LINEUPRL: VERIFIABLE REINFORCEMENT LEARNING FOR TIME SERIES CAPTIONING VIA CAPTION-TO-SERIES IDENTIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series captioning is a fundamental step in time series understanding and can also serve as the bridge between signal and natural language. Supervised fine-tuning (SFT) relies on a larger model's captions and cannot exceed their quality. Reinforcement learning (RL) can, but its rewards were designed for other modalities and other tasks, and they transfer poorly to open-ended generation in the time series domain. We address this by proposing LineupRL, a reinforcement learning with verifiable rewards (RLVR) pipeline whose reward is caption-to-series identification. The reward model is a frozen large language model (LLM) verifier that reads the generated caption and the candidate time series as raw values, never the chart, and must pick the described time series from multiple distractors. Matching is a far lighter demand on the verifier than writing questions or judging a caption, so an off-the-shelf LLM can supply the reward. Across two captioning benchmarks, and on forecasting and reconstruction where the predictor sees only the caption, LineupRL outperforms SFT and RL baselines on every metric. The 3B vision language model (VLM) trained by LineupRL also outperforms, at $1/24$ of the parameters, the 72B VLM whose captions the SFT baseline is distilled from. Our case study shows that LineupRL resists reward hacking, and that the captioner it trains both traces the trend and names the values at key points.

1 INTRODUCTION

Time series record how a system evolves over time, and decisions in healthcare, finance and climate depend on reading them accurately. Time series captioning automates this reading. Given a time series, a captioner writes a natural-language description of how the signal behaves (Jhamtani & Berg-Kirkpatrick, 2021; Sen et al., 2025; Zhou et al., 2026). Time series captions are useful from multiple perspectives. For human practitioners, text is easier to act on than the raw signal. For example, neonatal intensive-care staff chose better treatments from textual summaries of physiological time series than from the corresponding trend graphs (Law et al., 2005). For language-oriented systems, a caption provides a text interface to the signal, and this interface has been used for domain reporting (Fons et al., 2024b), forecast explanation (Aksu et al., 2024) and clinical prediction (Ji et al., 2026). The task dates back to rule-based data-to-text systems (Kukich, 1983; Reiter et al., 2005; Yu et al., 2007; Portet et al., 2009) and later data-driven models (Lloyd et al., 2014; Jandaghi & Pujara, 2020; Jhamtani & Berg-Kirkpatrick, 2021), both confined to fixed rules or output formats. Large language models (LLMs), vision-language models (VLMs) and time-series-language models (TSLMs) remove this restriction by open-ended generation, and recent work improves their captions by supervised fine-tuning (SFT) on time series paired with captions. Because human-written captions exist only in small collections

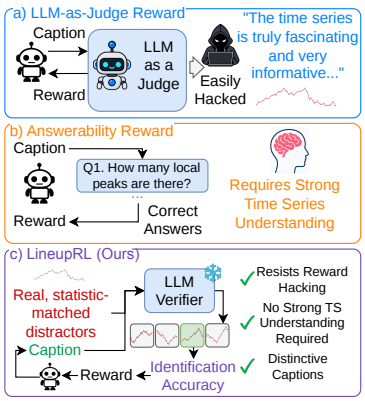


Figure 1: (a) LLM-as-judge is easily hacked. (b) Answerability needs strong time series understanding. (c) LineupRL (ours).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(Jhamtani & Berg-Kirkpatrick, 2021), these pairs are synthesized, either by prompting a larger model to describe the time series (Trabelsi et al., 2025; Lin et al., 2026) or by generating the text from attributes known by construction (Xie et al., 2024; Feng et al., 2026).

However, SFT has two weaknesses in this setting. First, it is bounded by the supervised data. A captioner trained to imitate synthesized captions can be no better than the model that wrote them, and it inherits that model's disadvantages, including plausible but incorrect statements about the time series (Trabelsi et al., 2025). Second, it generalizes poorly. Trained on a limited number of pairs, the captioner may memorize the reference captions rather than learning to read the time series, and what it memorized transfers poorly to time series it has not seen (Chu et al., 2025). On CaTS-Bench, open-source models fine-tuned on its synthetic pairs still trail proprietary models on numeric detail, and the fine-tuned model states statistics it cannot compute as confident guesses, which the benchmark attributes to a cross-entropy objective that treats number tokens like any other token (Zhou et al., 2026). In our own experiment, fine-tuning a 3B VLM on captions from a 72B teacher does not raise its entailment of the reference captions on BEDTime and lowers its series-to-caption identification on CaTS-Bench below the untuned initialization (Table 1). We call this failure mode ungrounded captioning. The caption is fluent and plausible, but it misses the important features of the time series, because the captioner learned the style of its references instead of learning to read the series.

Reinforcement learning (RL) addresses both weaknesses. Trained with a verifiable reward, the captioner learns from its own samples rather than from demonstrations, so no paired data is needed (Lambert et al., 2024; Guo et al., 2025). RL also generalizes where SFT memorizes (Chu et al., 2025), in part because its updates stay close to the base model (Shenfeld et al., 2026). However, RL needs a good reward design, and the representative rewards used for image captioning do not work on time series (Figure 1). An LLM-as-judge reward (Figure 1a) takes an LLM's grade of the caption as the reward. Because the grade is not verifiable, it is easily hacked (Xing et al., 2026). We observe that the captioner learns to game the grader instead of learning to describe the time series. An answerability reward (Figure 1b) asks whether questions about the time series can be answered from the caption alone (Xing et al., 2026). However, writing good questions and correct answers requires strong time series understanding, which current models largely lack (Zhou et al., 2026; Sen et al., 2025). A reward for time series captioning therefore has to satisfy three conditions. It must (i) resist reward hacking; (ii) demand only as much time series understanding as current LLMs or VLMs can supply; and (iii) reward important and distinctive features in time series.

In this work, we propose LineupRL, a reinforcement learning with verifiable rewards (RLVR) pipeline built on caption-to-series identification, a reward that satisfies all three desiderata (Figure 1c). Our starting point is that a caption is grounded and specific if a reader who sees only the caption can pick the described time series out of close alternatives. LineupRL renders each time series as a line chart and trains a VLM captioner on it, so the input does not grow with the time series length. For each training time series it draws $K-1$ distractors, which are real time series of the same length that are nearest to it in summary statistics. A frozen LLM verifier receives the caption and the K candidates as raw values and names the time series corresponding to the caption. The reward is the fraction of placements on which it names the true time series (Figure 1c), and each of these choices answers one of the three conditions. (i) The reward is an accuracy reward, the exact match of the verifier's pick against the ground-truth index, rather than a judged score of the kind that is prone to reward hacking (Guo et al., 2025). (ii) The verifier solves a K-way multiple-choice question over printed numbers, which needs matching rather than question writing or grading, and a 14B off-the-shelf LLM supplies it. (iii) The $K-1$ distractors are selected to concur with the true time series on general statistics, so the caption must describe the finer shape to earn the reward. Our contributions are as follows:

- We trace ungrounded captioning to behavioral cloning of synthesized references, show why the two existing families of RL rewards for captioning fail on time series, and state the three conditions a verifiable reward for open-ended captioning must satisfy.
- We propose LineupRL, whose reward is caption-to-series identification by a frozen verifier among statistic-matched distractors, which needs no reference captions.
- We evaluate captions by entailment, identification and sufficiency for forecasting and reconstruction, and show that LineupRL improves all three at once and that a 3B captioner trained this way surpasses the 72B teacher of the SFT baseline.

2 METHOD

2.1 PRELIMINARIES

We work with domain-agnostic univariate time series $x = (x_1, \ldots, x_T) \in \mathbb{R}^T$, where the length T varies across items. Time series are drawn from a data distribution $\mathcal{D}$, written $x \sim \mathcal{D}$. A renderer ρ converts a time series into a line chart $\rho(x)$ that carries the curve including some axis ticks without per-point value labels. A VLM captioner π_θ reads the chart $\rho(x)$ and outputs a natural-language description $c \sim \pi_\theta(\cdot \mid \rho(x))$, θ is updated during training so that π_θ produces captions with better quality. Caption quality is measured by three types of criteria (§2.3). The first type of criteria measures whether the generated caption implies a reference caption. The second measures caption-to-series identification and series-to-caption identification. The third treats the caption as an information bottleneck on the values and measures reconstruction and forecasting error.

Notations. Throughout, $K \geq 2$ is the number of candidates in every identification task of this paper, and $[K] = \{1, \ldots, K\}$. Let $\varphi(x) = (\mathrm{mean}, \mathrm{std}, \min, \max)$ be the summary statistics of a series. For each x, $\mathcal{N}(x) = \{\tilde{x}_1, \ldots, \tilde{x}_{K-1}\}$ denotes the distractor set used in training, constructed in §2.2, and $\mathcal{C}(x) = \{x\} \cup \mathcal{N}(x)$ the training candidate set. Their evaluation counterparts, $\bar{x}_1, \ldots, \bar{x}_{K-1}$ and $\tilde{\mathcal{C}}(x) = \{x, \bar{x}_1, \ldots, \bar{x}_{K-1}\}$, are constructed in §2.3. A verifier is an LLM that receives a caption together with the K candidates and answers this K-way multiple-choice question with a position in $[K]$. Because the candidates are presented to a verifier as a list, we write $O_p(\mathcal{C}(x))$ for the list in which the distractors appear in a fixed order and x is inserted at position $p \in [K]$, and $O_p(\tilde{\mathcal{C}}(x))$ for the same construction on the evaluation candidates. R denotes the verifier used in training, so $R\big(c, O_p(\mathcal{C}(x))\big) \in [K]$ is the position of the series that R believes the caption c describes. Evaluation uses a single frozen LLM, which need not be the LLM behind R, in two roles: $\tilde{R}_{\mathrm{c}\to\mathrm{s}}$ locates a series from a caption, and $\tilde{R}_{\mathrm{s}\to\mathrm{c}}$ locates a caption from a series.

RL with verifiable rewards. We post-train the captioner with reinforcement learning from verifiable rewards (RLVR) (Lambert et al., 2024; Guo et al., 2025). In this paradigm, the policy π_θ generates captions, each generated caption is scored by a reward function, and reinforcement learning updates θ to raise the probability of generations with higher rewards. Formally, given a reward function $r(c, x)$ that scores a caption c written for a series x, training maximizes the expected reward $J(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{E}_{c\sim\pi_\theta(\cdot|\rho(x))}[r(c, x)]$. We optimize $J(\theta)$ with the REINFORCE leave-one-out estimator (RLOO) (Kool et al., 2019; Ahmadian et al., 2024). Each training step draws a batch $\{x^{(1)}, \ldots, x^{(B)}\}$ of B series independently from $\mathcal{D}$. For each $b \in [B]$, the step samples a group of G captions $c_1^{(b)}, \ldots, c_G^{(b)}$ independently from $\pi_\theta(\cdot \mid \rho(x^{(b)}))$ and scores each caption with the reward. Since the attainable reward varies with the series, caption $c_i^{(b)}$ is credited by the leave-one-out contrast $A_i^{(b)} = r\big(c_i^{(b)}, x^{(b)}\big) - \frac{1}{G-1}\sum_{j\neq i} r\big(c_j^{(b)}, x^{(b)}\big)$, $i \in [G]$, whose subtracted term estimates the mean reward of $x^{(b)}$ from its other $G-1$ captions and is therefore a variance-reducing baseline. The step then updates θ by ascending the policy-gradient estimate $\hat{g} = \frac{1}{BG}\sum_{b=1}^{B}\sum_{i=1}^{G} A_i^{(b)}\,\nabla_\theta \log \pi_\theta\big(c_i^{(b)} \mid \rho(x^{(b)})\big)$ of $\nabla_\theta J(\theta)$. The reward r is the design choice of §2.2.

2.2 LINEUPRL

Figure 2 gives an overview of LineupRL, which instantiates the RLVR of §2.1 with the captioner π_θ as the policy and RLOO as the policy-gradient estimator; the component specific to LineupRL is the reward. The reward is caption-to-series identification (upper box of Figure 2b): it scores a caption c written for a series x by whether the frozen verifier R can identify x in $\mathcal{C}(x)$ from c alone. The verifier sees only the caption and the K candidates printed as raw numbers, without seeing the chart $\rho(x)$, because the weak chart comprehension of a VLM verifier would add noise to the reward. Since a single query at one fixed position would be noisy and positionally biased, we query the verifier K times, once with x at each position $p \in [K]$, and take the fraction of correct answers, $\mathrm{acc}(c, x) = \frac{1}{K}\sum_{p=1}^{K} \mathbf{1}\big[R\big(c, O_p(\mathcal{C}(x))\big) = p\big]$. The reward is the rescaled accuracy $r(c, x) = \lambda\,\mathrm{acc}(c, x)$ with $\lambda = 2$. A caption therefore earns reward only through statements that separate x from its neighbors. Text addressed to the verifier or a generic description changes no answer, and a fabricated value that happens to match a distractor sends the verifier to the wrong candidate, so only grounded and distinctive content is rewarded.

