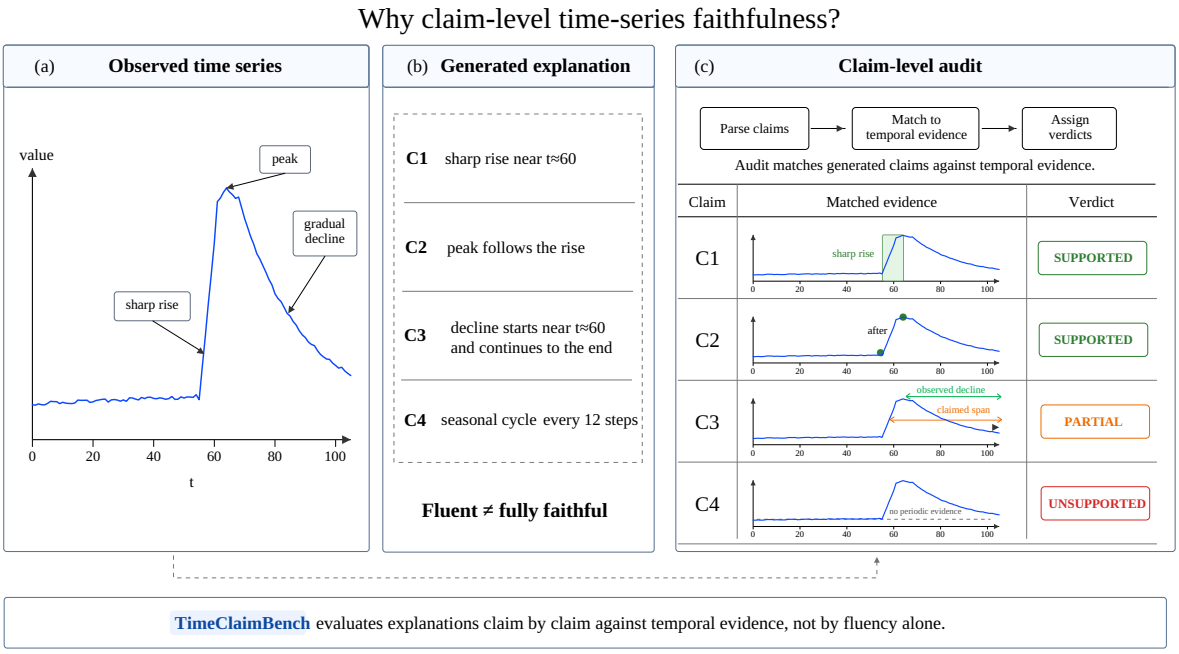

Figure 1: Motivation for claim-level time-series faithfulness evaluation. A fluent LLM-generated explanation may contain multiple temporal claims, some of which are supported by the observed sequence, while others are only partially supported or unsupported. TIMECLAIMBENCH parses the explanation into atomic claims, matches each claim to temporal evidence, and assigns claim-level verdicts.

Rashkin et al., 2023; Chen et al., 2020; Kamoi et al., 2023; Laban et al., 2022). Yet the evidence substrate is different: document and table verification can search for supporting sentences or cells, whereas time-series verification must first induce events from numerical values, intervals, slopes, changes, periodicity, and outliers.

We introduce TIMECLAIMBENCH, a claim-level benchmark for evaluating whether LLM-generated time-series explanations are grounded in the observed signal. Instead of scoring an explanation only as a whole, TIMECLAIMBENCH decomposes it into temporal claims and checks each claim against numerical evidence, exposing mixed cases where a response is partly correct but mislocalized, overstated, or hallucinated. TIMECLAIMBENCH combines a controlled synthetic suite covering eight temporal phenomena with a Real-NAB localization diagnostic for real anomaly-window behavior (Lavin and Ahmad, 2015), together with a fixed rule-based automatic verifier and human-audit calibration. Across five LLMs and four prompting conditions, verifier-unsupported temporal claims are common; event-grounded prompting improves automatic claim faithfulness by 18.3 points and reduces verifier-unsupported claims by 27.9 points. These results show that time-series explanation quality should be evaluated at the level of individual temporal claims, not only at the level of fluent paragraphs.

Our contributions are:

- We formulate claim-level temporal faithfulness for LLM-generated time-series explanations, distinguishing SUPPORTED, PARTIAL, and UNSUPPORTED claims with evidence intervals and error types.
- We construct TIMECLAIMBENCH, a benchmark with eight controlled temporal phenomena and a Real-NAB localization diagnostic for real anomaly-window behavior.
- We provide an executable non-LLM automatic verification protocol and a multi-model evaluation showing that verifier-unsupported temporal claims are common, chain-of-thought can amplify them, and event-grounded prompting substantially improves automatic claim faithfulness.

2 Related Work

Time-series language benchmarks and explanation evaluation. Recent work adapts language-model architectures and pretrained LLMs to time-

series inputs through prompting, reprogramming, embedding alignment, decoder adaptation, or foundation sequence models (Gruver et al., 2023; Jin et al., 2024; Cao et al., 2024; Sun et al., 2024; Pan et al., 2024; Liu et al., 2024; Zhou et al., 2023; Goswami et al., 2024; Das et al., 2024; Ansari et al., 2024). Closer to our setting, time-series captioning, description, QA, and chart-to-text work evaluate whether models describe or reason over temporal and numerical patterns (Jhamtani and Berg-Kirkpatrick, 2021; Kantharaj et al., 2022; Masry et al., 2022; Tang et al., 2023; Fons et al., 2024; Cai et al., 2024; Trabelsi et al., 2025; Sen et al., 2026; Zhou et al., 2026; Chen et al., 2025; Xie et al., 2025; Kong et al., 2025). LLM-as-a-judge methods further provide reference-free or preference-style evaluation for generated responses, including time-series explanations (Zheng et al., 2023; Liu et al., 2023; Kim et al., 2024; Sivalingam et al., 2026). These lines motivate language-based evaluation for time series, but they mainly score forecasts, answers, captions, or holistic rubrics rather than individual temporal claims inside explanations.

Fine-grained factuality and time-series verification. Fine-grained factuality and attribution work provides the evaluation unit most aligned with our goal: generated text is decomposed into smaller claims and checked against evidence (Jacovi and Goldberg, 2020; Atanasova et al., 2023; Rashkin et al., 2023; Laban et al., 2022; Min et al., 2023; Wei et al., 2024). Fact-checking benchmarks and revision or alignment pipelines extend this idea to documents, tables, and mixed evidence (Chen et al., 2020; Aly et al., 2021; Kamoi et al., 2023; Gao et al., 2023; Zha et al., 2023). Most closely, Strong and Vlachos (2025) introduce TSVer for verifying externally sourced factual claims against time-series evidence, whereas TIMECLAIMBENCH audits claims produced inside LLM-generated time-series explanations. The task is therefore not standalone fact-checking, but checking whether generated explanatory claims are grounded in temporal events, directions, magnitudes, and intervals induced from the observed sequence.

3 The TIMECLAIMBENCH Benchmark

3.1 Overview

Figure 2 gives the end-to-end workflow. The key design choice is to separate *evidence construction* from *language generation*: the same extracted events are used to condition event-grounded prompts and to verify claims after generation. This lets TIMECLAIMBENCH evaluate whether LLMs produce auditable temporal explanations grounded in numerical evidence, rather than merely fluent descriptions. Event-grounded prompting therefore tests evidence use, not a no-evidence fair baseline. Because the same operational extracted event graph is also used by the verifier, Section 5 includes mismatched-event, gold-event, and human-audit checks to characterize this coupling.

3.2 Design Goals

TIMECLAIMBENCH targets three design goals. First, it evaluates individual temporal claims because a single explanation can mix a correct trend description with a shifted interval, overstated magnitude, or unsupported pattern. Second, it pairs controlled synthetic series with real anomaly windows: synthetic examples provide known event evidence, while the Real-NAB localization diagnostic tests behavior under real noise. Third, it keeps evaluation executable but calibrated. The automatic verifier enables scalable prompt and model comparisons, and verifier ablations plus blind human audits expose the boundary of those automatic scores.

3.3 Benchmark Components

Time-series inputs. The benchmark contains a controlled synthetic split and a Real-NAB localization split. The synthetic split is the main benchmark and covers eight temporal phenomena: upward trends, downward trends, sharp rises, sharp drops, volatility shifts, change points, seasonality, and anomalies. Because these examples are generated with known event types, directions, intervals, and reference claims, they support auditable claim-level evaluation. The Real-NAB localization split is based on the Numenta Anomaly Benchmark (Lavin and Ahmad, 2015) and tests anomaly-window behavior under real noise. Detailed split sizes and construction choices are reported in Appendix A.1.

