Monitorable Chart Reasoning Agents via Verifiable Process Rewards

Anonymous ACL submission

Abstract

Chart reasoning agents are increasingly used to extract actionable insights in critical domains, achieving state-of-the-art performance on multiple benchmarks. Yet, high benchmark accuracy alone is insufficient for deployment, where stakeholders must be able to audit and verify how a model reaches its answer. Existing LVLM-based chart agents produce either answer-only predictions or free-form rationales that are hard to verify, obscuring whether an error arose from misreading the chart, extracting a wrong value, or miscomputing. We propose **Chart-RVR**, a reinforcement learning framework for training *monitorable chart agents* with verifiable process rewards. Chart-RVR decomposes chart reasoning into three auditable blocks: *Structure*, identifying the chart type; *Evidence*, reconstructing the underlying data table in JSON; and *Derivation*, exposing the stepwise trace that computes the answer. Across six in-domain and out-of-domain benchmarks, Chart-RVR attains state-of-the-art accuracy among comparable-sized LVLMs. Beyond accuracy, we assess monitorability using a triangulated protocol that combines ground-truth surrogate metrics, an oracle information-gain measure, and an LLM-as-auditor scoring *Process Verifiability* and *Evidence Localization*, showing that Chart-RVR yields rationales that are markedly more verifiable and evidence-grounded than those from CoT prompting, SFT, and existing chart-specific baselines. The code can be found at `https://anonymous.4open.science/r/chart-rvr-730D/`.

1 Introduction

Charts are a cornerstone of visual communication across finance, healthcare, public policy, and beyond, where experts and non-experts alike rely on them to allocate resources and drive strategic decisions. Automating their interpretation is therefore a high-value problem. Unlike natural images, typically described by high-level semantics (e.g., "a dog on a table"), charts encode information through *precise* spatial and numerically aligned relationships. This makes chart reasoning inherently *entangled*: structured data is tightly interwoven with visual design choices, and any chart-reasoning model must disentangle the two during decision making. Large Vision-Language Models (LVLMs), pre-trained on billions of image-text pairs, have shown strong performance on general visual question answering, including chart reasoning.

However, high benchmark accuracy alone is insufficient for deploying chart agents in high-stakes settings. Stakeholders in sensitive domains like health and finance demand correct answers and also accurate *intermediate evidence* and transparency into how it was reached. This entails understanding which *chart structure* was recognized, what *entities* were extracted, and how those entities were *combined*. We refer to this property as **monitorability** in chart reasoning, an emerging area of deployable agentic research (Guan et al., 2025). For chart agents, monitorability captures the extent to which intermediate reasoning is structured, verifiable, localizable, and understandable to end-users.

The majority of existing chart-reasoning agents fall into two categories. The first uses off-the-shelf pre-trained LVLMs inside a systematic harness. While effective on standard benchmarks, recent studies (Islam et al., 2024) expose two systematic weaknesses: even when these agents answer in-domain (ID) questions correctly, eliciting rationales via chain-of-thought (CoT) prompting often fails to improve accuracy and can even *harm* it (Zhang et al., 2025; Turpin et al., 2023), yielding incoherent or hallucinated traces. This brittleness is most acute for smaller LVLMs, which are attractive for efficient deployment but are the most unreliable at faithful multi-step reasoning. The second category fine-tunes agents on chart-specific data such as ChartGemma (Masry et al., 2025b) and BigCharts-R1 (Masry et al.), which first curate

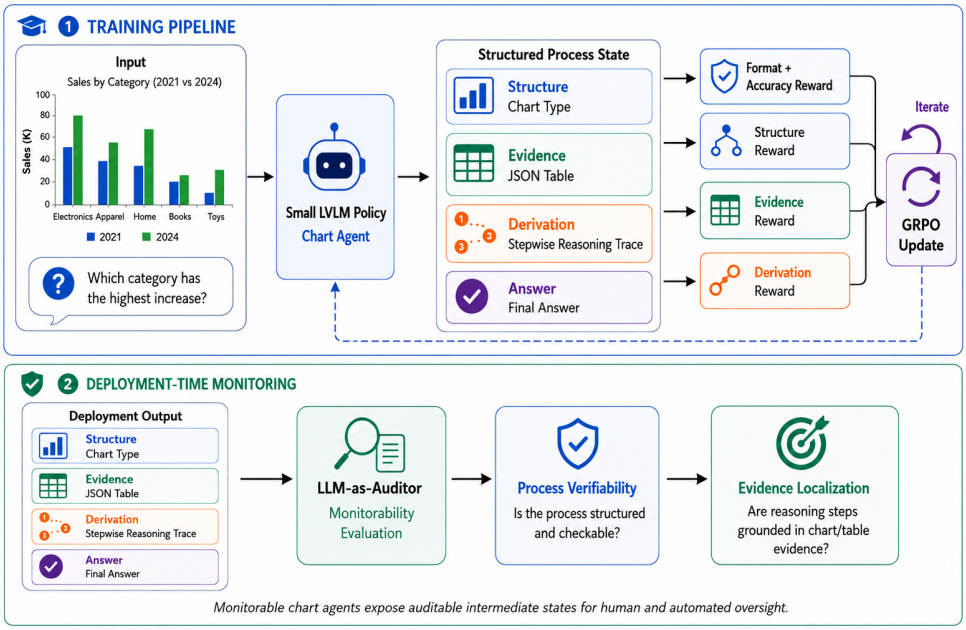


Figure 1: Overview of Chart-RVR. Top (training): a small LVLM policy emits a structured process state — chart type (Structure), reconstructed JSON table (Evidence), and a stepwise trace (Derivation) — and each block earns a verifiable reward that drives the GRPO update. Bottom (deployment): the same exposed blocks are scored by an LLM-as-auditor along Process Verifiability and Evidence Localization, so failures can be localized.

gold rationales and train via Supervised Finetuning (SFT) or RL-based approaches like GRPO. Even still, their rationales remain only *partially controllable*, and the rationale itself is never an explicit verification target - making it unsuitable for effective deployment.

As a consequence, fine-tuning chart agents on the final answer is not enough. Even if the answer is correct, the rationale may read fluently but misclassify the chart type, read a hallucinated value, select the wrong evidence, or miscompute the steps taken to get to the answer. SFT on reasoning traces (Masry et al., 2025b; Carbune et al., 2024; Zhang et al., 2024) imitates demonstration tokens, but token-level imitation does not guarantee verifiability, as a model can reproduce dataset-specific explanation styles while still incorrectly reading intermediate values. Current systems utilizing Reinforcement Fine-Tuning, like GRPO-style (Guo et al., 2025; Shao et al., 2024) outcome optimization, have shown to achieve high benchmark scores, but are even less monitorable as the rollouts generated are free form, without any constraints. Hence, systems like Masry et al. which use *answer-level* GRPO remain insufficient for monitorability as it offers no signal about whether perception, extraction, evidence selection, or arithmetic failed.

To mitigate this problem, we posit that monitorable chart agents require decomposing the reasoning process into three verifiable blocks: *Structure*, *Evidence*, and *Derivation*. The *Structure* block identifies the chart type, trivially verified by visual inspection, confirming whether values should be read from bars, lines, or sectors. Subsequently, the *Evidence* block reconstructs the underlying data table, exposing the precise values and labels used downstream and making hallucinated, missing, or inconsistent entries easy to detect. Finally, the *Derivation* block captures the algorithmic trace that composes this evidence into the answer via localization, arithmetic, averaging, etc., i.e., the **process** to compute the final answer. Together, these blocks *convert* unconstrained free-form reasoning into a structured process state that can be verified, localized, and monitored at deployment. A high-level schematic is shown in Figure 1.

Building on this premise, we propose **Chart-RVR**, a reinforcement learning framework for training *monitorable* chart agents with verifiable structure and process rewards. Chart-RVR pairs GRPO with chart-specific rewards that elicit a controlled CoT rationale comprising - (i) the chart type, (ii) underlying table reconstruction, (iii) a stepwise evidence-gathering and computation (derivation) trace, and (iv) the final answer. Unlike prior methods that curate gold rationales but reward only the final answer, Chart-RVR makes *each block of the process an explicit, checkable reward target*,

aligning the model with intermediate states that stakeholders can inspect and audit. Most CoT and curated-rationale baselines fall below even non-explanatory direct prompting, whereas Chart-RVR exposes a fully verifiable process while exceeding it across six benchmarks, with the largest margins under distribution shift. Beyond accuracy, we also propose quantifiable monitorability with a triangulated protocol: ground-truth *surrogate metrics* (chart-type accuracy and table reconstruction), an *oracle information-gain* measure quantifying how much a rationale raises a held-out LVLM's certainty in the correct answer, and an *LLM-as-auditor* scoring *Process Verifiability* and *Evidence Localization*. Our contributions are as follows:

- We formulate *monitorable chart reasoning* as a deployment-oriented task in which models must expose structured intermediate states that support verification and evidence localization.
- We propose **Chart-RVR**, a GRPO-based framework that trains small LVLMs with verifiable process rewards over chart-type prediction, JSON table reconstruction, and algorithmic process conformity, making the full reasoning process, not just the final answer, a checkable target.
- We show that Chart-RVR attains state-of-the-art accuracy among comparable LVLMs across six benchmarks *without* the explainability-accuracy tradeoff, with the strongest gains on Out of Domain (OOD) data with a comprehensive controlled reward-only comparison.
- We introduce a triangulated monitorability evaluation combining surrogate metrics, an oracle information-gain measure, and a two-axis *LLM-as-auditor* (*Process Verifiability* and *Evidence Localization*), and show that Chart-RVR produces markedly more verifiable and better-localized rationales than baselines.

