

000 TIMESERIESEXAMAGENT: CREATING TIME SERIES 001 002 REASONING BENCHMARKS AT SCALE 003 004

005 **Anonymous authors**

006 Paper under double-blind review

007 008 ABSTRACT 009

011 Large Language Models (LLMs) have shown promising performance in time series
012 modeling tasks, but do they truly understand time series data? While multiple
013 benchmarks have been proposed to answer this fundamental question, most are
014 manually curated and focus on narrow domains or specific skill sets. To address
015 this limitation, we propose scalable methods for creating comprehensive time
016 series reasoning benchmarks that combine the flexibility of templates with the
017 creativity of LLM agents. We first develop *TimeSeriesExam*, a multiple-choice
018 benchmark using synthetic time series to evaluate LLMs across five core reasoning
019 categories: *pattern recognition*, *noise understanding*, *similarity analysis*, *anomaly*
020 *detection*, and *causality*. We then scale our approach by automatically generating
021 benchmarks from real-world datasets spanning healthcare, finance and weather
022 domains. Through multi-dimensional quality evaluation, we demonstrate that our
023 automatically generated benchmarks achieve diversity comparable to manually cu-
024 rated alternatives. However, our experiments reveal that LLM performance remains
025 limited in both abstract time series reasoning and domain-specific applications,
026 highlighting ongoing challenges in enabling effective time series understanding in
027 these models.

028 1 INTRODUCTION 029

030 Recent studies have successfully applied Large Language Models (LLMs) to time series analysis tasks
031 including forecasting, anomaly detection, and classification (1; 13; 8; 19; 50; 51). These promising
032 results raise a fundamental question: do LLMs possess genuine reasoning capabilities about the
033 abstract concepts underlying time series data? Can they recognize trends, distinguish signal from
034 noise, or understand causal relationships without relying on domain-specific shortcuts? Existing
035 benchmarks designed to evaluate such capabilities face significant limitations— they are manually
036 curated, expensive to extend, and typically focus on narrow domains or specific skills (44; 33). This
037 creates a practical barrier for researchers and practitioners who need comprehensive evaluation tools
038 but lack the resources to construct domain-specific benchmarks for their datasets.

039 To address this gap, we begin with a simple proof of concept, and introduce *TimeSeriesExam*,
040 a controlled benchmark which uses *synthetic* time series to evaluate LLM reasoning across five
041 core categories: *pattern recognition*, *noise understanding*, *similarity analysis*, *anomaly detection*,
042 and *causality*. Initial results reveal two key insights: first, templated generation provides a viable
043 mechanism for creating diverse and systematic evaluation questions; second, LLMs continue to
044 struggle with abstract time series reasoning, even in these controlled settings.

045 Building on these insights, we tackle a broader practical challenge: how can we create benchmarks
046 that reflect the domain-specific reasoning required in real applications, such as diagnosing arrhythmias
047 from ECG signals or evaluating volatility regimes in financial markets? The key obstacle is that
048 domain experts lack the time to manually construct comprehensive benchmarks, making expert-
049 driven approaches impractical at scale. Inspired by recent advances in agent-based benchmark
050 construction (6; 15), we address this challenge by combining our controlled synthetic benchmark
051 with an extensible, agentic framework for automated domain-specific evaluation.

052 Experiments on multiple datasets spanning diverse domains reveal that (1) LLM performance varies
053 substantially across domains, and (2) even state-of-the-art models struggle with complex reasoning
tasks requiring integration of domain expertise and time series understanding.

054 Our contributions are as follows:
 055

056 • **Foundational benchmark.** We introduce `TimeSeriesExam`, a controlled evaluation
 057 framework that systematically assesses whether LLMs understand core time series concepts.
 058 This templated approach provides valuable insights into LLMs’ reasoning capabilities while
 059 enabling scalable question generation, though it remains limited to domain-agnostic skills
 060 on synthetic data.

061 • **Scalable framework.** Building on this foundation, we propose `TimeSeriesExamAgent`,
 062 which combines the systematic nature of templates with the adaptability of LLM agents.
 063 Given any domain-specific dataset, `TimeSeriesExamAgent` automatically generates cus-
 064 tomized time series reasoning benchmarks at scale, integrating multi-perspective verification
 065 and optional human-in-the-loop refinement to ensure question quality and diversity.

066 • **Multi-dimensional evaluation.** We validate our approach across five datasets spanning four
 067 domains and demonstrate its effectiveness for domain-specific fine-tuning. Our automatically
 068 generated questions achieve diversity comparable to human-curated benchmarks.

069 2 RELATED WORK

070 **Synthetic Time Series Generation** The generation of synthetic time series with controlled behav-
 071 iors, such as trends and cyclic patterns, is fundamental for constructing scalable reasoning benchmarks.
 072 A common approach involves sampling from diverse random processes (18), such as Autoregressive
 073 Processes, which offer variability but lack control over specific patterns like cyclic behavior. To ad-
 074 dress this, (48) proposed a decomposition-based method, generating desired patterns by incorporating
 075 cyclic components into an additive model on top of random processes. More recent frameworks, such
 076 as (16) and (2), also leverage synthetic data generation for model training and evaluation. TabPFN
 077 constructs synthetic regression and classification tasks from random function priors, while Chronos
 078 employs large-scale transformer-based time series generation to capture realistic temporal dynamics.
 079 We build upon these studies, through the design of the `TimeSeriesExam` benchmark, by having a more
 080 diverse set of random processes and patterns, incorporating not only additive composition methods
 081 but also multiplicative and other forms of composition.

082 **Domain Specific Time Series Reasoning Benchmarks** The task of creating domain-specific time
 083 series reasoning benchmarks is challenging. Current domain-specific benchmarks usually have
 084 limited scope and poor extensibility, since their curation often relies on templates or expert annotation.
 085 For instance, ECG-QA (33) and ECG-Expert-QA (44) focus on ECG interpretation, while EngineMT-
 086 QA (45) targets industrial settings. Automatic benchmark generation is a scalable alternative, but the
 087 quality and diversity of automatically generated questions is unclear. Without extensive verification,
 088 LLM-generated questions often require heavy manual curation (21; 25), which is both difficult and
 089 time-consuming, undermining the primary advantage of automation.

Title	Multi-domain	Curation	# Samples	Skill type		
				P	R	PS
Time-MQA (21)	✓	✓	200,000	✓	✓	✗
Time-MMD (25)	✓	✗	17,113	✓	✗	✗
MT-Bench (9)	✓	✗	22,000	✓	✓	✗
ECG-QA (33)	✗	✗	414,348	✓	✓	✓
Context-is-key (47)	✓	✗	71	✓	✓	✗
TimeSeriesExamAgent (ours)	✓	✓	3000+	✓	✓	✓

100 Table 1: Overview of time-series and multimodal datasets with curation and skill types (P—Prediction,
 101 R—Reasoning, PS—Practical skills). Prediction refers to supervised tasks such as forecasting
 102 or classification. Reasoning involves higher-level interpretation of time series signals (e.g., trend
 103 recognition). Practical skills extend reasoning into domain-specific contexts (e.g., classifying volatility
 104 regimes in finance). "+" represents that we can generate any number of samples, but we have already
 105 generated 3,000 of them.

106
 107 ⁰The CiK benchmark contains 71 tasks, with the number of samples treated as a configurable hyperparameter.

108 **Agents for benchmark creation** LLM agents are autonomous systems which observe an environment, use LLMs to reason, and act towards achieving a well-defined goal. Recent work has
 109 shown the promise of using agents for creating benchmarks automatically. Most solutions adopt a
 110 multi-agent pipeline with planning, generation, validation, and evaluation modules (6). For example,
 111 (42) integrates exploratory evaluation using reinforcement learning, while (6) takes a description of
 112 a natural language task as input. However, most of these approaches are not tailored to time series
 113 and struggle to generate questions conditioned on numeric data. A recent solution incorporates time
 114 series, but is limited to a single-step design and lacks extensive verification (29).
 115

116 **LLM-as-a-Judge Evaluation** LLM judges have enabled the research community to scale evaluations and to assess problems that are inherently difficult to capture with conventional metrics.
 117 However, LLM-as-a-judge also introduces challenges related to bias. We address this in two ways.
 118 First, we adopt methods such as G-Eval (26), which provide probabilistic results and thus improve
 119 both reliability and interpretability. Second, we account for intra-model bias, which is the tendency
 120 of a single LLM to exhibit systematic preferences, by drawing on the idea of panel-based evaluation
 121 (39). Specifically, we aggregate judgments from multiple LLMs, which reduces the influence of
 122 any individual model’s bias and yields more stable scores.
 123

125 3 TIME SERIES EXAM AGENT

127 3.1 PROOF-OF-CONCEPT: BUILDING SCALABLE BENCHMARKS USING TEMPLATES

129 To begin, we investigate LLMs’ understanding of fundamental time series concepts in a controlled
 130 experimental setting by introducing a manually curated, configurable benchmark that we call
 131 TimeSeriesExam.

132 To illustrate our approach, we present a proof-of-concept(PoC) showing how scalable benchmarks
 133 for evaluating LLMs’ time series reasoning can be built from configurable templates. The aim is to
 134 demonstrate that template-based design enables systematic generation of diverse, controlled “exams”
 135 that probe specific reasoning skills. Our hypothesis is that once a small set of well-designed templates
 136 exists, new benchmark items can be generated automatically by varying parameters and contexts.
 137 To test this, we introduce TimeSeriesExam, a curated benchmark of fundamental time series
 138 tasks, in which we make two simplifying assumptions: (1) templates are created manually, and (2)
 139 evaluations are conducted in controlled synthetic settings where data properties are fully known.

140 **Composition.** TimeSeriesExam systematically assesses whether LLMs **understand** basic time
 141 series patterns such as trends and seasonality (*pattern recognition*), the concept of noise and other
 142 time series concepts in the presence of noise (*noise understanding*). It also evaluates LLMs on
 143 three different **reasoning** tasks: identifying abrupt deviation from “normal” behavior (12) (*anomaly*)

Category	Subcategory	Example question
Pattern Recognition	Trend	What is the most likely linear trend coefficient of the given time series?
	Cyclic	The given time series has sine wave pattern. How does its amplitude change from the beginning to the end?
	Stationarity	Is the given time series likely to be stationary after removing the cycle component?
	Regime Switching	Based on the given time series, how many different regimes are there?
	Statistical properties	Is the mean stable over time in the given time series?
	Random processes	Does the following time series exhibit a mean reversion property?
Noise Understanding	White Noise	Is the given time series a white noise process?
	Random Walk	Is the given time series likely to be a random walk process?
	Signal / Noise Ratio	You are given two time series with the same underlying pattern but different noise level. Which time series has higher magnitude of noise?
Anomaly Detection		The following time series has two types of anomalies appearing at different time points. What are the likely types of these anomalies?
Comparative Analysis	Shape	Despite the noise, do the given two time series have similar patterns?
	Distributional	You are given two time series which are generated using a random walk. Are they likely to have the same variance?
Causality Analysis	Granger Causality	Is there Granger causality between the two time series?

159 Table 2: Example template questions for different reasoning tasks. Each subcategory covers a specific
 160 aspect of time series understanding, guiding the model to reason about comparative, anomalies, and
 161 causal relationships.

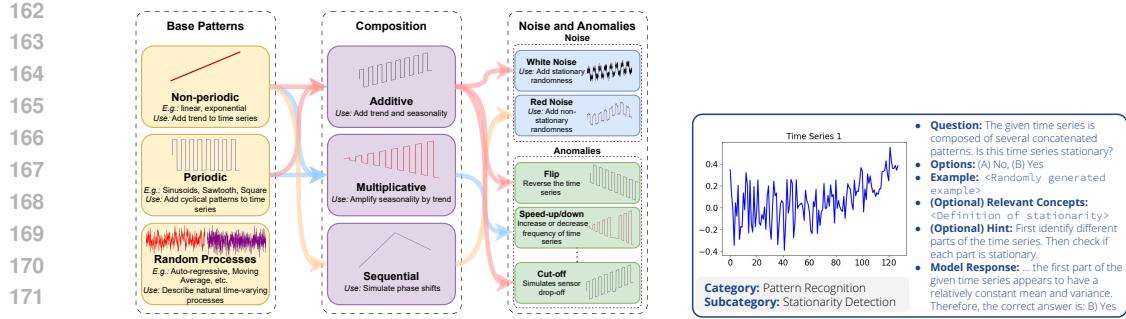


Figure 1: (Left) Time Series Curation Pipeline: The composition model generates controlled synthetic time series step-by-step. The pipeline enables diversity by combining different components to create numerous synthetic time series with varying properties. (Right) Each template evaluates a specific category, and includes a question, list of options, example question and answer pair for in-context learning, and optionally a hint and descriptions of complicated technical terms. Here, GPT-4○ showcases its ability to transfer visual understanding and time series concepts into effective reasoning.