Evidence construction. Each time series is converted by a deterministic rule-based event extractor into an extracted event graph. The extractor scans the normalized sequence with fixed temporal operators for trend, sharp change, volatility shift, change point, seasonality, anomaly, peak, and trough evidence; Appendix A.2 gives the exact settings. Each event node stores its type, optional direction, interval, score, and numeric attributes such as slope,

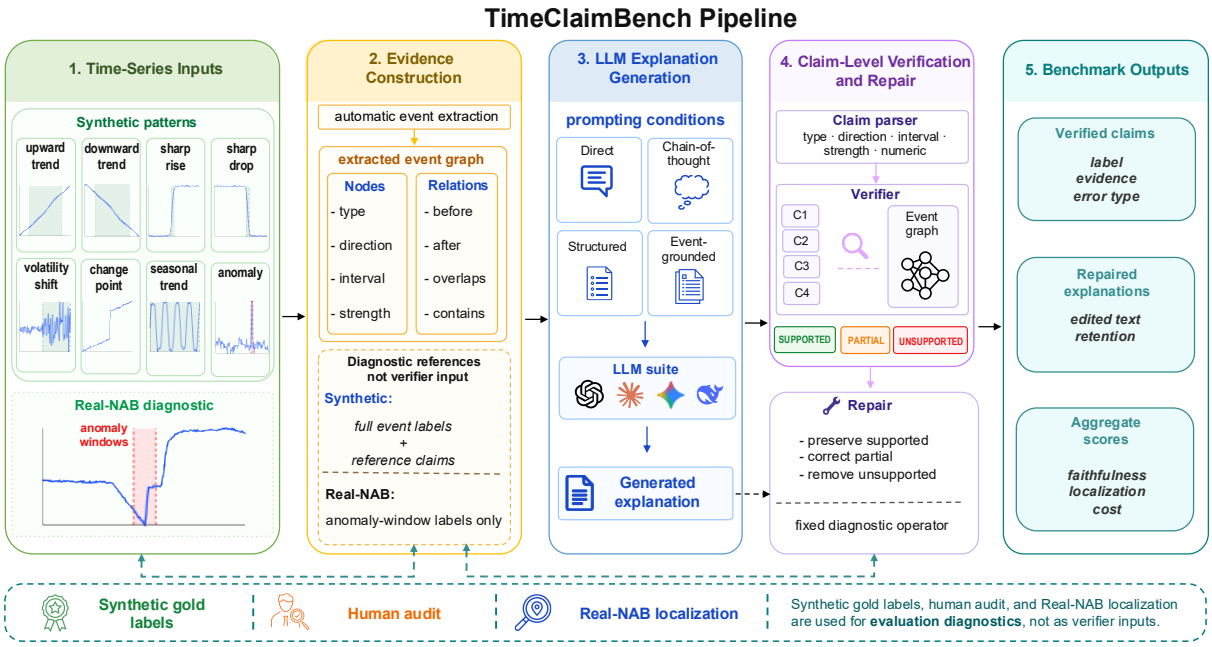

Figure 2: TIMECLAIMBENCH pipeline. Each input series is converted into an extracted event graph; LLM explanations are parsed into atomic claims, verified against the extracted event graph, optionally passed through verifier-guided repair, and summarized as claim-level outputs and aggregate metrics.

magnitude, period, or z-score. Relation edges encode temporal links such as BEFORE, AFTER, OVERLAPS, and CONTAINS. The default verifier uses automatically extracted events; synthetic gold events and NAB anomaly windows are reserved for diagnostics and oracle analyses.

LLM explanation generation. TIMECLAIMBENCH stores model explanations produced under alternative prompting conditions. Direct, chain-of-thought, and structured prompting expose only the raw series and question, while event-grounded prompting additionally exposes the extracted event list. The downstream evaluation protocol is specified in Section 4.

Claim-level verification and verifier-guided repair. The claim parser decomposes each explanation into atomic temporal claims and records attributes such as event type, direction, interval, strength, and numeric values when present. The automatic verifier retrieves compatible extracted events and checks claim–event agreement using type, direction, interval overlap, strength, and numeric consistency. Full support-label boundary cases, parser details, scoring rules, thresholds, and verifier-guided repair procedures are provided in Appendices A.3–A.5. The verifier-guided repair step preserves supported claims, corrects partially supported claims, and removes verifier-unsupported claims.

Outputs. The output layer returns the claim labels, matched evidence, repaired explanations, and token-cost records needed for evaluation. Section 4.2 explains how these outputs are aggregated in the main experiments, while Appendix A.6 provides the formal metric definitions.

4 Evaluation Protocol

4.1 Task Settings

We evaluate five LLM backbones under four prompting conditions: direct, chain-of-thought, structured, and event-grounded prompting. Each run uses a model-ready input containing the univariate series and question; the event-grounded prompting condition additionally receives an extracted event list. All runs use the same input order, generation settings, claim parser, automatic verifier, verifier-guided repair module, and cost-accounting pipeline.

The main synthetic setting uses the 400-example test split, with 50 examples from each of the eight synthetic temporal phenomena. We also include a Real-NAB localization diagnostic with 100 anomaly-centered windows and 100 normal windows.

4.2 Scoring and Statistical Reporting

For reporting the main synthetic results, we select three headline summaries from the benchmark outputs: automatic claim faithfulness, verifier-unsupported rate, and claims per explanation. These separate explicit support from the extracted event graph, claims unsupported under the extracted event graph, and response verbosity. Verifier-guided repair analyses additionally report retention and supported-claim preservation, while diagnostic analyses report partial rate, interval localization, coverage, cost, and Real-NAB localization when relevant. In result tables, arrows indicate the preferred direction of each metric. We use *Faith.* for automatic claim faithfulness, *Unsup.* for verifier-unsupported rate, *Claims/Expl.* for claims per explanation, and *Ret.* for claim retention after verifier-guided repair. Metrics marked with *+Rep.* are computed after verifier-guided repair; table-specific abbreviations are defined in the corresponding captions.

We aggregate prompt-level tables in two stages: first computing claim-level rates within each model–prompt run, then averaging over the five model runs. For uncertainty estimates, we use 5,000 bootstrap resamples of the 400 synthetic test series within each fixed model run; paired prompt comparisons preserve shared model–series identifiers. Appendix A.6 gives the formal metric definitions and dataset-specific diagnostics, and Appendix B provides the task notation, prompt templates, and generation settings.

4.3 Human Audit for Verifier Calibration

We include blind human audits to calibrate automatic verifier labels. For the synthetic benchmark, two annotators independently label 180 generated claims balanced by automatic verifier label; verifier alignment is computed on the 156 claims where the annotators agree. For Real-NAB, one blind annotator audits 90 event-grounded claims to test whether verifier labels remain plausible under real-world noise. Detailed annotation protocol, label distributions, and confusion matrices are provided in Appendix D and F.

4.4 Controls and Diagnostics

All full-scale prompt comparisons use the same fixed rule-based automatic verifier, claim parser, verifier-guided repair module, cost-accounting pipeline, and pre-specified thresholds. These controls make prompt and model comparisons reproducible under a single evaluation operator, but they do not make automatic scores human-level truth estimates. To evaluate the verifier itself, we construct 1,600 controlled claim-level cases from the 400 synthetic test examples and compare majority, type-only, type+interval, full extracted-event-graph, no-interval, and gold-event oracle variants. Auxiliary diagnostics include threshold calibration, coverage, pattern-wise breakdowns, gold-event versus extracted-event prompting, mismatched event lists, deterministic event verbalization, and Real-NAB localization. The retained 80-example multi-provider pilot is summarized at prompt level in Tables 27 and 28. Metric definitions, prompt templates, generation settings, calibration diagnostics, and reproducibility artifacts are provided in Appendices A.6, B, E, and I.

5 Experimental Results

We organize the results around three evaluation questions: **RQ1: Claim Faithfulness.** How often do LLM-generated time-series explanations receive explicit support from the extracted event graph? **RQ2: Evidence Use.** Does explicit temporal evidence improve automatic claim faithfulness and reduce verifier-unsupported claims? **RQ3: Verifier Calibration.** How reproducible and calibrated is the automatic verifier?

5.1 Claim Faithfulness

Prompt	Faith. ↑	ΔFaith.	Unsup. ↓	ΔUnsup.	Claims/Expl.	Ret. ↑
Direct	0.493	–	0.392	–	10.27	0.530
CoT	0.360	-0.133	0.558	+0.167	26.98	0.298
Structured	0.353	-0.140	0.468	+0.076	6.82	0.477
Event-grounded	0.676	+0.183	0.113	-0.279	8.46	0.850

Table 1: Main prompt-level synthetic results averaged over five LLMs on the 400-example test split. Faith. denotes automatic claim faithfulness, and Unsup. denotes verifier-unsupported rate under the extracted event graph. Δ columns are differences from direct prompting; the event-grounded differences are significant under paired bootstrap resampling ($p < 0.001$), with 95% CIs of [0.172, 0.195] for Faith. and [-0.288, -0.270] for Unsup. Retention is the fraction of claims preserved after verifier-guided repair.

Main synthetic scores. Free-form prompting often lacks extracted-event support. Table 1 summarizes the 400-example synthetic evaluation. Claims unsupported under the extracted event graph are common under free-form prompting: direct prompting yields 0.493 automatic claim faith-

fulness and a 0.392 verifier-unsupported rate, while CoT lowers automatic claim faithfulness to 0.360 and raises verifier-unsupported claims to 0.558. Event-grounded prompting is the only prompting condition that consistently improves both metrics, reaching 0.676 automatic claim faithfulness and 0.113 verifier-unsupported rate when averaged over the five LLMs. Appendix Tables 15, 25, and 26 provide per-model, pattern-by-prompt, repair, cost, and token diagnostics. The strongest single configuration is DeepSeek-V4-Flash with event-grounded prompting, which reaches 0.746 automatic claim faithfulness and 0.077 verifier-unsupported rate.

Pattern-level diagnostics. The pattern-wise breakdown in Table 2 shows that the gain is not concentrated in one synthetic phenomenon. Event-grounded prompting improves automatic claim faithfulness for all eight pattern types, with the largest gains on sharp drops (+0.307), seasonality (+0.273), volatility shifts (+0.227), and anomalies (+0.226). The remaining errors are informative: volatility shifts, seasonality, and anomalies still have the highest event-grounded verifier-unsupported rates, indicating that localized or composite temporal behavior remains harder to explain faithfully. Appendix Table 17 shows representative claim-level failures.

5.2 Evidence Use

Statistical and verifier-guided repair evidence. Event-grounded prompting consistently improves automatic claim faithfulness and reduces verifier-unsupported claims. The event-grounded gains are stable under paired bootstrap resampling: compared with direct prompting, event-grounded prompting improves automatic claim faithfulness by 0.183 and reduces verifier-unsupported rate by 0.279 ($p < 0.001$ for both; Appendix Table 13). CoT has low retention after verifier-guided repair (0.298), meaning that many generated claims must be removed, while event-grounded explanations retain 0.850 of claims. The mismatched-event stress test in Table 3 gives a complementary check: replacing the correct event list with events from another series drops automatic claim faithfulness by 0.481 for DeepSeek-V4-Flash and 0.422 for GPT-4.1-mini. This supports the interpretation that event-grounded improvements come from evidence use rather than prompt formatting alone; extractor-sharing effects are examined in the verifier controls below.