2 Related Work

Chart Reasoning. Chart reasoning has been an active area of research recently. Benchmarks for studying chart-related downstream tasks, such as chart-to-table conversion, chart captioning, chart factoid-based question answering, etc. are widely used to evaluate VLMs. Multiple chart-specific VLMs have been proposed, such as Unichart (Masry et al., 2023), MatCha (Liu et al., 2023), Pix2Struct (Lee et al., 2023), etc., with considerable success in some of the downstream tasks. However, most of the proposed models struggle when the complexity of the questions increases, which requires relatively deeper reasoning. Some new benchmarks (Hegde et al., 2025; Ma et al., 2025) have been proposed to measure both reasoning and accuracy performance in tandem. As a consequence, newer chart reasoning models such as Chartgemma (Masry et al., 2025b), TinyChart (Zhang et al., 2024), ChartAssistant (Meng et al., 2024), ChartBench (Xu et al., 2023), ChartInsights (Wu et al., 2024), etc., have been proposed to output rationales with their predictions. Newer approaches utilize contrastive learning (Dai et al., 2025) or use visual tools (Fu et al., 2025).

Chain-of-thought in LVLMs. Chain-of-thought entails prompting LLMs to think step by step before outputting the final prediction, and provides improvement in both performance and interpretability of LLMs through explicit natural-language reasoning traces (Wei et al., 2022). However, similar observations are not found in LVLMs, where CoT-type prompting significantly degrades performance (Zhang et al., 2025), especially in smaller models. Several approaches have attempted to improve CoT in LVLMs with further pre-training (Xu et al., 2024), reinforcement learning (Zhang et al., 2025; Xie et al., 2024; Liu et al., 2025b), etc. The degraded CoT performance also hinders the explainability of LVLMs (Jiaqi et al., 2025).

Comparison to Related Work. Concurrently, BigCharts-R1 (Masry et al.) introduces a large-scale chart-reasoning corpus constructed via web scraping and trains Qwen-family LVLMs with standard, answer-level GRPO. Our contribution is complementary along two axes. First, rather than a new dataset or a backbone-specific run, Chart-RVR is a model- and architecture-agnostic recipe built entirely from pre-existing datasets, yielding consistent gains across three LVLM families (Qwen2.5VL, Gemma3, InternVL3.5). Second, and more fundamentally, BigCharts-R1 rewards only the final answer, whereas Chart-RVR makes the intermediate process a checkable target, decomposing the rationale into verifiable Structure and Evidence blocks and a process-conformity Derivation reward. Finally, our out-of-domain evaluation targets in-the-wild benchmarks (EvoChart, ChartQAPro, ChartBench), whereas BigCharts-R1 reports OOD results primarily on synthetic corpora (e.g., FigureQA, DVQA). Hence, Chart-RVR is a transferable, process-supervised methodology that generalizes to realistic charts, rather than a dataset- or backbone-specific result.

3 Methodology

3.1 Problem Setup and Monitorable Decomposition

Let $\mathcal{D} = \{(x_i, q_i, y_i^*, a_i^*)\}_{i=1}^N$ be a dataset of chart images $x_i \in \mathcal{X}$, queries $q_i \in \mathcal{Q}$, ground-truth answers $y_i^* \in \mathcal{Y}$, and reference rationales $a_i^* \in \mathcal{A}$. Mirroring the three verifiable blocks from the Introduction, each rationale factorizes as $a_i^* = (c_i^*, T_i^*, w_i^*)$: the *Structure* block $c_i^* \in \mathcal{C}$ (chart type), the *Evidence* block $T_i^* \in \mathcal{T}$ (underlying data table), and the *Derivation* block $w_i^* \in \mathcal{W}$ (reasoning trace). A policy π_θ (an LVLM with parameters θ) produces completions $o_i = (\hat{a}_i, \hat{y}_i) \sim \pi_\theta(\cdot \mid x_i, q_i)$ with $\hat{a}_i = (\hat{c}_i, \hat{T}_i, \hat{w}_i)$, where $\hat{a}_i$ is the full structured rationale and $\hat{y}_i$ the final answer.

Chart-RVR elicits responses in a three-block monitorable format, as one tagged completion and assigns rewards to each block, so failures can be localized to Structure, Evidence, and Derivation rather than an unformatted opaque trace. **Structure** and **Evidence** are explicit ground-truth targets and are cast as two verifiable surrogate sub-tasks, namely - predicting the chart type $\hat{c} \in \mathcal{C}$ from a fixed type set (conditioning the model on type-specific cues, e.g. bar lengths vs. pie sectors), and reconstructing the data table $\hat{T} = (\hat{C}, \hat{R})$ as JSON with `'columns'` and `'rows'` (grounding reasoning in explicit tabular structure), since errors in $\hat{T}$ corrupt the prerequisites for $\hat{y}$). Finally, the Derivation block has no single ground-truth label and is instead shaped by our Process-Conformity signal (Section 3.5). All rewards are optimized jointly with GRPO.

3.2 Group Relative Policy Optimization (GRPO)

GRPO (Shao et al., 2024; Guo et al., 2025) extends PPO (Schulman et al., 2017) to group-relative advantage estimation with verifiable rewards. It elicits a set of samples forming a rollout group, computes rewards across them, and updates the policy with the objective discussed below.

Rollout groups. For each (x_i, q_i) we sample G rollouts $\{o_j\}_{j=1}^G \sim \pi_{\text{old}}(\cdot \mid x_i, q_i)$ from a frozen behavior policy π_{old}, each a completion $o_j = (\hat{a}_j, \hat{y}_j)$ parsed into its blocks by the schema tags. Let $\text{tok}(o_j) = (z_1^{(j)}, \ldots, z_{|o_j|}^{(j)})$ be its tokenization and $z_{<t}^{(j)} = (z_1^{(j)}, \ldots, z_{t-1}^{(j)})$ its prefix.

Objective. Within each group, absolute rewards $\{R_j\}_{j=1}^G$ are converted to relative advantages $\hat{A}_j$:

$$\bar{R} = \frac{1}{G}\sum_{j=1}^{G} R_j, \qquad s_R^2 = \frac{1}{G}\sum_{j=1}^{G}(R_j - \bar{R})^2,$$
$$\hat{A}_j = \frac{R_j - \bar{R}}{\max(1,\, s_R)}. \tag{1}$$

The policy is updated with a clipped surrogate and a token-averaged KL penalty:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(x_i,q_i,o_j)\sim\mathcal{D},\,\pi_{\text{old}}}\Big[L_{\text{clip}}(\theta) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\Big],$$
$$L_{\text{clip}}(\theta) = \frac{1}{G}\sum_{j=1}^{G}\frac{1}{|o_j|}\sum_{t=1}^{|o_j|}\min\Big\{\rho_{j,t}(\theta)\hat{A}_j,$$
$$\text{clip}\big(\rho_{j,t}(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_j\Big\}. \tag{2}$$

$$\rho_{j,t}(\theta) = \frac{\pi_\theta\Big(z_t^{(j)} \mid x_i, q_i, z_{<t}^{(j)}\Big)}{\pi_{\text{old}}\Big(z_t^{(j)} \mid x_i, q_i, z_{<t}^{(j)}\Big)}.$$

Here $\epsilon > 0$ is clip range, π_{ref} the initial reference policy, and $\beta > 0$ weights the token-averaged KL.

3.3 Schema Rewards: Format, Length, and Accuracy

Three independent rewards enforce output validity and a formatted answer block. The format reward also enforces the block schema that makes each block individually parseable and thus localizable.

Format reward. Regex validation of the two-block schema (`<think>` then `<answer>`) with nested `<type>` (chart type) and `<table>` (JSON table). We set $R_{\text{fmt}} = 1$ if all checks pass, else 0.

Length (sufficiency) reward. Overly short traces underspecify the reasoning while overly long ones overthink and hallucinate (Liu et al., 2025a); since our traces are conditioned on the type and table first, we reward tokenized lengths $\ell(o_j)$ within thresholds $0 < \eta_1 \le \eta_2$:

$$R_{\text{len}} = 1, \;\text{ if }\; \eta_1 \le \ell(o_j) \le \eta_2, \text{otherwise }\; 0.$$

Answer accuracy. Textual answers are normalized by norm (lowercase, strip trailing symbols) and matched exactly; numeric answers are matched within a scale-invariant tolerance τ to absorb imprecise post-computation values:

$$R_{\text{acc}} = \begin{cases} \mathbf{1}\{\text{norm}(\hat{y}) = \text{norm}(y^\star)\}, & \text{textual}, \\ \mathbf{1}\left\{\frac{|\hat{y}-y^\star|}{|y^\star|} \le \tau\right\}, & \text{numeric}. \end{cases} \tag{3}$$

The combined schema reward is $R_{schema} = R_{fmt} + R_{len} + R_{acc}$.

3.4 Structure and Evidence Rewards

The structure and evidence sub-blocks extracted from each response are then individually collected. These two sub-blocks are the verifiable core of monitorability - each is checked against ground truth, enabling a monitor to confirm directly what the model perceived. Their combined reward is $R_{\text{surr}} = R_{\text{type}} + R_{\text{table}}$.

Structure reward (chart type). For ground-truth type c^* and prediction $\hat{c}$, an exact match after normalization:

$$R_{\text{type}}(\hat{c}, c^*) = \mathbf{1}\big[\text{norm}(\hat{c}) = \text{norm}(c^*)\big].$$

Evidence reward (table reconstruction). The model emits a JSON table $\hat{T} = (\hat{C}, \hat{R})$ against ground truth $T^* = (C^*, R^*)$, with C^* the headers and R^* the rows, of the form {'columns':{.. , ..},'rows':{[..],[..],..,[..]} }.

$$R_{\text{table}}(\hat{T}, T^*) = \begin{cases} \underbrace{\frac{1}{|C^*|}\sum_{c \in C^*} \mathbf{1}\{c \in \hat{C}\}}_{\text{column header accuracy}} & \text{if valid table} \\ \quad + \underbrace{\frac{1}{|R^*|}\sum_{r \in R^*} \frac{1}{|r|}\sum_{j} \mathbf{1}\{r_j = \hat{r}_j\}}_{\text{cell accuracy}}, & \\ 0 \ \text{ otherwise.} & \end{cases} \tag{4}$$

Each correct header contributes $1/|C^*|$ and each correct cell $1/(|R^*| \times |r|)$. A parseable JSON adds a modest 0.5 (for reward smoothness); an unparseable one yields $R_{\text{table}} = 0$, encouraging syntactic validity.