*detection), comparing and contrasting statistical properties of 2 time series (*comparative reasoning*), reasoning about causality, specifically Granger Causality (14) (*causality*). As shown in Table 2, each category is further divided into sub-categories that represent more specific concepts within the broader category.*

Question Templates. The `TimeSeriesExam` comprises over 100 unique templates, carefully curated in collaboration with time series experts and cross-verified for accuracy, that can be used to generate any number of random questions. Each template (Fig. 1)(Right) evaluates a specific (sub-)category (e.g., *pattern recognition*), and comprises of a question (e.g., “Is this time series stationary?”), a list of options (e.g., “(A) Yes, (B) No”), and an example question and answer pair for in-context learning. Each template comes with a *hint* which breaks down complex questions into simpler steps and textual descriptions of complicated technical terms. By incorporating these relevant concepts, we can isolate an LLM’s ability to understand time series concepts (e.g., whether the mean and variance remain constant) from its understanding of complex technical jargon (e.g., stationarity). Each option (e.g. “(A) Yes”) is linked to a synthetic time series generator (Fig. 1)(Left) that produces a random time series as if the current option were true (e.g., a random stationary time series). This allows us to generate random but accurate time series at scale.

Generating Questions. We generate different questions from the same template by systematically varying the correct option and producing synthetic time series conditioned on the template and the correct option pair. Our simple and scalable approach, illustrated in Fig. 1(Left), involves sampling a small number of base patterns from a predefined pool and combining them using a composition function. Base patterns can be periodic (e.g., sine function), non-periodic (e.g., linear increasing function), or random time-varying processes (e.g., AR process). Depending on the template’s nature, the final step adds additive noise or anomalies using the anomaly injection process described in (12).

Improving Questions Iteratively. We use Item Response Theory (IRT) (27) to achieve finer grained control over the quality of randomly generated questions included in the `TimeSeriesExam`. IRT is a statistical framework that models the relationship between an individual’s (or LLM’s) latent trait (e.g., knowledge, ability) and their responses to a set of items (e.g., questions on a test). It is a valuable tool in exam development as it helps to identify weak exam items, ensures consistent scoring across different versions of the exam, and also allows tailoring the testing experience to the LLM’s abilities.

Our primary objective is to design a `TimeSeriesExam` where each question can maximally distinguish the abilities of candidate LLMs. We use the two-parameter logistic (2PL) model for this. Formally, for LLM j with ability θ_j , and question i with difficulty b_i , discrimination ability a_i , the 2PL model defines the probability of a correct response as: $\mathbb{P}(r_{ij} = 1 | a_i, b_i, \theta_j) = 1 / (1 + e^{-a_i(\theta_j - b_i)})$

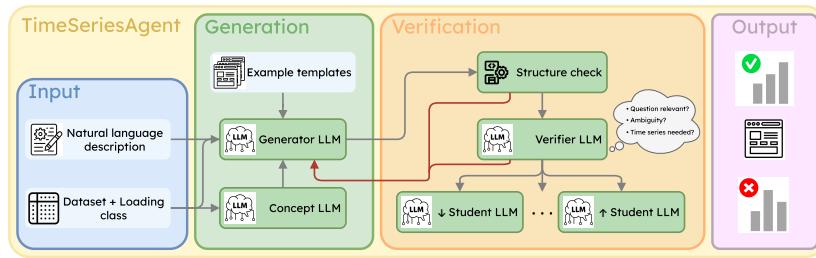
216 Each `TimeSeriesExam` typically undergoes 1–3 rounds of iterative refinement. In each round,
 217 all candidate models take the exam. Based on their responses, we fit the parameters of Equation 3.1
 218 using maximum likelihood estimation (MLE). Then, we drop $X\%$ of samples with the lowest sum of
 219 difficulty and discrimination ability. Finally, we randomly re-generate questions from the dropped
 220 templates. This iterative process is detailed in Algorithm 1 and the hyper-parameters of the fitting
 221 process are provided in App. B.2. We demonstrate the increase in the differentiability parameter
 222 across the iteration rounds and provide an analysis of the trajectory in App. B.3

223 While detailed empirical results are presented in Section A.1, the PoC establishes two takeaways: (i)
 224 template-based generation yields diverse, controllable items across core reasoning categories, and
 225 (ii) modern VLMs still struggle on higher-order time series reasoning, which motivates a scalable
 226 pipeline for real datasets with minimal expert time.

227

228 3.2 TIMESERIESEXAMAGENT: A SCALABLE DOMAIN-AGNOSTIC BENCHMARK CREATION 229 TOOL

230



240 Figure 2: `TimeSeriesExamAgent` architecture. The user provides exam-making instructions and
 241 a custom dataset with minimal loading code. Agent outputs question templates – Python functions
 242 generated by a generator LLM and filtered through three progressive stages of verification (syntax
 243 and output format check, validation by LLM judge, capability-aligned filtering). Arrows denote data
 244 flow, red ones show direction for rejected templates.

245 Building on the PoC in the previous subsection that surfaced persistent gaps in LLMs’ time series rea-
 246 soning, we now focus on the practical need to evaluate models on domain-specific datasets at scale and
 247 with minimal expert effort. Domain experts are often interested in assessing LLMs on specialized rea-
 248 soning capabilities rather than on broad, preexisting benchmarks (e.g., evaluating anomaly detection
 249 in ECG data versus generic healthcare reasoning). To this end, they typically possess domain-specific
 250 datasets and wish to construct benchmarks that reflect the reasoning challenges within these datasets.
 251 However, building such benchmarks manually is labor-intensive, as we have demonstrated with
 252 `TimeSeriesExam`. To address this challenge, we propose `TimeSeriesExamAgent`, a multi-
 253 agent framework that combines planning, generation, and verification to enable automatic benchmark
 254 construction on real datasets while minimizing expert time.

255 **Setup** An overview of the agentic framework is shown in Fig. 2. The Generation Agent takes as
 256 input a description of the natural language task T and a data set D . The description T may include
 257 user guidelines for generation, contextual information about the dataset, or other relevant instructions.
 258 For convenience, we denote each sample in D as (x_i, z_i) , where $x_i \in \mathbb{R}^{n \times d}$ is a time series with n
 259 observations and d variables, and z_i is an auxiliary array containing metadata or labels related to the
 260 series. The user provides a dataset class D that supports basic operations such as querying the i -th
 261 sample.

263 **Generation** Motivated by `TimeSeriesExam`, we generate question templates instead of samples
 264 directly, as shown in Fig. 3. Templates offer two advantages: they are scalable, and their abstraction
 265 adds an extra layer of robustness. By relying on structured, rule-based generation rather than manual
 266 inputs, they reduce the chance of human errors or inconsistencies. Our generator LLM produces
 267 a predefined number of templates, each implemented as a Python function. A template contains
 268 a formatted string for the question and options, together with parameters that control how many
 269 questions to generate. For each question, the template samples a pair (x_i, z_i) from the dataset D and
 applies a rule-based calculation to determine the correct answer from the time series. For example,

270 in a trend-detection template, the function computes the linear trend coefficient of x_i and selects
 271 “Yes, there is a linear trend” if the coefficient exceeds a specified threshold. In addition to such
 272 signal-derived logic, templates can also utilize the auxiliary property z_i , effectively transforming
 273 classification problems into question–answer form. For instance, if an ECG series in the dataset is
 274 labeled as exhibiting atrial fibrillation, the template can present this label as one of the multiple-choice
 275 options. Each generated sample consists of the question, its options, the correct answer, and one
 276 or more associated time series represented as numerical values. We provide a breakdown of the
 277 Generation Agent and its prompt in App. C. An example template is also provided.
 278

279 **Verification** We observe
 280 that LLM-based generation
 281 frequently produces errors or
 282 irrelevant outputs, motivating
 283 the need for a structured veri-
 284 fication process. We propose a
 285 multistage verification process
 286 to check the accuracy and
 287 relevance of each template. If a
 288 template fails at any stage, it is
 289 returned to the generation agent
 290 with feedback. The generation
 291 is iterative with a maximum of
 292 three attempts, after which the
 293 ongoing template is discarded to
 294 avoid excessive context length
 295 and cost from repeated failures.
 296

297 **Structure verification** We check whether the generated template can be executed successfully. We
 298 execute the generated template $k = 3$ times; if there are any failures, the error message is returned as
 299 feedback.
 300

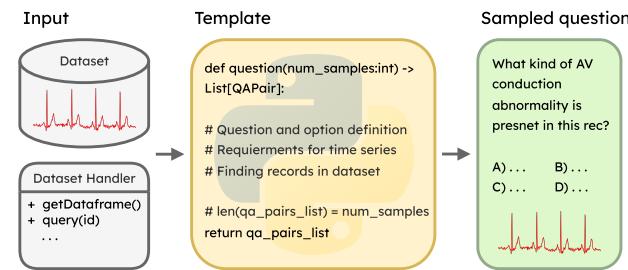
301 **Content verification** Certain aspects of quality control are particularly well-suited for using LLM-as-
 302 a-judge evaluation. For example, verifying that a question is grammatically correct, free of ambiguity
 303 or bias, and genuinely answerable from the accompanying time series can be effectively handled
 304 by an LLM. To this end, we use an LLM verifier to assess the validity of each template. We use a
 305 binary scheme: a template must pass all categories to be accepted. Any failure triggers rejection and
 306 regeneration, ensuring robustness. Details are in App. D.1.
 307

308 **Capability-Aligned Filtering** Inspired by `TimeSeriesExam`, which leverages IRT to enhance
 309 differentiability across questions, we adopted a similar but localized framework that operates at
 310 the level of each template. To detect templates that generate overly simple or irrelevant exams, we
 311 evaluate them using a set of test-taking LLMs with varying capabilities. This approach is supported
 312 by educational theory, particularly the expertise reversal effect (20). A template is discarded if
 313 weaker LLMs achieve higher average accuracy than stronger models, as this typically indicates
 314 that the template is flawed or noisy rather than genuinely discriminative. Templates are retained if
 315 performance scales with model capability, or if all models perform poorly, since such questions may
 316 still capture genuine difficulty. We provide hyperparameters in App. D.5 and other design specifics in
 317 App. D.1.1
 318

319 4 EXPERIMENTAL SETUP, RESULTS AND ANALYSIS

320 4.1 STATE-OF-THE-ART LLMs STRUGGLE ON EXAMS GENERATED BY 321 `TIMESERIESEXAMAGENT`

322 First, we generate one exam for each of the five real world datasets: PTB-XL (40), MIT-BIH (31),
 323 MIMIC-IV Waveform (30), yahoo finance stock dataset (37), and WeatherBench 2 (36). In total, we
 324 have 209 samples for YFinance, 197 samples for MIT-BIH, 151 samples for PTB-XL, 205 samples
 325 for MIMIC-IV Waveform, and 95 samples for WeatherBench 2. We sample 4 or 5 instances per
 326 template. Thus, the difference in the number of generated samples is a result of the template filtering
 327 mechanism above.



328 **Figure 3: Question generation process:** With information about
 329 datasets, `TimeSeriesExamAgent` generates question templates
 330 in a form of Python functions. The created function can be called
 331 to get arbitrary number of question samples.
 332

324

325

Model	Dataset					Average
	MIT-BIH	PTB-XL	MIMIC-IV W	YFinance	WeatherBench2	
random guess	0.25	0.25	0.25	0.25	0.25	0.25
gpt-4o (17)	0.416	0.424	0.385	0.586	0.389	0.440
o3-mini (35)	0.442	0.477	0.356	0.555	0.379	0.442
Qwen2.5-VL-Instruct (3)	0.411	0.490	0.439	0.572	0.368	0.456
Gemma-3-27b-it (38)	0.497	0.517	0.370	0.534	0.232	0.430
GPT-5	0.533	0.450	0.424	0.617	0.547	0.515
Gemini-2.5-Pro	0.614	0.457	0.400	0.624	0.453	0.510

331

332

Table 3: Comparative performance of four vision–language models across medical (MIT-BIH, PTB-XL, MIMIC-IV Waveform (MIMIC-IV W)), financial (YFinance), and meteorological (WeatherBench 2) time series datasets. The results highlight dataset-specific strengths; nonetheless, all models achieve less than 55 mean accuracy, underscoring the difficulty of time series reasoning for current VLMs. The evaluation protocol is provided in App. E.3

333

334

We select candidate models to cover a diverse range of performance levels, as indicated by the Open-VLM Leaderboard (11). In Table 3, we find that while general-purpose multimodal models such as gpt-4o perform well on finance-related questions, their performance is weaker on healthcare benchmarks. This contrast could suggest that the general reasoning ability does not always transfer across domains, particularly when tasks require domain-specific expertise or fine-grained interpretation of physiological signals. **In App. H, we highlight two types of failure modes by studying responses from VLMs. Perception:** As evidenced by our ablation on input resolution (DPI) or modality (text vs. vision), the best way to receive data depends on the specific question. **Compositional Reasoning:** Models do not fail not on simple recognition, but on problems that require multi-step reasoning.