Automatic claim faithfulness versus informativeness. Appendix Table 16 shows why automatic claim faithfulness should not be conflated with informativeness. CoT covers more gold events than direct prompting because it generates many more claims, but it also produces the highest verifier-unsupported rate. Event-grounded prompting is more conservative: it has higher automatic claim faithfulness and lower verifier-unsupported rate, but lower gold-event coverage than CoT. The deterministic event verbalizer provides an extractor-aligned reference: direct template verbalization reaches 0.936 automatic claim faithfulness and 0.040 verifier-unsupported rate, with high event coverage and no API generation cost. This shows that strong automatic scores are possible when language is tightly constrained to the extracted event graph, but it should be read as an extractor-aligned reference rather than an LLM explanation method. It is not an upper bound on explanation quality: it verbalizes the extractor's own event representation and omits richer but harder-to-verify explanatory language. Its high score shows that verifier-aligned language can be recognized reliably, but TIME-CLAIMBENCH should not be used to optimize automatic verifier scores alone.

5.3 Verifier Calibration

Controlled verifier and event-list checks. Table 5 first checks the verifier on controlled synthetic claim cases. Type-only matching reaches only 0.456 macro-F1, while adding interval and extracted-event-graph checks raises macro-F1 to 0.733; the gold-event oracle reaches 0.884, indicating that automatic event extraction is a major remaining bottleneck.

The generation-side event-list ablation in Table 4 provides a separate check. Replacing the richer extracted event list with sparse synthetic gold events reduces claim volume and automatic claim faithfulness for both tested models. Thus, gold events are useful for isolating verifier and extractor effects, but they are not a simple upper bound for generation: the representation and amount of evidence supplied to the LLM also affect how many auditable claims it writes.

Human calibration. Table 6 reports the calibrated human audit. The two annotators reach substantial agreement on the 180-claim sample (three-way $\kappa = 0.680$; binary $\kappa = 0.696$), showing that the support-label task is annotatable after guideline

Pattern	Direct Faith.	Event Faith.	Δ Faith.	Event Unsup.	Event Part.	Event Claims/Expl.
Upward trend	0.519	0.691	+0.172	0.008	0.301	8.70
Downward trend	0.493	0.685	+0.192	0.023	0.292	8.54
Sharp rise	0.552	0.714	+0.162	0.047	0.240	8.96
Sharp drop	0.452	0.759	+0.307	0.059	0.183	8.74
Volatility shift	0.332	0.558	+0.227	0.254	0.187	8.40
Change point	0.513	0.719	+0.206	0.063	0.218	7.92
Seasonality	0.385	0.658	+0.273	0.214	0.128	10.19
Anomaly	0.410	0.636	+0.226	0.235	0.129	6.26

Table 2: Pattern-wise diagnostic breakdown on the 400-example synthetic test split, pooled over the five full-test models. Direct and event-grounded columns compare automatic claim faithfulness before verifier-guided repair. The event-grounded columns also show remaining verifier-unsupported and partial-support rates, exposing which temporal phenomena remain difficult after grounding.

Model	Event evidence	Faith. ↑	ΔFaith.	Unsup. ↓	Part.	Claims/Expl.	Ret.	Cost/Expl.
DeepSeek-V4-Flash	Matched events	0.734	+0.000	0.078	0.188	8.51	0.881	$0.00043
DeepSeek-V4-Flash	Mismatched events	0.253	-0.481	0.294	0.453	8.85	0.571	$0.00047
GPT-4.1-mini	Matched events	0.710	+0.000	0.085	0.205	9.26	0.865	$0.00137
GPT-4.1-mini	Mismatched events	0.288	-0.422	0.302	0.410	9.30	0.589	$0.00134

Table 3: Mismatched event-list ablation on the 80-example stratified synthetic subset. Faith. denotes automatic claim faithfulness, and Unsup. denotes verifier-unsupported rate under the extracted event graph. Matched events use the correct extracted event list in the event-grounded prompt. Mismatched events keep the same raw series but supply events from another series, then verification is performed against the correct extracted event graph. The large Faith. drop shows that event-grounded prompting is sensitive to the supplied evidence rather than simply benefiting from prompt formatting.

Model	Event list	Faith. ↑	Unsup. ↓	Part.	Claims/Expl.	Faith.+Rep. ↑	Ret. ↑	Sem. Pres. ↑	Cost/Expl. ↓
GPT-4.1-mini	Extracted events	0.706	0.082	0.212	9.71	0.992	0.881	1.000	$0.00140
	Gold events	0.503	0.444	0.053	1.23	0.882	0.859	1.000	$0.00033
DeepSeek-V4-Flash	Extracted events	0.746	0.077	0.176	8.88	0.992	0.899	1.000	$0.00044
	Gold events	0.556	0.421	0.023	1.09	0.886	0.924	1.000	$0.00015

Table 4: Gold-event versus extracted-event ablation on the 400-example synthetic test split. Faith. denotes automatic claim faithfulness, and Unsup. denotes verifier-unsupported rate under the corresponding event list. Extracted-event rows reuse the original event-grounded generations and recompute verification with the same parser used for the gold-event rows. Gold-event rows rerun event-grounded generation with synthetic gold events as the event list.

Verifier	Macro-F1	Sup. F1	Part. F1	Unsup. F1	IoU
Majority unsupported	0.222	0.000	0.000	0.667	0.000
Type-only	0.456	0.552	0.000	0.817	0.531
Type+direction	0.456	0.552	0.000	0.815	0.531
Type+interval	0.733	0.819	0.564	0.817	0.604
Event graph (no interval)	0.458	0.552	0.005	0.817	0.537
Event graph	0.733	0.819	0.564	0.817	0.601
Gold-event oracle	0.884	0.856	0.797	1.000	0.651

Table 5: Verifier baseline and ablation results on 1,600 synthetic claim-level cases derived from the 400-example test split. The benchmark contains one supported claim, one interval-mislocalized partial claim, and two unsupported claims per series. Sup., Part., and Unsup. F1 are class-specific F1 scores; IoU is computed for supported and partial cases with predicted support intervals.

Comparison	N	Exact	Macro-F1	κ	Auto-Sup. Accept	Non-Sup. F1
Annotators (3-way)	180	0.867	–	0.680	–	–
Annotators (binary)	180	0.883	–	0.696	–	–
Verifier vs consensus gold	156	0.410	0.353	0.090	0.807	0.333

Table 6: Calibrated human audit on the balanced 180-claim sample. Consensus gold uses the 156 claims with annotator agreement. Auto-Sup. Accept is the fraction of verifier-supported claims judged supported by humans; Non-Sup. F1 collapses PARTIAL and UNSUPPORTED into a non-supported class. Additional metric definitions and the remaining 24 cases are summarized in Appendix A.6 and Appendix D.

calibration. Because the audit sample is balanced by automatic verifier labels, it diagnoses label reliability rather than estimating corpus-level human label prevalence. Verifier alignment with the 156 consensus-gold claims is much lower: exact agreement is 0.410, macro-F1 is 0.353, and $\kappa = 0.090$. We therefore interpret the automatic verifier as a fixed, human-calibrated evaluation operator rather than a human-level judge. This calibration re-

veals that the verifier is precision-oriented rather than recall-oriented. Automatic SUPPORTED labels align reasonably with human support, but many automatic PARTIAL and UNSUPPORTED labels are judged supported by humans, often because humans accept broader temporal phrasing or weaker evidence matches (Appendix Table 14). Therefore, verifier-unsupported-rate results should be interpreted as conservative extracted-event-graph mismatch rates rather than absolute hallucination rates. Detailed audit appendices report label distributions, unresolved disagreement types, and human–human and verifier–human confusion matrices in Appendix 18–20.

Together, these checks support relative prompt and model diagnosis under a fixed verifier, while event extraction and human calibration limit absolute truth estimation. Threshold tuning does not close the human-verifier gap: cross-validated thresholds slightly reduce exact agreement and macro-F1 relative to the fixed thresholds used in the main experiments. Finer error-type and score-bin calibration results are reported in Appendix Tables 21 and 22.

Model	Faith. ↑	Unsup. ↓	Pos. Hit ↑	FA ↓	Prec. ↑	IoU ↑
DeepSeek-V4-Flash	0.671	0.049	0.960	0.020	0.505	0.080
Event verbalizer	0.961	0.015	0.960	0.560	0.505	0.080
GPT-4.1-mini	0.617	0.049	0.960	0.010	0.505	0.080

Table 7: Real-NAB localization diagnostic results. Faith. denotes automatic claim faithfulness, and Unsup. denotes verifier-unsupported rate under the extracted event graph. Pos. Hit is the fraction of positive anomaly windows with an overlapping anomaly-like claim; FA is explicit false-anomaly rate on normal windows; Prec. is broad anomaly-like localization precision over all windows, where normal-window anomaly-like claims contribute to the denominator but cannot count as NAB-overlap matches; IoU is interval intersection-over-union with the NAB anomaly window on positive windows.

Slice	N	Supp. (%)	Part. (%)	Unsup. (%)	Macro-F1
All audited claims	90	36.7	27.8	35.6	0.390
DeepSeek-V4-Flash	45	42.2	28.9	28.9	0.403
GPT-4.1-mini	45	31.1	26.7	42.2	0.367
Anomaly windows	45	31.1	31.1	37.8	0.355
Normal windows	45	42.2	24.4	33.3	0.418

Table 8: Single-annotator Real-NAB audit for 90 event-grounded claims. Supp., Part., and Unsup. are human-label percentages; Macro-F1 is reported on a 0–1 scale.

Real-world stress test. The Real-NAB localization diagnostic evaluates real anomaly windows rather than complete synthetic event labels. Table 7 shows that both representative LLMs identify 0.960 of positive anomaly windows with an overlapping anomaly-like claim and keep explicit normal-window false-anomaly rates low (0.020 for DeepSeek-V4-Flash and 0.010 for GPT-4.1-mini). However, broad anomaly-like precision is only 0.505 and interval IoU is only 0.080, meaning that many real windows contain sharp changes or change-like claims that do not align cleanly with NAB anomaly intervals. A 90-claim blind human audit gives a stricter check: verifier-human exact agreement is 0.389, macro-F1 is 0.390, and $\kappa = 0.083$ (Table 8). Appendix Tables 23 and 24 report the Real-NAB by-label calibration and full three-way confusion matrix. The controlled synthetic split remains the main setting for full claim-grounding evaluation.