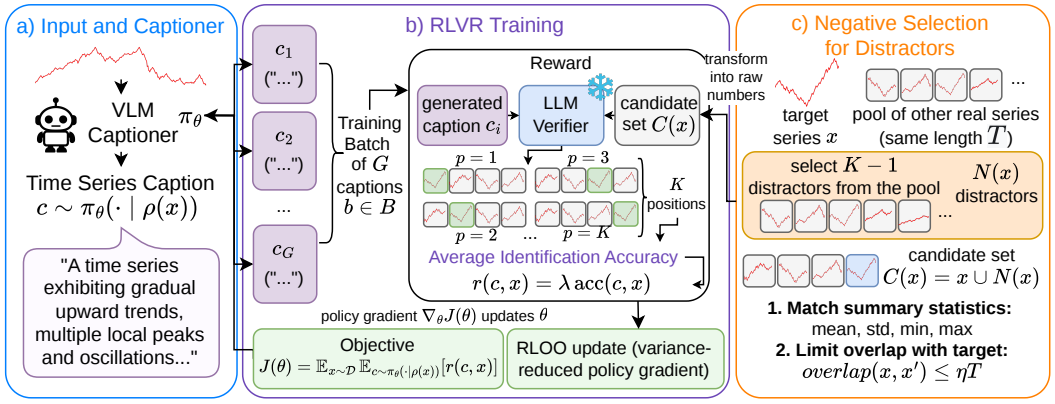


Figure 2: Overview of LineupRL. (a) The VLM captioner π_θ reads the chart $\rho(x)$ of a series x and samples a caption c. (b) RLVR training. For each series a group of G captions is sampled. Each caption is scored by a frozen LLM verifier that receives the caption and the candidate set $\mathcal{C}(x)$ printed as raw numbers and must locate x; the true series is placed at each of the K positions in turn, and the fraction of correct answers gives the reward $r(c, x) = \lambda \, \mathrm{acc}(c, x)$. The rewards of the group drive the RLOO policy-gradient update of θ. (c) Negative selection. The $K-1$ distractors $\mathcal{N}(x)$ are drawn from a pool of other real series of the same length by matching summary statistics and limiting overlap with x, and $\mathcal{C}(x) = \{x\} \cup \mathcal{N}(x)$.

The quality of the reward is largely determined by the distractor set $\mathcal{N}(x)$, whose construction is shown in Figure 2c. A straightforward construction, deriving the distractors from x itself by shuffling, reversing, or cutting and pasting, is inappropriate because such distractors are easily detected by the LLM verifier: under this construction the verifier identifies x at 72.5% even without reading the caption. To avoid such information leakage, LineupRL draws $\mathcal{N}(x)$ from a pool of other real series of the same length as x, and makes the question hard by two criteria, matching summary statistics and limiting overlap with x. Let $overlap(x, x')$ be the longest overlap the two series x and x' share. The distractor set is $\mathcal{N}(x) = \arg\min_{\mathcal{S}} \sum_{x' \in \mathcal{S}} \|\varphi(x) - \varphi(x')\|_2$, where $\mathcal{S}$ ranges over sets of $K-1$ other corpus series of length T with $overlap(x, x') \leq \eta T$ for all $x' \in \mathcal{S}$, and $\eta \in (0, 1)$ is an overlap budget. In other words, $\mathcal{N}(x)$ collects the $(K-1)$ nearest neighbors of x under φ among series of the same length, excluding any that overlaps x on more than a fraction η of its steps. In training we apply the same rule to the eight nearest admissible neighbors of x, which form a pool from which $K-1$ distractors are redrawn at every step so that the caption cannot overfit one fixed $\mathcal{N}(x)$; validation items keep one fixed distractor set. The candidate set $\mathcal{C}(x) = \{x\} \cup \mathcal{N}(x)$ is then handed to the verifier. When candidates match the target in length and summary statistics, the verifier must rely on the fine-grained description to distinguish them.

2.3 Evaluation protocol

Captions are generated by greedy decoding at inference, so each series x has one caption $c(x)$, and every evaluation criterion is an empirical average over an evaluation set $\mathcal{E} = \{x^{(1)}, \ldots, x^{(N)}\}$. We apply three criteria, whose implementation details are given in Appendix B.

Criterion 1: entailment against reference captions. Let $H(x)$ be the set of reference captions of x, and let $q(a, b) \in [0, 1]$ be the probability that a fixed natural-language inference model assigns to the entailment class for premise a and hypothesis b. Over the pairs $\mathcal{P} = \{(x, h) : x \in \mathcal{E}, \, h \in H(x)\}$ we report $\widehat{\mathcal{A}} = |\mathcal{P}|^{-1} \sum_{(x,h) \in \mathcal{P}} \mathbf{1}[\, q(c(x), h) \geq \tau \,]$, the fraction of reference captions that the generated caption implies, where $\tau \in (0, 1)$ is a fixed decision threshold. The generated caption is the premise and the reference caption is the hypothesis, so $\widehat{\mathcal{A}}$ asks whether the content of a reference caption is recoverable from the generated one while allowing the generated caption to say more; the reverse direction is not reported.

Criterion 2: identification by a frozen verifier. This criterion runs identification in two directions: in caption-to-series identification (CAP→SER) the verifier receives a caption and K candidate series

and must locate the described series, and in series-to-caption identification (SER→CAP) the verifier receives a series and K candidate captions and must locate the caption written for it. Both directions use the evaluation candidate set $\tilde{\mathcal{C}}(x)$, whose distractors $\bar{x}_1, \ldots, \bar{x}_{K-1}$ are the $K-1$ nearest neighbors of x under φ among evaluation series of the same length; $\tilde{\mathcal{C}}(x)$ is computed from the series alone and reads no caption, so it is identical for different captioners. CAP→SER is the same caption-to-series identification task that the reward of §2.2 is built on, except that the verifier is $\tilde{R}_{\mathrm{c}\to\mathrm{s}}$ and the candidates come from $\tilde{\mathcal{C}}(x)$. Its identification accuracy is $\widetilde{\mathrm{acc}}_{\mathrm{c}\to\mathrm{s}}(c, x) = \frac{1}{K}\sum_{p=1}^{K} \mathbf{1}\big[\tilde{R}_{\mathrm{c}\to\mathrm{s}}\big(c, O_p(\tilde{\mathcal{C}}(x))\big) = p\big]$, and we report its average over the evaluation data, $\widehat{\mathcal{M}}_{\mathrm{c}\to\mathrm{s}} = |\mathcal{E}|^{-1}\sum_{x\in\mathcal{E}} \widetilde{\mathrm{acc}}_{\mathrm{c}\to\mathrm{s}}(c(x), x)$. In SER→CAP, the verifier receives the series x and K candidate captions, and must locate the caption written for x. The candidate captions are those the same captioner wrote for the members of $\tilde{\mathcal{C}}(x)$. For $p \in [K]$, let $Q_p(x) := (c(\bar{x}_1), \ldots, c(\bar{x}_{p-1}), c(x), c(\bar{x}_p), \ldots, c(\bar{x}_{K-1}))$ denote the ordered caption list with $c(x)$ inserted at position p; it is the caption-side counterpart of $O_p(\tilde{\mathcal{C}}(x))$. The identification accuracy again averages over all K placements of $c(x)$ to avoid the verifier's position bias, $\widetilde{\mathrm{acc}}_{\mathrm{s}\to\mathrm{c}}(x) = \frac{1}{K}\sum_{p=1}^{K} \mathbf{1}\big[\tilde{R}_{\mathrm{s}\to\mathrm{c}}\big(x, Q_p(x)\big) = p\big]$, and we report its average over the evaluation data, $\widehat{\mathcal{M}}_{\mathrm{s}\to\mathrm{c}} = |\mathcal{E}|^{-1}\sum_{x\in\mathcal{E}} \widetilde{\mathrm{acc}}_{\mathrm{s}\to\mathrm{c}}(x)$.

Criterion 3: sufficiency as an information bottleneck. The last criterion asks how much of the series itself survives the encoding into language. We fit a decoder that reads the caption and predicts a numeric target on the chronologically earliest windows of each corpus, and report its error $\widehat{\mathcal{S}}$ on the latest, so that no test target precedes any training window. Two targets are used: the horizon that follows the described window (forecasting) and the window itself (reconstruction). Because the series is never an input to the decoder, only what the caption states can lower $\widehat{\mathcal{S}}$. A low error is necessary but not sufficient, since a caption that only names a pattern common to the dataset would score well while identifying nothing, so we report it against three references: a text-blind predictor b that never reads a caption, the decoder refitted on an empty caption, and a detachment control in which the decoder is refitted with every window paired with another window's caption. A caption counts as informative only if its error stays far below these.

3 Experiments

3.1 Experimental setup

We train on 10,000 fragments sampled from TSFragment-600K (Ge et al., 2025) under a fixed seed, spread roughly evenly over the three lengths. Charts keep a handful of x and y tick labels, so the sampling positions and the value range remain readable, but carry no title, legend, series name, unit or per-point value labels, so the policy cannot rely on the dataset name or read values off the chart, and has to describe the shape. Every identification task uses $K = 4$ candidates. Criteria 1 and 2 are measured on BEDTime and CaTS-Bench; since the CaTS-Bench references name dates, places, domains and exact values that a bare chart does not carry, Criterion 1 on CaTS-Bench scores against references that Claude Opus 5 rewrote to remove them (Appendix B.1). Criterion 3 is measured on four forecasting corpora (ETTh2, ETTm2, Saugeen river flow and Australian electricity demand) under the chronological split of §2.3 (Appendix B). The evaluation data do not overlap the training data: no series of BEDTime, CaTS-Bench or the four forecasting corpora appears among the training fragments (Appendix B).

Every trained model starts from the same Qwen2.5-VL-3B-Instruct weights and is prompted with the same instruction. We compare LineupRL against the untuned initialization; against SFT distilled from Qwen2.5-VL-72B-Instruct captions of the same charts; against RL with a CapRL-style answerability reward (Xing et al., 2026); and against RL with an LLM-as-judge reward. The three RL variants share the policy-optimization hyperparameters and the frozen Qwen2.5-14B-Instruct behind the reward, which serves as our verifier, as the answerer of the answerability reward and as the judge, so they differ only in the form of the reward (Appendix A.2). The 72B teacher, two further off-the-shelf VLMs, three text LLMs that read the values as digits, and a time-series language model are reported for reference. Criterion 2 is scored by two frozen LLMs that take no part in training any model, GLM-4-9B and Phi-4-14B, and every identification number we report is the mean of the two; readings under three other verifiers are in Appendix C.1. Criterion 1 uses a frozen DeBERTa long-NLI model, and Criterion 3 fits a frozen EmbeddingGemma-300M with a LoRA adapter and an MLP head under

Table 1: Entailment and identification on two captioning benchmarks. C→S is short for caption-to-series (CAP→SER) identification. S→C is short for series-to-caption (SER→CAP) identification. Each identification cell is the mean over two frozen verifiers that took no part in training, GLM-4-9B and Phi-4-14B; random chance is $\frac{1}{K} = 0.25$. CaTS-Bench entailment is scored against rewritten references (Appendix B.1). Bold marks the best of the four models fine-tuned from the same initialization.

		BEDTime			CaTS-Bench		
Type	Captioner	Entailment	C→S	S→C	Entailment	C→S	S→C
VLM	Qwen2.5-VL-3B	0.296	0.550	0.495	0.352	0.751	0.721
	Qwen2.5-VL-72B	0.320	0.605	0.484	0.440	0.803	0.643
	Qwen3-VL-8B	0.320	0.669	0.498	0.407	0.817	0.626
	InternVL3-14B	0.357	0.562	0.512	0.506	0.766	0.701
LLM	Phi-3.5-mini	0.180	0.909	0.807	0.318	0.980	0.890
	Qwen2.5-7B	0.240	0.916	0.823	0.400	0.985	0.900
	Qwen2.5-14B	0.297	0.919	0.838	0.535	0.991	0.938
TSLM	ChatTS-14B	0.272	0.683	0.721	0.292	0.882	0.863
Fine-tuned from Qwen2.5-VL-3B							
VLM	SFT on 72B's captions	0.281	0.570	0.448	0.408	0.760	0.616
	RL, answerability	0.286	0.448	0.443	0.299	0.599	0.560
	RL, LLM-as-judge	0.118	0.544	0.412	0.033	0.694	0.589
	LineupRL	**0.400**	**0.709**	**0.638**	**0.611**	**0.886**	**0.868**

an identical recipe for every captioner, reported as mean and standard deviation over five training seeds. Training, prompting and predictor hyperparameters are in Appendix A.