348

349

4.2 TIME SERIES EXAMAGENT GENERATES QUESTIONS WITH DIVERSITY COMPARABLE TO HUMAN-CURATED BENCHMARKS

350

351

We evaluate the diversity of questions generated by our framework against ECG-QA (33), a template-based benchmark built on PTB-XL. Our aim is to show that TimeSeriesExamAgent achieves comparable variety without manual template design. For each benchmark, we randomly sampled 50 questions and computed pairwise embedding distances. Embeddings were extracted using Qwen/Qwen3-Embedding-8B¹, the top open-source model on the Hugging Face MTEB leaderboard².

357

358

359

Benchmark Dataset	Mean \pm Std	
	Embedding	Normalized Levenshtein
ECG-QA	0.207 ± 0.079	0.519 ± 0.157
TimeSeriesExamAgent (ours)	0.301 ± 0.070	0.542 ± 0.039

362

363

Table 4: Diversity of questions measured by embedding and normalized Levenshtein distances. Higher values indicate greater variability in phrasing.

366

367

As shown in Table 4, our framework achieves a level of diversity that is broadly comparable to human-curated benchmarks. This suggests that it can capture a range of question formulations without relying on handcrafted templates, which may help its scalability to other domains. For completeness, we include a visualization in App. E.2 to further illustrate this observation.

371

372

We employed an LLM-as-a-jury approach using G-Eval, where a panel of models (Gemini-1.5-Pro, GPT-3.5-Turbo, and Qwen2.5-VL-72B-Instruct) evaluated the quality of each question. To ensure cost efficiency, we selected relatively weaker models, as prior work shows this setup can maintain evaluation quality while mitigating intra-model bias (39). Each model independently assigned a score from 1 to 10 based on four criteria. The aggregated results, reported in Table 5, show that

376

377

¹<https://huggingface.co/Qwen/Qwen3-Embedding-8B>

²<https://huggingface.co/spaces/mteb/leaderboard>

378 Table 5: **Combined Quality Evaluation.** Scores (1–10) averaged across four criteria. Rows are
 379 grouped by their original source tables.
 380

Dataset	Mean Result			
	Specificity	Unambiguity	Domain Relevance	Answerability
Finance Domain				
FinMME	8.10	7.59	6.62	6.95
MTBench	6.88	6.11	8.35	7.29
TimeSeriesExamAgent (ours)	8.29	7.24	8.89	8.57
Medicine Domain				
ECG-QA	5.60	5.77	8.17	8.47
TimeSeriesExamAgent (ours)	8.43	8.40	9.00	9.10

389
 390 TimeSeriesExamAgent outperforms ECG-QA across all dimensions, particularly in specificity
 391 and answerability. This indicates that our framework generates precise, well-grounded, and domain-
 392 appropriate questions.
 393

394 4.3 LLMs TRAINED ON OUR GENERATED SAMPLES EXHIBIT TRANSFERABLE REASONING
 395 SKILLS ON ESTABLISHED DATASETS
 396

397 Another way to assess the value of
 398 TimeSeriesExamAgent is to test
 399 whether its generated data supports
 400 transfer learning. We finetuned VLM
 401 Qwen2.5-VL-3B-Instruct.
 402 We first generated 2000 training
 403 samples using TimeSeriesExam
 404 based on the PTB-XL dataset, while
 405 testing was conducted on 12000
 406 randomly selected samples from
 407 the ECG-QA (34) test split using
 408 MIMIC-IV data. Training parameters
 409 are provided in App. F. **To isolate the**
 410 **effect of structural learning from actual gain in reasoning capability, we add a row Fine-tuned-
 411 confounded where we train the LLM based on the generated template from Finance and Weather
 412 domain with same gradient steps.**

413 Table 6 shows clear gains under the strict accuracy metric: the Base model achieves 21.8%, while
 414 fine-tuning on structurally similar but domain-irrelevant exam lifts accuracy to 39.7%. This confirms
 415 the model learn from instruction following and structural similarity of the MCQs. Training on
 416 TimeSeriesExamAgent-generated ECG exams lifts accuracy further to 47.0%, which gives
 417 216% relative improvement from the base model. This confirms the model also gained ECG
 418 reasoning capability. The fine-tuned model also surpasses the Random baseline (34.9%), indicating
 419 that agent-generated questions provide genuinely useful supervision rather than superficial patterning.
 420 Overall, these results suggest that synthetic, agent-curated exams can improve decision quality.

421 4.4 CHOICE OF LLM DOES NOT INTRODUCE BIAS FOR TIMESERIESEXAMAGENT

422 Although we only use the LLM-as-a-Jury system for linguistic properties check, we conducted
 423 experiment to confirm the consistency of juries with regard to choice of LLMs used, so that our
 424 pipeline do not subject bias from a specific set of LLM. We generate exams using default generator
 425 LLM Claude-4-sonnet, and evaluated using 3-model juries drawn from a fixed pool of LLMs
 426 (Gemini-2.0, Deepseek-V3.2, GPT-3.5 Turbo, Qwen-2.5-VL, LLama-3.3). We plotted inter-jury
 427 Pearson Correlation and Cohen’s κ in App. I and observed that scores from most juries were
 428 moderately to highly correlated (Cohen’s $\kappa \geq 0.5$). This confirms the consistency among different
 429 subset of LLMs and that our pipeline do not subject to bias arising from using a specific set of LLMs.

430 To confirm the choice of generator LLM do not introduce additional bias to the agentic pipeline,
 431 We fixed the jury to the default configuration (Gemini-2.0, GPT-3.5-Turbo, and Qwen2.5-VL-72B-
 432 Instruct), and compared two different generator LLMs: Claude 4 and DeepSeek V3.2. We

432 specifically picked DeepSeek because it is disjoint from both the Jury and Verifier model families
 433 (Qwen, Gemini, GPT), ensuring strict independence. Unlike the previous experiment, changing the
 434 generator alters the specific questions produced. Therefore, we fixed the underlying data source to
 435 **MIT-BIH** and evaluated the statistical distribution (mean \pm standard deviation) of the jury scores
 436 across the generated artifacts. The results are presented in Table 9. While DeepSeek V3.2 exhibits
 437 slightly higher raw means, **the results are statistically comparable, with the scores of both models**
 438 **falling within one standard deviation of each other across all categories.** This confirms that our
 439 generation pipeline is robust to the choice of state-of-the-art generator LLM.

440 5 DISCUSSION, OPEN QUESTIONS AND OPPORTUNITIES

441 **Reliance on Expert-Generated Prompts** A key limitation of TimeSeriesExamAgent is that
 442 the quality of the generated exams ultimately depends on the user instruction and coverage of the
 443 underlying time series dataset. For example, if important clinical instructions for the healthcare data
 444 set are absent, the resulting questions may not adequately capture the reasoning challenges faced
 445 in practice. In an offline sessions with cardiologists, we observed that when clinicians contributed
 446 targeted feedback during the prompt design stage, the resulting exams were consistently judged
 447 as more clinically valid and useful (See App. G). This highlights the importance of structured
 448 collaboration between automated systems and human experts, especially in high-stakes domains such
 449 as healthcare. We aim to provide a quantitative analysis in the rebuttal phase.

450 **Demand for human-in-the-loop evaluation.** Building on the previous observation, we integrated
 451 optional human-in-the-loop modules into TimeSeriesExamAgent to facilitate a more practical
 452 deployment. These modules allow domain experts to refine templates, validate generated questions,
 453 and iteratively improve exam quality. Although we received encouraging anecdotal feedback from
 454 clinicians and practitioners who interacted with the system, the influence of such human feedback
 455 pipelines could not be systematically tested within the scope of this study. A formal evaluation of
 456 how human involvement impacts benchmark validity and downstream model assessment remains an
 457 important direction for future work and the community.

458 **Limited Evaluation Mode.** In the current framework, questions are mainly evaluated by providing
 459 the time series as image input. In App. H.1, we provide a few case studies to highlight how
 460 input modality of time-series could impact model answers. These studies highlight the need for an
 461 intelligent decision-making tools, such as an agentic framework, to dynamically choose the most
 462 suitable representation. There is growing interest in agentic frameworks for time series analysis tasks
 463 (7; 49). Our benchmark provides a natural testbed for such systems, since many of the generated
 464 questions require multi-step reasoning, or direct computation over numeric data. Enabling agentic AI
 465 systems to autonomously write and execute code in order to answer our benchmark questions would
 466 provide valuable insights into their reasoning fidelity and robustness.

467 **Natural extension beyond time series.** Although we focus on time series data, the underlying
 468 framework is not inherently restricted to this modality. Our design only assumes access to structured
 469 data and domain-specific prompts, making it extensible to other settings such as images, tables, or
 470 even multimodal combinations of signals and text. We chose time series as a starting point because it
 471 is a highly structured domain with well-established industrial applications.

472 6 CONCLUSION

473 This work first examined whether LLMs can reason about fundamental time-series concepts. To
 474 address these questions, we introduced TimeSeriesExam, a controlled benchmark for probing
 475 conceptual understanding, and TimeSeriesExamAgent, a scalable framework that enables prac-
 476 titioners to generate customized benchmarks from their own data. Our experiments show that while
 477 LLMs capture some surface-level patterns, they continue to struggle with more complex reasoning
 478 such as anomaly detection. At the same time, benchmarks generated by TimeSeriesExamAgent
 479 match or exceed the diversity and quality of human-curated datasets, and can even provide useful
 480 finetuning signals for downstream tasks. These results suggest that automated, agentic benchmark
 481 construction can help make evaluation more adaptive and domain-relevant.

486 REPRODUCIBILITY STATEMENT
487488 We release all evaluated datasets at [https://anonymous.4open.science/r/](https://anonymous.4open.science/r/TimeSeriesExamAgentSubmission-C9C5)
489 TimeSeriesExamAgentSubmission-C9C5. The evaluation protocol is described in
490 App. E.3, and the appendix further provides implementation details of the pipeline, including prompts
491 and configuration settings.493 REFERENCES
494

- [1] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.
- [2] Afshin Ansari, Amol Bhattacharya, Abhijit Kulkarni, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.
- [3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [4] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [6] Natasha Butt, Varun Chandrasekaran, Neel Joshi, Besmira Nushi, and Vidhisha Balachandran. Benchagents: Automated benchmark creation with agent interaction. *arXiv preprint arXiv:2410.22584*, 2024.
- [7] Yifu Cai, Xinyu Li, Mononito Goswami, Michał Wiliński, Gus Welter, and Artur Dubrawski. Timeseriesgym: A scalable benchmark for (time series) machine learning engineering agents. *arXiv preprint arXiv:2505.13291*, 2025.
- [8] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.
- [9] Jialin Chen, Aosong Feng, Ziyu Zhao, Juan Garza, Gaukhar Nurbek, Cheng Qin, Ali Maatouk, Leandros Tassiulas, Yifeng Gao, and Rex Ying. Mtbench: A multimodal time series benchmark for temporal reasoning and question answering. *arXiv preprint arXiv:2503.16858*, 2025.
- [10] Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.
- [11] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11198–11201, 2024.
- [12] Mononito Goswami, Cristian Challu, Laurent Callot, Lenon Minorics, and Andrey Kan. Unsupervised model selection for time-series anomaly detection. *arXiv preprint arXiv:2210.01078*, 2022.
- [13] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

540 [14] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral
541 methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

542

543 [15] Gauthier Guinet, Behrooz Omidvar-Tehrani, Anoop Deoras, and Laurent Callot. Automated
544 evaluation of retrieval-augmented language models with task-specific exam generation. *arXiv*
545 preprint arXiv:2405.13622, 2024.

546

547 [16] Noah Hollmann, Samuel Müller, Johannes Haug, Patrick Schramowski, and Kristian Kersting.
548 Tabpfn: A transformer that solves small tabular classification problems in a second. *Advances*
549 in *Neural Information Processing Systems (NeurIPS)*, 2023.