5.4 Summary of Findings

Fluency does not imply claim faithfulness. CoT increases claim volume without improving grounding, while event-grounded prompting narrows generation around explicit temporal evidence. The verifier and verifier-guided repair results also show why paragraph-level scores are insufficient: verifier-unsupported claims can be removed or corrected only after decomposition into auditable units. Human audits clarify the intended use of the scores: they support reproducible model and prompt diagnosis, while human calibration remains necessary for estimating absolute claim truth in a new domain.

6 Conclusion

We introduced TIMECLAIMBENCH, a claim-level benchmark for testing whether time-series explanations are grounded in numerical temporal evidence. Across five LLMs, many generated temporal claims fail to receive support from the fixed extracted-event verifier, and chain-of-thought increases claim volume without improving automatic claim faithfulness. Event-grounded prompting improves automatic claim faithfulness, reduces verifier-unsupported claims, and enables less destructive verifier-guided repair. Human audits further calibrate the automatic verifier as a fixed evaluation operator rather than a substitute for human judgment. The same claim-level evaluation framework can be extended to multivariate series, domains, and domain-specific event types.

Limitations

TimeClaimBench currently focuses on univariate time-series explanations and a fixed rule-based event representation. This controlled setting makes claim grounding measurable, but domain-specific temporal concepts, multivariate dependencies, and expert semantics may require additional event types and annotation protocols. Claim-level grounding also does not measure forecasting quality, downstream utility, or causal validity unless such information is explicitly present in the input.

Future extensions can add broader domains, multivariate series, and larger expert annotation studies. These directions are complementary to the present goal: making temporal explanation faithfulness measurable at the claim level.

Ethics

This work studies faithfulness auditing for natural-language explanations of time-series data. The benchmark is intended for evaluation and diagnostic analysis, not for replacing domain expertise or making automated decisions in high-stakes settings. The experiments use synthetic data and public time-series data; no private or personally identifiable information is released.

Human annotation is used only for calibration of aggregate benchmark behavior. Reported audit results are aggregated or de-identified, and any released annotation artifacts will follow annotator consent and venue artifact policies. We will document data sources, licenses, and evaluation scripts to support reproducible and responsible use.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification*, FEVER '21, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Abdul Fatir Ansari, Lorenzo Stella, Ali 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 Bernie Wang. 2024. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*. Expert Certification.

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '23, pages 283–294, Toronto, Canada. Association for Computational Linguistics.

Yifu Cai, Arjun Choudhry, Mononito Goswami, and Artur Dubrawski. 2024. TimeSeriesExam: A time series understanding exam. *ArXiv preprint*, arXiv:2410.14752.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. 2024. TEMPO: Prompt-based generative pre-trained transformer for time series forecasting. In *Proceedings of the Twelfth International Conference on Learning Representations*, ICLR '24, Vienna, Austria.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. TabFact: A large-scale dataset for table-based fact verification. In *Proceedings of the International Conference on Learning Representations*, ICLR '20, Online.

Ziwen Chen, Xiaoyuan Zhang, and Ming Zhu. 2025. TS-CLIP: Time series understanding by CLIP. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, EMNLP '25, pages 4646–4664, Suzhou, China. Association for Computational Linguistics.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, ICML '24, Vienna, Austria. JMLR.org.

Elizabeth Fons, Rachneet Kaur, Soham Palande, Zhen Zeng, Tucker Balch, Manuela Veloso, and Svitlana Vyetrenko. 2024. Evaluating large language models on time series feature understanding: A comprehensive taxonomy and benchmark. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, EMNLP '24, pages 21598–21634, Miami, Florida, USA. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '23, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML '24, Vienna, Austria. JMLR.org.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23, New Orleans, LA, USA. Curran Associates Inc.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 4198–4205, Online. Association for Computational Linguistics.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2021. Truth-conditional captions for time series data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, EMNLP '21, pages 719–733, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-LLM: Time series forecasting by reprogramming large language models. In *Proceedings of the Twelfth International Conference on Learning Representations*, ICLR '24, Vienna, Austria.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. WiCE: Real-world entailment for claims in Wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, EMNLP '23, pages 7561–7583, Singapore. Association for Computational Linguistics.

Shankar Kantharaj, Rixie Tiffany Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. Chart-to-text: A large-scale benchmark for chart summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '22, pages 4005–4023, Dublin, Ireland. Association for Computational Linguistics.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024. Prometheus: Inducing fine-grained evaluation capability in language models. In *Proceedings of the Twelfth International Conference on Learning Representations*, ICLR '24, Vienna, Austria.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. 2025. Time-MQA: Time series multi-task question answering with context enhancement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '25, pages 29736–29753, Vienna, Austria. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Alexander Lavin and Subutai Ahmad. 2015. Evaluating real-time anomaly detection algorithms – the Numenta Anomaly Benchmark. In *Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications*, ICMLA '15, pages 38–44, Miami, FL, USA.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, EMNLP '23, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2024. AutoTimes: Autoregressive time series forecasters via large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NeurIPS '24, Vancouver, BC, Canada. Curran Associates Inc.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, ACL '22, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, EMNLP '23, pages 12076–12100, Singapore. Association for Computational Linguistics.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. 2024. S^2IP-LLM: Semantic space informed prompt learning with LLM for time series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, ICML '24, Vienna, Austria. JMLR.org.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David

Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840.

Medhasweta Sen, Zachary Gottesman, Jiaxing Qiu, C. Bayan Bruss, Nam Nguyen, and Tom Hartvigsen. 2026. BEDTime: A unified benchmark for automatically describing time series. *ArXiv preprint*, arXiv:2509.05215.

Preetham Sivalingam, Murari Mandal, Saurabh Deshpande, and Dhruv Kumar. 2026. LLM-as-a-judge for time series explanations. *ArXiv preprint*, arXiv:2604.02118.

Marek Strong and Andreas Vlachos. 2025. TSVer: A benchmark for fact verification against time-series evidence. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, EMNLP '25, pages 29906–29926, Suzhou, China. Association for Computational Linguistics.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. 2024. TEST: Text prototype aligned embedding to activate LLM's ability for time series. In *Proceedings of the Twelfth International Conference on Learning Representations*, ICLR '24, Vienna, Austria.

Benny Tang, Angie Boggust, and Arvind Satyanarayan. 2023. VisText: A benchmark for semantically rich chart captioning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '23, pages 7268–7298, Toronto, Canada. Association for Computational Linguistics.

Mohamed Trabelsi, Aidan Boyd, Jin Cao, and Huseyin Uzunalioglu. 2025. Time series language model for descriptive caption generation. *Engineering Applications of Artificial Intelligence*, 162:112673.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NeurIPS '24, Vancouver, BC, Canada. Curran Associates Inc.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. 2025. ChatTS: Aligning time series with LLMs via synthetic data for enhanced understanding and reasoning. *Proceedings of the VLDB Endowment*, 18(8):2385–2398.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with A unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '23, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, volume 36 of *NeurIPS '23*, pages 46595–46623, New Orleans, LA, USA. Curran Associates, Inc.

Luca Zhou, Pratham Yashwante, Marshall Fisher, Alessio Sampieri, Zihao Zhou, Fabio Galasso, and Rose Yu. 2026. CaTS-Bench: Can language models describe time series? *ArXiv preprint*, arXiv:2509.20823.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023. One fits all: Power general time series analysis by pretrained LM. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23, New Orleans, LA, USA. Curran Associates Inc.

A Details for TIMECLAIMBENCH

A.1 Benchmark Construction

Synthetic split. The synthetic benchmark is designed to evaluate whether natural-language explanations are grounded in observable time-series evidence. We use controlled synthetic data so that event type, interval, direction, and reference claims are known by construction. This enables claim-level verification at a finer granularity than explanation-level preference judgments.

The synthetic split covers eight first-release temporal phenomena: upward trend, downward trend, sharp rise, sharp drop, volatility shift, change point, seasonality, and anomaly. Each JSONL record contains a time series, metadata, gold events, a question, and reference claims. The full synthetic split contains 1,200 training examples, 200 development examples, and 400 test examples. The main experiments use the full 400-example test split with 50 examples per phenomenon. We also retain a stratified 80-example pilot subset for faster provider checks.

Split	Size	Use
Synthetic train	1,200	development and prompt checks
Synthetic dev	200	validation and debugging
Synthetic test	400	main evaluation
Synthetic pilot subset	80	fast provider checks
Real-NAB localization diagnostic	200	real-world localization check

Table 9: Benchmark splits. The synthetic test split is the main evaluation set and contains 50 examples for each of the eight synthetic phenomena. The Real-NAB localization diagnostic contains 100 anomaly-centered windows and 100 normal windows.

Real-NAB localization diagnostic. To test whether the pipeline transfers beyond synthetic control, we add a real-world localization diagnostic based on the Numenta Anomaly Benchmark (NAB) (Lavin and Ahmad, 2015). We use the real NAB subsets `realKnownCause`, `realTraffic`, `realTweets`, `realAWSCloudwatch`, and `realAdExchange`. NAB provides timestamped univariate series and labeled anomalous windows. We convert these labels into gold ANOMALY events for localization diagnostics while still using the same automatic event extractor to detect trends, sharp changes, seasonality, volatility shifts, change points, and anomalies.

The Real-NAB localization diagnostic contains 200 fixed-length windows: 100 anomaly-centered windows and 100 normal windows. Values are normalized within each window for compatibility with the event operators. Original values, timestamps, source file names, and NAB label windows are preserved in metadata.