3.5 Derivation Reward: Process Conformity

The Structure and Evidence rewards discussed above incentivize accurate perception by the agent which is a prerequisite for accurate reasoning. The Derivation block governs *how* the evidence is utilized to compute the final answer. As Answer-only rewards are sparse and gameable, models can guess or retrofit a fluent rationale (Lee and Hockenmaier, 2025) around a correct answer, hence undermining the monitorability of the agent. Similarly, *token-level* imitation (SFT) entrenches dataset-specific styles without ensuring a faithful process. We instead reward *procedural alignment* to a gold reasoning skeleton *in embedding space rather than by surface form*. As chart reasoning is a semi-structured problem, the model may phrase steps in the derivation freely, but must gather the right evidence and apply operations in a consistent order. This gives dense, step-localized credit, discourages decorative CoT. Enforcing an algorithmic skeleton rather than a lexical template improves robustness under domain shift. Conformity *shapes* the derivation to be auditable, while the hard Structure, Evidence, and answer rewards remain the bedrock of correctness in the Derivation process.

Given a text-embedding model ϕ, we score two sentences a, b by a rescaled cosine similarity $s(a, b) = \frac{1}{2}\big(1 + \cos(\phi(a), \phi(b))\big) \in [0, 1]$, and split the derivation into two stages: gathering the appropriate data, then reasoning over it.

Stage 1: Stepwise grounding (R_{eg}). We split the derivation into steps and score the first m steps $\hat{w}_{[:m]}$ *stepwise* against the reference $w^*_{[:m]}$, giving localized credit for correctly grounded evidence-gathering.

Stage 2: Reasoning alignment (R_{rs}). The remaining derivation $\hat{w}_{[m:]}$ is scored as a whole against $w^*_{[m:]}$, preventing drift into degenerate traces.

$$R_{eg} = \tfrac{1}{m}\sum_{i}^{m} s(\hat{w}_{[:m](i)}, w^*_{[:m](i)}); \quad R_{rs} = s(\hat{w}_{[m:]}, w^*_{[m:]}). \tag{5}$$

The process-conformity reward is $R_{proc} = R_{eg} + R_{rs}$.

Total reward. The final reward is a weighted sum of the Schema rewards ($R_{schema} \in [0, 3]$), the Structure-and-Evidence rewards (R_{surr}), and the Process-Conformity reward ($R_{proc} \in [0, 2]$), with $\lambda_1, \lambda_2 > 0$:

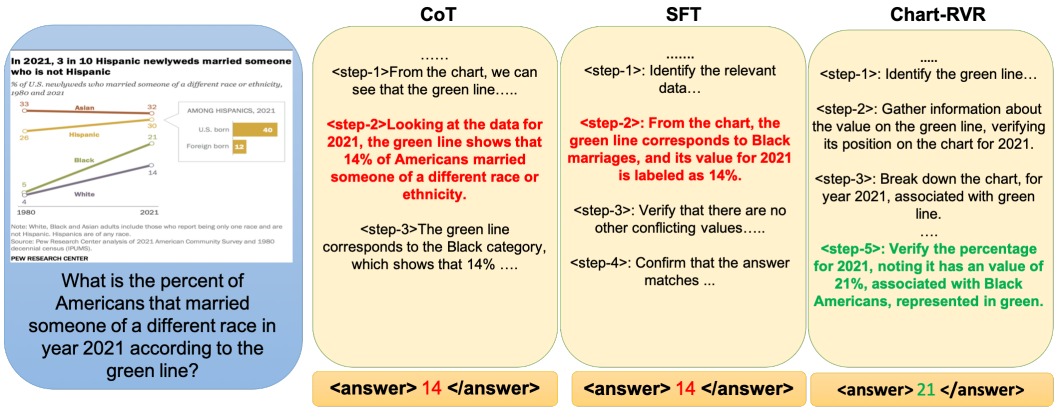
$$R = R_{schema} + \lambda_1 R_{surr} + \lambda_2 R_{proc}. \tag{6}$$

4 Experiments and Results

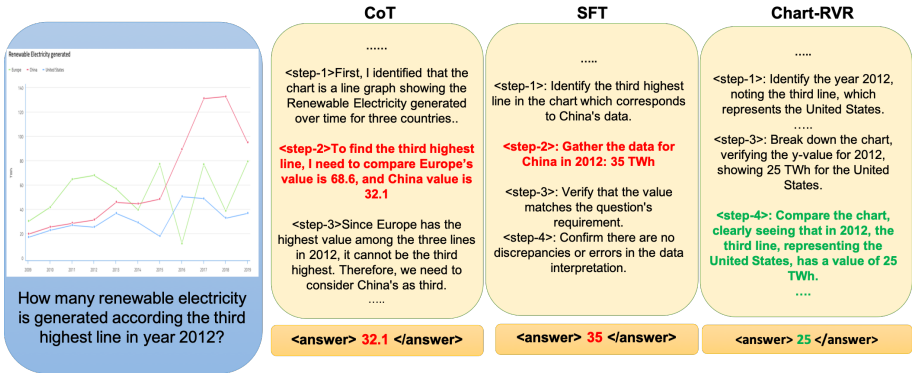
4.1 Dataset and Model Settings

Datasets. We build our CoT training data from three datasets: ChartQA (Masry et al., 2022) (factual and reasoning questions), PlotQA (Methani et al., 2020) (synthetic factoid questions), and ChartFC (Akhtar et al., 2023) (binary reasoning questions); their test splits serve as our **in-domain** benchmarks. For **out-of-domain** evaluation we use EvoChart (Huang et al., 2025) (irregular in-the-wild charts), ChartQAPro (Masry et al., 2025a) (harder charts and questions), ChartBench (Xu et al., 2023) (complex reasoning), and the reasoning-only CharXiv (Wang et al., 2024).

Models and baselines. All main experiments use Qwen2.5VL-3B-Instruct (Bai et al., 2025); to show generalizability, we also train Gemma3-3B-it (Team et al., 2025) and InternVL3.5-4B



(a) **Evidence failure: wrong value off the correct series.** All three identify the green (Black-Americans) line, but CoT and SFT extract the wrong value at the evidence-gathering step (red), reading 14% instead of 21%. Notably, SFT's later "verification" steps are decorative—they affirm the fabricated value rather than re-reading the chart, illustrating how a fluent, self-confirming rationale conceals its own error. Chart-RVR re-grounds in the chart, recovers the correct value (green), and confines the decision to a single checkable step.



(b) **Evidence failure: wrong series identified.** A distinct failure mode—selecting which line is third-highest in 2012. CoT picks Europe and SFT picks China (red), then each confidently computes a value from the wrong series; Chart-RVR correctly localizes the third-highest line to the United States and reads 25 TWh (green). Because the error is confined to the series-identification step, an auditor can pinpoint and repair it without re-deriving the trace.

Figure 2: **Localizable failures on EvoChart (OOD).** Verbatim rationale excerpts (elisions marked "...") for three conditions: the base model with CoT prompting, the SFT model, and Chart-RVR. Erroneous steps are in red, correctly grounded steps in green. In both cases the baselines fail at the *Evidence* block—value extraction in (a), series identification in (b)—and the failure pins to a single, identifiable step rather than diffusing across a fluent trace. Crucially, step structure alone is insufficient: the SFT traces are equally step-structured yet still fail, and their verification steps merely rubber-stamp the hallucinated value; Chart-RVR's Structure and Evidence rewards instead force the trace to re-ground in the chart. See Appendix for additional examples.

(Wang et al., 2025). We compare against Chart-Gemma (Masry et al., 2025b), a PaliGemma-based model emitting executable-Python rationales and the strongest explainable baseline at 3-4B scale. We further compare against BigCharts-R1 (Masry et al.); as neither its data nor models are public, we reimplement their reward recipe faithfully and train it on the *same backbone and data* as ours (the `Q2.5VL-SFT+GRPO` row), isolating the comparison to the reward design.

Evaluation. We report Relaxed Accuracy (Masry et al., 2022), counting a numeric prediction correct within a 5% tolerance,

$$\texttt{RelAcc}(\hat{y}, y^*) = \mathbf{1}\frac{|y^* - \hat{y}|}{|y^*|}[\leq 0.05] \quad \text{for } y^* \neq 0$$

matching the tolerance τ of our accuracy reward (Equation 3); non-numeric answers use exact match. For ChartFC, since some semantically correct "True/False" replies are mis-scored under Yes/No matching, we append "Answer Yes/No" to the prompt.

Hyperparameters. The process-conformity step count is $m = 3$ (the average trace length), and the length-reward band is $[\eta_1, \eta_2] = [150, 400]$ tokens. Remaining settings and reward curves are in the Appendix.

4.2 Benchmark Performance

Table 1 reports accuracy on six benchmarks. Chart-RVR outperforms all explainable baselines - in-

Table 1: Main benchmark results across 6 diverse chart datasets. The 'Exp?' column signifies if the approach is explainable (i.e., outputs CoT rationales or equivalent, like Python programs). The performance improvement is more pronounced on out-of-domain (OOD) datasets (last 3 columns).