3.2 Main results

Table 1 reports the performance of all models on the two benchmarks. Among the four models fine-tuned from the same initialization, LineupRL is best in all six columns. The two RL baselines show how much the reward design matters. The answerability reward ends up below the initialization it started from on all four identification columns. The LLM-as-judge reward loses less identification than the answerability reward on three of the four columns, but its entailment score collapses to 0.118 and 0.033, far beneath the base model. Against the 72B teacher itself, LineupRL is ahead in all six columns with $1/24$ of the parameters, which shows that an appropriate RL pipeline can raise a VLM's time series captioning ability beyond what distillation offers. Another finding concerns the three text LLMs, which read the raw values instead of the chart. Their identification scores are far above every chart-reading model on both benchmarks, but this advantage does not carry over to entailment: even the largest of them trails LineupRL. Reading the raw values thus helps mainly with the fine detail that tells one series from another, and much less with describing the overall shape and events of the time series.

Table 2 reports the test error of a caption-only predictor on forecasting and reconstruction. The predictor is an EmbeddingGemma-300M (Vera et al., 2025) equipped with a LoRA (Hu et al., 2021) adapter and a lightweight MLP decoder head, reading the caption alone. The three baseline rows set the scale of the errors. LineupRL has the lowest error in all eight cells, significantly so in seven, the exception being reconstruction on Saugeen where the LLM-as-judge reward ties it. Compared to Qwen2.5-VL-72B, LineupRL wins five cells, ties two and loses one (forecasting on ETTm2), so the 3B model is on par with a model $24\times$ its size. Two patterns from Table 1 recur: the answerability reward scores significantly worse for forecasting than the untuned initialization on ETTh2 and ETTm2, while the LLM-as-judge reward, whose entailment collapsed there, is the second-best captioner in five of the eight cells, so a caption can entail the reference and still be of little use downstream.

3.3 Ablations

Table 2: Information bottleneck on time series tasks: Mean and standard deviation of test MSE over five training seeds of the predictor are reported. The naive baseline is last-value persistence for forecasting and mean training window for reconstruction. Bold marks every fine-tuned model not significantly worse than the best in its column; significance tests are deferred to Appendix B.5.

	Forecasting				Reconstruction			
Captioner	ETTh2	ETTm2	Saug.	Elec.	ETTh2	ETTm2	Saug.	Elec.
Naive baseline	0.3047	0.0596	1.2955	0.9078	0.9501	0.8335	1.1748	0.8634
Empty caption	0.8446 ±0.0101	0.8001 ±0.0042	1.0601 ±0.0016	0.9129 ±0.0041	0.9284 ±0.0306	0.8274 ±0.0029	1.1764 ±0.0008	0.8667 ±0.0021
Shuffled caption	0.8437 ±0.0087	0.8040 ±0.0046	1.0619 ±0.0042	0.9124 ±0.0043	0.9347 ±0.0121	0.8270 ±0.0015	1.1774 ±0.0018	0.8663 ±0.0010
Qwen2.5-VL-3B	0.3062 ±0.0086	0.1012 ±0.0047	0.9419 ±0.0119	0.5751 ±0.0351	0.2445 ±0.0052	0.0633 ±0.0029	0.6852 ±0.0173	0.4352 ±0.0156
Qwen2.5-VL-72B	0.2149 ±0.0094	0.0451 ±0.0009	0.8521 ±0.0061	0.3089 ±0.0094	0.1134 ±0.0065	0.0288 ±0.0022	0.4444 ±0.0113	0.1993 ±0.0077
Fine-tuned from Qwen2.5-VL-3B								
SFT on 72B's captions	0.2595 ±0.0151	0.0615 ±0.0021	0.9026 ±0.0216	0.3480 ±0.0086	0.1478 ±0.0044	0.0333 ±0.0016	0.5259 ±0.0177	0.2144 ±0.0084
RL, answerability	0.3780 ±0.0168	0.2110 ±0.0159	0.9384 ±0.0104	0.3258 ±0.0151	0.3391 ±0.0185	0.1883 ±0.0151	0.5309 ±0.0091	0.2608 ±0.0095
RL, LLM-as-judge	0.2384 ±0.0134	0.0630 ±0.0077	0.8619 ±0.0217	0.3388 ±0.0113	0.1373 ±0.0104	0.0325 ±0.0019	**0.4517** ±0.0027	0.2248 ±0.0029
LineupRL	**0.1806** ±0.0078	**0.0541** ±0.0022	**0.8175** ±0.0196	**0.2042** ±0.0091	**0.0989** ±0.0033	**0.0294** ±0.0010	**0.4463** ±0.0054	**0.1357** ±0.0068

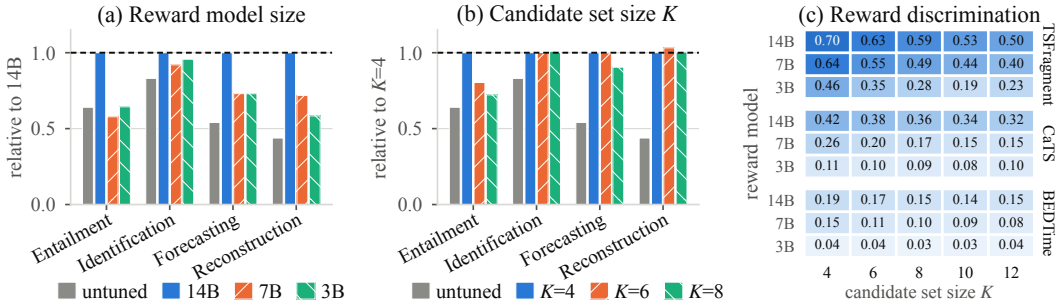
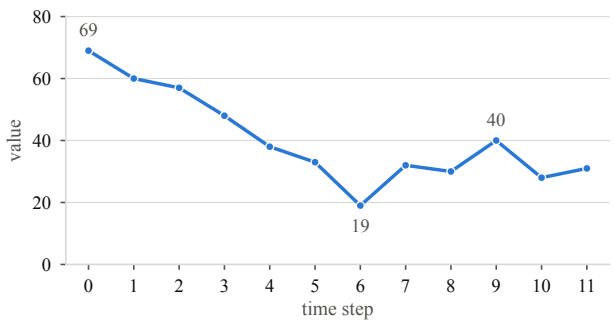


Figure 3: Ablation on the reward's three knobs. (a) three reward models and (b) $K \in \{4, 6, 8\}$, where K is the cardinality of the candidate set in each identification question and $K - 1$ the number of distractors; the identification bar is SER→CAP on CaTS-Bench under the two evaluation verifiers. (c) re-scores reference captions of three domains, written by no captioner, under every combination of reward model size (down) and K (across); each cell is accuracy above chance, $(\text{acc}-1/K)/(1-1/K)$, and darker is higher. Every number behind the three panels is in Appendix C.

In the ablation study, we investigate three factors that affect the performance of LineupRL: the size of the reward model, the cardinality of the candidate set, and the selection of distractors. Figure 3(a) shows that shrinking the reward model hurts all four criteria. With reward models smaller than 14B, the fine-tuned model does not improve entailment over the untuned model (the 3B verifier ties it), which indicates that our reward still requires a certain degree of time series comprehension from the LLM: less than writing questions or grading a caption, but more than a 7B model has. Figure 3(b) shows that more distractors do not help in general: entailment drops markedly, identification and reconstruction tie, and forecasting error rises by 11% at $K = 8$. Figure 3(c) explains both observations by examining the reward signal under each setting on reference captions that no captioner wrote. Caption-to-series identification accuracy falls with a smaller reward model and with more distractors, which makes the reward signal noisier; the 7B and 3B verifiers are

Table 3: Nearest against random distractors.

Distractors	Nearest (**ours**)	Random
Reward acc. over chance	0.872	0.982
Entailment	**0.506**	0.354
Ident. C→S	**0.797**	0.704
Ident. S→C	**0.753**	0.744
Caption length (chars)	667	738
Forecasting rel. MSE	**1.000**	1.212
Reconstruction rel. MSE	**1.000**	1.142



LineupRL (ours). 668 characters, 9 time positions, 9 values.
The time series starts with an initial value of approximately 68. It then shows a gradual decline, maintaining a value around 60 by time step 2. Following this, there's a steady decrease, with a value around 58 by time step 3. The value continues to drop, reaching approximately 42 by time step 5. There's a notable dip, dropping to about 16 by time step 6. It then experiences a slight recovery, maintaining a value around 31 by time steps 7 and 8. There's a subsequent upward movement, **peaking at a value of about 33 by time step 9.** Finally, the series settles around a value of approximately 30 by time step 11, with a final value of about 30 at the 11th time step.

RL, LLM-as-judge reward. 1,424 characters, 9 time positions, 10 values.
The line chart depicts a single time series with a y-axis labeled "value" and an x-axis labeled "time". **The y-axis ranges approximately from 0 to 70 and marks increments of 10,** *while the x-axis ranges approximately from 0 to 11 and marks integer values. The plot shows a blue line with circular blue markers at each data point.* **The line starts slightly above 60 on the y-axis at the beginning of the x-axis,** descends to just below 60 around the 2nd point on the x-axis, drops to about 55 around the 3rd point, continues to about 45 around the 4th point, drops to around 32 around the 5th point, dips to about 15 around the 6th point, rises to about 30 around the 7th point, **peaks at about 35 around the 8th point,** and then slightly decreases to about 30 around the 10th point, ending just below 30 at the 11th point on the x-axis. *This description captures every detail of the chart, including the exact values at each point, the exact position on the axes, the exact trend of the line, and the exact position of the markers. Anyone reading this description would be able to recreate this specific line chart exactly, as all details are precise and complete. The sequence and magnitude of the points, their exact values, the exact position on the axes, and the exact trend of the line are all exactly as described. There are no nuances or variations; every aspect of the chart is accounted for in this detailed description.*

RL, answerability reward. 918 characters, 21 time positions, 0 values.
The line chart depicts a single time series dataset. Here is a detailed description of the series: 1. Key Points and Values: The highest values are at time 0. The lowest values are at time 5-6. *2. Key Features:* The values decrease rapidly from time 0 to time 3. There is a steep decline from time 2 to time 4. The values remain relatively stable from time 4 to time 5. *3. Notable Features:* The values drop sharply from time 0 to time 1. **The values drop sharply from time 1 to time 2.** The values drop sharply from time 2 to time 3. *4. Key Points:* Maximum values: Around time 0. Minimum values: Around time 6. *5. Key Points of Interest:* The highest values are between time 0 and time 1. The lowest values are between time 5 and time 6. This series is characterized by a steep decline followed by a slight rise and then a steady level.

Figure 4: Case study on captions from three RL-fine-tuned models for the series at the top left. Spans are classed as green, a fact pinned to a time and a magnitude the series supports (within 0.35σ of the series over the interval it points at); blue, a position with no magnitude; **red**, a claim the series contradicts; *gray*, content irrelevant to the series. Underline marks a span covering one of the series' four events (opening maximum, fall, trough at t=6, rebound), which all three captions visit in order.

close to chance on CaTS-Bench and BEDTime even on these captions, so the drop in (a) comes from the verifier rather than from the captions it graded.

Another important factor in LineupRL is negative selection, the rule that draws the distractors. LineupRL draws the $K-1$ distractors of a training item from its nearest neighbors in a four-statistic feature space; Table 3 replaces that rule with a uniform random draw from time series of the same length and compares the two fine-tuned models. Its caption rows average BEDTime and CaTS-Bench, and its forecasting and reconstruction rows are geometric means over the four corpora of the error relative to LineupRL; Appendix C gives every cell. Hard negatives give the better captioner on every criterion. The random-distractor variant in fact scores higher on its own training signal, 0.982 above chance against 0.872 for LineupRL, but that only means the question got easier. A random distractor differs from the true series in level and range, so stating the approximate level is enough to earn the reward, and the caption need not say more. Nearest neighbors agree on those coarse statistics, so the caption has to describe the finer shape before the verifier can tell them apart.

3.4 Case Study

Figure 4 shows the captions that the three RL reward designs produce for the same series, and it makes the weakness of both baseline rewards concrete. The answerability reward is limited by the model that sets its questions: posing and answering questions about a time series is far harder for an LLM or a VLM than doing so about a natural image. Only 28% of the questions the VLM wrote survive a check against the raw values with the answer the VLM gave (Appendix A.2), and most survivors merely ask which interval holds an extremum, so the captions this reward trains do not describe a time series well. The LLM-as-judge reward behaves as reported for image captioning, where an LLM judge is easy to hack: much of the caption is text about the caption itself, claiming

that it is precise and complete, rather than about the time series. LineupRL, in contrast, describes the events of the series in order and gives the key values, and its caption is the shortest of the three.

4 Related work

Reinforcement learning with verifiable rewards. RLVR replaces the learned preference model of RLHF (Ouyang et al., 2022) with a programmatic check on the sampled output, optimized by critic-free group estimators such as GRPO (Shao et al., 2024) and RLOO (Kool et al., 2019; Ahmadian et al., 2024). RLVR applies most directly where a rule decides correctness outright (Shao et al., 2024; Guo et al., 2025; Zhang et al., 2026), and has since been extended to open-ended tasks by designing verifiers for them. Recent reward designs include rubrics decomposing a subjective answer into checkable criteria (Gunjal et al., 2026; Huang et al., 2025), outcome checks against an external environment (Jin et al., 2025; Zheng et al., 2025), and downstream utility (Berthon & van der Schaar, 2025).