We treat Real-NAB as a localization diagnostic rather than as the primary benchmark. NAB primarily supports anomaly-window and change-like evaluation under real noise. It does not provide complete gold labels for all eight synthetic phenomenon types, and its anomaly windows can be coarser than the temporal spans expressed in natural-language claims. Therefore, Real-NAB results should be read as localization and robustness diagnostics, not as complete human-equivalent claim verification.

Subset	Windows	Annotation used
Anomaly-centered	100	NAB windows converted to gold anomaly events
Normal	100	no NAB anomaly window in selected span

Table 10: Real-NAB localization diagnostic construction.

A.2 Event Extraction and Extracted Event Graph

The verifier converts each time series into temporal events using deterministic operators with fixed settings across all models and prompts; Table 11 summarizes the default settings. The extractor emits typed events with intervals, scores, and attributes such as direction, magnitude, slope, period, or z-score. Overlapping events of the same type are deduplicated when interval IoU is at least 0.8, keeping the higher-scoring event.

The extracted events form an extracted event graph whose nodes are temporal events. Edges encode temporal relations such as BEFORE, AFTER, OVERLAPS, and CONTAINS. The verifier retrieves candidate events by claim type and uses the graph to check whether the claimed temporal structure is compatible with the extracted evidence.

A.3 Claim-Level Labels and Error Taxonomy

Table 12 expands the three-way support labels used in the main evaluation into operational boundary cases for annotation and verifier diagnostics. The benchmark assigns one primary label to each claim, namely SUPPORTED, PARTIAL, or UNSUPPORTED, and records finer error types when a non-supported claim can be localized to a specific failure mode. These error types are not additional classes in the main metric. Instead, they are used to interpret failures such as interval shifts, overstatement, numeric

Operator	Evidence emitted	Key fixed settings
Trend	Upward/downward trend events with slope, interval, and strength.	Least-squares slopes over full, first-half, second-half, and middle-half spans; minimum length 8; slope threshold 0.03; weak < 0.08, moderate < 0.16, strong otherwise.
Sharp rise/drop	Local sharp-change events with direction, magnitude, and interval.	Difference window 5; z-normalized change at least 2.0 relative to first-difference standard deviation.
Volatility shift	Intervals where local variability changes substantially.	Before/after window 10; variance-ratio-style volatility ratio at least 1.8.
Change point	Mean-shift events with split point and affected interval.	All candidate splits scanned; at least four points per side; best mean-shift score > 1.0.
Seasonality	Periodic events with period and strength.	Periods 4–24, capped at half the sequence length; autocorrelation at least 0.45.
Anomaly	Point or short-window anomaly events.	Absolute z-score at least 3.0.
Peak/trough	Local extrema used as auxiliary evidence.	Prominence at least 1.0.
Deduplication	Removal of redundant events.	Same-type events deduplicated when interval IoU is at least 0.8; higher-scoring event is kept.

Table 11: Fixed event-extraction settings used by the default TIMECLAIMBENCH verifier.

mismatch, hallucinated events, unsupported causal statements, and chronology violations.

A.4 Claim Parsing and Verification

Claim parsing. The claim parser maps free-form explanations to atomic temporal claims. It detects event types with lexical patterns, including trends, rises, drops, volatility changes, change points, seasonality, anomalies, peaks, troughs, and recoveries. It extracts explicit index spans such as "indices 20–30", maps coarse phrases such as "early", "middle", "late", "first half", and "final segment" to approximate spans, and records optional direction, strength, and numeric values.

The parser is conservative. If a sentence contains multiple temporal phenomena, it is split into separate claims when possible. If no temporal phenomenon can be identified, the claim type is marked as unknown and normally requires strong matching evidence to be labeled supported. Unsupported causal language is marked when the explanation introduces an external cause not observable from the univariate sequence.

Claim–event scoring. Let c be a parsed claim and e a candidate event from the extracted event graph. For each claim c, the verifier retrieves candidate events whose types are compatible with the claim. It computes a weighted agreement score:

$$\begin{aligned} S(c,e) = {} & w_t S_{\text{type}}(c,e) \\ & + w_d S_{\text{direction}}(c,e) \\ & + w_i S_{\text{interval}}(c,e) \\ & + w_s S_{\text{strength}}(c,e) \\ & + w_n S_{\text{numeric}}(c,e). \end{aligned}$$

The fixed weights are $(w_t, w_d, w_i, w_s, w_n) = (.35, .20, .25, .10, .10)$. The type score measures compatibility between the claim type and event type. The direction score checks sign or trend direction when present. The interval score is interval IoU when a claim has an explicit or inferred span. The strength score compares coarse strength or magnitude categories. The numeric score checks extracted values when the claim contains a concrete value, count, index, or period.

Optional attributes absent from the claim receive a neutral component score of 1.0 rather than triggering weight renormalization. If the claim specifies direction, strength, or numeric content but the matched event lacks the corresponding attribute, the verifier assigns a conservative default score of 0.8 for that component. If a claim has no explicit or inferred interval, the interval component is therefore neutral and the interval score is not used as a hard support gate, but the matched event must still be compatible in type and direction.

Label assignment. The best event match determines the automatic label. A claim is labeled SUPPORTED when the best score is at least 0.75 and, for claims with explicit intervals, interval IoU is at least 0.5. After parsing, a direction-incompatible atomic claim is gated to UNSUPPORTED. Otherwise, a claim is labeled PARTIAL only when a compatible candidate event exists and the best score is at least 0.45, but the claim fails full support because of a non-direction grounding defect. Operationally, this includes an explicit or inferred interval whose IoU with the matched event is below 0.5 but still indicates a shifted or broadened span, compatible

Label	Error type	Definition	Typical example	Boundary rule
SUPPORTED	–	The main temporal phenomenon, direction, and approximate interval are grounded in the series.	"A sharp rise occurs around indices 20–25."	Exact wording need not match the event name if the phenomenon and interval are compatible.
PARTIAL	interval shift	The right phenomenon is present but localized to a shifted, too broad, or too narrow interval.	"The increase lasts through the final third" when the event ends near the middle.	Use PARTIAL when the claim would become supported after interval correction.
	overstatement	A grounded pattern is described with excessive strength, magnitude, duration, or frequency.	"A strong trend" for a weak but detectable trend.	Use PARTIAL when the phenomenon exists but a salient attribute is wrong.
	numeric error	The event is present, but the claim gives an inaccurate value, count, period, or index.	"The cycle repeats every 8 steps" when the detected period is closer to 12.	Use PARTIAL when the temporal pattern is grounded but the numeric detail is wrong.
	mixed claim	One part of a claim is grounded, but another part adds unsupported content.	"The series rises near $t = 20$ because of an external shock."	Use PARTIAL when supported and unsupported parts are inseparable in the same claim.
UNSUPPORTED	pattern hallucination	The explanation introduces a temporal pattern absent from the evidence.	Claiming seasonality or a change point when no compatible event is detected.	Use UNSUPPORTED when no compatible event can ground the main claim.
	wrong direction or timing	The claimed event has the wrong sign or occurs in a region where no compatible event is visible.	Claiming a sharp drop where the evidence shows a sharp rise.	Direction-incompatible atomic claims are gated to UNSUPPORTED; use PARTIAL only for inseparable mixed claims with a grounded subclaim.
	causal claim	The explanation infers an external cause not observable from the univariate series.	"The drop is caused by an external shock."	Causal language is unsupported unless causal evidence is explicitly provided.
	chronology violation	The claim uses future-window evidence to justify an earlier observation under a decision time.	"At $t = 30$, the series has entered a sustained decline" when the decline is only visible after $t = 50$.	Use this error when the claim may be true with full-sequence access but invalid at the specified decision time.

Table 12: Full claim-level support taxonomy and boundary rules used for annotation and verifier diagnostics.

type and direction with mismatched strength or magnitude, numeric mismatch for a claimed value, count, index, or period, or an inseparable mixed claim whose grounded subclaim can be matched while the remaining content is unsupported. For mixed sentences, the parser first splits separable temporal claims when possible; remaining inseparable mixed claims are treated as PARTIAL only when a grounded subclaim is matched. Remaining claims are labeled UNSUPPORTED. All thresholds are fixed before model comparison and are used consistently across models, prompts, and diagnostic studies.

A.5 Verifier-Guided Repair

The verifier-guided repair module uses verifier output to produce a conservative edited explanation. It preserves claims labeled SUPPORTED, corrects claims labeled PARTIAL using the matched grounded event when possible, and removes claims labeled UNSUPPORTED. The repair prompt is instructed not to introduce new patterns, numeric values, or causal explanations that are not present in the evidence.

Because the repaired output is guided and re-evaluated by the same verifier, automatic claim faithfulness after verifier-guided repair is treated as a fixed-criterion measurement rather than independent human evidence. We therefore report claim retention and supported-claim preservation together with post-repair automatic claim faithfulness.

A.6 Metric Definitions

Table abbreviations. Across tables, *Faith.* denotes automatic claim faithfulness, *Unsup.* verifier-unsupported rate, *Part.* partial rate, *Expl.* explanation, *Ret.* claim retention after verifier-guided repair, *Sem. Pres.* supported-claim preservation, *Chron. Viol.* chronology violations, *Rep.* verifier-guided repair, *Sup./Supp.* supported, *Non-Sup.* non-supported, and *Cov.* coverage. *Claims/Expl.*, *Cost/Expl.*, and *Tokens/Expl.* are per-explanation averages. *Faith.+Rep.* and *Unsup.+Rep.* report the corresponding metric after verifier-guided repair. For Real-NAB and calibration tables, *Pos. Hit*, *FA*, *Prec.*, *IoU*, and *Macro-F1* denote positive-window hit rate, false-anomaly rate, precision, interval intersection-over-union, and macro-averaged F1. *Non-Sup. F1* is the binary F1 after collapsing PARTIAL and UNSUPPORTED into a single non-

supported class.

Automatic claim-label rates are computed as claim-level micro-averages over the evaluation set. Let S, P, and U denote SUPPORTED, PARTIAL, and UNSUPPORTED. Let $\mathcal{D}$ be a set of generated explanations and

$$N_C = \sum_{r \in \mathcal{D}} |\mathcal{C}(r)|$$

be the total number of extracted claims. If $N_C = 0$, claim-level rates are defined as 0.