Approach	Exp?	ChartQA	PlotQA	ChartFC	EvoChart	ChartQAPro	ChartBench
Direct Prompting							
Q2.5VL-Ins	✗	82.0	80.5	74.4	48.72	25.7	66.04
Explainable Models with Rationales							
Q2.5VL-Ins (CoT)	✔	73.12	52.72	69.20	29.6	15.80	51.16
ChartGemma (Masry et al., 2025b)	✔	76.44	33.28	70.33	36.96	10.93	40.56
Fine-tuned Models with Rationales							
Q2.5VL-SFT	✔	83.08	74.18	77.30	46.08	23.56	64.64
Q2.5VL-SFT + GRPO ((Masry et al.))	✔	84.18	77.51	77.45	45.14	24.92	66.32
Q2.5VL-Ins (A+F+L)	✔	76.72	56.22	58.58	38.88	17.55	48.1
Q2.5VL-Ins (A+F+L+Tasks)	✔	81.8	76.24	63.85	51.68	27.66	65.28
Chart-RVR-3B (Ours)	✔	84.56	**78.68**	77.62	53.36	28.38	68.32
Curated Data Fine-tuned Models with Rationales							
Q2.5VL-SFT-Hard	✔	84.28	75.54	77.90	49.36	23.20	65.12
Chart-RVR-3B-Hard (Ours)	✔	**85.76**	77.9	**80.07**	**54.24**	**28.64**	**69.46**

cluding ChartGemma and the BigCharts-R1 reward recipe and uniquely among explainable methods, *exceeds non-explainable direct prompting* (85.76 vs. 82.0 on ChartQA), whereas CoT and ChartGemma fall well below it. Gains are observed in in-domain 1–3% and OOD datasets EvoChart **+7.28%**, ChartQAPro **+4.82%**, ChartBench **+3.68%**. The Hard variant adds a further 1-2% gains, implying the effectiveness of data curation. (Note: PlotQA is not reasoning-heavy). On the significantly hard OOD reasoning-only dataset - CharXiv (Table 2), Chart-RVR-Hard improves **+4.7%** over SFT (Note the accuracy is adjudged by LLM judge as defined in CharXiv protocol).

Reward ablation. The rows in Table 1 isolate each reward group. Standard GRPO (A+F+L) underperforms SFT - **naive GRPO is ineffective**; the surrogate-task rewards (A+F+L+Tasks) recover and surpass it OOD; and the process-conformity reward delivers the final, consistent gains across all benchmarks, confirming its effectiveness over generic GRPO.

Table 2: Performance on the CharXiv reasoning subset.

Method	CharXiv (reasoning)
CoT	26.2
Q2.5VL-SFT-Hard	29.6
Chart-RVR-3B-Hard	**34.3**

4.3 Robustness across Architectures and Seeds

To confirm the gains are not artifacts of a single split or backbone, across backbones (Table 3), Chart-RVR yields consistent OOD improvements on both Gemma3 and InternVL3.5, where naive GRPO+Tasks is inconsistent, trailing only on InternVL ChartQAPro (at par), evidencing architecture-agnostic generalization. Next, we (i) retrain on three random 6k subsets of ChartQA/PlotQA/ChartFC. Across seeds, Chart-RVR beats SFT by **1–2%** ID and **4–6%** OOD with low variance (Table 9) (Appendix), matching the single-split results and indicating robust convergence.

Table 3: **Ablation across baselines on OOD benchmarks.** Subcolumns are Gemma-3 and InternVL-3.5 (IVL). Chart-RVR outperforms across datasets.

	EvoChart		ChartQAPro		ChartBench	
Setting	**Gemma**	**IVL**	**Gemma**	**IVL**	**Gemma**	**IVL**
CoT	21.04	45.36	24.82	15.91	42.28	54.42
SFT	21.92	44.08	23.56	**29.34**	54.14	63.16
GRPO (A+F+L+Tasks)	42.26	50.04	29.51	20.43	53.18	64.62
Chart-RVR (Ours)	**42.48**	**50.24**	**32.59**	**29.34**	**58.18**	**64.78**

4.4 Monitorability

Beyond accuracy, we assess whether rationales are *monitorable* via a triangulated protocol covering the three properties: *grounded* metrics computed against ground-truth/oracle targets (no LLM judgment) for **(1) verifiability** and **(2) localizability**; an *LLM-as-auditor*, validated against those grounded signals, for **(3) auditability**; and an oracle information-gain measure of rationale **(4) usefulness**.

(1) Verifiability (grounded). Structure accuracy $\mathrm{Acc}_c = \mathbf{1}[\mathrm{norm}(\hat{c}) = \mathrm{norm}(c^*)]$ and table fidelity (normalized edit distance ED, Eq. 7) mea-

sure whether the Structure and Evidence blocks check out against ground truth (Table 4).

$$E_T(\hat{T}, T^*) = \frac{1}{|C^*|} \sum_{c \in C^*} \mathbf{1}[c \notin \hat{C}] + \frac{1}{|R^*|r} \sum_{r \in R^*} \sum_j \mathbf{1}[r_j \neq \hat{r}_j]. \tag{7}$$

As shown in Table 4, SFT and Chart-RVR both lift chart-type accuracy (the base model is already decent), while Chart-RVR improves table reconstruction on EvoChart (OOD) by 0.06 over SFT - explicitly grounding reasoning in verifiable data.

Table 4: Verifiability and HR metrics: Chart Type Accuracy (Acc, ↑) and table reconstruction (Tab, edit-distance ↓). Last column represents Hallucination Rate (HR, ↓). Dataset order follows Table 1.

Dset	CoT			SFT			Chart-RVR		
	Acc	Tab	HR	Acc	Tab	HR	Acc	Tab	HR
CQA	0.87	0.46	0.15	0.94	**0.38**	0.11	**0.95**	0.49	**0.10**
PQA	0.70	0.65	0.12	**0.78**	1.13	0.08	0.77	**1.03**	**0.08**
CFC	1.00	0.65	0.18	1.00	0.20	0.14	**1.00**	**0.20**	**0.13**
EC	0.74	0.99	0.31	0.81	1.28	0.26	**0.84**	**1.22**	**0.18**
CPro	0.69	0.72	0.46	0.70	1.05	0.41	**0.72**	**1.03**	**0.34**

(2) Localizability. To compute the fidelity of each evidence gathering step, we reuse the tolerant matching of numerical values in the first m steps i.e. the set $\mathcal{V}(\hat{w})$ of numeric quantities cited in the derivation (extracted by a deterministic parser), a value v is *supported* when some cell u of the reference table T^* lies within relative tolerance τ,

$$\text{match}(v, T^*) = \mathbf{1}\Big[\exists\, u \in T^* : \frac{|v-u|}{\max(|u|,\varepsilon)} \leq \tau\Big],$$

where ε is a small constant guarding against division by zero, grounding precision is calculated as $\text{GP}(\hat{w}) = |\mathcal{V}|^{-1} \sum_{v \in \mathcal{V}} \text{match}(v, T^*)$ and hallucination rate $\text{HR} = 1 - \text{GP}$ then measure how many cited values are genuinely traceable to the chart - under the same 5% band used for answer scoring, so a step counts as grounded iff its number is as close to a real data point as a correct answer must be. Chart-RVR attains the highest GP and lowest HR, with the largest margins on the OOD benchmarks (Table 4).

Auditability (LLM-as-auditor). An oracle $\mathcal{O} \in \{\text{GPT-4o}, \text{LLaVA-Next-72B}\}$, given $(x, q, \hat{a})$, scores *Process Verifiability* (PV: is the trace decomposed into checkable steps stating intermediate quantities?) and *Evidence Localization* (EL: is each step tied to a specific chart/table element?) on 1–5 (rubric in Appendix). Chart-RVR scores highest on both axes under both oracles (Table 5), with the +Tasks → Ours jump isolating process-conformity as the driver.

Table 5: **LLM-as-auditor scores** (1–5 ↑), averaged over six benchmarks under two oracles. **PV**: Process Verifiability; **EL**: Evidence Localization. The "Surrogate (no PC)" row is the `A+F+L+Tasks` ablation; the gain to ours isolates the process-conformity (PC) reward.

	GPT-4o		LLaVA-Next-72B	
Method	**PV ↑**	**EL ↑**	**PV ↑**	**EL ↑**
CoT prompting	2.41	2.18	2.55	2.30
SFT	3.12	2.96	3.20	3.05
Surrogate (no PC)	3.58	3.41	3.63	3.49
Chart-RVR (Ours)	**4.36**	**4.28**	**4.41**	**4.33**

Usefulness ($\Delta \log P$). Finally, the Explainable Information Gain $\Delta \log P = \log P_{\mathcal{O}}(y^* \mid x, a) - \log P_{\mathcal{O}}(y^* \mid x)$ measures how much a rationale raises an oracle's certainty in the correct answer (DeYoung et al., 2020; Tutek et al., 2025; Paul et al., 2024) (Qwen2.5VL-72B oracle). Table 6 shows CoT rationales *reduce* certainty (negative), while Chart-RVR *adds* information, especially OOD (EvoChart +0.02 vs. CoT −9.13; PlotQA +3.66; ChartQAPro +2.41). All methods dip on ChartQA (base-model memorization) but improve on the harder ChartQA-human split. A second oracle (LLaVA-Next-72B, App. Table 10) and a human study (App. Fig. 13) corroborate, and Figure 2 gives a qualitative example.

Table 6: Explainable Information Gain ($\Delta \log P$) on 'Hard' variants; positive means the rationale raises certainty in the correct answer. We observe identical result trends on LLaVA-72B in the Appendix.