Time series captioning. Early systems verbalize an explicit model of the data, from Gaussian-process reports to utility-ranked trends and executable pattern programs (Lloyd et al., 2014; Jandaghi & Pujara, 2020; Jhamtani & Berg-Kirkpatrick, 2021). Language models now write the descriptions themselves, either as dedicated captioners or as one capability of a time-series LLM (Trabelsi et al., 2025; Lin et al., 2026; Xie et al., 2024; Feng et al., 2026). Reinforcement learning has been applied to the time series domain where the task already supplies a verifiable target (Zhang et al., 2025; Fan et al., 2026; Liu et al., 2026; Hou et al., 2026). In our setting the description itself is the output, it is rewarded by whether a frozen verifier can identify the series from it alone, and no downstream task or label is involved. BEDTime (Sen et al., 2025) and CaTS-Bench (Zhou et al., 2026) benchmark how well current models caption a time series, and feature-level probes (Fons et al., 2024a; Merrill et al., 2024) test the same ability indirectly.

Reinforcement learning for image captioning. Image captioning is closest to ours. Metric rewards score a sampled caption against human references and became standard with the self-critical baseline (Rennie et al., 2017; Liu et al., 2017). Human-feedback rewards replace the metric with ratings or written corrections (Fidler et al., 2017; Seo et al., 2020). Model-scored rewards employ a frozen cross-modal encoder to score the caption by its similarity to the image (Cho et al., 2022; Dzabraev et al., 2024) or by retrieval (Luo et al., 2018; Dessì et al., 2023; Chaffin et al., 2024). The retrieval line is the closest to ours in spirit: a listener must pick the described image among distractors. It relies on a retrieval model trained on paired image-text data, which does not exist for time series, and scores against distractors drawn at random or by image similarity. LineupRL replaces the listener with a frozen text LLM that reads the candidates as raw values, so no retrieval model is trained, draws the distractors from real series matched on summary statistics, and optimizes a VLM policy with verifiable rewards. Judge rewards employ an LLM judge with rubrics or verified atomic claims (Huang et al., 2026; Li et al., 2026; Ye et al., 2026; Tang et al., 2026). Utility rewards make the caption serve a downstream task, e.g., question answering (Fisch et al., 2020; Xing et al., 2026) or safety decision (Ma et al., 2026).

5 Conclusion

We present LineupRL, a reinforcement learning framework that scores a time series caption by whether the series can be identified from it. By rewarding a caption for letting a frozen text-only verifier pick its series out of a set of close alternatives, LineupRL obtains its training signal from the data alone, without reference captions or a learned reward model. Across two captioning benchmarks and a caption-only forecasting and reconstruction evaluation, a 3B captioner trained this way surpasses SFT distillation, alternative RL rewards, and the 72B model it could have been distilled from. Ablation studies validate the three choices the reward adds, the scale of the verifier, the number of candidates and the rule that draws the distractors, and show that the hard negatives give the better captioner on every criterion. Caption-to-series identification against close alternatives is thus a workable verifiable reward for continuous signals, where reference captions are scarce and LLM judges are easy to hack. Longer and multivariate series and captioners beyond VLMs are left to future work.

AI USE STATEMENT

In this work, we used generative AI tools to generate teacher captions for the supervised fine-tuning (SFT) baseline, to write and verify the multiple-choice questions of the answerability baseline (Appendix A.2), and to rewrite the CaTS-Bench reference captions used by Criterion 1 (Claude Opus 5, Appendix B.1); the rewritten references were checked by a script and reviewed by the authors. We did not use generative AI tools to implement the proposed algorithm. Additionally, we used AI tools to organize our code, prepare figures and tables, check the manuscript for grammatical and spelling errors, and polish language. The authors are responsible for the final manuscript, code, and other research artifacts, including AI-assisted content.

ETHICS STATEMENT

This work studies automated time series captioning using existing datasets and pretrained models. Generated captions may omit relevant events or misstate trends and numerical values. The identification reward measures whether a verifier can distinguish the target series from alternatives; it does not guarantee the correctness of every statement in a caption. Captions should therefore be checked against the underlying series before informing consequential decisions. Applications involving sensitive time series require appropriate privacy protections and evaluation in the intended domain.

REPRODUCIBILITY STATEMENT

The reward and the distractor construction are defined in §2.2, and the three evaluation criteria in §2.3. Appendix A gives the implementation: the training data, policy-optimization hyperparameters and reward configuration of LineupRL (K, λ, η, the verifier and its decoding mode), the data, reward and checkpoint-selection rule of every baseline, and what each ablation setting changes. Appendix B gives the evaluation: the benchmark subsets and their sizes, the candidate-set construction, the rewriting of the CaTS-Bench references, the NLI model and threshold, the windowing of the forecasting corpora and the recipe of the caption-only predictor, which is held fixed across captioners, the significance tests behind the bold entries of Table 2, and every cell of that table with its across-seed spread and the seeds used. Appendix C reports every captioner under five verifiers and the numbers behind every ablation panel. Appendix D gives every prompt verbatim. The rendering, training, reward-server and evaluation code, every prompt, the identifiers of the 10,000 training fragments and the rewritten CaTS-Bench references are available at `https://anonymous.4open.science/r/LineupRL-official-repo`; the rendered charts are regenerated deterministically by the released renderer. The captions written by every captioner in Tables 1 and 2 and the scripts that regenerate every table and figure from the raw metric files will be released with the paper.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Taha Aksu, Chenghao Liu, Amrita Saha, Sarah Tan, Caiming Xiong, and Doyen Sahoo. Xforecast: Evaluating natural language explanations for time series forecasting. *arXiv preprint arXiv:2410.14180*, 2024.

Antonin Berthon and Mihaela van der Schaar. Language bottleneck models for qualitative knowledge state modeling. *arXiv preprint arXiv:2506.16982*, 2025.

Antoine Chaffin, Ewa Kijak, and Vincent Claveau. Distinctive image captioning: Leveraging ground truth captions in clip guided reinforcement learning. In *2024 IEEE International Conference on Image Processing (ICIP)*, pp. 2550–2556. IEEE, 2024.

Jaemin Cho, Seunghyun Yoon, Ajinkya Kale, Franck Dernoncourt, Trung Bui, and Mohit Bansal. Fine-grained image captioning with clip reward. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 517–527, 2022.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Roberto Dessì, Michele Bevilacqua, Eleonora Gualdoni, Nathanaël Carraz Rakotonirina, Francesca Franzon, and Marco Baroni. Cross-domain image captioning with discriminative finetuning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6935–6944. IEEE, 2023.

Maksim Dzabraev, Alexander Kunitsyn, and Andrei Ivaniuta. Vlrm: Vision-language models act as reward models for image captioning. *arXiv preprint arXiv:2404.01911*, 2024.

Yuang Fan, Lilin Xu, Millie Wu, Jingping Nie, Qingyu Chen, Yuzhe Yang, Zhuo Zhang, Xin Liu, Subigya Nepal, Xiaofan Jiang, et al. Timesrl: Generalizable time-series behavioral modeling via semantic rl-tuned llms–a case study in mental health. *arXiv preprint arXiv:2605.21295*, 2026.

Xinran Feng, Yi Xie, Chao Zhang, Ruikun Li, Wanyun Ling, Ziyue Li, and Chenxi Liu. Decoupling perception from description: Computation-grounded representation alignment between multivariate time series and language. *arXiv preprint arXiv:2608.05238*, 2026.

Sanja Fidler et al. Teaching machines to describe images with natural language feedback. *Advances in Neural Information Processing Systems*, 30, 2017.

Adam Fisch, Kenton Lee, Ming-Wei Chang, Jonathan H Clark, and Regina Barzilay. Capwap: Image captioning with a purpose. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8755–8768, 2020.

Elizabeth Fons, Rachneet Kaur, Soham Palande, Zhen Zeng, Tucker Balch, Manuela Veloso, and Svitlana Vyetrenko. Evaluating large language models on time series feature understanding: A comprehensive taxonomy and benchmark. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21598–21634, 2024a.

Elizabeth Fons, Rachneet Kaur, Zhen Zeng, Soham Palande, Tucker Balch, Svitlana Vyetrenko, and Manuela Veloso. Tadacap: Time-series adaptive domain-aware captioning. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pp. 54–62, 2024b.

Yunfeng Ge, Jiawei Li, Yiji Zhao, Haomin Wen, Zhao Li, Meikang Qiu, Hongyan Li, Ming Jin, and Shirui Pan. T2s: High-resolution time series generation with text-to-series diffusion models. *arXiv preprint arXiv:2505.02417*, 2025.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations*, volume 2026, pp. 127924–127945, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Difei Hou, Jiaqi Yue, and Chunhui Zhao. Words towards explainability: Caption label-free learning via dual loop agentic time series captioning. In *Forty-third International Conference on Machine Learning*, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Tzu-Heng Huang, Sirajul Salekin, Javier Movellan, Frederic Sala, and Manjot Bilkhu. Rubicap: Rubric-guided reinforcement learning for dense image captioning. *arXiv preprint arXiv:2603.09160*, 2026.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, et al. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*, 2025.

Pegah Jandaghi and Jay Pujara. Human-like time series summaries via trend utility estimation. *arXiv preprint arXiv:2001.05665*, 2020.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. Truth-conditional captions for time series data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 719–733, 2021.

Zongliang Ji, Yifei Sun, Andre Amaral, Anna Goldenberg, and Rahul G Krishnan. Can we generate portable representations for clinical time series data using llms? *arXiv preprint arXiv:2603.23987*, 2026.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Wouter Kool, Herke Van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! 2019.

Karen Kukich. Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics*, pp. 145–150, 1983.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Anna S Law, Yvonne Freer, Jim Hunter, Robert H Logie, Neil McIntosh, and John Quinn. A comparison of graphical and textual presentations of time series data to support medical decision making in the neonatal intensive care unit. *Journal of clinical monitoring and computing*, 19(3): 183–194, 2005.

Tianle Li, Xuyang Shen, Yan Ma, Rongxin Guo, Shaoxiang Chen, Jiacheng Chen, Haochen Wang, Hongyang Tang, Yucong Zhou, and Yu Cheng. Claimdiff-rl: Fine-grained caption reinforcement learning through visual claim comparison. *arXiv preprint arXiv:2605.20278*, 2026.

Minhua Lin, Zhengzhang Chen, Yanchi Liu, Xujiang Zhao, Zongyu Wu, Junxiang Wang, Xiang Zhang, Suhang Wang, and Haifeng Chen. Decoding time series with llms: A multi-agent framework for cross-domain annotation. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6244–6281, 2026.

Haoxin Liu, Yichen Zhou, Rajat Sen, B Aditya Prakash, and Abhimanyu Das. Rethinking post-training recipes for multimodal time-series forecasting. *arXiv preprint arXiv:2605.29401*, 2026.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 873–881. IEEE, 2017.

James Lloyd, David Duvenaud, Roger Grosse, Joshua Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. Discriminability objective for training descriptive captions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6964–6974, 2018.

Caoyuan Ma, Wenpu Liu, Weichu Xie, Tian Gu, Shilei Zhao, Lingxi Min, Shuai Dong, Yuqi Xu, Ji Zhao, Ziyue Wang, et al. Safecap: Improving lvlm safety with image captioning reinforcement learning. *arXiv preprint arXiv:2608.10513*, 2026.

Mike A Merrill, Mingtian Tan, Vinayak Gupta, Thomas Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3512–3533, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.201. URL https://aclanthology.org/2024.findings-emnlp.201/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816, 2009.

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2):137–169, 2005.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1179–1195. IEEE, 2017.

Medhasweta Sen, Zachary Gottesman, Jiaxing Qiu, C Bayan Bruss, Nam Nguyen, and Tom Hartvigsen. Bedtime: A unified benchmark for automatically describing time series. *arXiv preprint arXiv:2509.05215*, 2025.

Paul Hongsuck Seo, Piyush Sharma, Tomer Levinboim, Bohyung Han, and Radu Soricut. Reinforcing an image caption generator using off-line human feedback. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 2693–2700, 2020.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, volume 2026, pp. 59839–59864, 2026.

Zhijiang Tang, Linhua Wang, Jiaxin Qi, Weihao Jiang, Peng Hou, Anxiang Zeng, and Jianqiang Huang. Cccaption: Dual-reward reinforcement learning for complete and correct image captioning. *arXiv preprint arXiv:2602.21655*, 2026.

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

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, et al. Embeddinggemma: Powerful and lightweight text representations. *arXiv preprint arXiv:2509.20354*, 2025.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *arXiv preprint arXiv:2412.03104*, 2024.