Automatic claim faithfulness. The main automatic claim faithfulness metric is the supported-claim rate:

$$\text{Faith} = \frac{1}{N_C} \sum_{r \in \mathcal{D}} \sum_{c_i \in \mathcal{C}(r)} \mathbf{1}[\hat{y}_i = S].$$

Partial claims do not receive credit in the main score. We report them separately:

$$\text{PartialRate} = \frac{1}{N_C} \sum_{r \in \mathcal{D}} \sum_{c_i \in \mathcal{C}(r)} \mathbf{1}[\hat{y}_i = P],$$

$$\text{UnsupportedRate} = \frac{1}{N_C} \sum_{r \in \mathcal{D}} \sum_{c_i \in \mathcal{C}(r)} \mathbf{1}[\hat{y}_i = U].$$

Claims per explanation. Claims per explanation measures how many auditable temporal claims a prompt produces:

$$\text{Claims/Expl.} = \frac{1}{|\mathcal{D}|} \sum_{r \in \mathcal{D}} |\mathcal{C}(r)|.$$

This metric is reported because a prompt may generate more coverage by producing many more claims, while also increasing unsupported content.

Macro-F1. For a label set $\mathcal{Y}$, class-wise precision, recall, and F1 are

$$P_y = \frac{\text{TP}_y}{\text{TP}_y + \text{FP}_y}, \quad R_y = \frac{\text{TP}_y}{\text{TP}_y + \text{FN}_y},$$

$$F1_y = \frac{2 P_y R_y}{P_y + R_y}.$$

Macro-F1 is the unweighted average over labels:

$$\text{MacroF1} = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} F1_y.$$

For three-way support-label evaluation, $\mathcal{Y} = \{S, P, U\}$. For binary non-supported F1, PARTIAL and UNSUPPORTED are collapsed into the positive non-supported class. If a precision, recall, or F1 denominator is empty, the corresponding value is defined as 0.

Interval localization. Intervals are zero-based closed index spans. For $I_1 = [s_1, t_1]$ and $I_2 = [s_2, t_2]$,

$$a = \max\{0, \min(t_1, t_2) - \max(s_1, s_2) + 1\},$$
$$b = \max(t_1, t_2) - \min(s_1, s_2) + 1,$$
$$\text{IoU}(I_1, I_2) = a/b.$$

If a claim has no explicit or inferred interval, the interval-IoU gate is not applied, but the verifier must still find compatible evidence for the claim type and direction.

Coverage. Coverage measures whether faithful claims mention important events. Let $E_{\leq K}(X)$ denote the available highest-scoring extracted events for a series X, up to K events. Top-K supported coverage is

$$\text{TopKCov}_S = \frac{1}{|\mathcal{D}|} \sum_{r \in \mathcal{D}} q_r,$$
$$q_r = \frac{\sum_{e \in E_{\leq K}(X_r)} \mathbf{1}[A(e, r)]}{\max(1, |E_{\leq K}(X_r)|)}.$$

Here, $A(e, r)$ holds if there exists a claim $c_i \in \mathcal{C}(r)$ such that $\hat{y}_i = S$ and c_i matches event e.

Top-K grounded coverage replaces $\hat{y}_i = S$ with $\hat{y}_i \in \{S, P\}$. If fewer than K events are extracted, the denominator uses the number of available events; if no events are extracted, that example contributes 0.

Gold coverage is computed only for synthetic data and measures whether each gold event is matched by a compatible SUPPORTED or PARTIAL claim with interval IoU of at least 0.3. If a series has no gold events for a gold-coverage diagnostic, its contribution is defined as 0.

Repair diagnostics. Let r' be the repaired explanation. Claim retention is

$$\text{Retention} = \frac{|\mathcal{C}(r')|}{|\mathcal{C}(r)|}.$$

If the original explanation contains no extracted claims, retention is defined as 0. In the full synthetic evaluation, this case does not occur (0/8,000 model–prompt outputs), so the convention does not affect aggregate retention trends. Supported-claim preservation measures whether originally supported claims remain present after verifier-guided repair:

$$\text{SupportedPreservation} = \frac{N_{\text{pres}}}{N_S}.$$

Here N_S is the number of originally supported claims, and N_{pres} counts those claims when r' contains a supported claim grounded by an event overlapping the original evidence event. If $N_S = 0$, supported-claim preservation is defined as 0. This is an operational semantic-preservation diagnostic, not an independent human evaluation of repair quality.

Chronology violations. When a decision time t_0 is specified, a claim violates chronology if its supporting evidence interval starts after t_0, or if the claim requires future-window evidence to justify an earlier observation. Chronology violations are reported as an error-type diagnostic.

Cost. Cost per explanation is computed from provider token usage and the pricing configuration:

$$\text{Cost/Expl.} = \frac{\text{TotalProviderCost}}{|\mathcal{D}|}.$$

Real-NAB localization diagnostics. For Real-NAB, let W^+ be anomaly-centered windows and W^- be normal windows. Let $\mathcal{A}$ denote anomaly-like claim types, including ANOMALY, SHARPRISE, SHARPDROP, and CHANGEPOINT. A claim is considered grounded for these diagnostics if it is labeled SUPPORTED or PARTIAL.

Positive hit rate measures whether positive windows contain at least one grounded anomaly-like claim overlapping the NAB anomaly window:

$$\text{PosHit} = \frac{1}{|W^+|} \sum_{w \in W^+} \mathbf{1}[P(w)].$$

Here, $P(w)$ holds if there exists an anomaly-like claim $c_i \in \mathcal{A}$ such that $\hat{y}_i \in \{S, P\}$ and $I(c_i) \cap I_{\text{NAB}}(w) \neq \emptyset$.

Explicit false-anomaly rate measures whether normal windows contain an explicit grounded anomaly claim:

$$\text{ExplicitFA} = \frac{1}{|W^-|} \sum_{w \in W^-} \mathbf{1}[F(w)].$$

Here, $F(w)$ holds if there exists a claim c_i such that $\tau(c_i) = \text{ANOMALY}$ and $\hat{y}_i \in \{S, P\}$.

Anomaly-like precision is a broad window-level localization diagnostic:

$$\text{AnomLikePrec} = \frac{\sum_{w \in W^+ \cup W^-} \mathbf{1}[M(w)]}{\sum_{w \in W^+ \cup W^-} \mathbf{1}[G(w)]}.$$

Here, $G(w)$ holds if the explanation contains at least one grounded anomaly-like claim in the selected Real-NAB window. $M(w)$ is defined only as a true match for anomaly-centered windows: it holds when $w \in W^+$, $G(w)$ is true, and at least one grounded anomaly-like claim overlaps the NAB anomaly interval $I_{\text{NAB}}(w)$. For normal windows $w \in W^-$, there is no NAB anomaly interval in the selected span, so $M(w) = 0$ by definition. Thus normal windows with grounded anomaly-like claims contribute to the denominator but not the numerator. If no window contains a grounded anomaly-like claim, anomaly-like precision is defined as 0. For positive windows, we also report mean interval IoU with the NAB anomaly window and normalized center distance:

$$\text{CenterDist}(I_1, I_2) = \frac{|m(I_1) - m(I_2)|}{T},$$

where $m(I)$ is the interval midpoint and T is the window length.

Aggregation and uncertainty. For prompt-level results, we first compute claim-level rates for each model–prompt run and then average over model runs. Bootstrap confidence intervals resample the 400 synthetic test series within each fixed model run. Paired prompt comparisons preserve shared model–series identifiers.

B Evaluation Protocol

B.1 Task and Generation Settings

The full synthetic evaluation uses the 400-example synthetic test split, with 50 examples for each of the eight temporal phenomena. Each example is converted into a model-ready input consisting of the raw univariate series and a question asking for a concise time-series explanation. For the Real-NAB localization diagnostic, we use 200 fixed-length windows: 100 anomaly-centered windows and 100 normal windows.

We evaluate five LLM backbones from four providers under four prompting conditions. Direct, chain-of-thought, and structured prompts receive only the raw time series and the question. The event-grounded prompt additionally receives the automatically extracted event list and instructs the model to cite listed evidence. All provider calls use temperature 0, the same input order, cached retry logic, and the same downstream claim parser, verifier, verifier-guided repair module, and cost-accounting pipeline. Thus, prompt comparisons

differ in prompting conditions, while verification is performed by the same fixed automatic verifier.

B.2 Formal Evaluation Task

Using the metric notation from Appendix A.6, each instance contains a zero-indexed univariate time series $X = (x_0, \ldots, x_{T-1})$ and a generated explanation r. The claim parser maps r to atomic temporal claims:

$$\mathcal{C}(r) = \{c_i\}_{i=1}^{n_r}.$$

The event extractor maps X to an extracted event graph:

$$G_X = (\mathcal{V}_X, \mathcal{R}_X),$$

where each event $e \in \mathcal{V}_X$ has an event type $\tau(e)$, optional direction $d(e)$, interval $I(e) = [s_e, t_e]$, score, and numeric attributes such as slope, magnitude, period, or z-score. Relations in $\mathcal{R}_X$ encode temporal links such as BEFORE, AFTER, OVERLAPS, and CONTAINS.

For each claim c_i, the verifier assigns a support label

$$\hat{y}_i \in \{\text{SUPPORTED}, \text{PARTIAL}, \text{UNSUPPORTED}\},$$

and, when applicable, an evidence event $\hat{e}_i$ and support interval $\hat{I}_i$. The verifier also records error types for non-supported claims when an error category can be identified.

B.3 Model and Prompt Details

The full synthetic evaluation uses five LLM backbones from four providers: GPT-4.1-mini, GPT-5.4-mini, DeepSeek-V4-Flash, Gemini-3-Flash, and Claude Sonnet 4.5. We compare four prompting conditions. Direct, chain-of-thought, and structured prompts receive only the raw time series and the question. The event-grounded prompt additionally receives the automatically extracted event list and is instructed to use only listed evidence. Generation settings are summarized in Appendix B.1; the prompt templates below give the exact user-facing instructions.