Dataset	Δ CoT	Δ SFT	Δ Chart-RVR
ChartQA	-5.04	**-2.22**	-4.3
ChartQA (human)	-3.46	+0.09	**+0.3**
PlotQA	-2.25	+1.13	**+3.66**
EvoChart	-9.13	-6.82	**+0.02**
ChartQAPro	-0.04	+1.75	**+2.41**

5 Conclusion

We introduced Chart-RVR, a reinforcement learning framework that turns chart reasoning into a monitorable process by augmenting GRPO with verifiable rewards over chart-type prediction and table reconstruction, together with a process-conformity objective that shapes faithful, step-by-step derivations. Across six benchmarks and three LVLMs, Chart-RVR improves accuracy while producing markedly more verifiable and better-localized rationales, with the largest gains under distribution shift — evidence of stronger OOD generalization than SFT and vanilla GRPO.

Limitations

Our study targets small (3–4B) LVLMs across three backbones; gains at larger closed-source model scales are untested. Our LLM-as-auditor axes inherit judge subjectivity, mitigated by two oracles, grounded-signal corroboration, and a (small-scale) human study. We evaluate only English chart-QA, and leave a rigorous *Intervention Utility* protocol, repairing localized failures from the exposed process state to future work.

References

Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. 2023. Reading and reasoning over chart images for evidence-based automated fact-checking. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 399–414.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Victor Carbune, Hassan Mansoor, Fangyu Liu, Rahul Aralikatte, Gilles Baechler, Jindong Chen, and Abhanshu Sharma. 2024. Chart-based reasoning: Transferring capabilities from LLMs to VLMs. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 989–1004, Mexico City, Mexico. Association for Computational Linguistics.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Yue Dai, Soyeon Caren Han, and Wei Liu. 2025. Graph-based multimodal contrastive learning for chart question answering. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2658–2663.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4443–4458.

Xingyu Fu, Minqian Liu, Zhengyuan Yang, John Richard Corring, Yijuan Lu, Jianwei Yang, Dan Roth, Dinei Florencio, and Cha Zhang. 2025. Refocus: Visual editing as a chain of thought for structured image understanding. In *Forty-second International Conference on Machine Learning*.

Melody Y Guan, Miles Wang, Micah Carroll, Zehao Dou, Annie Y Wei, Marcus Williams, Benjamin Arnav, Joost Huizinga, Ian Kivlichan, Mia Glaese, and 1 others. 2025. Monitoring monitorability. *arXiv preprint arXiv:2512.18311*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Shamanthak Hegde, Pooyan Fazli, and Hasti Seifi. 2025. Chartqa-x: Generating explanations for charts. *arXiv preprint arXiv:2504.13275*.

Muye Huang, Han Lai, Xinyu Zhang, Wenjun Wu, Jie Ma, Lingling Zhang, and Jun Liu. 2025. Evochart: A benchmark and a self-training approach towards real-world chart understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3680–3688.

Mohammed Saidul Islam, Raian Rahman, Ahmed Masry, Md Tahmid Rahman Laskar, Mir Tafseer Nayeem, and Enamul Hoque. 2024. Are large vision language models up to the challenge of chart comprehension and reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3334–3368, Miami, Florida, USA. Association for Computational Linguistics.

WANG Jiaqi, Kevin Qinghong Lin, James Cheng, and Mike Zheng Shou. 2025. Think or not? selective reasoning via reinforcement learning for vision-language models. In *The Exploration in AI Today Workshop at ICML 2025*.

Jinu Lee and Julia Hockenmaier. 2025. Evaluating step-by-step reasoning traces: A survey. *arXiv preprint arXiv:2502.12289*.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR.

Chengzhi Liu, Zhongxing Xu, Qingyue Wei, Juncheng Wu, James Zou, Xin Eric Wang, Yuyin Zhou, and Sheng Liu. 2025a. More thinking, less seeing? assessing amplified hallucination in multimodal reasoning models. *arXiv preprint arXiv:2505.21523*.

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. 2023. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12770.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. 2025b. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*.

Chenghao Ma, Junpeng Ding, Jun Zhang, Ziyan Ma, Huang Qing, Bofei Gao, Liang Chen, Meina Song, and 1 others. 2025. Sci-reason: A dataset with chain-of-thought rationales for complex multimodal reasoning in academic areas. *arXiv preprint arXiv:2504.06637*.

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

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, Megh Thakkar, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2025a. ChartQAPro: A more diverse and challenging benchmark for chart question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19123–19151, Vienna, Austria. Association for Computational Linguistics.

Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. 2023. Unichart: A universal vision-language pretrained model for chart comprehension and reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14662–14684.

Ahmed Masry, Abhay Puri, Masoud Hashemi, Juan A Rodriguez, Megh Thakkar, Khyati Mahajan, Vikas Yadav, Sathwik Tejaswi Madhusudhan, Alexandre Piché, Dzmitry Bahdanau, and 1 others. Bigcharts-r1: Enhanced chart reasoning with visual reinforcement finetuning. In *Second Conference on Language Modeling*.

Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. 2025b. ChartGemma: Visual instruction-tuning for chart reasoning in the wild. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 625–643, Abu Dhabi, UAE. Association for Computational Linguistics.

Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. Chartassistant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7775–7803.

Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. *arXiv preprint arXiv:2402.13950*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasović, and Yonatan Belinkov. 2025. Measuring chain of thought faithfulness by unlearning reasoning steps. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9946–9971.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformer Reinforcement Learning. `https://github.com/huggingface/trl`.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, and 1 others. 2024. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting

elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Yifan Wu, Lutao Yan, Leixian Shen, Yunhai Wang, Nan Tang, and Yuyu Luo. 2024. ChartInsights: Evaluating multimodal large language models for low-level chart question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12174–12200, Miami, Florida, USA. Association for Computational Linguistics.

Yuxi Xie, Guanzhen Li, Xiao Xu, and Min-Yen Kan. 2024. V-DPO: Mitigating hallucination in large vision language models via vision-guided direct preference optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13258–13273, Miami, Florida, USA. Association for Computational Linguistics.

Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. 2024. Llava-o1: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*.

Zhengzhuo Xu, Sinan Du, Yiyan Qi, Chengjin Xu, Chun Yuan, and Jian Guo. 2023. Chartbench: A benchmark for complex visual reasoning in charts. *arXiv preprint arXiv:2312.15915*.

Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. 2024. TinyChart: Efficient chart understanding with program-of-thoughts learning and visual token merging. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1898, Miami, Florida, USA. Association for Computational Linguistics.

Ruohong Zhang, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. 2025. Improve vision language model chain-of-thought reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1662, Vienna, Austria. Association for Computational Linguistics.

A Appendix

A.1 Why Structured Process States Improve Monitorability

The three blocks are motivated by an elementary information-theoretic observation. Let Q be the query, Y the answer, and $Z^* = (C^*, T^*, D^*)$ the structured process state - chart type, underlying table, and derivation. By the chain rule for conditional mutual information, conditioning the answer on these blocks is non-increasing in uncertainty,

$$H(Y \mid Q) \;\geq\; H(Y \mid Q, C^*) \;\geq\; H(Y \mid Q, C^*, T^*) \;\geq\; H(Y \mid Q, Z^*),$$

and each step is strict exactly when that block carries an answer-relevant information not already present in the previous ones (e.g., $I(Y; T^* \mid Q, C^*) > 0$). In the limit, when the answer is a deterministic function of the query, table, and derivation, $Y = f(Q, T^*, D^*)$, the residual uncertainty vanishes, $H(Y \mid Q, T^*, D^*) = 0$. This motivates rewarding the blocks jointly rather than optimizing the final answer alone: each block is incentivized precisely to the extent that it supplies answer-relevant evidence. We emphasize that this characterizes the *oracle* state Z^*; our method emits estimates $\hat{Z} = (\hat{C}, \hat{T}, \hat{D})$, and the empirical sections (Sec. 4.4) measure how faithfully the emitted state recovers this information and how verifiable and localizable it is in practice.

A.2 Salient Dataset Construction Details

Chart-RVR CoT Reasoning Datasets. Although the training datasets discussed contain plenty of examples, there is a distinct lack of a reliable source of ground-truth rationales, data tables, and chart-type annotations associated with them. As a consequence, we generate a CoT Chart dataset sampled from the training splits of ChartQA, ChartFC, and PlotQA. To generate faithful CoT rationales, inferring the chart type and generating the associated data tables, we utilize a large-scale SOTA LVLM - Qwen2.5VL-72B. The prompt template for generating the dataset is provided in Figure 4 (Appendix), where both the query and label are provided to the model. (1) *CoT Datasets:* For the CoT dataset, we randomly sample 2,000 datapoints each from the aforementioned datasets based on a specific seed for a total of 6,000 training samples. (2) *CoT-Hard Dataset*: A significant issue in randomly sampling training points from the datasets is the lack of diversity and dominance of easy samples, which constitute queries that require no reasoning, e.g., 'title of the chart. To alleviate this, we specifically sample data from the human-annotated reasoning subset of ChartQA (labeled 'human') and random samples from the ChartFC and PlotQA data. We filter out overly simplistic questions from the dataset for a total of 30,000 training samples.

Chart Types ($\mathcal{C}$). Even though a large variety of chart types exist, most of the commonly found charts can be categorized into 10 fundamental categories. We instantiate a controlled set of chart families: *Line*, *Bar*, *Stacked Bar*, *Pie*, *Histogram*, *Scatter*, *Area*, *Stacked Area*, *Bubble*, and *Treemap*. [1]

Construction. To generate the Chart-RVR CoT datasets, we utilize a large open-source LVLM as an oracle, namely Qwen2.5VL-72B-Instruct (Bai et al., 2025). For the Chart-RVR datasets, the prompt template utilized is shown in Figure 4.

1. *Normalization.* Images are resized with aspect-ratio preservation to a minimum of 512 leading edge size; tables are canonicalized to a header (columns) + rectangular body (rows) with string conversion where applicable. We map the dataset's native chart label to the chart types above.
2. *Templated prompting.* We instantiate Figure 4 with (a) the canonical chart label, (b) a compact task description, and (c) formatting constraints (typed tags, JSON schema, operation tokens). The template explicitly separates (i) *data gathering* steps from (ii) *computation* steps to enable process supervision.
3. *Sampling and pre-screen.* For each (image, question) pair, we output rationales via deterministic inference with temperature set to 0. Candidates failing structural checks (JSON parse, length bounds, required tags) are discarded before reward computation.