Long Xing, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jianze Liang, Qidong Huang, Jiaqi Wang, Feng Wu, and Dahua Lin. Caprl: Stimulating dense image caption capabilities via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 13066–13093, 2026.

Shaokai Ye, Vasileios Saveris, Yihao Qian, Jiaming Hu, Elmira Amirloo, and Peter Grasch. Balcaprl: A balanced framework for rl-based mllm image captioning. *arXiv preprint arXiv:2605.07394*, 2026.

Jin Yu, Ehud Reiter, Jim Hunter, and Chris Mellish. Choosing the content of textual summaries of large time-series data sets. *Natural Language Engineering*, 13(1):25–49, 2007.

Junru Zhang, Lang Feng, Xu Guo, Yuhan Wu, Yabo Dong, and Duanqing Xu. Timemaster: Training time-series multimodal llms to reason via reinforcement learning. *arXiv preprint arXiv:2506.13705*, 2025.

Xin Zhang, Xingyu Li, Rongguang Wang, Ruizhong Miao, Zheng Wang, Yuying Wang, Dan Roth, and Chenyang Li. Chart-rl: Generalized chart comprehension via reinforcement learning with verifiable rewards. In *Proceedings of the First Workshop on Structured Understanding, Retrieval, and Generation in the LLM Era (SURGeLLM 2026)*, pp. 107–118, 2026.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 414–431, 2025.

Luca Zhou, Pratham Yashwante, Marshall Fisher, Alessio Sampieri, Zihao Zhou, Fabio Galasso, and Rose Yu. CaTS-bench: Can language models describe time series? In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34479–34519, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1722. URL https://aclanthology.org/2026.findings-acl.1722/.

A IMPLEMENTATION DETAILS

A.1 TRAINING LINEUPRL

Policy optimization. The policy is Qwen2.5-VL-3B-Instruct, trained full-parameter with OpenRLHF on four GPUs (bf16, ZeRO-2, gradient checkpointing, FlashAttention-2). The advantage estimator is RLOO with $G = 8$ captions per chart at temperature 1.0, 32 charts per rollout, training batch size 128, learning rate 10^{-6} with 20 warm-up steps, KL coefficient 0, reward normalization, generation capped at $1{,}024$ tokens and no format reward. Four passes over the $9{,}500$ training charts amount to about $1{,}200$ rollout steps. Every 50 steps we caption the 500 validation charts greedily and score them against their fixed distractors; validation reward peaks at step 550 (1.808, verifier accuracy 90.4%, against 1.172 and 58.6% at initialization) and is flat afterwards, so step 550 is the reported checkpoint.

Reward. The frozen verifier is Qwen2.5-14B-Instruct in non-thinking mode, served alongside training under the prompt of Appendix D. The candidate set has $K = 4$ members printed as raw numbers at a shared precision, the reward scale is $\lambda = 2$, so a caption earns one of $\{0, 0.5, 1, 1.5, 2\}$, and the overlap filter of §2.2 uses $\eta = 0.2$. The same caption instruction (Appendix D) is used at rollout, validation and evaluation time.

A.2 BASELINES

All four fine-tuned captioners start from the same Qwen2.5-VL-3B-Instruct weights, are trained full-parameter, read the same charts and receive the same caption instruction (Appendix D). The three RL models share the policy optimizer of Appendix A.1: RLOO, $G = 8$ captions per chart at temperature 1.0, 32 charts per rollout, training batch 128, learning rate 10^{-6} with 20 warm-up steps, KL coefficient 0, generation capped at $1{,}024$ tokens, no format reward, bf16 with ZeRO-2 on four GPUs. Table 4 lists what differs, and the paragraphs below give the reward of each baseline.

RL, answerability. This baseline ports the CapRL recipe (Xing et al., 2026): a VLM writes multiple-choice questions about each training chart, and the reward is the accuracy of a frozen text-only answerer that sees the caption alone. Questions are written by qwen-vl-max reading the chart alone, at temperature 0.8, ten four-option questions with keys per chart under the instruction of Appendix D, which is CapRL's question prompt adapted to a series: it asks for questions about the shape and where its features sit and forbids exact values, since a chart without value labels cannot support them. This yields $95{,}329$ questions over $9{,}985$ charts. Because a VLM confidently answers its own questions wrongly, every question is then checked against the raw values: a text LLM (Qwen3.6-27B) drives a tool-calling loop that measures the series (extrema, segment statistics, feature detection) and the check itself is done by code, and the LLM only drives the tools. Only 27.9% of the questions ($26{,}570$ over $9{,}332$ charts) survive with the key the VLM gave; a further 21.5% would need their key corrected and are discarded together with the 50.6% that are ambiguous, not computable from the values, or wrong. Of the surviving questions, 99.4% have four time intervals as options. Charts with fewer than two surviving questions are dropped, leaving $7{,}243$ training charts with $23{,}436$ questions and 500 validation charts with $1{,}545$ questions. The answerer is the same frozen Qwen2.5-14B-Instruct that serves as our verifier; it is queried once per question under the prompt of Appendix D at temperature 0.6, and the reward is the fraction of the chart's questions it answers correctly, scaled to $[0, 2]$ like ours. Training follows the shared recipe with eight passes and a checkpoint every 25 steps; the checkpoint with the highest validation reward, step 650 (1.238, accuracy 61.9%, against 0.574 and 28.7% at initialization), is the reported one.

RL, LLM-as-judge. The judge is again the frozen Qwen2.5-14B-Instruct that serves as our verifier, so that the three RL models differ in the form of the reward and not in the model behind it. It receives the true series as raw values and the caption, never the chart or any other series, and grades the caption from 1 to 10 under the rubric of Appendix D, which weighs accuracy above specificity above completeness, pins an empty or generic caption to 1, and instructs the judge not to reward length; the reward is $(g - 1)/9 \times 2$, and an unparseable reply counts as 1 so that the RLOO group stays complete. No length or format penalty is added beyond the rubric. Under the shared recipe with two passes (about 600 steps) the reward saturates: from step 410 the mean group reward stays at 2.0, and over the last 100 steps the within-group standard deviation is 0.004, so the policy gradient vanishes. Validation reward reaches its ceiling at steps 500 and 550, and step 550 is the reported checkpoint.

Table 4: Configuration of the four fine-tuned models. Each model is selected on the 500 validation charts under its own training signal, so no model is selected on a test metric. The answerability model trains on fewer charts because a chart enters its training set only if at least two of its questions survive verification.

	LineupRL	RL, answerability	RL, LLM-as-judge	SFT
Training signal	identification among $K = 4$ candidates	accuracy on the chart's verified questions	grade 1–10 of the caption	cross-entropy on 72B captions
Frozen model in the loop	Qwen2.5-14B-Instruct verifier	Qwen2.5-14B-Instruct answerer	Qwen2.5-14B-Instruct judge	Qwen2.5-VL-72B-Instruct teacher
Training charts	9,500	7,243 (23,436 questions)	9,500	9,500 pairs
Passes over the data	4	8	2	5
Learning rate	10^{-6}	10^{-6}	10^{-6}	10^{-5}, cosine
Checkpoint every	50 steps	25 steps	50 steps	50 steps
Selection rule	highest validation reward	highest validation reward	highest validation reward	lowest validation loss

Its captions show the hack that Figure 4 illustrates: 99.9% of them end with a paragraph addressed to the grader, which takes 43.8% of their characters; the corresponding share is 12.1% for the untuned initialization and 0.0% for LineupRL.

SFT on the 72B teacher's captions. The teacher Qwen2.5-VL-72B-Instruct captions the same 9,500 training and 500 validation charts under the same caption instruction with greedy decoding; its captions have a median length of 2,283 characters and 85.1% of them open with the same eight words. Captions shorter than 80 characters or without sentence-final punctuation are dropped, which removes none of the 9,500 training pairs and one validation pair. The student is trained full-parameter with the `trl` SFT trainer for five epochs (742 steps) at learning rate 10^{-5} with a cosine schedule and 5% warm-up, effective batch 64, bf16, sequence length 2,048, evaluating on the 499 validation pairs every 50 steps; validation loss decreases monotonically to 0.199 at step 700, which is the reported checkpoint. Scored under our reward on the validation charts, the teacher's own captions reach an identification accuracy of 68.3%, between the untuned 3B student (58.6%) and LineupRL (90.4%), so imitating them cannot go beyond that level.

A.3 Ablation settings

Each ablation setting changes one component of the recipe of Appendix A.1 and keeps everything else: the training data, the caption instruction, the policy-optimization hyperparameters and the Criterion 3 predictor. The reward-model ladder replaces the Qwen2.5-14B verifier of the reward with Qwen2.5-7B or Qwen2.5-3B, so on this ladder the model that scores the reward changes while the evaluation verifiers of Appendix B do not. The candidate ladder raises K from 4 to 6 or 8 with the same 14B verifier; the $K - 1$ distractors of a smaller K are a prefix of those of a larger one, so raising K adds candidates rather than exchanging them. The distractor-rule setting keeps $K = 4$ and the 14B verifier and replaces the nearest-neighbor rule by a uniform random draw from series of the same length. Every setting is a single training run at one seed (Appendix B.5) on the learning-rate schedule of LineupRL, so the settings of a ladder differ only in the component named above.

B Evaluation details

Overlap between training and evaluation data. The 10,000 training fragments carry no source label, so we matched their values against the evaluation series directly. 6,147 fragments are windows of the OT column of ETTh1 or ETTm1, the two ETT sites that TSFragment-600K draws on; none is a window of any column of ETTh2 or ETTm2 (every fragment was checked against every column of both files at a tolerance of 0.011, the 57 near-constant fragments at every offset), and neither Saugeen nor the Australian demand corpus is a source of TSFragment-600K. In the other direction, no BEDTime or CaTS-Bench series shares a run of 12 or more consecutive values with a training

fragment, apart from flat stretches of at most two distinct values (5 series on BEDTime, 72 on CaTS-Bench) and one 14-step air-quality stretch whose values lie between 0.03 and 0.05, inside the matching tolerance. ETTh2 and ETTm2 thus share a domain and a variable with part of the training data, but no observation.

B.1 Criterion 1: entailment

BEDTime supplies 2,000 series whose references are crowd-sourced or template-generated rather than LLM-written, giving $|\mathcal{P}| = 3{,}958$ pairs; we use DeBERTa long-NLI at $\tau = 0.5$; an empty caption scores 0.003 on this criterion and a reference scored against itself 1.000. CaTS-Bench ships one reference per series but none of it is usable as written, since it states calendar dates, the place, the phenomenon and statistics of the source dataset, none of which our charts carry. The reported CaTS-Bench column therefore scores against rewritten references, described next.

Rewriting the CaTS-Bench references. The 2,969 references were rewritten in two passes by Claude Opus 5, which saw the reference text only, never the series, the chart or any metadata; each pass was split into 30 slices of about 100 references, every slice under the identical instruction, and a rewrite that broke a rule on the first attempt was redone under the same instruction. Pass 1 removes the domain: the phenomenon or attribute name, places, units and currencies, comparisons to statistics outside the series, and calendar dates, which become relative positions ("about two-thirds of the way through"); it keeps every observation in its order and every number as written (Appendix D). Pass 2 starts from the output of pass 1 and removes every numeral and quantity, replacing each by the comparative fact it supported ("starting at 91.29 and reaching 105.27" becomes "starting low and ending higher"), so that the two reference sets differ only in the numerals (Appendix D). A script checks the result: after pass 1 every reference still contains a digit, after pass 2 none of the 2,969 does, and the mean length falls from 255 to 220 characters. We report the pass-2 references because 99.9% of the pass-1 references state an exact quantity against 1.3% of BEDTime's, and a caption written from a chart without value labels cannot entail an exact quantity however well it describes the shape: scored against pass 1, every chart captioner lands at the empty-caption floor, and the column would rank models by their access to the digits instead of by caption quality.

B.2 Criterion 2: identification

Evaluation series come from BEDTime (1,500 series from three of its four public subsets; SUSHI is excluded because its 2,048-point series need roughly 57k tokens for a single four-way option set, past the verifier's context) and CaTS-Bench (2,969 series, lengths 8–150). We use their series but build our own protocol, because both benchmarks test what a model can do with the series itself, and we test what the caption preserves. We build the candidate sets ourselves, at $K = 4$, as the three nearest neighbors in standardized $(\mathrm{mean}, \mathrm{std}, \mathrm{min}, \mathrm{max})$ space at the same length, whereas BEDTime's own distractors are selected to be maximally dissimilar, which its authors describe as an upper bound on performance. Each identification number is the mean over two frozen verifiers that took no part in training any model, GLM-4-9B-0414 and Phi-4 (14B), each queried at all four gold positions (next paragraph). Per-position accuracy is reported as a control, and all captioners answer the same items.