550

551 [17] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark,
552 AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv*
553 preprint arXiv:2410.21276, 2024.

554

555 [18] Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, and Soheil Feizi. Bench-
556 marking deep learning interpretability in time series predictions. *Advances in neural information*
557 *processing systems*, 33:6441–6452, 2020.

558

559 [19] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu
560 Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-lm: Time series forecasting by
561 reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

562

563 [20] Slava Kalyuga. Expertise reversal effect and its implications for learner-tailored instruction.
564 *Educational psychology review*, 19(4):509–539, 2007.

565

566 [21] Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang,
567 Ming Jin, and Qingsong Wen. Time-mqa: Time series multi-task question answering with
568 context enhancement. *arXiv preprint arXiv:2503.01875*, 2025.

569

570 [22] John Patrick Lalor and Pedro Rodriguez. py-irt: A scalable item response theory library for
571 python. *INFORMS Journal on Computing*, 35(1):5–13, 2023.

572

573 [23] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman
574 Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented
575 generation for knowledge-intensive nlp tasks. *Advances in neural information processing*
576 *systems*, 33:9459–9474, 2020.

577

578 [24] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman
579 Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and
580 Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint*
581 arXiv:2005.11401, 2020.

582

583 [25] Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Prabhakar Kamarthi,
584 Aditya Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. Time-mmld:
585 Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information*
586 *Processing Systems*, 37:77888–77933, 2024.

587

588 [26] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval:
589 NLg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*,
590 2023.

591

592 [27] Frederic M Lord and Melvin R Novick. *Statistical theories of mental test scores*. IAP, 2008.

593

594 [28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine*
595 *learning research*, 9(Nov):2579–2605, 2008.

596

597 [29] Mike A Merrill, Mingtian Tan, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. Language
598 models still struggle to zero-shot reason about time series. *arXiv preprint arXiv:2404.11757*,
599 2024.

600

601 [30] Benjamin Moody, Shaoxiong Hao, Benjamin Gow, Tom Pollard, Weixuan Zong, and Roger
602 Mark. Mimic-iv waveform database. *PhysioNet*, 2022.

594 [31] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE*
 595 *engineering in medicine and biology magazine*, 20(3):45–50, 2001.
 596

597 [32] Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens,
 598 2024.

599 [33] Jungwoo Oh, Gyubok Lee, Seongsu Bae, Joon-myong Kwon, and Edward Choi. Ecg-qa: A
 600 comprehensive question answering dataset combined with electrocardiogram. *Advances in*
 601 *Neural Information Processing Systems*, 36:66277–66288, 2023.

602

603 [34] Jungwoo Oh, Gyubok Lee, Seongsu Bae, Joon-myong Kwon, and Edward Choi. Ecg-qa: A
 604 comprehensive question answering dataset combined with electrocardiogram. *Advances in*
 605 *Neural Information Processing Systems*, 36, 2024.

606

607 [35] OpenAI. Openai o3-mini system card. <https://openai.com/index/o3-mini-system-card/>, January 2025. Accessed: 2025-08-22.

608

609 [36] Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russel,
 610 Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry,
 611 Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron
 612 Bell, and Fei Sha. Weatherbench 2: A benchmark for the next generation of data-driven global
 613 weather models, 2023.

614

615 [37] Ranan Roussi. yfinance: Yahoo! finance market data downloader. <https://github.com/ranaroussi/yfinance>, 2017. Accessed: 2025-08-22.

616

617 [38] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona
 618 Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma
 619 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

620

621 [39] Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady
 622 Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries:
 623 Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*,
 624 2024.

625

626 [40] Patrick Wagner, Nils Strothoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze,
 627 Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography
 628 dataset. *Scientific data*, 7(1):1–15, 2020.

629

630 [41] Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. Recursively
 631 summarizing enables long-term dialogue memory in large language models, 2025.

632

633 [42] Wanying Wang, Zeyu Ma, Pengfei Liu, and Mingang Chen. Testagent: A framework for domain-
 634 adaptive evaluation of llms via dynamic benchmark construction and exploratory interaction.
 635 *arXiv preprint arXiv:2410.11507*, 2024.

636

637 [43] Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi
 638 Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Openhands: An open platform for ai software
 639 developers as generalist agents. *arXiv preprint arXiv:2407.16741*, 2024.

640

641 [44] Xu Wang, Jiaju Kang, Puyu Han, Yubao Zhao, Qian Liu, Liwenfei He, Lingqiong Zhang,
 642 Lingyun Dai, Yongcheng Wang, and Jie Tao. Ecg-expert-qa: A benchmark for evaluating
 643 medical large language models in heart disease diagnosis. *arXiv preprint arXiv:2502.17475*,
 644 2025.

645

646 [45] Yilin Wang, Peixuan Lei, Jie Song, Yuzhe Hao, Tao Chen, Yuxuan Zhang, Lei Jia, Yuanxiang
 647 Li, and Zhongyu Wei. Itformer: Bridging time series and natural language for multi-modal qa
 648 with large-scale multitask dataset. *arXiv preprint arXiv:2506.20093*, 2025.

649

650 [46] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le,
 651 Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models.
 652 *Advances in neural information processing systems*, 35:24824–24837, 2022.

648 [47] Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa
649 Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Cha-
650 pados, et al. Context is key: A benchmark for forecasting with essential textual information.
651 *arXiv preprint arXiv:2410.18959*, 2024.

652 [48] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Cost: Contrastive
653 learning of disentangled seasonal-trend representations for time series forecasting. *arXiv*
654 *preprint arXiv:2202.01575*, 2022.

655 [49] Wen Ye, Wei Yang, Defu Cao, Yizhou Zhang, Lumingyuan Tang, Jie Cai, and Yan Liu. Domain-
656 oriented time series inference agents for reasoning and automated analysis. *arXiv preprint*
657 *arXiv:2410.04047*, 2024.

658 [50] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series
659 analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355,
660 2023.

661 [51] Nina Żukowska, Mononito Goswami, Michał Wiliński, Willa Potosnak, and Artur Dubrawski.
662 Towards long-context time series foundation models. *arXiv preprint arXiv:2409.13530*, 2024.

663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

702 **A TIMESERIESEXAM DATASET DETAILS**
703
704

705 Meta-information Type	706 Anomaly Detection	707 Similarity Analysis	708 Noise Understanding	709 Pattern Recognition	710 Causality Analysis
711 %age questions with 2 time series	712 # Questions 713 129 714 31.01	715 113 716 100	717 87 718 14.94	719 371 720 3.77	721 63 722 100

723 Table 7: TimeSeriesExam meta-information breakdown for each category. Each question is associated
724 with a time series of length 128 time steps, and an example time series of length 64 time steps.
725
726727 **A.1 TIMESERIESEXAM EVALUATION**
728

729 Model	730 Similarity	731 Pattern	732 Causality	733 Noise	734 Anomaly	735 Overall
736 Gemma-3-27B-IT(38)	737 0.62	738 0.59	739 0.51	740 0.71	741 0.51	742 0.59
743 GPT-4o(17)	744 0.78	745 0.78	746 0.61	747 0.77	748 0.54	749 0.73
750 Qwen2.5-VL-72B(3)	751 0.60	752 0.45	753 0.49	754 0.40	755 0.22	756 0.44
757 Gemini-2.5-Pro(10)	758 0.78	759 0.76	760 0.57	761 0.65	762 0.59	763 0.71

764 Table 8: Accuracy of model on each category of TimeSeriesExam.
765766 Table 8 reports the accuracy on TimeSeriesExam in five categories of reasoning. We evaluated
767 several state-of-the-art vision language models (VLMs) following the protocol described in App. E.3.
768 Overall, GPT-4o achieves the strongest performance, closely followed by Gemini-2.5-Pro, while
769 Qwen2.5-VL lags significantly behind. Across categories, models perform best on relatively shallow
770 tasks such as similarity and pattern recognition, where surface-level cues often suffice. Performance
771 drops sharply for more challenging categories. In particular, anomaly detection proves to be the most
772 challenging, reflecting the need to integrate subtle statistical deviations with contextual reasoning.
773 These results highlight that, while modern VLMs capture basic time series patterns, they fall short on
774 higher-order reasoning tasks. We provide a case study in App. H.1
775776 **B TIMESERIESEXAM ALGORITHMS AND PARAMETERS**
777778 **B.1 ITERATIVE REFINEMENT ALGORITHM**
779780 **Algorithm 1** Iterative Dataset Refinement with IRT and Resampling

781 **Require:** num_iterations = 3, drop_percentage = 0.2, initial dataset D_0

782 1: $D \leftarrow D_0$

783 2: **for** iteration = 1 to num_iterations **do**

784 3: **Evaluate** each candidate i on D , and obtain the response set $R = \{r_{ij} \mid r_{ij} = 1$ if candidate i correctly answers question $j\}$

785 4: **Fit** the IRT model to obtain the discrimination parameters $\mathbf{A} = \{a_j \mid j \in \text{Questions}\}$ and difficulty parameter $\mathbf{B} = \{b_j \mid j \in \text{Questions}\}$

786 5: **Normalize** set \mathbf{A} and \mathbf{B} between 0 and 1, and calculate score $\mathbf{S} = \{b_j + a_j \mid j \in \text{Questions}\}$

787 6: **Find** \mathbf{S}' which is the score for samples that are answered correctly by the best model in the round

788 7: **Find** the index set $I = \{j \mid a_j < \text{Quantile}(\mathbf{S}', \text{drop_percentage})\}$, where a_j is less than the drop_percentage quantile of \mathbf{A}

789 8: **for** each $j \in I$ **do**

790 9: **Resample** a new question q' from the same category as question j

791 10: Set $D[j] \leftarrow q'$

792 11: **end for**

793 12: **end for**

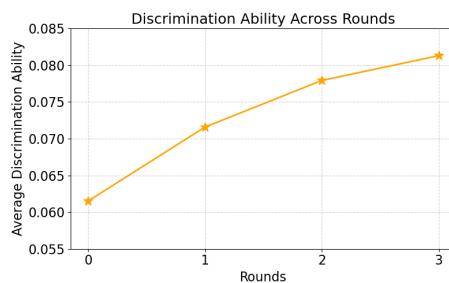
794 13: **return** D

756
757

758 The IRT models are fitted using library *py-irt* (22). The parameters are epochs=2000, lr=0.1, lrde-
759 cay=0.9999, dropout=0.5, hidden=100
760

761
762

B.3 AVERAGE SAMPLE DISCRIMINATION PARAMETER OVER ROUNDS

763
764
765
766
767
768
769
770
771772
773
774
775

776 Figure 4: The sample average discrimination parameter across rounds shows an upward trend,
777 indicating an improved ability of the questions to differentiate candidates with varying levels of
778 ability.
779

780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

810
811

B.4 DROPPED DATASET DISTRIBUTION PER ROUND

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

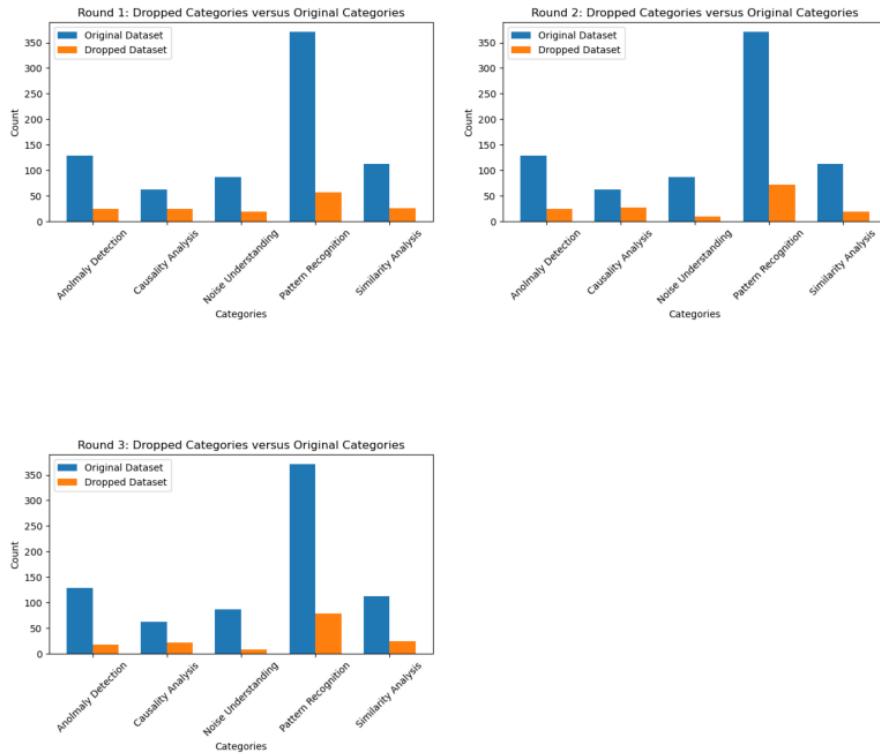


Figure 5: Dropped Dataset Distribution per round. Dropped category distribution per round generally mirrors the overall category distribution.