NOTE: As opposed to many other approaches, our dataset can be constructed from pre-existing datasets, imparting transparency.

Manual Filtering of sub-par rationales. Although our rewards prune many low-quality generations automatically, we perform a light manual pass to remove pathological rationales that would otherwise pollute training data points. A rationale

[1] We normalize synonyms to a canonical label: *Column*→*Bar*, *Donut*→*Pie*, *Point*→*Scatter*. Grouped bars are labeled *Bar*; cumulative variants are *Stacked Bar*.

is marked *sub-par* and discarded if any of the following holds:

1. *Unparseable structure:* the emitted JSON blocks (table tag) fail to parse or violate schema (missing keys, non-rectangular rows).
2. *Data hallucination:* The rationale cites a category/series not present in the ground-truth table, i.e., non-sensical characters, special symbols, etc.
3. *Length:* We filter out all rationales with fewer than 3 steps of reasoning.

We utilize the following procedure for final dataset validation:

1. For pass 1, the prompt structure shown in Figure 4 is used to generate the chain-of-thought data.

2. For pass 2, all samples with overtly short/long and incorrect reasoning traces are filtered (i.e. trace length less than 3 or more than 8 based on manual inspection and outlier rejection). To validate if a trace is wrong, we algorithmically check if the last CoT line trace contains the correct answer. A total of 358 samples are filtered out in this step out of 15000.

3. For pass 3, three doctoral-level researchers are employed to validate the reasoning traces of a randomly sampled data subset (1000 samples). Each researcher filtered fewer than 6 samples (1%) in total, with a 100% agreement between them.

We will release the filtered dataset upon acceptance for transparency and faithful reproduction. Additionally, the smaller CoT Dataset with appropriate seeds will also be released.

Agreement of annotation done using human validation

We manually validated 100 randomly sampled oracle-annotated charts, each from the EvoChart and ChartQAPro datasets (note no ground truth table annotations are available for either). We categorize each data sample into 2 categories: (1) Needs Approximation? - which entails guessing the correct values from the chart axes rather than labelled counterparts, "Factoid/Direct" - which entails looking at axes marks to determine the correct values. In Table 7, we report the Cohen's κ: the inter-annotator agreement score b/w human (Annotator-1) and Oracle model (Annotator-2). We also show examples of the categorization in Figure 3.

Category	#Samples	Cohen's κ
Needs Approximation	48	0.91
Factoid/Direct	152	0.98
Overall	200	0.97

Table 7: Inter-annotator agreement between human annotators and the Oracle model.

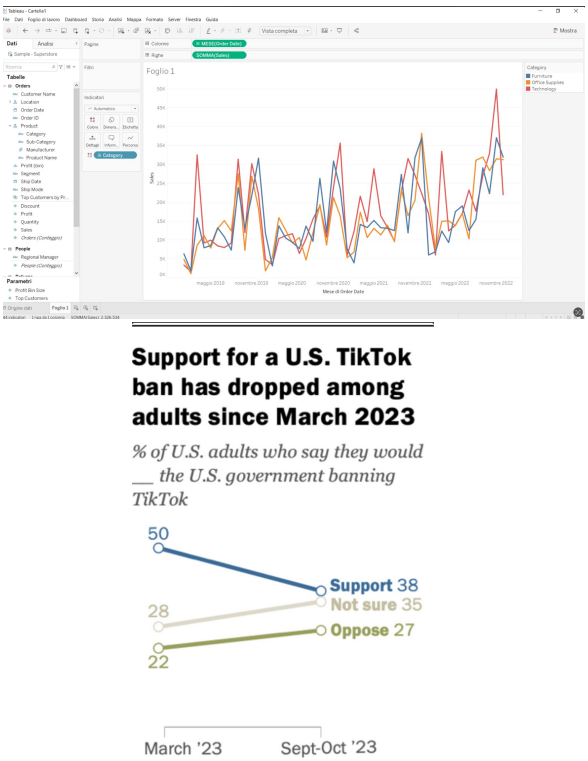


Figure 3: Examples of charts where the Oracle and human annotations (TOP) **need approximation** and (BOTTOM) are **factoid/direct**; the corresponding annotated data tables may differ.

Emphasizing OOD benchmarks. Many widely used chart reasoning datasets like ChartQA, PlotQA, and ChartFC have been repeatedly incorporated (in whole or in part) into instruction-tuning corpora, synthetic data expansions, and public multimodal mixtures that contemporary LVLMs are trained or aligned on, together with related datasets such as FigQA and DVQA[2]. This creates a realistic risk of distributional familiarity and data leakage at pretraining/finetuning time, yielding optimistic "in-distribution" estimates. To assess

[2]Note: Because most LVLM training mixtures are only partially disclosed, we cannot exhaustively audit every provider. Our designation reflects (i) the public availability and long shelf-life of ChartQA/PlotQA/ChartFC, (ii) their documented use in multiple open instruction-tuning recipes as documented in InternVL's technical report (Chen et al., 2024)

generalization beyond such exposure, we therefore evaluate on 'truly' out-of-distribution (OOD) results on EvoChart, ChartBench, and ChartQA-Pro, which were not used in model pretraining or alignment and contain chart styles, templates, and question programs that differ from the legacy sets. As shown in the Appendix, it can be observed that datasets like EvoChart possess significantly harder chart images than benchmark sets like ChartQA. Throughout the paper, we treat ChartQA/PlotQA/ChartFC as 'ID' baselines and use EvoChart/ChartBench/ChartQA-Pro to measure robustness, reasoning fidelity, and explainability under genuine distribution shift. Truly generalizable chart reasoning with explanations remains an open problem for frontier LVLMs.

A.3 Structured Prompt Evaluation

First, we evaluated how chain-of-thought prompting affects off-the-shelf LVLMs in Table 8. Although CoT prompting shows gains in LLMs, we observed (Liu et al., 2025b; Xu et al., 2024) an opposite trend in LVLMs, where standard CoT prompting degrades performance by a large margin compared to direct prompting. Next, we examined how structured prompting (i.e., instructing the model to emit chart type, table, and the reasoning process along with the answer) using the prompt structure shown in Figure 5improves the results over standard CoT prompting. As can be seen, our structured prompting approach improved performance, but is still significantly less than direct prompting.

Table 8: Comparison of Direct, CoT, and Structured Prompt on off-the-shelf Qwen2.5VL-3B-Instruct. Our prompt template is detailed in Figure 5.

Dataset	Direct	CoT	Structured
ChartQA	82.0	41.8	**73.12**
PlotQA	80.5	31.82	**52.72**
ChartFC	74.4	48.02	**69.20**
EvoChart	48.72	18.72	**29.60**
ChartQAPro	25.7	12.01	**15.80**
ChartBench	66.04	29.4	**51.16**

A.4 Implementation Details

Training Details. SFT: We utilize the same system prompt format as in Figure 5 for fine-tuning. We train the entire model for 3 epochs, with a learning rate of $1e-5$ for the LLM and projector, while $2e-6$ for the vision tower, with a warm-up ratio of 0.03. **Chart-RVR:** We utilized the TRL (von Werra et al., 2020) implementation of GRPO with a maximum prompt length of 4096, maximum completion length 768, and number of generations (rollouts) 4 per sample. To further reduce prompt lengths, we utilize the JSON notation to represent the underlying chart tables. For Process Conformity Rewards, we used sentence embeddings from a lightweight embedding model, MiniLM-L6-v2 (Reimers and Gurevych, 2019). We train all models for 4 epochs with a learning rate set as $1e-6$. For SFT, each training sample's CoT trace and answer is appropriately formatted into the expected prompt format and adhering to `<think><type></type><table></table> ...</think><answer></answer>`. The validation dataset is created using **500** randomly sampled data points from the training sets of ChartQA, ChartFC, and PlotQA.

- **SFT:** For SFT, we train the model for a maximum of 3 epochs and select the model with the highest accuracy on the validation set. Additionally, we utilize the AdamW optimizer for training with a training batch size of 4 across 2 NVIDIA H100 GPUs. The learning rate schedule utilized is linear with a maximum learning rate of $1e-5$ and 1000 warmup steps. During training, the entire model is trained with FP16 precision, and the vision tower is trained using an LR of $2e-6$.

- **Chart-RVR:** For Qwen2.5VL models, we utilize a maximum completion length of 768 and a total prompt length of 4096. For Gemma and InternVL models, due to more efficient vision token computations, we utilize a maximum prompt length of 3072 with a maximum completion length of 512. For Qwen2.5VL, we utilize a maximum learning rate of $1e-6$ while for Gemma and InternVL, the learning rate is reduced to $5e-7$ and $2e-7$ respectively. The training process takes place on 4 NVIDIA H100 GPUs with a per-device batch size of 2. The number of rollouts is set to 4. The total number of epochs is set at 4 for all models and configurations.

Hyperparameter Tuning. Note that the Chart-RVR setup has 3 major hyperparameters to tune, which assign relative weights to each component of the reward design. Unless otherwise stated, we use fixed values across all model families: $\lambda_1 = 1$ for the surrogate-task reward and $\lambda_2 = 1$ for the process-conformity reward. In practice, we found

the method to be insensitive to modest perturbations of λ_1, λ_2, and the compute budget (within ± 1 for λ_1, λ_2), due to the smooth normalization of the combined reward. With sufficient optimization steps, each reward term saturates, and the final ranking of methods remains unchanged. We therefore fix λ_1 and λ_2 to simplify training and ensure comparability across backbones.