B.4 Prompt Templates

Direct prompt.

```
You are a time-series analyst.
Given the following time series, describe
the major temporal patterns.
Focus on trends, peaks, drops, volatility,
anomalies, seasonality, and change points.
Time series:
{series}
Question:
{question}
Answer with a concise explanation.
```

Chain-of-thought prompt.

```
You are a time-series analyst.
Analyze the time series step by step.
First inspect the overall trend.
Then inspect local peaks and drops.
Then inspect volatility and anomalies.
Finally provide a concise explanation.
Time series:
{series}
Question:
{question}
```

Structured prompt.

```
You are a time-series analyst.
Return a concise bullet list of grounded
temporal patterns.
Only include claims that can be checked
against the series.
Time series:
{series}
Question:
{question}
```

Event-grounded prompt.

```
You are a time-series analyst.
You are given a time series and
automatically extracted events.
Write an explanation using only the
listed events.
Do not mention patterns not supported
by the event list.
Cite event IDs in brackets.
Time series:
{series}
Extracted events:
{events}
Question:
{question}
Output format:
Explanation:
- [E?] ...
- [E?] ...
```

C Additional Empirical-Finding Tables

Prompt	Faith. 95% CI ↑	Unsup. 95% CI ↓	ΔFaith. (EG–Prompt)	ΔUnsup. (EG–Prompt)
Direct	0.493 [0.485, 0.501]	0.392 [0.383, 0.400]	0.183 [0.172, 0.195]	-0.279 [-0.288, -0.270]
CoT	0.360 [0.355, 0.365]	0.558 [0.552, 0.564]	0.316 [0.306, 0.325]	-0.446 [-0.454, -0.438]
Structured	0.353 [0.344, 0.361]	0.468 [0.458, 0.477]	0.323 [0.311, 0.335]	-0.355 [-0.366, -0.345]
Event-grounded	0.676 [0.667, 0.684]	0.113 [0.105, 0.120]	–	–

Table 13: Bootstrap uncertainty for prompt-level full synthetic results. We resample series within each fixed model run, compute claim-level rates per model, and average over the five models. Difference columns report paired event-grounded-minus-prompt comparisons; all three non-event-grounded comparisons have $p < 0.001$ for both automatic claim faithfulness and verifier-unsupported rate. Negative ΔUnsup. means fewer verifier-unsupported claims for event-grounded prompting.

Verifier label	N	Human sup.	Human part.	Human unsup.	Gate-consistent
Supported	57	0.807	0.035	0.158	0.807
Partial	59	0.797	0.153	0.051	0.203
Unsupported	40	0.750	0.025	0.225	0.250

Table 14: Verifier calibration on the 156 consensus-gold human-audit claims. Human columns show the consensus label distribution within each automatic verifier label. Gate-consistent counts verifier-supported claims as accepted only when humans also mark them supported, and counts verifier-partial/unsupported claims as accepted when humans mark them non-supported.

D Human Audit Details

Annotation protocol. We conduct the human audit as a calibration study for claim-level time-series grounding. We first sample 180 claims from the full synthetic verification outputs, balanced by automatic verifier label: 60 SUPPORTED, 60 PARTIAL, and 60 UNSUPPORTED. Annotators receive a written guideline defining the three support labels, the error-type taxonomy, and practical boundary rules for interval shifts, magnitude overclaims, mixed claims, and causal statements. During annotation, they see the claim text, the plotted time series, extracted events, gold/reference events, and the full explanation for context, but they do not see the automatic verifier label or each other's labels. After guideline calibration, two annotators independently label all 180 claims and agree on 156; Table 19 reports the annotator confusion matrix. We use these 156 agreement cases as consensus gold for verifier-alignment reporting; the 24 disagreements are excluded from verifier-alignment metrics and summarized separately. Table 18 reports the resulting label distributions. The unresolved disagreements mainly reflect boundary cases rather than random label noise: 13 involve SUPPORTED/PARTIAL disagreements, 8 involve SUPPORTED/UNSUPPORTED disagreements, and 3 involve PARTIAL/UNSUPPORTED disagreements. Among these 24 cases, the automatic verifier labeled 20 as UNSUPPORTED, 3 as SUPPORTED, and 1 as PARTIAL, indicating that unresolved annotation cases are concentrated in verifier-unsupported predictions.

Source	N	Supported	Partial	Unsupported
Annotator A	180	131	20	29
Annotator B	180	136	20	24
Consensus gold	156	123	12	21
Verifier sample	180	60	60	60

Table 18: Label distributions in the calibrated 180-claim human audit. Consensus gold contains only the 156 claims on which the two annotators agree.

Annotator A label	B supported	B partial	B unsupported
supported	123	6	2
partial	7	12	1
unsupported	6	2	21

Table 19: Human-human confusion matrix for the two calibrated annotators. Rows are Annotator A labels and columns are Annotator B labels.

Reference	True label	Verifier supported	Verifier partial	Verifier unsupported
Consensus gold	supported	46	47	30
	partial	2	9	1
	unsupported	9	3	9
Annotator A	supported	48	48	35
	partial	3	9	8
	unsupported	9	3	17
Annotator B	supported	47	47	42
	partial	4	10	6
	unsupported	9	3	12

Table 20: Verifier-human confusion matrices grouped by reference label source. Rows are human or consensus labels, and columns are verifier labels. Consensus gold uses the 156 human-agreement cases; Annotator A and Annotator B use all 180 claims.

E Verifier Calibration Details

The main calibration table groups consensus-gold human labels by the verifier's three-way label. For a finer diagnostic, Table 21 groups non-supported verifier predictions by automatic error type, and Table 22 groups the same consensus-gold audit claims by verifier score. Because the audit sample is intentionally balanced by verifier label, these tables should be read as threshold diagnostics rather than estimates of natural corpus prevalence. We also tested whether simple threshold tuning explains the human-verifier gap. The fixed thresholds used in all main experiments obtain 0.410 exact agreement, 0.353 macro-F1, $\kappa = 0.090$, and 0.333 non-supported F1 on the 156 consensus-gold claims. Four-fold cross-validated threshold selection lowers these values to 0.391, 0.335, 0.066, and 0.316, while the best in-sample setting only marginally reaches 0.417 exact agreement and 0.357 macro-F1. Thus, the gap is not primarily a threshold-selection artifact.

Model	Prompt	Faith. ↑	Unsup. ↓	Claims/Expl.	Faith.+Rep. ↑	Unsup.+Rep. ↓	Ret. ↑	Cost/Expl. ↓
Claude Sonnet 4.5	Direct	0.401	0.542	20.18	0.898	0.041	0.191	$0.00674
	CoT	0.345	0.590	34.09	0.870	0.039	0.134	$0.00991
	Structured	0.314	0.514	9.61	0.787	0.063	0.249	$0.00584
	Event-grounded	0.697	0.140	7.89	0.986	0.004	0.797	$0.01104
DeepSeek-V4-Flash	Direct	0.544	0.290	5.44	0.975	0.018	0.705	$0.00051
	CoT	0.429	0.460	16.52	0.974	0.015	0.450	$0.00088
	Structured	0.407	0.424	5.16	0.987	0.011	0.571	$0.00064
	Event-grounded	0.746	0.077	8.88	0.992	0.006	0.899	$0.00044
Gemini-3-Flash	Direct	0.450	0.399	9.96	0.954	0.027	0.536	$0.00379
	CoT	0.319	0.610	25.30	0.966	0.022	0.303	$0.00476
	Structured	0.368	0.448	6.90	0.960	0.028	0.542	$0.00441
	Event-grounded	0.636	0.098	7.84	0.997	0.002	0.871	$0.00554
GPT-4.1-mini	Direct	0.625	0.297	5.55	0.981	0.011	0.700	$0.00044
	CoT	0.351	0.552	30.39	0.967	0.018	0.333	$0.00105
	Structured	0.349	0.474	5.99	0.968	0.021	0.518	$0.00050
	Event-grounded	0.706	0.082	9.71	0.992	0.001	0.881	$0.00140
GPT-5.4-mini	Direct	0.445	0.432	10.20	0.949	0.032	0.515	$0.00150
	CoT	0.357	0.580	28.59	0.952	0.024	0.270	$0.00267
	Structured	0.326	0.479	6.41	0.970	0.029	0.505	$0.00151
	Event-grounded	0.595	0.166	7.99	1.000	0.000	0.804	$0.00260

Table 15: Full synthetic test results on 400 examples across five LLMs. Faith. denotes automatic claim faithfulness before verifier-guided repair. Retention measures the fraction of extracted claims retained after verifier-guided repair. Costs are estimated from provider token usage and the pricing configuration.

Prompt	Runs	Faith. ↑	Unsup. ↓	Claims/Expl.	Top-5 Sup. Cov. ↑	Top-5 Grounded Cov. ↑	Gold Sup. Cov. ↑	Gold Grounded Cov. ↑
Direct	2000	0.459	0.439	10.27	0.243	0.300	0.721	0.797
CoT	2000	0.354	0.567	26.98	0.337	0.391	0.811	0.868
Structured	2000	0.347	0.474	6.82	0.176	0.250	0.532	0.654
Event-grounded	2000	0.679	0.111	8.46	0.368	0.421	0.611	0.675
Event verbalizer	400	0.936	0.040	11.35	0.832	0.832	0.765	0.767

Table 16: Coverage and informativeness diagnostics on the 400-example synthetic test set. Top-5 coverage is the mean fraction of the five highest-scoring extracted events that are mentioned by supported claims, or by supported-or-partial claims for grounded coverage. Gold coverage is the mean fraction of synthetic gold events covered by a compatible claim with interval IoU at least 0.3. Sup. Cov. uses supported claims only, while Grounded Cov. uses supported-or-partial claims. Automatic claim faithfulness and verifier-unsupported rates are recomputed over the coverage-analysis records for within-table comparison; the main prompt-level estimates are reported in Table 1. The deterministic event verbalizer is included as an extractor-aligned no-API reference for event coverage.