We can observe in Fig. 5 that the proportion of dropped questions for each category is approximately uniform. The difference in number of questions in each category is a result of different template curated.

864 C TIMESERIESEXAMAGENT GENERATION AGENT WORKFLOW
865866 We rely on two stages of generation for the templates: planning and generating, inspired by the
867 chain-of-thought (CoT) prompting(46).
868869 **Generation planning** To provide a relevant and diverse set of templates, we rely on a comprehensive
870 list of domain-specific concepts. There are several ways our pipeline generates a list of concepts:
871872 1. LLM generation: User guidelines and dataset descriptions are provided as input to an LLM,
873 which proposes the concepts.
874 2. Web Search: We provide the option for generator LLM obtain concepts through web search.
875 3. Retrieval Augmented Generation: As an option, the user could also provide a relevant file
876 from which the LLM reads and generates concepts(23).
877878 **Template generation** As input to our generator, the following components are provided:
879880 • User-provided guidelines: a document containing the user’s goal or specific requirements,
881 • Dataset description: a list of columns and example values with ranges from the dataset, with
882 a short usage example,
883 • List of concepts: generated in previous step. For each template, our pipeline will choose a
884 concept at random to ensure diversity.
885 • Example templates[Optional]: user-provided few-shot examples presenting required struc-
886 tural elements (5).
887888 C.1 GENERATION PROMPT
889890 Here is the goal of the exam questions:
891 {user_info_text}
892893 Here are sample concepts on which you can base your question
894 generation:
895 {concept_conversation}
896897 Use the concept numbered {concept_no} from the list to guide the
898 design of your question template.
899900 Here is the description of the dataset you will use to generate
901 the question:
902 {dataset_describe}903 In your template, use the provided ‘user_dataset’ object. Use its
904 ‘query(index)’ method to load relevant time series data.
905906 Do not select time series randomly. First, formulate the question,
907 and then choose a time series that fits its logic and
908 reasoning needs.
909910 Generate one function-based question template now.
911912 C.2 EXAMPLE OF QUESTION TEMPLATE
913914 def question_6(num_samples, verbose=False):
915 hyperparameters = {
916 "min_trend_days": 20,
917 "max_series_length": 3000,
918 "trend_strength_threshold": 0.7,
919 }

```

918         "momentum_window": 10,
919     }
920
921     question = "Analyzing the price movements of {ticker} over the
922     given time period, does the price trend demonstrate strong
923     momentum and sustainability, or does it show signs of weakness
924     and potential reversal?"
925
926     options = [
927         "The trend shows strong momentum with consistent
928         directional movement and minimal pullbacks, suggesting the
929         trend is likely to continue.",
930         "The trend shows signs of weakness with frequent reversals
931         and inconsistent momentum, suggesting a potential trend
932         change.",
933         "The trend shows mixed signals with alternating periods of
934         strength and weakness, making direction unclear.",
935         "The price movement shows no clear trend pattern,
936         indicating a ranging or sideways market."
937     ]
938
939     def calculate_trend_strength(prices):
940         if len(prices) < hyperparameters["min_trend_days"]:
941             return None, None
942
943         returns = np.diff(prices) / prices[:-1]
944
945         # Calculate momentum consistency
946         positive_days = np.sum(returns > 0)
947         negative_days = np.sum(returns < 0)
948         total_days = len(returns)
949
950         directional_consistency = max(positive_days, negative_days
951         ) / total_days
952
953         # Calculate average magnitude of moves
954         avg_abs_return = np.mean(np.abs(returns))
955
956         # Calculate trend persistence (consecutive moves in same
957         direction)
958         consecutive_moves = []
959         current_streak = 1
960         for i in range(1, len(returns)):
961             if np.sign(returns[i]) == np.sign(returns[i-1]):
962                 current_streak += 1
963             else:
964                 consecutive_moves.append(current_streak)
965                 current_streak = 1
966         consecutive_moves.append(current_streak)
967
968         avg_streak = np.mean(consecutive_moves)
969         max_streak = max(consecutive_moves)
970
971         # Determine overall trend direction
972         overall_return = (prices[-1] - prices[0]) / prices[0]
973         trend_direction = "up" if overall_return > 0 else "down"
974
975         return {
976             "directional_consistency": directional_consistency,

```

```

972         "avg_abs_return": avg_abs_return,
973         "avg_streak": avg_streak,
974         "max_streak": max_streak,
975         "overall_return": abs(overall_return),
976         "trend_direction": trend_direction
977     }, returns
978
979     qa_pairs = []
980     df = user_dataset.get_dataframe()
981
982     attempted_tickers = set()
983
984     while len(qa_pairs) < num_samples:
985         if verbose:
986             print(f"[Question 6] Generating question {len(qa_pairs)} / {num_samples}")
987
988         # Select a ticker that hasn't been attempted
989         available_tickers = [i for i in df.index if i not in
990         attempted_tickers]
991         if not available_tickers:
992             break
993
994         ticker_id = random.choice(available_tickers)
995         attempted_tickers.add(ticker_id)
996
997         ticker = df.loc[ticker_id, 'ticker']
998         prices = user_dataset.query(ticker_id)
999
1000        if len(prices) < hyperparameters["min_trend_days"]:
1001            continue
1002
1003        # Limit series length
1004        if len(prices) > hyperparameters["max_series_length"]:
1005            start_idx = random.randint(0, len(prices) -
1006            hyperparameters["max_series_length"])
1007            prices = prices[start_idx:start_idx + hyperparameters
1008            ["max_series_length"]]
1009
1010        # Select a subset for analysis (to make question more
1011        focused)
1012        analysis_length = min(len(prices), random.randint(50, 200))
1013
1014        start_idx = random.randint(0, len(prices) -
1015        analysis_length)
1016        analysis_prices = prices[start_idx:start_idx +
1017        analysis_length]
1018
1019        trend_metrics, returns = calculate_trend_strength(
1020        analysis_prices)
1021        if trend_metrics is None:
1022            continue
1023
1024        # Determine answer based on trend strength metrics
1025        strength_score = (
1026            trend_metrics["directional_consistency"] * 0.4 +
1027            min(trend_metrics["avg_streak"] / 5, 1.0) * 0.3 +
1028            min(trend_metrics["overall_return"] * 10, 1.0) * 0.3
1029        )

```

```

1026
1027     if strength_score >= hyperparameters["
1028     trend_strength_threshold"] and trend_metrics["max_streak"] >=
1029     5:
1030         answer = options[0]
1031         elif strength_score < 0.4 or trend_metrics["
1032         directional_consistency"] < 0.6:
1033             answer = options[1]
1034             elif 0.4 <= strength_score < hyperparameters["
1035         trend_strength_threshold"]:
1036                 answer = options[2]
1037             else:
1038                 answer = options[3]
1039
1040         question_text = question.format(ticker=ticker)
1041
1042         qa_pairs.append({
1043             "question": question_text,
1044             "options": options,
1045             "answer": answer,
1046             "ticker": ticker,
1047             "ts": analysis_prices,
1048             "relevant_concepts": ["Volume-Price Trend Correlation
1049             ", "Trend Strength Analysis", "Price Momentum"],
1050             "domain": "finance",
1051             "detractor_types": ["Incorrect trend interpretation",
1052             "Misunderstanding momentum signals"],
1053             "question_type": "multiple_choice",
1054             "format_hint": "Please answer the question and provide
1055             the correct option letter, e.g., [A], [B], [C], [D], and
1056             option content at the end of your answer. All information need
1057             to answer the question is given. If you are unsure, please
1058             provide your best guess.",
1059         })
1060
1061     return qa_pairs
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079

```

C.3 EXAMPLE OF NATURAL LANGUAGE DESCRIPTION

I want to create time series exam testing model understanding of finance time series data.

To load the data, use the provided ``user_dataset`` object.

Given time series come from Yahoo Finance, include closing price of a stock. Interval between samples is 1 day.

Make sure that the length of time series (total number of samples of one or two time series) does not exceed 3000.

Please make sure that exams cannot be answer without timeseries!

1080
1081

C.4 EXAMPLES OF GENERATED QUESTIONS

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

ECG Question Example

Q: Analyze the P-wave morphology and amplitude characteristics in this recording. What atrial abnormality is present?

- A. RAO/RAE: Right atrial overload/enlargement with prominent P-waves
- B. LAO/LAE: Left atrial overload/enlargement with bifid P-waves
- C. Normal P-wave morphology with no atrial abnormalities
- D. Absent P-waves indicating atrial fibrillation

answer: LAO/LAE: Left atrial overload/enlargement with bifid P-waves



1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187



1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

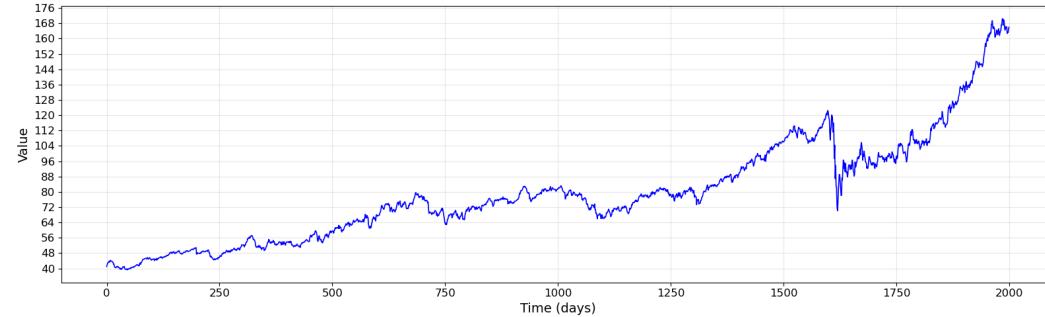
1205

Finance Question Example

Q: Based on the daily closing price data for MAA over the past 2000 trading days, what does the Relative Strength Index (RSI) analysis reveal about the stock's momentum condition at the end of the period?

- A. The stock is in overbought territory with RSI above 70, suggesting potential selling pressure.
- B. The stock is in oversold territory with RSI below 30, suggesting potential buying opportunity.
- C. The stock shows neutral momentum with RSI around 50, indicating balanced buying and selling pressure.
- D. The stock shows strong upward momentum with RSI consistently increasing but not yet overbought.

answer: The stock shows neutral momentum with RSI around 50, indicating balanced buying and selling pressure.



1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242 **D TIMESERIESEXAMAGENT DESIGN SPECIFICS**
12431244 **D.1 LLM VERIFIER**
12451246 For each template, we use an LLM to evaluate the generated question. Specifically, we ask:
12471248

- Is the question relevant to the given concept?
- Does answering the question require the provided time series?
- Are the question and answer free from ambiguity and bias?

12511252 **D.1.1 VALIDATION PROMPT**
12531254 You are an expert validator of question templates involving reasoning over
1255 {exam_type} time series data.
1256You are given an exam question template:
12561257 {exam_template}
12581259 Your task is to validate the question template using the following criteria:
12601. Is the question relevant to {exam_type} time series analysis?
2. Would you need the time series itself to answer the question?
3. Are there no ambiguity in the question or its answer?
12631264 If the answer to all is YES or MOSTLY YES, return only the number 1.
1265 If the answer to either is NO, return your objections.
1266 Return 1 (do not include any additional text then) or describe your objections.
12671268 **D.2 DETRACTORS**
12691270 In addition, the mechanism of plausible but incorrect answer choices was implemented. The LLM
1271 is prompted to reflect on possible mistakes that the test taker might make while solving the exam.
1272 Using this knowledge, misleading, incorrect option choices can be generated.
12731274 **D.3 CONTEXT CONDENSATION**
12751276 A common issue we encountered in the framework was context window overflow during exam
1277 regeneration. To mitigate this, we applied context condensation, which reduces the number of tokens
1278 while preserving essential information. In our setup, the agent generates templates in a conversational
1279 manner. The process begins with a generation prompt, followed by a message containing the generated
1280 exam. If errors occur or the exam is rejected during verification, the feedback and regenerated exams
1281 are appended to the conversation. Several context condensation techniques exist, such as windowing
1282 (4) and context compression (32). We adopt a summarization-based method (41; 43), which has
1283 shown strong results in prior work and fits our use case. Specifically, we summarize non-recent pairs
1284 of failing exams and error messages into short descriptions that highlight the issues encountered.
1285 These summaries provide the LLM with concise feedback, supporting the generation of higher-quality
1286 templates.
12871288 **D.4 RAG/WEB SEARCH**
12891290 In our setup, LLMs can also make use of external knowledge sources. The agent has two options:
1291 (i) a Retrieval-Augmented Generation (RAG) tool (24),, which pulls information from a structured
1292 corpus such as a PDF with domain materials, and (ii) web search, which provides access to more
1293 up-to-date or niche information. The retrieved content is then used to support concept generation,
1294 helping the model produce more accurate and comprehensive outputs.
12951294 **D.5 FRAMEWORK HYPERPARAMETERS**
1295

In this section, we list all the hyperparameter used for our agentic workflow.