Evaluation Setup. For all our evaluations, we utilize the public test sets of the datasets. For ChartQA[3], ChartQAPro[4], EvoChart[5], and ChartFC[6], we utilize the entire test sets sampled from sources as listed in the footnotes. For ChartBench[7] and PlotQA[8], due to their massive sizes, we sample a 5000-sized subset using a fixed seed. For all models, we resize the leading image edges to a minimum and maximum of 448 and 812, respectively, using Bicubic interpolation. For all structured CoT, SFT, and GRPO evaluations, we utilize the prompt template in Figure 5 as the system prompt for all models.

A.5 Practical Reward Hacking and Mitigation Strategies

Reward hacking is a common phenomenon in Reinforcement Fine-Tuning (RFT), In this section, we demonstrate common reward-hacking behavior observed in our experiments and mitigation strategies employed by us to alleviate these concerns. Note that the reward hacking behavior is usually observed during the early stages of the training and can cause a sudden training collapse, from which the policy never recovers.

Stacked Rewards on Length. As the models are incentivized to output longer traces (rollouts) via the length reward, in some cases, this results in unrelated characters being outputted to 'fill' in the extra tokens by new line tokens ('\n') or repeating multiple identical redundant steps. This is often observed when smaller LVLMs, which do not output their CoT traces reliably, are suddenly incentivized to output much longer traces. If left unmitigated, this can cause a training collapse where the model can never recover from this particular policy, which technically still maximizes the reward. To alleviate this, we utilize a stacked reward design, where a partial reward is assigned for reasoning lengths above certain thresholds. This treats the length reward maximization as a 'warm' start process and nudges the model to output longer traces gradually and not immediately upon the start of training. We utilize a 0.5 reward for token lengths of 100 or more, and finally, the full 1.0 is set when the length exceeds η_1 tokens. Additionally, we also penalize 'filler' tokens by checking if more than 5 '\n' occur in contiguous tokens; we provide a 0 reward.

Stacked Rewards on Table Reconstruction. As detailed in the main text, we utilize the Table reconstruction rewards in tranches - a successful JSON parse of the tokens inside each rollout's <table> tags gives a 0.5 reward. As the reward function on table reconstruction is extremely dense, we warm start the process by assigning a 0.5 extra reward if 'columns' and 'rows' appear in the rollout's JSON parsing.

A.6 Training Curves and Reward Maximization Behavior

In Figure 6, we visually demonstrate the reward maximization during the Chart-RVR training process on 3 splits of the CoT dataset. We observe that the format rewards are maximized very early in the training with minimal changes observed throughout the process. The accuracy reward gradually improves and stagnates after about 60k steps (about 2 epochs). Similarly, the Table Rewards and Process Conformity Rewards demonstrate a smooth increase initially and then are maximized around the same number of steps. All rewards show a massive improvement during early training.

Table 9: Comparison between fine-tuned models using 3 distinct seeds of the ChartRVR-CoT Dataset. Chart-RVR improves performance over SFT across all datasets, with OOD improvements more pronounced.

Dataset	Direct	CoT	SFT	Chart-RVR
ChartQA	82.0	73.12	83.18 ± 0.33	**83.87 ± 0.68**
PlotQA	**80.50**	52.72	76.05 ± 1.65	**78.71±0.20**
ChartFC	74.4	69.20	76.67 ±0.54	**78.08±1.36**
EvoChart	48.72	29.6	46.50 ± 0.36	**52.16±0.86**
ChartQAPro	25.7	15.80	24.52 ±0.48	**28.30±1.00**

[3] https://huggingface.co/datasets/HuggingFaceM4/ChartQA
[4] https://huggingface.co/datasets/ahmed-masry/ChartQAPro
[5] https://github.com/MuyeHuang/EvoChart
[6] https://github.com/mubasharaak/ChartCheck
[7] https://huggingface.co/datasets/SincereX/ChartBench
[8] https://github.com/NiteshMethani/PlotQA/blob/master/PlotQA_Dataset.md

A.7 Consistent Improvement by Chart-RVR

We report the average performance and standard deviation in Table 9 over all 3 seeds. As can be seen, Chart-RVR consistently outperformed SFT across both ID and OOD datasets by **1-2%** and **4-6%** respectively, demonstrating the efficacy of our method over SFT. In addition, the average results are at par with the results on a single CoT dataset split (as shown in Table 1), highlighting robust convergence.

A.8 Additional Explainability Results

Table 10 reports the improvement on the explainability metric for LLaVA-Next-72B under three training modes (CoT, SFT, Chart-RVR). Chart-RVR delivers the largest gains on 4/5 datasets, notably +0.38 on ChartQA and +0.23 on EvoChart—while SFT narrowly leads on PlotQA (+0.1642 vs +0.1632). Overall gains are modest on harder sets (e.g., ChartQAPro = +0.058), but Chart-RVR provides the most consistent uplift in explainability.

Table 10: Usefulness Results using LLaVA-Next-72B.

Dataset	Δ CoT	Δ SFT	Δ Chart-RVR
ChartQA	+0.27	+0.29	**+0.38**
ChartQA (human)	+0.21	+0.21	**+0.25**
PlotQA	+0.1605	+0.1642	+0.1632
EvoChart	+0.16	+0.14	**+0.23**
ChartQAPro	+0.049	+0.043	**+0.058**

A.9 LLM-as-Auditor Rubric

We score two monitorability axes - Process Verifiability (PV) and Evidence Localization (EL) - with the system prompts below, issued to each oracle $\mathcal{O} \in \{\text{GPT-4o}, \text{LLaVA-Next-72B}\}$. The chart image x, query q, and rationale $\hat{a}$ are supplied as the user turn; the oracle returns a single integer 1–5 per axis. Identical prompts are used across oracles and conditions; scores are parsed by regex and averaged over the eval set.

System Prompt — Process Verifiability (PV)

```
You are a strict auditor of chart-reasoning
rationales.  You are given a chart image,
a question,  and a model's step-by-step
rationale. Judge ONLY PROCESS VERIFIABILITY:
whether the rationale is decomposed into
discrete, independently checkable steps that
explicitly state the intermediate quantities
and the operations applied to them.
Do NOT reward fluency, length, or confidence.
Do NOT judge whether the final answer is
correct. A long, well-written rationale with
no stated numbers or unseparated steps must
score LOW.
Score on this 1-5 scale:  5 - Every step
is discrete and self-checkable; each cited
value and each operation (read, compare, add,
average,  etc.)  is explicitly stated, so
a reader could re-verify the whole chain
without guessing.  4 - Mostly discrete and
checkable; one step omits a value or merges two
operations, but the chain is still auditable.
3 - Partially structured; some steps state
quantities/operations,  others are vague or
bundle several actions into one. 2 - Largely
unstructured;  quantities or operations are
mostly implicit and the chain cannot be
checked step by step. 1 - A single block or
free-form paragraph with no separable steps
or stated intermediate quantities.
Output format: return ONLY the integer score,
e.g. "4". No explanation.
```

System Prompt — Evidence Localization (EL)

```
You are a strict auditor of chart-reasoning
rationales.  You are given a chart image,
a question,  and a model's step-by-step
rationale. Judge ONLY EVIDENCE LOCALIZATION:
whether each reasoning step is grounded in a
specific, identifiable element of THIS chart
(a named series, bar, slice, axis tick, or
table cell) rather than referring to the
figure vaguely or citing values that are not
visible in it.
Use the chart image to verify grounding. Do
NOT reward fluency or confidence. Do NOT judge
whether the final answer is correct. A step
that asserts a number with no identifiable
source in the chart must score LOW, even if
the number happens to be right.
Score on this 1-5 scale: 5 - Every step names
the specific chart element it reads from, and
every cited value is traceable to a visible
point/label in the chart. 4 - Nearly all steps
are grounded to specific elements;  at most
one value is stated without a clear visual
referent.  3 - Mixed;  some steps point to
specific elements while others reference the
chart only generally ("the data shows...").
2 - Mostly ungrounded;  few steps identify
a specific element and several values have
no visible source.  1 - No step is tied to
an identifiable chart element; values appear
without any traceable source.
Output format: return ONLY the integer score,
e.g. "4". No explanation.
```

A.10 Visual Comparison between CoT, SFT, and Chart-RVR

In this section, we demonstrate visual examples where Chart-RVR outperforms both CoT and SFT. First, we demonstrate Chart-RVR's effectiveness on OOD data on the EvoChart benchmark in Figure 7. As can be seen, CoT and SFT misidentify the number of categories as they only look at the leg-

end on the left, while Chart-RVR correctly counts the number of colors. In the second case, Chart-RVR correctly identifies the value of the second April, which is a duplicate entry on the x-axis.

Next, in Figure 8, Chart-RVR correctly identifies the 'BeReal' category, which is not picked up by both CoT and SFT, as it is only one data point on the line chart. This reinforces the need to condition reasoning on faithful table reconstruction. In the next image, Chart-RVR is able to capture the entity '3.6' accurately, where both CoT and SFT fail.

In Figures 9,11, and 12, we report cases where Chart-RVR outperforms both CoT and SFT on the ChartQA dataset. We observe that Chart-RVR is particularly accurate in cases where the chart is extremely complex.

A.11 Failure Cases

Finally, in Figure 10, we report 2 failure cases of Chart-RVR on the OOD dataset EvoChart. The top chart is composed of both line and bar graphs together, making the reasoning process confounded. Note that Chart-RVR correctly identifies the chart type, i.e., line, which is the correct chart to look for, but answers the approximate value 6, which is very close to 5 and can be attributed to a scaling issue. In the next chart, the values of both lines in 2023 are extremely overlapping; hence, all approaches make errors.