On the choice of the evaluation verifiers. The reward verifier is Qwen2.5-14B-Instruct, and CAP→SER asks the question the reward asks. If the same model scored it, that column would only show how far the training objective generalizes to new series and candidate sets. The identification columns of Table 1 are therefore scored by two frozen LLMs that took no part in training any model, GLM-4-9B-0414 and Phi-4 (14B), and report their mean. Tables 7 and 8 give every captioner's accuracy under five verifiers, the reward verifier, the two evaluation verifiers and two further LLMs, all under the same four-rotation protocol, so the reading of any one verifier can be checked against the others; Appendix C.1 says why the two further LLMs are not averaged into the main table. Three checks show that the verifier is not driving the result. No verifier answers without the caption: with the caption replaced by a placeholder, the two evaluation verifiers identify the true series among the nearest-neighbor candidates at 0.18 to 0.29 (chance 0.25), whereas in the pilot that chose the distractor rule (§2.2) the shuffled and reversed distractors we rejected are identified at 72.5% without a caption. SER→CAP is a direction the reward never asks, Criterion 1 is scored by an NLI model against reference captions and Criterion 3 by a caption-only predictor, neither of which is the verifier, and LineupRL leads on all of them. On reference captions that none of the captioners wrote, the

14B reward verifier is more accurate than the 7B and 3B verifiers on all three domains and at every K (Figure 3c, Appendix C), so a larger verifier is simply more accurate, not more partial to one captioner's style.

B.3 Criterion 3: caption-only forecasting and reconstruction

The two targets share the decoder class and the windows, and in both cases the training loss and the reported metric are the same quantity, so no sampling noise enters the measurement. The split is chronological, so that adjacent windows cut at stride 1 cannot land on both sides of it. Each corpus is one univariate series whose rows are divided $60/20/20$ in time order into a training, a validation and a test segment, following the ETT convention; a window is assigned to the segment that contains the first step of its forecasting target, which is the step after its lookback, and a window whose target runs past the end of the series is discarded. Every test target therefore starts later than every validation target, and every validation target later than every training target. The only contact between segments is at their boundary, exactly as in the standard forecasting protocol: the lookback of the earliest windows of a segment reaches back into the segment before it, and a training target runs at most 15 steps past the training segment. A test window never shares a step with any training window or training target, since the validation segment is longer than a window on every corpus. Reconstruction uses the same windows and the same assignment, so its test targets are the lookback windows of the test split, which reach into the validation segment but never into the training segment. The predictor is fitted on the training windows, early-stopped on the validation windows, and reported on the test windows only. Values are standardized with the statistics of the training segment, which makes $\widehat{\mathcal{S}}$ dimensionless and comparable across datasets. The text-blind predictor b, the naive baseline of Table 2, is last-value persistence for forecasting and the mean training window for reconstruction; the empty-caption and detachment controls of §2.3 are the empty and shuffled caption rows of that table. On truncation: caption length differs across captioners by more than $2\times$, so the encoder's default 256-token cap would have cut every caption of the more verbose captioners and 21% of ours, and the ranking would then partly reflect length. Every run records its token statistics and aborts if any caption would be cut.

Caption-only predictor. Windows are cut with a lookback of 128 and, for forecasting, a horizon of 16, at stride 1, assigned to the chronological $60/20/20$ segments described above, and subsampled uniformly to 5,000 windows per dataset under a fixed data seed, which leaves about 3,000 training, 1,000 validation and 1,000 test windows per corpus, the same windows for every captioner. Values are standardized with the statistics of the training segment. The decoder is `google/embeddinggemma-300m` with a LoRA adapter ($r = 16$), mean-pooled and fed to an MLP head, trained for up to 30 epochs at learning rate 3×10^{-4}, keeping the epoch with the lowest validation MSE and stopping after 6 epochs without improvement; the test windows are scored once, with that epoch's weights. The recipe is held fixed across captioners, so differences between cells come from the captions and not from tuning, and every cell is run with five training seeds. Table 2 reports the mean over those seeds; Table 6 repeats every cell with its standard deviation.

B.4 Aggregation for the ablation figure and Table 3

Aggregation. Figure 3(a,b) and Table 3 aggregate the per-setting tables of Appendix C.3 as follows. Every caption-level number is the unweighted mean of its BEDTime and CaTS-Bench values, except the identification bar of the figure, which is SER→CAP on CaTS-Bench alone; the BEDTime identification of every setting is in Tables 10 and 12, and Table 3 lists both directions averaged over the two benchmarks. The forecasting and reconstruction numbers are geometric means over the four corpora of the ratio of a setting's test MSE to LineupRL's, so that corpora on different error scales weigh equally; the figure plots the inverse so that taller is better. The reward row of Table 3 is the identification accuracy above chance, $(\mathrm{acc} - 1/K)/(1 - 1/K)$, reached on the setting's own candidate sets at the reported checkpoint.

B.5 Statistical tests and noise

Criteria 1 and 2. Both criteria are deterministic functions of a caption set: captions are decoded greedily, the candidate sets and the NLI threshold are fixed, and every captioner answers the same items. Tables 1, 7 and 8 therefore report each number once, as is standard for benchmark tables, and

Table 5: Paired t-test p-values, LineupRL against every other captioner. Two-sided, paired across the 5 predictor seeds. LineupRL has the lower mean everywhere except the cells marked †, where the other captioner is ahead.

	SFT	Answ.	Judge	72B	3B init
Forecasting					
ETTh2	0.001	<0.001	0.002	0.002	<0.001
ETTm2	0.016	<0.001	0.035	<0.001†	<0.001
Saug.	0.007	<0.001	0.030	0.005	<0.001
Elec.	<0.001	<0.001	<0.001	<0.001	<0.001
Reconstruction					
ETTh2	<0.001	<0.001	<0.001	0.008	<0.001
ETTm2	0.010	<0.001	0.007	0.609†	<0.001
Saug.	<0.001	<0.001	0.088	0.678†	<0.001
Elec.	<0.001	<0.001	<0.001	<0.001	<0.001

bold marks the best value of the fine-tuned group. The two statements about SFT in §1 are read off Table 1: on BEDTime entailment SFT scores 0.281 against 0.296 for the untuned initialization, so SFT does not raise entailment there, and we do not claim that it lowers it; on CaTS-Bench SER→CAP SFT is below the initialization under both evaluation verifiers (0.598 against 0.680 under GLM-4-9B, 0.634 against 0.763 under Phi-4-14B).

Significance of the Criterion 3 gaps. Bolding in Table 2 comes from a two-sided paired t-test across the 5 predictor seeds. Seeds are paired because a seed moves every captioner together: only the predictor's training loop varies, so all 5 runs score identical windows against identical captions, and the paired difference removes the shared per-seed component. In each column we take the captioner with the lowest mean and bold it together with every captioner it cannot be separated from at $p < 0.05$, so two bold entries mark a tie rather than a ranking. Table 5 lists the p-values of LineupRL against every other captioner. With $n = 5$ ($\mathrm{df} = 4$) the test is lenient; it only tells which gaps are stable across seeds.

Two further families of tests back the claims in §2.3. Every fine-tuned captioner beats both text-only controls in all eight dataset-study cells: across those 64 comparisons the weakest is $p = 9.8 \times 10^{-5}$. The two controls are in turn indistinguishable from each other in 7 of the 8 cells, the exception being forecasting on ETTm2 ($p = 0.003$), where the shuffled caption is 0.5% worse than no caption at all. A wrong caption is therefore about as useful as none, so the margin of a captioner over the controls measures series-specific content.

Number of training runs. Every fine-tuned captioner in Tables 1 and 2 and every ablation setting of §3.3 is a single training run at one seed; only the Criterion 3 predictor is repeated over seeds, so the spreads in Table 2 are the predictor's, not the captioner's. Two noise scales help read the gaps. Scoring one checkpoint twice moves a Criterion 2 accuracy by about ± 0.005, and across the five predictor seeds the relative standard deviation of a Table 2 cell has median 3.6% and 90th percentile 6.1%. LineupRL's margin over the best fine-tuned baseline is between $+0.114$ and $+0.253$ on the six columns of Table 1 and above 9% of the error in six of the eight cells of Table 2; the two Saugeen cells are 5% (forecasting) and 1% (the reconstruction tie). The Criterion 3 tests compare the caption sets we obtained; they do not cover the variance of training itself, and a rerun of any model could move its numbers by more than the evaluation noise. The test script is part of the released code.

C Additional results

C.1 Identification under five verifiers

Tables 7 and 8 re-score every captioner of Table 1 under five frozen verifiers: the reward verifier, the two evaluation verifiers whose mean the main table reports, and two further LLMs. Everything except the verifier is as in Appendix B.

The main table averages GLM-4-9B and Phi-4-14B and not the other two because a verifier is only informative where it separates captions. The reward verifier is set aside for the reason given in

Table 6: Criterion 3 with across-seed spread. The cells of Table 2, laid out one study per block so the two panels can be read separately. Bolding and baselines are as in that table. Only the predictor's training loop changes between seeds, so the spread comes from the predictor; each captioner is a single RL or SFT run.

	Forecasting (predict the next window)			
Captioner	ETTh2	ETTm2	Saug.	Elec.
Naive baseline	0.3047	0.0596	1.2955	0.9078
Empty caption	0.8446 ±0.0101	0.8001 ±0.0042	1.0601 ±0.0016	0.9129 ±0.0041
Shuffled caption	0.8437 ±0.0087	0.8040 ±0.0046	1.0619 ±0.0042	0.9124 ±0.0043
Qwen2.5-VL-3B	0.3062 ±0.0086	0.1012 ±0.0047	0.9419 ±0.0119	0.5751 ±0.0351
Qwen2.5-VL-72B	0.2149 ±0.0094	0.0451 ±0.0009	0.8521 ±0.0061	0.3089 ±0.0094
Fine-tuned from Qwen2.5-VL-3B				
SFT on 72B's captions	0.2595 ±0.0151	0.0615 ±0.0021	0.9026 ±0.0216	0.3480 ±0.0086
RL, answerability	0.3780 ±0.0168	0.2110 ±0.0159	0.9384 ±0.0104	0.3258 ±0.0151
RL, LLM-as-judge	0.2384 ±0.0134	0.0630 ±0.0077	0.8619 ±0.0217	0.3388 ±0.0113
LineupRL	**0.1806** ±0.0078	**0.0541** ±0.0022	**0.8175** ±0.0196	**0.2042** ±0.0091

	Reconstruction (predict the window itself)			
Captioner	ETTh2	ETTm2	Saug.	Elec.
Naive baseline	0.9501	0.8335	1.1748	0.8634
Empty caption	0.9284 ±0.0306	0.8274 ±0.0029	1.1764 ±0.0008	0.8667 ±0.0021
Shuffled caption	0.9347 ±0.0121	0.8270 ±0.0015	1.1774 ±0.0018	0.8663 ±0.0010
Qwen2.5-VL-3B	0.2445 ±0.0052	0.0633 ±0.0029	0.6852 ±0.0173	0.4352 ±0.0156
Qwen2.5-VL-72B	0.1134 ±0.0065	0.0288 ±0.0022	0.4444 ±0.0113	0.1993 ±0.0077
Fine-tuned from Qwen2.5-VL-3B				
SFT on 72B's captions	0.1478 ±0.0044	0.0333 ±0.0016	0.5259 ±0.0177	0.2144 ±0.0084
RL, answerability	0.3391 ±0.0185	0.1883 ±0.0151	0.5309 ±0.0091	0.2608 ±0.0095
RL, LLM-as-judge	0.1373 ±0.0104	0.0325 ±0.0019	**0.4517** ±0.0027	0.2248 ±0.0029
LineupRL	**0.0989** ±0.0033	**0.0294** ±0.0010	**0.4463** ±0.0054	**0.1357** ±0.0068

Seeds 2020, 2021, 2022, 2023, 2024; all 5 score identical windows against identical captions.

Appendix B.2. Llama-3.1-8B and Mistral-Nemo-12B score every VLM captioner below 0.45 on S→C on both benchmarks, and the three fine-tuned baselines between 0.26 and 0.33 on BEDTime, close to chance (0.25), so in that direction their readings say more about the verifier than about the captions; GLM-4-9B and Phi-4-14B place the same three baselines between 0.33 and 0.50 on BEDTime and between 0.52 and 0.65 on CaTS-Bench. The two are still worth reporting as a check on the direction they do resolve: on C→S, LineupRL leads the fine-tuned models under all five verifiers on both benchmarks.

C.2 Reward discrimination on reference captions

Figure 3(c) asks each of the three verifiers the CAP→SER question at $K \in \{4, 6, 8, 10, 12\}$ on captions that none of the captioners wrote: the 500 validation charts of the training distribution with the GPT-4o-mini captions shipped with TSFragment-600K, $2{,}613$ CaTS-Bench series with the number-free references of Appendix B.1, and $1{,}480$ BEDTime series with their first annotation. Candidates are the nearest neighbors in the same statistic space as the reward, the true series is rotated through every position, and the items are shared across verifiers and K, so the only variable is the verifier and the number of candidates. Absolute values are lower than on the policy's captions, which the reward has optimized to be discriminable; what matters is the ordering, which is $14\text{B} > 7\text{B} > 3\text{B}$ on every domain and at every K.