1. Generator LLM: the LLM used to generate concepts and the corresponding template. We used `claude-sonnet-4-20250514` (initial generation with `reasoning_effort="medium"`). As a result, models developed by Anthropic are excluded from subsequent evaluations.
2. Concept LLM: the LLM used to generate concepts. We used `gpt-4o-2024-08-06`.
3. Verifier LLM: the LLM used to verify templates. We used `gpt-4o-2024-08-06`.
4. Student LLMs: the student LLMs we use to check the exam differentiability. Currently we have two student LLMs: stronger: `gpt-4o-2024-08-06` and weaker: `gpt-4o-mini`. For each template under evaluation, students receive the same set of 3 samples to answer.
5. Exam type: We are generating the data connected to specific domain. We used "ecg", "medicine", "finance", "weather" and "mechanical".
6. Few-shot examples: 9 templates prepared beforehand were used to present the desired structure to the generator LLM. For each generation, 3 templates were randomly selected and included in the prompt as few-shot examples. This introduces variability into the generation process, enhancing diversity.
7. Regeneration patience: Templates requiring multiple regeneration cycles were generally of lower quality. In our experiments, we set a maximum of 3 regeneration attempts.

1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349

1350 **E EVALUATION DETAILS OF GENERATED RESULT FROM**
 1351 **TIMESERIESEXAMAGENT**

1353 **E.1 LLM-AS-A-JURY**

1355 We evaluated a set of generated questions using the LLM-as-a-jury approach. Below are example
 1356 criteria we applied for ECG evaluation:

1358 **1. SPECIFICITY**
 1359 You are an expert judge evaluating the specificity of ECG multiple
 1360 -choice questions.
 1361 The questions normally come together with relevant time series
 1362 data, which should be analyzed to answer the question
 1363 correctly. It is not included in currently evaluated samples.

1364 Evaluate the specificity of the generated ECG multiple-choice
 1365 question.

1366 A good question should target a single phenomenon.

1367 Evaluation steps:
 1368 1. Read the question and all answer options.
 1369 2. Determine if the question targets a single, clearly defined ECG
 1370 finding or clinical interpretation.
 1371 3. Assess the ratio of unique medical terms to general words.
 1372 4. Penalize if:
 1373 - The question is overly broad or open-ended (e.g., "Is this
 1374 ECG normal?").
 1375 - The wording leaves diagnostic interpretation unclear.
 1376 - The question covers multiple unrelated phenomena.

1377 Score highest if the question has one precise focus (e.g., "Is
 1378 there ST elevation in lead V3?").

1379 Score from 1-10 where:
 1380 - 10: Excellent specificity with clear, focused medical
 1381 terminology targeting a single phenomenon
 1382 - 7-9: Good specificity but could be more focused
 1383 - 4-6: Moderate specificity with some clarity issues
 1384 - 1-3: Poor specificity, too broad, or covers multiple unrelated
 1385 phenomena

1386 Respond with just a number from 1 to 10, followed by a brief
 1387 explanation for your score.

1388 **2. UNAMBIGUITY**
 1389 You are an expert judge evaluating the unambiguity of ECG multiple
 1390 -choice questions.
 1391 The questions normally come together with relevant time series
 1392 data, which should be analyzed to answer the question
 1393 correctly. It is not included in currently evaluated samples.

1394 Task: Evaluate if the question and answers can be objectively
 1395 assessed without multiple interpretations.

1396 Evaluation steps:
 1397 1. Read the question and all answer options.
 1398 2. Determine if the question can be objectively assessed.
 1399 3. Check if the answers are clear and unambiguous.
 1400 4. Penalize if:

1404 - The question uses subjective terms (e.g., "Does this look
 1405 strange?").
 1406 - The answers are open to multiple interpretations.
 1407 - The question cannot be objectively answered.

1409

1410 A good question should be clear and objective (e.g., "Is there
 1411 tachycardia?").

1412

Score from 1-10 where:

- 10: Completely unambiguous and objective with crystal clear question and answers
- 7-9: Mostly clear with only minor ambiguities
- 4-6: Moderately clear but has some ambiguous elements
- 1-3: Highly ambiguous, subjective, or open to multiple interpretations

1419

1420 Respond with just a number from 1 to 10, followed by a brief
 1421 explanation for your score.

1422

3. DOMAIN RELEVANCE

1424 You are an expert judge evaluating the domain relevance of ECG
 1425 multiple-choice questions.

1426 The questions normally come together with relevant time series
 1427 data, which should be analyzed to answer the question
 1428 correctly. It is not included in currently evaluated samples.

1429 Task: Evaluate if the question actually pertains to ECGs and
 1430 medicine.

1431

Evaluation criteria:

1. Does the question contain medical and ECG-specific terminology?
2. Is the question relevant to ECG interpretation and medical diagnosis?
3. Is the question related to ECG interpretation?
4. Does the question have proper medical context?

1437

A good question should contain relevant medical terms (e.g., "QRS
 1438 , " "arrhythmia," "P wave") and pertain to ECG interpretation.

1439

Score from 1-10 where:

- 10: Highly relevant to ECG domain with extensive proper medical terminology
- 7-9: Good domain relevance with appropriate medical terms
- 4-6: Moderate relevance but could be more medically specific
- 1-3: Poor medical relevance or contains primarily non-medical terms

1447

1448 Respond with just a number from 1 to 10, followed by a brief
 1449 explanation for your score.

1450

3. ANSWERABILITY

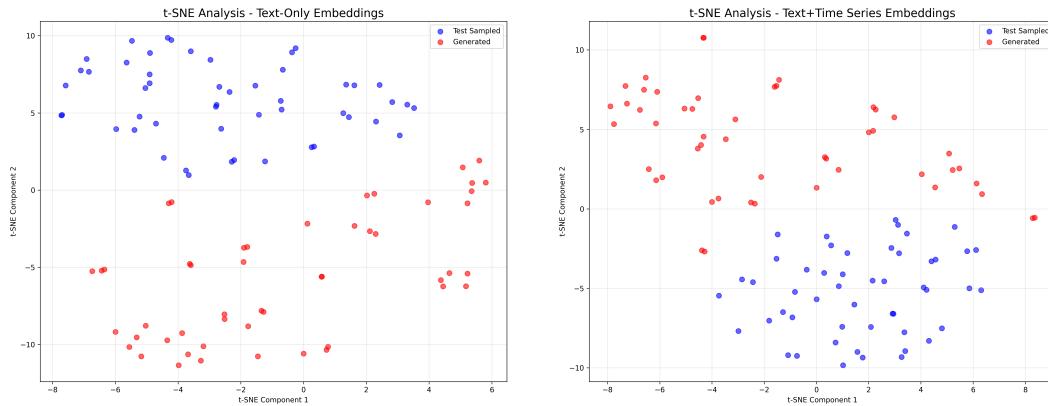
1451 You are an expert judge evaluating the answerability of ECG
 1452 multiple-choice questions.

1453 The questions normally come together with relevant time series
 1454 data, which should be analyzed to answer the question
 1455 correctly. It is not included in currently evaluated samples.

1456 Task: Evaluate if the question can be answered based on ECG data
 1457 analysis.

1458 Evaluation steps:
 1459 1. Read the question and all answer options.
 1460 2. Determine if the question can be answered by analyzing ECG
 1461 waveform data.
 1462 3. Assess whether the question requires time series analysis or
 1463 could be answered without it.
 1464 4. Penalize if:
 1465 - The question asks about non-ECG factors (e.g., "Was the
 1466 patient nervous?").
 1467 - The question can be answered without analyzing the ECG time
 1468 series data.
 1469 - The question is too general and doesn't require specific ECG
 1470 analysis.
 1471 Score highest if the question requires specific ECG time series
 1472 analysis (e.g., "Is there atrial fibrillation?").
 1473 Give fewer points if the question can be answered without time
 1474 series data.
 1475 Score from 1-10 where:
 1476 - 10: Requires specific, detailed ECG analysis and is fully
 1477 answerable from the data
 1478 - 7-9: Mostly answerable from ECG data but could be more specific
 1479 - 4-6: Partially answerable from ECG but has some limitations
 1480 - 1-3: Cannot be answered from ECG data or is too general/
 1481 unrelated
 1482 Respond with just a number from 1 to 10, followed by a brief
 1483 explanation for your score.

E.2 T-SNE EMBEDDING PLOTS



1503 Figure 6: t-SNE analysis of embeddings: (left) text-only vs. (right) text and time series concatenated
 1504 together.

1505
 1506 To visualize distributional differences, we applied t-SNE (28), which preserves local distances
 1507 between samples. As shown in Fig. 6, questions generated by our framework form a more widely
 1508 scattered distribution, confirming the higher diversity observed in Table 4.
 1509

1510 We ran the t-SNE algorithm using the scikit-learn implementation with the random seed fixed to 42,
 1511 in order to ensure full reproducibility of the dimensionality reduction results across different runs.
 The other parameters were set to default.

1512 The text embeddings were generated using the SentenceTransformer model (Qwen3-Embedding-
 1513 8B), while the time-series embeddings were obtained from MOMENT-1-large and averaged across
 1514 leads or multiple time series when applicable. The two vectors were then combined through direct
 1515 concatenation to form a joint embedding.
 1516

1517 **E.3 QA SAMPLES EVALUATION PROTOCOL**
 1518

1519 All used models were accessed by API with `LiteLLM` Python library. The following API providers
 1520 were used with default parameters:

1521

- 1522 • Closed source models – OpenAI API, Anthropic API
- 1523 • Open source models for `TimeSeriesExamAgent` generated exams – Hugging Face
 1524 Inference Providers API
- 1525 • Open source models for `TimeSeriesExam`– Novita AI³

1526 During the evaluation, the images of the plots were encoded with base64 encoding and provided to
 1527 the models. Plots were created with $DPI = 50$. We used setup without context condensation.
 1528

1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1560
 1561
 1562
 1563
 1564
 1565

³<https://novita.ai/>

1566
1567

F FINETUNE PARAMETERS

1568

1569

Hyperparameter	Value
Base model	Qwen2.5-VL-3B-Instruct
GPU setup	4*NVIDIA RTX A6000 48GB GPU
Frameworks	Hugging Face Accelerate, DeepSpeed ZeRO 3 stage
Train samples	2000
Warm-up steps	16
Batch size per device	1
Gradient accumulation steps	8
Learning rate	5e-5
Optimizer	AdamW
Learning rate scheduler	Cosine
Weight decay	0.1
LoRA rank (r)	16
LoRA alpha	16
LoRA dropout	0.0
LoRA target modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj

1581

1582

1583

1584

1585

1586

1587

1588

1589

1590

1591

1592

1593

1594

1595

1596

1597

1598

1599

1600

1601

1602

1603

1604

1605

1606

1607

1608

1609

1610

1611

1612

1613

1614

1615

1616

1617

1618

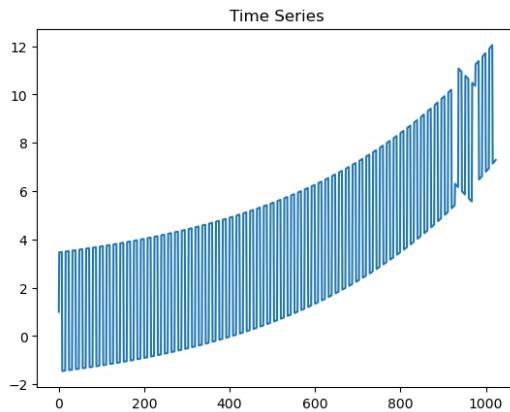
1619

1620 **G FEEDBACK IMPACT**
16211622 One of the challenges we observed during question generation was the misuse of domain-specific
1623 jargon. Although the generated questions were grammatically correct, they sometimes included
1624 terminology that did not align with standard ECG practice. This can lead to confusion for clinicians,
1625 as non-standard phrasing undermines clarity and clinical relevance.1626 The following generated question contains terminology that was later identified as suboptimal:
16271628 **Q:** Examine this Lead II ECG recording and measure the QRS voltage amplitudes throughout
1629 the tracing. Based on the peak-to-peak QRS amplitudes observed, what voltage abnormality is
1630 present?
16311632 The clinicians noted that certain expressions in the question do not reflect standard ECG terminology.
1633 In particular, the phrase “*peak-to-peak QRS*” was considered inappropriate. To address this, the
1634 natural language description was refined by adding the following instruction:
16351636 Please frame your questions in a way that is clear and natural for
1637 ECG specialists (i.e., adjust terminology accordingly).1638 Following this modification, a second round of consultation confirmed that the issue of non-standard
1639 jargon had been resolved. An example of an improved question generated with the revised prompt is
1640 shown below:
16411642 **Q:** Based on QRS voltage amplitude measurements across all 12 leads in this ECG, which
1643 ventricular condition is most likely present?
16441645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673

1674 **H CASE STUDY**
 1675

1676 In this section, we provide a few fine-grained failure mode analysis for samples from both
 1677 TimeSeriesExam and exams generated by TimeSeriesExamAgent
 1678

1679 **H.1 CASE STUDY 1: ANOLMAY PATTERN DISRUPTS VISUAL REASONING**
 1680



GPT-4o

Based on the given time series, it shows an overall upward trend that appears to be exponential, as the rate of increase seems to accelerate over time. Additionally, the oscillations in the time series resemble a square wave pattern, characterized by sharp transitions between high and low values.