Failure mode	Run	Series	Problematic claim	Verifier diagnosis
Unsupported pattern	GPT-4.1-mini *Direct*	test_000003 *upward trend*	*"Around the 50th point, there is a sharp and significant drop... indicating a major change point or anomaly."*	Unsupported: no anomaly or change-point event supports the claim.
Interval mislocalization	DeepSeek-V4-Flash *Event-grounded*	test_000002 *upward trend*	*"A moderate downward trend follows from index 20 to 60..."*	Partial: the phenomenon is plausible, but the span is shifted or too broad.
Causal hallucination	GPT-4.1-mini *CoT*	test_000032 *upward trend*	*"...abruptly interrupted by a significant drop, suggesting a possible external shock or regime change."*	Unsupported: the explanation adds an external cause not present in the event evidence.

Table 17: Representative claim-level failure cases selected from the full synthetic verification outputs. The cases illustrate unsupported pattern introduction, interval mislocalization, and causal over-interpretation.

Verifier error type	N	Human sup.	Human part.	Human unsup.	Non-sup. accept
interval_mislocalization	56	0.786	0.161	0.054	0.214
pattern_hallucination	26	0.654	0.000	0.346	0.346
direction_error	10	1.000	0.000	0.000	0.000
pattern_confusion	7	0.857	0.143	0.000	0.143

Table 21: Human calibration of verifier non-supported error types on the consensus-gold subset.

Score bin	N	Mean score	Human sup.	Human part.	Human unsup.
[0.00,0.25)	40	0.000	0.750	0.025	0.225
[0.25,0.50)	1	0.480	1.000	0.000	0.000
[0.50,0.75)	49	0.699	0.796	0.163	0.041
[0.75,0.90)	16	0.806	0.812	0.125	0.062
[0.90,1.00]	50	0.961	0.800	0.020	0.180

Table 22: Verifier score calibration on consensus-gold human-audit claims. Scores are produced by the rule verifier before human labels are used.

F Real-NAB Human Audit Details

We additionally audit 90 Real-NAB event-grounded claims sampled evenly from the verifier's SUPPORTED, PARTIAL, and UNSUPPORTED labels. Table 8 appears in the main text; the appendix reports the calibration-by-label and confusion-matrix details. This is a single-annotator diagnostic used to test real-world label plausibility, not a replacement for the calibrated two-annotator synthetic audit.

Verifier label	N	Human supp.	Human part.	Human unsupp.	Exact	Gate-consistent
Supported	30	33.3	33.3	33.3	0.333	0.333
Partial	30	36.7	36.7	26.7	0.367	0.633
Unsupported	30	40.0	13.3	46.7	0.467	0.600

Table 23: Real-NAB human-audit calibration by automatic verifier label. The sample is intentionally balanced by verifier label, so these rates diagnose label reliability rather than corpus prevalence.

Human label	Verifier supported	Verifier partial	Verifier unsupported
Supported	10	11	12
Partial	10	11	4
Unsupported	10	8	14

Table 24: Three-way confusion matrix for the Real-NAB human audit. Rows are human labels and columns are automatic verifier labels.

G Additional Synthetic Diagnostics

This appendix contains two supplementary synthetic diagnostics tables. Table 25 reports pattern-by-prompt behavior before verifier-guided repair, and Table 26 reports additional full synthetic metrics including partial rates, claim counts after verifier-guided repair, supported-claim preservation, total cost, and token use.

Pattern	Prompt	Faith.	Unsup.	Part.	Claims/Expl.
Upward trend	Direct	0.519	0.338	0.143	10.07
	CoT	0.374	0.528	0.098	28.11
	Structured	0.417	0.334	0.250	6.80
	Event-grounded	0.691	0.008	0.301	8.70
Downward trend	Direct	0.493	0.308	0.199	10.69
	CoT	0.382	0.475	0.143	26.84
	Structured	0.392	0.303	0.305	6.80
	Event-grounded	0.685	0.023	0.292	8.54
Sharp rise	Direct	0.552	0.364	0.085	10.38
	CoT	0.435	0.477	0.088	26.40
	Structured	0.381	0.393	0.226	6.34
	Event-grounded	0.714	0.047	0.240	8.96
Sharp drop	Direct	0.452	0.447	0.101	10.58
	CoT	0.381	0.537	0.082	25.30
	Structured	0.304	0.508	0.188	6.34
	Event-grounded	0.759	0.059	0.183	8.74
Volatility shift	Direct	0.332	0.529	0.139	9.80
	CoT	0.285	0.616	0.099	27.11
	Structured	0.215	0.612	0.173	7.38
	Event-grounded	0.558	0.254	0.187	8.40
Change point	Direct	0.513	0.417	0.070	10.94
	CoT	0.383	0.552	0.065	26.52
	Structured	0.392	0.431	0.177	6.80
	Event-grounded	0.719	0.063	0.218	7.92
Seasonality	Direct	0.385	0.590	0.025	10.53
	CoT	0.254	0.721	0.025	29.32
	Structured	0.343	0.616	0.041	7.35
	Event-grounded	0.658	0.214	0.128	10.19
Anomaly	Direct	0.410	0.535	0.054	9.14
	CoT	0.351	0.615	0.034	26.22
	Structured	0.345	0.567	0.088	6.72
	Event-grounded	0.636	0.235	0.129	6.26

Table 25: Full pattern-by-prompt diagnostics on the synthetic test split, pooled over the five full-test models. Metrics are computed before repair from claim-level verifier labels.

H Additional Multi-Provider Pilot Diagnostics

The retained 80-example pilot uses the same prompt-level aggregation as the main synthetic setting. Table 27 reports prompt-level averages, and Table 28 reports the corresponding prompt-by-model diagnostics.

Prompt	Faith. ↑	Unsup. ↓	Claims/Expl.	Faith.+Rep. ↑	Unsup.+Rep. ↓	Ret. ↑	Cost/Expl. ↓
Direct	0.490	0.404	10.54	0.956	0.025	0.519	$0.00260
CoT	0.367	0.551	26.69	0.948	0.022	0.306	$0.00387
Structured	0.339	0.480	6.89	0.932	0.036	0.462	$0.00259
Event-grounded	0.663	0.118	8.20	0.989	0.004	0.852	$0.00417

Table 27: Prompt-level pilot results averaged over five LLMs on the 80-example stratified synthetic subset. Faith. denotes automatic claim faithfulness before verifier-guided repair, Ret. measures the fraction of extracted claims retained after verifier-guided repair, and Cost/Expl. is the estimated API cost per generated explanation.

Prompt	Part.	Claims/Expl. +Rep.	Sem. Pres. ↑	Avg. Total Cost ↓	Tokens/Expl.
Direct	0.106	4.41	0.890	$0.2081	1313.7
CoT	0.082	7.51	0.877	$0.3093	1818.2
Structured	0.180	2.99	0.880	$0.2074	1429.8
Event-grounded	0.219	6.99	0.998	$0.3337	2646.8

Table 28: Additional prompt-level pilot diagnostics averaged over five LLMs. Sem. Pres. denotes supported-claim preservation: an originally supported claim is counted as preserved when the repaired explanation contains a supported claim grounded by an overlapping support event. Chronology violations were negligible: all pilot rates were at most 0.002. Avg. Total Cost is the mean cost of an 80-example model–prompt run.

Model	Prompt	Part.	Claims/Expl. +Rep.	Sem. Pres. ↑	Total Cost ↓	Tokens/Expl.
Claude Sonnet 4.5	Direct	0.057	3.86	0.557	$2.6975	895.9
	CoT	0.066	4.58	0.569	$3.9652	1106.4
	Structured	0.172	2.39	0.422	$2.3380	820.8
	Event-grounded	0.163	6.29	0.959	$4.4150	2280.5
DeepSeek-V4-Flash	Direct	0.166	3.83	0.999	$0.2051	2102.4
	CoT	0.111	7.44	0.974	$0.3525	3416.7
	Structured	0.169	2.95	1.000	$0.2578	2564.7
	Event-grounded	0.176	7.98	1.000	$0.1757	2533.0
Gemini-3-Flash	Direct	0.151	5.34	0.941	$1.5148	2075.5
	CoT	0.070	7.67	0.975	$1.9057	2400.4
	Structured	0.184	3.74	0.994	$1.7646	2270.4
	Event-grounded	0.266	6.83	1.000	$2.2153	4152.8
GPT-4.1-mini	Direct	0.078	3.89	1.000	$0.1754	682.8
	CoT	0.097	10.11	0.965	$0.4206	1062.9
	Structured	0.176	3.10	0.999	$0.1990	707.6
	Event-grounded	0.212	8.55	1.000	$0.5582	2298.8
GPT-5.4-mini	Direct	0.123	5.25	0.952	$0.6001	786.6
	CoT	0.063	7.73	0.869	$1.0674	1043.0
	Structured	0.195	3.24	1.000	$0.6049	776.0
	Event-grounded	0.240	6.42	1.000	$1.0419	2163.1

Table 26: Additional full synthetic diagnostics on 400 examples. Sem. Pres. denotes supported-claim preservation: an originally supported claim is counted as preserved when the repaired explanation contains a supported claim grounded by an overlapping support event. Chronology violations were negligible in this setting: all model–prompt rates were at most 0.002.

I Reproducibility Artifacts

The anonymized release will include the benchmark data, extracted event graphs, prompt templates, model generations, automatic verifier outputs, verifier-guided repair outputs, aggregate metric files, and scripts needed to regenerate the tables in the paper. The main synthetic evaluation uses the 400-example synthetic test split; the Real-NAB localization diagnostic uses the 200-window split described in Appendix A.1.

Human-audit materials are handled separately from the public benchmark release. The paper reports aggregated annotation statistics, consensus labels, calibration summaries, and confusion matrices; raw annotator identifiers and private annotation files are not required for interpreting the results and are not listed here. If accepted, we will release de-identified audit records and analysis scripts to the extent permitted by the annotators' consent and the venue's artifact policy.