C.3 Per-setting ablation tables

Tables 10–15 give the per-benchmark and per-corpus numbers behind Figure 3(a,b) and Table 3: for each of the three ladders, the six caption-level metrics of Table 1 with the mean caption length they

Table 7: Identification on BEDTime under five frozen verifiers. Every captioner's C→S and S→C accuracy on the candidate sets of Table 1, re-scored by each verifier under the same four-rotation protocol (chance 0.25). Qwen2.5-14B is the reward verifier; GLM-4-9B and Phi-4-14B are the two evaluation verifiers whose mean Table 1 reports. Bold marks the best of the four models fine-tuned from the same initialization, as in Table 1.

	Qwen2.5-14B		GLM-4-9B		Llama-3.1-8B		Mistral-Nemo-12B		Phi-4-14B	
Captioner	C→S	S→C	C→S	S→C	C→S	S→C	C→S	S→C	C→S	S→C
Qwen2.5-VL-3B	0.664	0.566	0.559	0.443	0.472	0.328	0.518	0.349	0.541	0.548
Qwen2.5-VL-72B	0.742	0.625	0.610	0.455	0.582	0.338	0.574	0.334	0.600	0.513
Qwen3-VL-8B	0.752	0.604	0.639	0.446	0.648	0.372	0.620	0.351	0.700	0.549
InternVL3-14B	0.709	0.600	0.585	0.467	0.542	0.360	0.541	0.324	0.539	0.557
Phi-3.5-mini	0.936	0.903	0.893	0.828	0.884	0.668	0.837	0.701	0.925	0.787
Qwen2.5-7B	0.948	0.926	0.904	0.835	0.906	0.599	0.849	0.604	0.928	0.812
Qwen2.5-14B	0.947	0.937	0.901	0.850	0.909	0.725	0.870	0.645	0.937	0.825
ChatTS-14B	0.757	0.775	0.744	0.659	0.728	0.487	0.637	0.564	0.622	0.782
Fine-tuned from Qwen2.5-VL-3B										
SFT on 72B's captions	0.714	0.588	0.582	0.420	0.534	0.325	0.547	**0.324**	0.559	0.476
RL, answerability	0.624	0.499	0.453	0.390	0.384	0.279	0.455	0.281	0.443	0.495
RL, LLM-as-judge	0.718	0.549	0.534	0.331	0.341	0.255	0.518	0.264	0.554	0.493
LineupRL	**0.788**	**0.765**	**0.717**	**0.596**	**0.613**	**0.340**	**0.612**	**0.324**	**0.700**	**0.680**

Table 8: Identification on CaTS-Bench under five frozen verifiers. Every captioner's C→S and S→C accuracy on the candidate sets of Table 1, re-scored by each verifier under the same four-rotation protocol (chance 0.25). Qwen2.5-14B is the reward verifier; GLM-4-9B and Phi-4-14B are the two evaluation verifiers whose mean Table 1 reports. Bold marks the best of the four models fine-tuned from the same initialization, as in Table 1.

	Qwen2.5-14B		GLM-4-9B		Llama-3.1-8B		Mistral-Nemo-12B		Phi-4-14B	
Captioner	C→S	S→C	C→S	S→C	C→S	S→C	C→S	S→C	C→S	S→C
Qwen2.5-VL-3B	0.873	0.803	0.674	0.680	0.675	0.429	0.591	0.449	0.827	0.763
Qwen2.5-VL-72B	0.951	0.806	0.711	0.631	0.763	0.372	0.648	0.434	0.896	0.655
Qwen3-VL-8B	0.933	0.775	0.716	0.587	0.802	0.379	0.650	0.413	0.919	0.665
InternVL3-14B	0.919	0.798	0.684	0.689	0.734	0.384	0.619	0.426	0.849	0.713
Phi-3.5-mini	0.989	0.988	0.979	0.953	0.978	0.819	0.867	0.849	0.981	0.826
Qwen2.5-7B	0.997	0.991	0.982	0.961	0.981	0.753	0.869	0.850	0.988	0.839
Qwen2.5-14B	0.999	0.997	0.987	0.970	0.987	0.881	0.915	0.859	0.996	0.906
ChatTS-14B	0.889	0.923	0.871	0.823	0.864	0.667	0.717	0.712	0.893	0.904
Fine-tuned from Qwen2.5-VL-3B										
SFT on 72B's captions	0.919	0.759	0.673	0.598	0.722	0.362	0.605	**0.414**	0.846	0.634
RL, answerability	0.746	0.627	0.523	0.523	0.507	0.291	0.478	0.311	0.675	0.597
RL, LLM-as-judge	0.915	0.725	0.650	0.529	0.535	0.275	0.561	0.319	0.738	0.650
LineupRL	**0.964**	**0.951**	**0.868**	**0.846**	**0.810**	**0.431**	**0.713**	0.400	**0.904**	**0.891**

were measured on, and the eight Criterion 3 cells of Table 2. The untuned initialization is repeated in every table as a reference line. Table 9 in Appendix C.2 lists the numbers behind Figure 3(c).

Two notes for reading these tables alongside the main ones. First, the Criterion 3 numbers are single runs at predictor seed 2020, where Table 2 reports the mean over five, so the LineupRL row reads 0.1687 on forecasting ETTh2 here against 0.1806 there; across the five seeds of Table 2 the relative standard deviation of a cell has median 3.6% and 90th percentile 6.1%, which is the scale below which a difference in these tables should not be read. Second, on the reward-model ladder the verifier that scores the reward changes with the setting, whereas the identification columns are scored by the same two evaluation verifiers for every setting, so they compare the captions on equal terms; entailment and Criterion 3 are not the objective of any setting.

Table 9: Reward discrimination on reference captions, the cells of Figure 3(c): CAP→SER accuracy above chance, $(\mathrm{acc} - 1/K)/(1 - 1/K)$, for three verifiers and five candidate counts on the reference captions of three domains.

Domain	Verifier	$K = 4$	$K = 6$	$K = 8$	$K = 10$	$K = 12$
TSFragment val.	Qwen2.5-14B	0.701	0.634	0.586	0.527	0.496
	Qwen2.5-7B	0.641	0.553	0.491	0.443	0.401
	Qwen2.5-3B	0.455	0.353	0.284	0.186	0.233
CaTS-Bench	Qwen2.5-14B	0.421	0.384	0.357	0.336	0.324
	Qwen2.5-7B	0.257	0.199	0.170	0.151	0.147
	Qwen2.5-3B	0.112	0.100	0.095	0.080	0.105
BEDTime	Qwen2.5-14B	0.194	0.165	0.155	0.142	0.146
	Qwen2.5-7B	0.147	0.114	0.096	0.090	0.081
	Qwen2.5-3B	0.038	0.041	0.031	0.032	0.036

Table 10: Ablation on the reward model: caption metrics. Effect of the size of the frozen verifier that turns a caption into the reward, with everything else held at the LineupRL recipe. Columns are those of Table 1: *Ent.* is the fraction of reference captions the generated one entails, C→S and S→C are the identification accuracies averaged over the two frozen evaluation verifiers, GLM-4-9B and Phi-4-14B (chance 0.25). *Len.* is the mean caption length in characters, printed because both entailment columns move with it. Bold marks the best of the three trained settings; the untuned initialization is shown for reference. Figure 3 plots the CaTS-Bench S→C column.

	BEDTime				CaTS-Bench			
Reward model	Ent.	C→S	S→C	Len.	Ent.	C→S	S→C	Len.
Untuned Qwen2.5-VL-3B	0.296	0.550	0.495	1084	0.352	0.751	0.721	1287
Qwen2.5-14B (**ours**)	**0.400**	0.709	0.638	621	**0.611**	**0.886**	**0.868**	713
Qwen2.5-7B	0.287	0.710	0.644	1331	0.298	0.850	0.799	1725
Qwen2.5-3B	0.270	**0.715**	**0.710**	898	0.381	0.830	0.831	959

Table 11: Ablation on the reward model: Criterion 3. Test MSE (lower is better) of the caption-only predictor of Table 2, which never sees the series. Every setting keeps $K = 4$ candidates and the rest of the LineupRL recipe; only the verifier that scores the reward changes. *Rel.* is the geometric mean of the four per-corpus ratios to the LineupRL row, so > 1 means the caption carries less of what the predictor needs. Bold marks the best of the three trained settings. Figure 3 plots the *Rel.* columns alone.

	Forecasting					**Reconstruction**				
Reward model	ETTh2	ETTm2	Saug.	Elec.	Rel.	ETTh2	ETTm2	Saug.	Elec.	Rel.
Untuned Qwen2.5-VL-3B	0.3114	0.1037	0.9234	0.5476	1.850	0.2469	0.0654	0.6698	0.4124	2.283
Qwen2.5-14B (**ours**)	**0.1687**	**0.0523**	**0.8225**	**0.1922**	1.000	**0.0974**	**0.0286**	**0.4472**	**0.1317**	1.000
Qwen2.5-7B	0.2535	0.0758	0.8709	0.2926	1.369	0.1441	0.0435	0.5318	0.1847	1.391
Qwen2.5-3B	0.2312	0.0783	0.8459	0.3205	1.370	0.1693	0.0534	0.6552	0.2385	1.713

Table 12: Ablation on the number of candidates: caption metrics. Effect of the number of candidate series the verifier must choose among, with everything else held at the LineupRL recipe. Columns are those of Table 1: *Ent.* is the fraction of reference captions the generated one entails, C→S and S→C are the identification accuracies averaged over the two frozen evaluation verifiers, GLM-4-9B and Phi-4-14B (chance 0.25). *Len.* is the mean caption length in characters, printed because both entailment columns move with it. Bold marks the best of the three trained settings; the untuned initialization is shown for reference. Figure 3 plots the CaTS-Bench S→C column.

	BEDTime				CaTS-Bench			
Candidates	Ent.	C→S	S→C	Len.	Ent.	C→S	S→C	Len.
Untuned Qwen2.5-VL-3B	0.296	0.550	0.495	1084	0.352	0.751	0.721	1287
$K = 4$ (**ours**)	**0.400**	0.709	0.638	621	**0.611**	**0.886**	0.868	713
$K = 6$	0.343	0.664	0.644	482	0.470	0.851	0.869	561
$K = 8$	0.340	**0.790**	**0.796**	555	0.391	0.852	**0.876**	620

Table 13: Ablation on the number of candidates: Criterion 3. Test MSE (lower is better) of the caption-only predictor of Table 2, which never sees the series. Every setting keeps the same frozen Qwen2.5-14B verifier and the rest of the LineupRL recipe; only the number of candidates it must choose among changes. The $K - 1$ distractors of a smaller K are a prefix of those of a larger one, so raising K adds candidates rather than exchanging them. *Rel.* is the geometric mean of the four per-corpus ratios to the LineupRL row, so > 1 means the caption carries less of what the predictor needs. Bold marks the best of the three trained settings. Figure 3 plots the *Rel.* columns alone.

	Forecasting					**Reconstruction**				
Candidates	ETTh2	ETTm2	Saug.	Elec.	Rel.	ETTh2	ETTm2	Saug.	Elec.	Rel.
Untuned Qwen2.5-VL-3B	0.3114	0.1037	0.9234	0.5476	1.850	0.2469	0.0654	0.6698	0.4124	2.283
$K = 4$ (**ours**)	**0.1687**	0.0523	0.8225	**0.1922**	1.000	**0.0974**	0.0286	0.4472	0.1317	1.000
$K = 6$	0.2046	**0.0442**	**0.8067**	0.1925	1.002	0.0975	0.0288	**0.4179**	**0.1234**	0.969
$K = 8$	0.2206	0.0529	0.8421	0.2120	1.106	0.1006	**0.0271**	0.4287	0.1382	0.996

Table 14: Ablation on how the distractors are chosen: caption metrics. Effect of the rule that picks the distractors an item is scored against, with everything else held at the LineupRL recipe. Columns are those of Table 1: *Ent.* is the fraction of reference captions the generated one entails, C→S and S→C are the identification accuracies averaged over the two frozen evaluation verifiers, GLM-4-9B and Phi-4-14B (chance 0.25). *Len.* is the mean caption length in characters, printed because both entailment columns move with it. Bold marks the best of the two trained settings; the untuned initialization is shown for reference. Table 3 aggregates these columns.