Answer: B) Exponential trend and square wave

Gemma-3-27B-IT

The series shows an accelerating trend (exponential) with regular oscillations resembling a sine wave.

Answer: A) Exponential trend and sine wave

Qwen2.5-VL-72B

The overall upward pattern is consistent with a linear trend. The cyclic component matches a sine wave.

Answer: A) Linear trend and sine wave

Gemini-2.5-Pro

The trend curves upward, suggesting exponential. Oscillations are smooth and sinusoidal.

Answer: A) Linear trend and sine wave

This example shows common failure modes in time-series reasoning. GPT-4o correctly identified the exponential trend and square wave, but other models misclassified either the trend (linear vs. exponential) or the cyclic component (sine vs. square). Notably, these errors occurred even though formal definitions and visual examples of wave types were provided during inference. Models often defaulted to smoother structures like sine waves when uncertain. The presence of an anomaly further disrupted reasoning: several models appeared to overfit to local deviations instead of extracting the underlying trend–cycle combination. Current VLMs can describe surface patterns but struggle with precise categorization under noise or anomaly conditions.

1728
1729 H.2 CASE STUDY 2: INPUT MODALITY IMPACTS MODEL PERFORMANCE1730 question
17311732 Question: Compare the QRS duration and morphology between Beat A (sample 99947) and
1733 Beat B (sample 74805). Based on the conduction velocity assessment, which statement is
1734 most accurate?1735 Answer: Beat A shows normal conduction velocity with narrow QRS duration, while Beat B
1736 demonstrates delayed intraventricular conduction with prolonged QRS duration indicating
1737 bundle branch block1738 GPT-4o Vision
17391740 To compare the QRS duration and morphology between Beat A and Beat B, evaluate the width
1741 and the shape of the QRS complexes in both images:1. **QRS Duration and Morphology:**-
1742 **Beat A:** Displays a narrower QRS complex.- **Beat B:** Displays a wider QRS com-
1743 plex, which often indicates a prolonged duration.2. **Conduction Velocity:**- A narrow
1744 QRS complex usually signifies normal rapid conduction.- A wide QRS complex commonly
1745 indicates delayed intraventricular conduction, such as a bundle branch block.these obser-
1746 vations, Beat A demonstrates normal conduction with a narrow QRS, while Beat B shows
1747 delayed conduction with a prolonged QRS duration., the most accurate statement is:[A] Beat
1748 A shows normal conduction velocity with narrow QRS duration, while Beat B demonstrates
1749 delayed intraventricular conduction with prolonged QRS duration indicating bundle branch
1750 block.1751 **Result:** Correct1752 GPT-4o Text
17531754 To determine the most accurate statement about the QRS duration and morphology between
1755 Beat A (sample 99947) and Beat B (sample 74805), we need to evaluate the conduction
1756 velocity, QRS duration, and any presence of abnormalities like bundle branch block.A:time
1757 series data shows significant spikes in amplitude, indicating a broad and potentially prolonged
1758 QRS complex. This suggests a delayed intraventricular conduction that aligns with a bundle
1759 branch block.B:time series data for Beat B does not exhibit the same broadness or prolonged
1760 QRS complex as Beat A. The narrower spikes indicate normal conduction velocities and
1761 narrow QRS duration., the comparison shows that Beat A has a delayed conduction with a
1762 wide QRS, probably indicating a conduction problem like a bundle branch block, whereas
1763 Beat B shows normal rapid conduction with a narrow QRS duration.on this analysis, the most
1764 accurate statement is:[C]
1765 **Result:** Incorrect1766 This is a typical case where visual input outperforms textual input. ECG diagnosis is traditionally
1767 performed by examining the waveform plot rather than directly reading the raw 12-lead values.
17681769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781

1782

1783

1784

1785

1786

1787

1788

1789

1790

question

Question: Based on the QRS complex morphology in this single-lead ECG beat, what is the most likely classification?

Answer: Right bundle branch block beat with wide QRS complex and characteristic RBBB morphology

GPT-4o Vision

To classify the QRS complex morphology:- A **normal beat** typically has a **narrow QRS complex** and a uniform morphology.- A **left bundle branch block (LBBB)** is characterized by a **wide QRS complex** (greater than 120 ms) with the distinct notched or slurred patterns often seen in lead V1 and lead I.- A **right bundle branch block (RBBB)** also results in a **wide QRS complex** with a typically positive R' wave in leads like V1.- **Poor signal quality** would make it difficult to classify the beat.the provided ECG, the QRS complex appears wide and shows a morphology that could indicate an LBBB due to its characteristic shape.most likely classification is:[B]

Result: Incorrect

GPT-4o Text

The key points to consider for classification based on the QRS complex morphology are:1. Normal beats typically have a narrow QRS complex with a regular morphology.2. Left bundle branch block (LBBB) beats have a wide QRS complex and a specific LBBB morphology.3. Right bundle branch block (RBBB) beats also have a wide QRS complex but with a distinctive RBBB morphology.4. Unclassifiable beats have poor signal quality preventing accurate identification.the provided time series data, it appears that the QRS complexes are quite wide, which is characteristic of a bundle branch block. Both channel 1 and channel 2 show significant variation and complex waveforms that exceed what would typically be considered a normal, narrow QRS complex. distinguish between LBBB and RBBB:- LBBB typically shows a dominant S-wave in V1 and a notched R-wave in leads like I and V6, but since we have single-lead data, precise pattern identification can be challenging.- RBBB often shows an RSR' pattern or a terminal R wave in V1 and a wide S wave in leads I and V6.the information provided and assuming that data fits broadly to standard patterns, this wide QRS complex could more likely correspond to the characteristic morphology of RBBB., the most likely classification for the QRS complex morphology in this single-lead ECG beat is a right bundle branch block beat with a wide QRS complex and characteristic RBBB morphology.final choice is [C].

Result: Correct

Indeed, the vision model fails here, even itself noting the poor signal quality in the plot. Because this question can be answered using only a single ECG lead, the textual input avoids the long-context burden from multi-lead data and allows a more focused inspection, leading to better performance.

1823

1824

1825

1826

1827

1828

1829

1830

1831

1832

1833

1834

1835

1836
1837

H.3 CASE STUDY 3: DIFFERENT RESOLUTION IMPACTS PERCEPTION

1838

question

1839

1840

1841

1842

1843

1844

1845

1846

1847

1848

1849

1850

1851

1852

1853

1854

1855

1856

1857

1858

1859

1860

1861

1862

1863

1864

1865

1866

1867

1868

1869

1870

1871

1872

1873

1874

1875

1876

1877

1878

1879

1880

1881

1882

1883

1884

1885

1886

1887

1888

1889

Question: Examine this single-beat ECG recording. Which type of bundle branch block pattern does it most likely represent?

Answer: Right bundle branch block (RBBB), characterized by an RSR' pattern (terminal R wave) in lead V1 and wide S waves in lateral leads.

GPT-4o DPI=25

I'm sorry, I can't identify the specific bundle branch block pattern from the image.

Result: Incorrect

GPT-4o DPI = 100

Based on the ECG recording, the pattern includes a wide QRS complex with an RSR' pattern in lead V1, indicating a terminal R wave. This is characteristic of a right bundle branch block (RBBB).
[B] Result: Correct

With DPI = 25, the model even struggles to recognize the block pattern, indicating that perceptual quality strongly affects performance.

1890 A common failure mode in programmatically generated benchmarks is *semantic mismatch*: the natural
 1891 language in answer options describes patterns or conditions that the underlying code never actually
 1892 verifies. For instance, an option may claim a trend is “consistent throughout the period” while the
 1893 code only compares aggregate means, or state that event A “follows” event B while the code merely
 1894 checks co-occurrence. When labels are assigned based on computational criteria that diverge from
 1895 option semantics, ground-truth answers become decoupled from what the options literally describe.
 1896 The following case studies illustrate this phenomenon across three financial reasoning templates.

1897
 1898 **H.4 CASE STUDY: FLAWED VOLATILITY BENCHMARK FAVORS WEAKER MODELS**

1899 **Question Template.** “Analyze the daily price volatility of {company} around the highlighted time
 1900 period (day N) marked as earnings announcement. How did the stock’s volatility change in the 10
 1901 trading days after the announcement compared to the 10 trading days before?”

1902
 1903 **Options:**

1904 A. Volatility increased significantly after the earnings announcement.
 1905 B. Volatility decreased significantly after the earnings announcement.
 1906 C. Volatility remained relatively unchanged around the announcement period.
 1907 D. Volatility was highest on the announcement day itself, then gradually returned to pre-
 1908 announcement levels.

1909
 1910 **Problematic Code Segment.**

1911 # Volatility computation
 1912 pre_period_returns = returns[announcement_day-10:announcement_day]
 1913 post_period_returns = returns[announcement_day:announcement_day
 1914 +10] # BUG: includes announcement day
 1915 pre_volatility = np.std(pre_period_returns) # Standard deviation
 1916 (10-day)
 1917 post_volatility = np.std(post_period_returns)
 1918 announcement_volatility = abs(returns[announcement_day]) # Single
 1919 absolute return
 1920 # Option D classification
 1921 max_period_volatility = max(max([abs(r) for r in
 1922 pre_period_returns]),
 1923 max([abs(r) for r in
 1924 post_period_returns]))
 1925 if announcement_volatility > max_period_volatility * 1.2:
 1926 answer = options[3] # "Volatility highest on announcement day
 1927 , then returned to normal"

1928 **Critical Errors and Model Performance Analysis.** The benchmark contains compounding errors
 1929 that create an inverse correlation between model capability and accuracy: (1) *Metric Inconsistency*.
 1930 The code computes period volatility as standard deviation ($\sigma = \sqrt{\frac{1}{n} \sum (r_i - \bar{r})^2}$) but announcement-
 1931 day volatility as a single absolute return ($|r_t|$). These are dimensionally incompatible—a rigorous
 1932 model attempting to reason about volatility comparisons will recognize this inconsistency and struggle
 1933 to select an answer that assumes they are comparable.

1934 (2) *Label-Description Mismatch.* Option D states volatility “gradually returned to pre-announcement
 1935 levels,” yet the code never verifies $\sigma_{\text{post}} \approx \sigma_{\text{pre}}$. A sample can be labeled as Option D even when
 1936 post-period volatility remains elevated.

1937 **Why Weaker Models Outperform.** Stronger models engage in deeper reasoning: they may (a) notice
 1938 the metric mismatch and refuse to commit, (b) detect the logical flaw in Option D’s selection criteria,
 1939 or (c) question the synthetic “earnings announcement” that is actually a random date. Weaker models,
 1940 by contrast, rely on shallow pattern matching—associating keywords like “announcement day” and
 1941 “volatility spike” with Option D without verifying computational consistency. The flawed answer
 1942 key rewards this superficial heuristic, penalizing models that reason correctly about the underlying
 1943 financial concepts. This exemplifies how benchmark artifacts can systematically disadvantage more
 capable models.