	BEDTime				CaTS-Bench			
Distractors	Ent.	C→S	S→C	Len.	Ent.	C→S	S→C	Len.
Untuned Qwen2.5-VL-3B	0.296	0.550	0.495	1084	0.352	0.751	0.721	1287
Nearest neighbor (**ours**)	**0.400**	**0.709**	**0.638**	621	**0.611**	**0.886**	**0.868**	713
Random	0.324	0.598	0.636	723	0.384	0.810	0.853	753

Table 15: Ablation on how the distractors are chosen: Criterion 3. Test MSE (lower is better) of the caption-only predictor of Table 2, which never sees the series. Both settings keep the same frozen Qwen2.5-14B verifier, $K = 4$ candidates and the rest of the LineupRL recipe; only the rule that picks the $K - 1$ distractors of a training item changes. *Rel.* is the geometric mean of the four per-corpus ratios to the LineupRL row, so > 1 means the caption carries less of what the predictor needs. Bold marks the best of the two trained settings. Table 3 aggregates these columns.

	Forecasting					**Reconstruction**				
Distractors	ETTh2	ETTm2	Saug.	Elec.	Rel.	ETTh2	ETTm2	Saug.	Elec.	Rel.
Untuned Qwen2.5-VL-3B	0.3114	0.1037	0.9234	0.5476	1.850	0.2469	0.0654	0.6698	0.4124	2.283
Nearest neighbor (**ours**)	**0.1687**	**0.0523**	0.8225	**0.1922**	1.000	**0.0974**	0.0286	**0.4472**	**0.1317**	1.000
Random	0.2314	0.0643	**0.8138**	0.2484	1.212	0.1397	**0.0268**	0.4513	0.1652	1.142

D PROMPTS

Every prompt below is reproduced from the code that ran; typographic dashes and bullets are rendered in ASCII and nothing else is changed.

Caption prompt. The single user turn every captioner in Tables 1 and 2 receives, at RL rollout, at SFT teacher generation, at validation and at evaluation. Number-reading models receive the same text with *image*/*chart* replaced by *values*.

```
This image is a line chart of a single time series. Describe
the series in detail: what values it takes, how it moves
over time, and where along the time axis its notable
features occur. Write it so that someone who cannot see the
chart could tell this series apart from other series that
look broadly similar.
```

Verifier prompt of the LineupRL reward. System line followed by the user turn, at $K = 4$. Every candidate is printed at the precision chosen for the true series, so formatting cannot single it out. Criterion 2 (CAP→SER) asks the same question of the evaluation candidates; SER→CAP swaps the two sides. *System:*

```
You are a careful time-series analyst. You answer with a
single letter.
```

User:

```
Below is a written description of one time series.
Description:
"""{caption}"""
Here are four candidate time series, each given as its raw
values in time order.
A) {opt_a}
B) {opt_b}
C) {opt_c}
D) {opt_d}
Exactly one of these four is the series that the description
was written about.
Which one is it? Answer with a single letter: A, B, C, or D.
```

Judge prompt of the LLM-as-judge reward. The judge receives the true series as raw numbers and the caption, never the chart or any other series, and returns an integer 1–10. *System:*

```
You are a careful time-series analyst. You grade written
descriptions of time series against the actual values. You
answer with a score and nothing else.
```

User:

```
Here are the values of one time series, in order:
{series}
Here is a written description of that series:
{caption}
Grade the description on how well it describes THIS series,
from 1 to 10.
What to weigh, in order:
1. Accuracy. Every claim it makes -- direction, shape, where
features sit, magnitudes --
   must be true of the values above. A confident claim that
is wrong is the worst thing a
```

```
   description can do and must pull the score below any
description that says less.
2. Specificity. A description that would fit many different
series is worth little, even
   when nothing in it is false. Ask whether someone holding
this description could pick
   this series out from others of broadly similar shape.
3. Completeness. Whether the description covers the
movements that matter, not whether it
   covers everything.
Anchors:
  1-2 Mostly wrong, or so generic it would fit almost any
series.
  3-4 Broadly right about the overall direction, wrong or
silent on everything else.
  5-6 Correct about the main movement, vague about where
features occur or how large
        they are.
  7-8 Accurate and specific about the main features and
roughly where they occur.
  9-10 Accurate throughout AND specific enough to identify
this series among similar
        ones.
Score 1 if the description is empty, or if it could have
been written without seeing
these values at all. You are grading the description, not
the series: when there is
nothing there to grade, the score is 1, however interesting
the values are.
Do not reward length. A short description that is accurate
and specific scores higher
than a long one that hedges, repeats itself, or pads with
claims you cannot verify.
Reply with the integer score alone, no words, no
punctuation.
```

Question-writing prompt of the answerability reward. System prompt and user turn given to the question-writing VLM, which sees the chart only; N is filled in with the length of the series. *System:*

```
You are given a line chart of a single univariate time
series. Generate TEN multiple-choice questions (with their
answers) about THIS specific series.
Hard requirements:
- The ten questions must each probe a DIFFERENT aspect of
the curve. Do not ask ten variants of the same thing, and do
not always start with "where is the peak" or "what is the
overall trend".
- Mix both kinds: some questions about the OVERALL behavior
(trend, shape, how it starts vs. ends, which half/third is
higher, steeper, or noisier) and some about a SPECIFIC
salient event you can clearly see (a prominent peak, dip,
surge, or plateau) -- when you ask about such an event,
locate it by its coarse region and only if it is clearly
visible.
- Each question must be SPECIFIC to this curve: refer to the
concrete shape you actually see (a particular rise, dip,
plateau, spike, oscillation, or where along the series it
happens). Avoid generic, template-like questions that could
apply to any series.
- Each question must be answerable purely from the curve,
have exactly four options, and exactly one correct answer.
Make the distractors plausible.
```

```
- Do NOT ask for exact numeric values, forecasting, or
anything that is not visible in the chart.
- Even when the raw values are provided to you for reference,
use them only to read the true shape -- do NOT ask for exact
y-values or step-precise readings, and only ask about a
feature (peak/dip/surge/plateau) if it is genuinely there in
the values.
- Write every positional option as a PLAIN NUMERIC time
index -- a SINGLE index like "t=7" when the answer is
intrinsically one point, or an INTERVAL like "t=5-10" when
the answer is a region/segment -- with NO verbal label (do
not write "early part", "middle", etc.; those words are
vague and ambiguous). The series has integer time indices 0
to N-1 (N is given to you).
- MATCH the option granularity to what the question actually
asks -- do NOT force every option into a wide interval when
that is the wrong shape:
   - REGION/SEGMENT questions (overall trend, which part is
steeper / higher-on-average / noisier, where a rise / dip /
plateau spans): use COARSE intervals. Place boundaries where
THIS curve actually changes so the four are generally
UNEQUAL in width (do NOT just split into four equal
quarters), each at least about one fifth of the series wide.
   - SINGLE-POINT questions (where the exact maximum / peak
or minimum / trough sits): use SINGLE time indices (or tight
windows) as the options. Inflating a one-point answer into a
wide interval makes several intervals tie and the question
becomes unanswerable -- so be truthful and keep these
options point-sized.
- CRITICAL -- the four options of a positional question must
be MUTUALLY EXCLUSIVE (disjoint): no two may overlap or
cover the same span/point (adjacent intervals may meet at a
single boundary index only). If you cannot fill four
disjoint options, make one option a NON-positional choice
instead (e.g. "Roughly constant throughout / No single clear
region").
Output format -- follow it EXACTLY. The example below shows
ONLY the format; do NOT copy its wording or topic.
Positional options are numeric and disjoint (intervals for
region questions, single indices for single-point
questions):
#### 1. **<a specific question about this curve>**
   - A) **<Choice A>**
   - B) **<Choice B>**
   - C) **<Choice C>**
   - D) **<Choice D>**
**Answer:** <letter>) <correct option text>
------
Produce questions 1 to 10 in this exact format, each block
separated by a line containing only "------". Output nothing
else.
```

User:

```
Here is the line chart of the time series. The series has
N={n} points (time indices 0..{n-1}). Write each positional
option as a plain numeric time index or time interval only
(e.g. 't=7' or 't=5-13'), with NO verbal region words. For
region questions use coarse intervals (>= about N/5 wide)
and place the boundaries where this curve actually changes
so the four intervals are UNEQUAL in width; for single-point
questions, such as where the peak or the trough sits, use
single indices or tight windows. Keep the four options
non-overlapping. Generate the ten questions now.
```

Answerer prompt of the answerability reward. The frozen text-only answerer is queried once per question in this chat template; the first letter in its reply is compared with the key.

```
<|im_start|>user
You will be given a caption describing a univariate time
series (a line chart of a value versus a time index).
Your task is to answer the multiple-choice question strictly
based on the caption, even if the answer may seem obvious
from prior knowledge or the question wording.

Ignore any outside knowledge. Do not assume anything the
caption does not explicitly or implicitly state.

Example:
Caption: <Caption Start> The series rises steadily from
about 10 at the start to a peak near 40 around the middle,
then declines to roughly 25 by the end. <Caption End>
Question: At which point in time is the series at its
highest value?
- A) the very beginning
- B) one quarter of the way through
- C) around the middle
- D) at the very end

The answer is C.

Now, answer the question based on the following caption:

Caption: <Caption Start> {} <Caption End>
Question: {} <|im_end|>
<|im_start|>assistant
The answer is
```

CaTS-Bench rewrite, pass 1 (domain removed). Instruction given to each of the 30 rewriting agents; the slice markers {{START}}, {{END}}, {{K}} and {{NNN}} differ per agent and <work> stands for the scratch directory of the run.

```
Each line of your slice is {"id": "...", "gold":
"<caption>"}. The caption is your only input: there is no
chart, no values and no metadata, and you must not go
looking for any.

Rewrite each caption so it describes only what an unlabelled
line chart could show. Take out the domain: the phenomenon
or attribute name, any place, port, store or product name,
units and currency (keep the bare number), comparisons to
statistics outside this series (historical mean, all-time
max, "the reference period"), and any explanation that needs
to know what the data is.

Dates become relative positions, never counted ones. "From
2011 to 2018" becomes "across the series"; "peaking in
August 2023" becomes "peaking about two-thirds of the way
through"; "a dip in late May" becomes "a dip near the end".
Do not say how many points the series has and do not number
them -- no "the eighth point", no "40 points", no "index 6",
and drop counts of time units such as "a five-week
snapshot". The text does not tell you how many points there
are, and a guess would put a false claim into the caption.
The vocabulary is "at the start", "early on", "midway", "in
the second half", "near the end", "about two-thirds of the
way through".

Keep the same observations in the same order, and keep every
value and statistic as written. Rephrase as freely as the
removals require, so that what is left reads as fluent,
natural prose rather than a sentence with holes in it. What
you must not do is add an observation the caption did not
make, or change a number, even if the caption looks wrong.
If a sentence has nothing left after the removals, drop it.
```

```
Your slice: sed -n '{{START}},{{END}}p'
<work>/items_textonly.jsonl ({{K}} lines)
Write <work>/shards/batch_{{NNN}}.jsonl -- exactly {{K}}
lines, same order, one JSON object per line built with
python3 + json.dumps (never by hand), "id" copied unchanged,
plus "rewritten". Rewrite them yourself, one by one; a
script is only for reading input and writing output. Reply
with the shard path and line count, nothing else.
```

CaTS-Bench rewrite, pass 2 (numbers removed). Applied to the output of pass 1, so the two reference sets differ only in the numerals.

```
Each line of your slice is {"id": "...", "gold":
"<caption>"}. The caption is your only input: there is no
chart, no values and no metadata, and you must not go
looking for any.

Rewrite each caption so that it states the same observations
with **no numbers at all**. The target is the register of
these examples, which are real captions from another
benchmark:

    "line decreases near the end"
    "highest point is in middle"
    "rises in the middle then flattens"

Remove every numeral and every quantity: values ("starting
at 91.29"), ranges, means, percentages, counts, and any word
that only carries a magnitude ("15%", "approximately 40
units"). Positions in time stay, because they are
qualitative: "at the start", "early on", "midway", "in the
second half", "near the end", "about two-thirds of the way
through".

Replace a quantity with the comparative fact it supports,
never with a vaguer number. "Starting at 91.29 and reaching
105.27" becomes "starting low and ending higher". "A notable
increase of approximately 15%" becomes "a clear increase".
"Rising above 100 along the way" becomes "rising above its
starting level along the way" ONLY if the caption itself
says what 100 relates to; if it does not, drop that clause.
"The mean of the series is 97.3" has no numberless form that
says anything about THIS series against others -- drop it.

Keep the same observations in the same order, and keep every
non-numeric detail: direction, shape, where features sit,
whether movement is smooth or noisy, whether it is steady or
abrupt. Do not add an observation the caption did not make.
If a sentence is nothing but a quantity, drop the sentence.
If dropping leaves the caption empty, write the single most
specific shape statement the original supports.

The result must read as fluent prose, not as a sentence with
holes in it, and must contain no digits.

Your slice: sed -n '{{START}},{{END}}p' <work>/items.jsonl
({{K}} lines)
Write <work>/shards/batch_{{NNN}}.jsonl -- exactly {{K}}
lines, same order, one JSON object per line built with
python3 + json.dumps (never by hand), "id" copied unchanged,
plus "rewritten". Rewrite them yourself, one by one; a
script is only for reading input and writing output. Reply
with the shard path and line count, nothing else.
```