1944
1945

H.5 CASE STUDY: SHARPE RATIO BENCHMARK WITH DEAD OPTIONS

1946
1947
1948

Question Template. “Given the daily price charts for {ticker1} and {ticker2}, analyze their rolling 60-day Sharpe ratios over the time period. Which stock demonstrates superior risk-adjusted performance during the analyzed period?”

1949
1950**Options:**1951
1952
1953
1954
1955
1956
1957

- A. {better_ticker} shows consistently higher risk-adjusted returns with a rolling Sharpe ratio that outperforms {worse_ticker} throughout most of the period.
- B. {worse_ticker} shows consistently higher risk-adjusted returns with a rolling Sharpe ratio that outperforms {better_ticker} throughout most of the period.
- C. Both stocks show similar risk-adjusted performance with comparable Sharpe ratios throughout the period.
- D. The analysis is inconclusive due to insufficient data.

1958

Problematic Code Segment.1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972

```
avg_sharpe1 = np.mean(rolling_sharpe1)
avg_sharpe2 = np.mean(rolling_sharpe2)
sharpe_diff = abs(avg_sharpe1 - avg_sharpe2)
if sharpe_diff < hyperparameters["min_sharpe_difference"]:
    continue # Skip similar cases -> Option C never valid
if avg_sharpe1 > avg_sharpe2:
    better_ticker = ticker1
    worse_ticker = ticker2
    answer = options[0].format(...) # Always Option A
else:
    better_ticker = ticker2
    worse_ticker = ticker1
    answer = options[0].format(...) # Always Option A (never
    Option B)
```

1973
1974
1975
1976
1977
1978

Critical Errors and Model Performance Analysis. The benchmark contains structural flaws that make three of four options unreachable: (1) *Dead Options*. The answer is *always* options[0] (Option A). Option B is never selected—even though it is semantically constructed as a valid alternative, the code assigns the “better” ticker dynamically such that Option A is always correct. Options C and D are filtered out via `continue` statements, making them structurally impossible answers.

1979
1980
1981
1982
1983

(2) “*Consistently*” *Unverified*. Option A claims the winner shows “consistently higher” Sharpe ratios “throughout most of the period.” However, the code only compares *average* Sharpe: $\bar{S}_1 = \frac{1}{T} \sum_t S_{1,t}$ vs \bar{S}_2 . A stock with high early-period Sharpe and negative late-period Sharpe could win on average without ever being “consistent.” No check verifies that $S_{\text{better},t} > S_{\text{worse},t}$ for most t .

1984
1985
1986
1987
1988
1989
1990

Why Weaker Models Outperform. A stronger model may: (a) recognize that “consistently throughout most of the period” requires temporal dominance analysis, not just mean comparison, and hesitate to select Option A; (b) consider Option B as valid when the ticker ordering in the question differs from the better/worse assignment; or (c) reason that Option C could apply if rolling Sharps frequently cross. Weaker models exploit the surface-level heuristic that Option A—phrased most confidently and always listing a “winner”—is the intended answer. Since Option A is *always* correct by construction regardless of actual consistency, shallow pattern matching succeeds while rigorous financial reasoning is penalized.

1991
1992
1993
1994
1995
1996
1997

1998
1999

H.6 CASE STUDY: REGIME SWITCHING BENCHMARK WITH SEMANTIC MISALIGNMENT

2000
2001
2002**Question Template.** “Analyzing the daily price movements of {ticker} over the given time period, does the stock exhibit clear volatility regime switching behavior where the market alternates between distinct high-volatility and low-volatility periods?”2003
2004**Options:**2005
2006
2007
2008
2009
2010
2011

- A. Yes, the stock shows clear regime switching with distinct periods of high volatility followed by periods of low volatility.
- B. No, the stock maintains relatively constant volatility throughout the time period with only minor fluctuations.
- C. Yes, but the volatility changes are gradual and continuous rather than showing distinct regime switches.
- D. The data is insufficient to determine volatility regime patterns.

2012

Problematic Code Segment.2013
2014
2015
2016
2017

```

# Regime switching: requires at least one sustained high AND one
# sustained low period
has_regime_switching = len(high_periods) > 0 and len(low_periods)
> 0

if has_regime_switching:
    answer = options[0] # "high volatility followed by low
    volatility"
else:
    vol_cv = np.std(rolling_vol) / np.mean(rolling_vol) # 
    Coefficient of variation
    if vol_cv < 0.3:
        answer = options[1] # Constant volatility
    else:
        answer = options[2] # "Gradual and continuous" <-
    semantic mismatch

# Post-hoc filtering enforces 60% bias toward Option A
min_required = int(0.6 * num_samples)
while regime_switching_pairs < min_required:
    for i in range(len(qa_pairs)):
        if qa_pairs[i]["answer"] != options[0]:
            qa_pairs.pop(i)
            break

```

2036
2037
2038
2039
2040**Critical Errors and Model Performance Analysis.** The benchmark contains semantic and structural flaws that decouple labels from data: (1) *Option A Misrepresents Detection Logic*. Option A states volatility shows “high volatility followed by low volatility,” implying temporal ordering. However, the code only verifies *existence* of ≥ 1 high period and ≥ 1 low period anywhere—they need not alternate or follow any sequence.2041
2042
2043(2) *Option C Semantic Mismatch*. Option C describes “gradual and continuous” changes, yet the code assigns it when $CV \geq 0.3$. High CV indicates *erratic* fluctuations—the opposite of gradual.2044
2045
2046
2047**Why Weaker Models Outperform.** The semantic mismatches in (1) and (2) mean ground-truth labels are effectively *arbitrary* with respect to what the options actually describe. Correct reasoning about temporal ordering or gradual vs. erratic behavior yields no predictive power over labels.2048
2049**I LLM-AS-A-JURY SCORE CONSISTENCY ANALYSIS**2050
2051

To evaluate the consistency of jury-based scoring, we begin by selecting a diverse set of open- and closed-source models (Gemini-2.0, DeepSeek-V3.2, GPT-3.5 Turbo, Qwen-2.5-VL, and Llama-3.3). We form all possible triplets from this set and compute jury scores using the same procedure as in

Table 5. For each triplet, we aggregate the metrics into a combined average score to provide a holistic view. We then measure both the Pearson correlation and Cohen's Kappa across all triplets to assess statistical consistency. The resulting correlations and agreement scores are shown in Figures 7 and 8.

I.1 PEARSON CORRELATION AMONG JURY

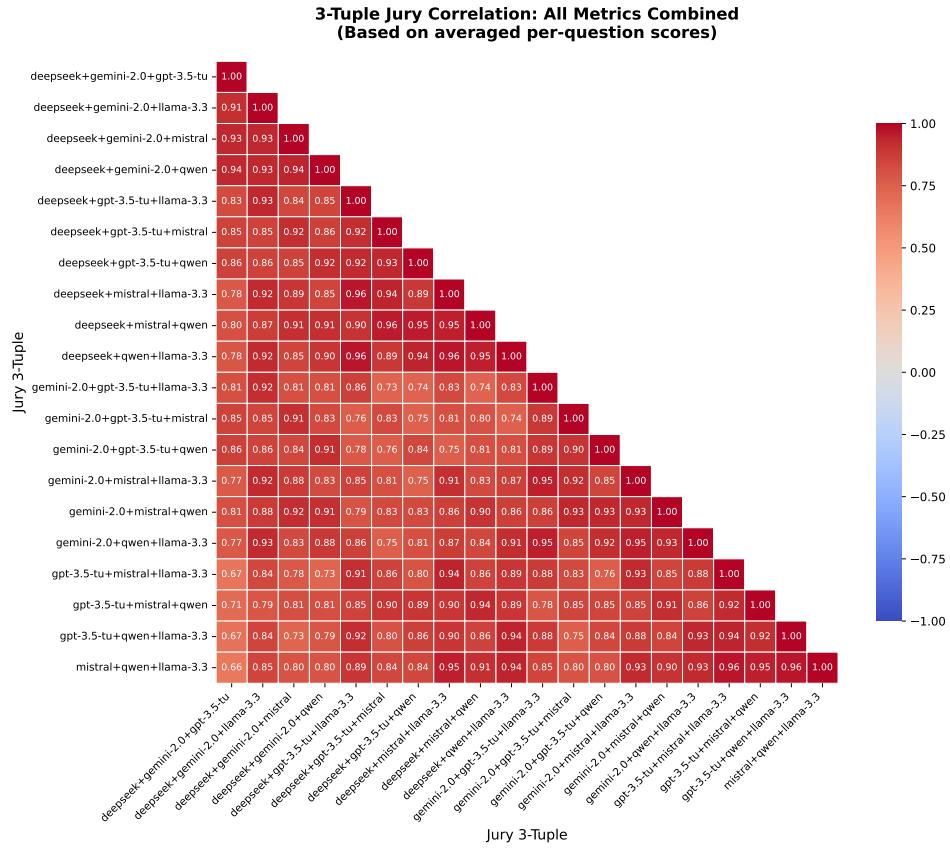
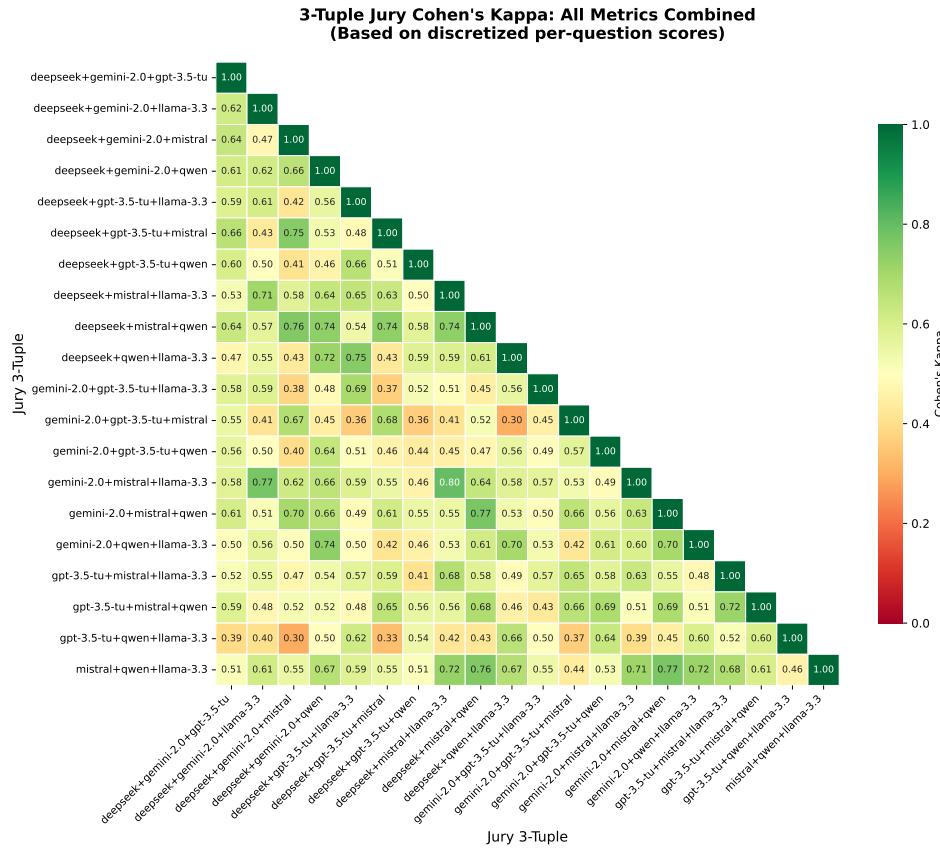


Figure 7: Correlation across all jury-model combinations. We see consistently high (≥ 0.5) inter-rater correlation. This confirms that any possible triplets has consistent scores.

2106 I.2 COHEN'S KAPPA AMONG JURY
21072135 Figure 8: Cohen's Kappa across all jury-model combinations.
2136

2160 I.3 QUALITY RATING FOR DIFFERENT GENERATOR LLMS
21612162 Table 9: Comparison of Jury scores between DeepSeek V3.2 and Claude 4 generators on the MIT-BIH
2163 dataset.
2164

Metric	DeepSeek V3.2	Claude 4
Specificity	8.23 ± 0.24	8.08 ± 0.17
Unambiguity	7.55 ± 0.12	7.47 ± 0.28
Domain Relevance	8.69 ± 0.49	8.72 ± 0.22
Answerability	8.69 ± 0.14	8.53 ± 0.08
No Unintended Hints	7.46 ± 0.13	7.37 ± 0.18