A.12 Human Study

Finally, to ascertain the interpretability of the rationales generated by our approach as compared to SFT and CoT, we conduct a human study with 5 graduate-level volunteers. Our study has 5 data samples from the ChartQA and EvoChart datasets, which are **correctly** classified by all 3 approaches - CoT, SFT, and Chart-RVR, and the traces output by these approaches. We do not disclose which method produces what trace, and also shuffle the options. We show the instructions for the study, a sample question, and the final results in Figure 13. As can be observed, most people prefer Chart-RVR responses for 4/5 questions as compared to SFT and CoT, implying a clear preference.

System Prompt template for CoT Dataset Generation

You are helping me answer questions on charts.
You have to look both at the chart picture and the question.
The question and the answer will be provided to you.
First you have to recover the table data from the chart image in JSON format.
For the chart image, output only a JSON object with:
"columns": list of column headers,
"rows": list-of-lists, one per data row
No prose, no comments.
1. Respond with **only** a JSON object inside a "'json code fence.
2. The JSON must use exactly this schema:
{
"columns": [...],
"rows": [...]
}
3. Do NOT output HTML, Markdown, or commentary. Any deviation gets zero reward.
Next, think step by step in as many small steps as required to answer the question based on the chart.
Lastly, also predict the type of chart out of the following:
"line", "bar", "stacked bar", "pie", "histogram", "scatterplot", "area", "stacked area", "bubble", "treemap"
Format:
Question: <question>
Answer: <answer>
Table: <json table>
Reasoning:
<step-1>: Provide a description of reasoning
<step-2>: Gather ALL the appropriate data from the chart
<step-3>: Break down the query into smaller parts and verify each part with the data
...
<step-n>: Do the final calculation or reasoning to derive the answer
<step-n+1>: VERIFY the final answer is correct for no hallucinations
Type: <type of chart>

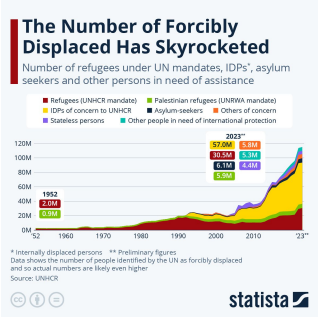
Figure 4: **System prompt template** used across Structured CoT, SFT, GRPO, and Chart-RVR.

System Prompt template for Structured CoT / SFT / GRPO / Chart-RVR

You are a vision-language assistant. You are given a chart image and a query about the chart.
Think step-by-step about how to answer the query based on the chart image and then provide the final answer.
Output format:
Respond **with exactly two blocks in order and nothing else**:
<think>
First output the type of chart in <type>, then output the underlying data table and finally, think step-by-step about how to answer the query based on the chart image and then provide the final answer. <type>
Type of chart - one word from line, bar, stacked bar, pie, histogram, scatterplot, area, stacked area, bubble, treemap. </type>
Next output the data table in the <table></table> tags
<table>
json table - for the chart image, output only a JSON object with: "columns": list of column headers, "rows": list-of-lists, one per data row
No prose, no comments.
1. Respond with **only** a JSON object
2. The JSON must use exactly this schema: { "columns": [...], "rows": [[...], [...],..., [...]] }
3. Do NOT output HTML, Markdown, or commentary. Any deviation gets zero reward.
</table>
Provide your reasoning here in steps:
<step-1>: Provide a description of reasoning
<step-2>: Gather ALL the appropriate data from the chart
<step-3>: Break down the query into smaller parts and verify each part with the data
...
<step-n>: Do the final calculation or reasoning to derive the answer
</think>
<answer>
Final answer on a single line
</answer>

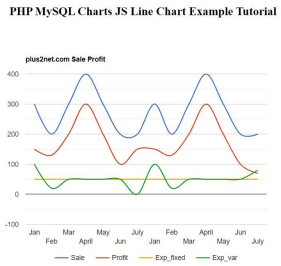
Figure 5: **System prompt template** used across Structured CoT, SFT, GRPO, and Chart-RVR.

Figure 6: Chart-RVR Reward maximization during training on 3 separate CoT datasets on Qwen2.5VL-3B.



Query: ['How many refugee categories were mentioned in the chart?']
Ground Truth: ['7']

CoT Rationale:
<type>line</type>
</step-1>From the chart, we can see that there are six categories mentioned: refugees (UNHCR mandate), Palestinian refugees (UNRWA mandate), IDPs of concern to UNHCR, Asylum-seekers, Others of concern, and Stateless persons.
</step-2>The query asks specifically about the number of refugee categories mentioned in the chart. Based on the information provided in the chart, there are two main refugee categories: refugees (UNHCR mandate) and Palestinian refugees (UNRWA
</step-3> Therefore, the answer to the question is that there are two refugee categories mentioned in the chart.
<answer>2</answer>
SFT Rationale:
<type>stacked bar</type>
<table>{'columns': ['Year', 'Refugees (UNHCR mandate)', 'Palestinian refugees (UNRWA mandate)',...</table>
<step-1>: Identify the distinct categories listed under the 'Refugees' section in the chart.
<step-2>: Count the number of unique categories identified.
<step-3>: Verify that all categories listed under 'Refugees' are accounted for without any duplicates.
<step-4>: Confirm the count matches the answer provided.
<answer>2</answer>
Chart-RVR Rationale:
<type>Stacked Bar</type>
<table>\n{"columns": ["Year", "Refugees (UNHCR mandate)", "Palestinian refugees (UNRWA mandate)",...</table>
<step-1>: Count the number of categories represented on the chart, each corresponding to a refugee category.
<step-2>: Identify the legend, noting it lists the categories: Refugees (UNHCR mandate), Palestinian refugees (UNRWA mandate), IDPs of concern to UNHCR, Asylum-seekers, Others of concern, Stateless persons, and Other people in need of international protection.
<step-3>: Verify each category, confirming there are seven categories.
<step-4>: Examine the chart, recognizing them as the refugee categories.
<step-5>: Verify the legend, confirming there are seven categories.
<step-6>: Provide a count, answering that there were 7 refugee categories mentioned in the chart.
<answer>7</answer>



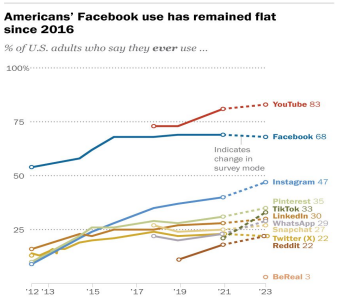
Query: ['What is the value of the red line at the second April?']
Ground Truth: ['300']

CoT Rationale:
<type>line</type>
<table>{ "columns": [\n "Month",\n "Sale",\n "Profit",\n "Exp_fixed",\n "Exp_var"\n</table>
<step-1> The chart is a line chart showing sales profit over time.
<step-2> From the chart, we can see that the red line represents the \'Profit\' category.
<step-3> We need to find the value of the red line at the second April (which is May).
<step-4> Looking at the chart, the red line reaches its peak at around 350 units in May.
<step-5> Therefore, the value of the red line at the second April (May) is approximately 350 units.
<answer>350</answer>

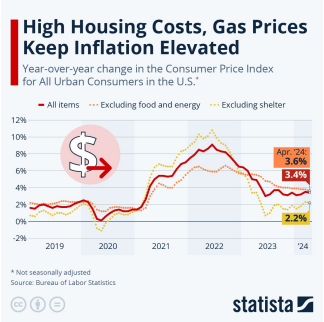
SFT Rationale:
<type>line</type>
<table>{'columns': ['Month', 'Sale', 'Profit', 'Exp_fixed', 'Exp_var'], 'rows': [['Jan', '300', '150', '50', '70'],.. </table>
<step-1>: Identify the red line which represents 'Profit'.
<step-2>: Locate the second occurrence of 'April' on the x-axis.
<step-3>: Read the corresponding y-value for the red line at that point.
<step-4>: Verify that the value is approximately 150.
<step-5>: Confirm there are no other conflicting values or errors."
<answer>150</answer>

Chart-RVR Rationale:
<type>Line</type>
<table>{"columns": ["Month", "Sale", "Profit", "Exp_fixed", "Exp_var"],"rows": [["Jan", 300, 150, 50, 100],</table>
<step-1>: Identify the red line, which represents the Profit data.
<step-2>: Examine the chart, noting the second occurrence of April. This is the second April after January.
<step-3>: Verify the value associated with this red line, comparing it to the graph.
<step-4>: Determine the value at the second April, comparing it to the data: the profit value is 300.
<step-5>: Confirm it is the red line, corresponding to the second April.
<step-6>: Provide a answer to confirm the value is 300, associated with this red line.
<answer>300</answer>

Figure 7: Example failure cases from the EvoChart dataset (OOD) where Chart-RVR outperforms CoT and SFT on 2 challenging examples.



Query: ['How many apps are mentioned in the chart?']
Ground Truth: ['11']



Query: ['What is the percentage value of excluding food and energy in April 2024?']
Ground Truth: ['3.6']

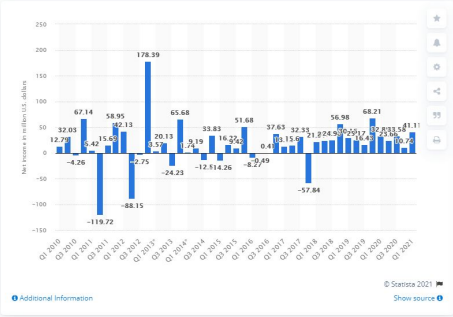
CoT Rationale:
<type>line</type>
<table>{ "columns": ["App","Year"],"rows": [["YouTube", "2016"], ["Facebook", "2016"], ["Instagram", "2016"],["Pinterest"
<step-1>The chart is a line graph showing the percentage of U.S. adults who say they ever use various social… will count the number of distinct apps mentioned in the legend at the top of the chart.\n</step-2>\n<step-3>\nI will verify that all the apps listed in the legend are indeed mentioned in the data rows.\n</step-3>\n<step-4>\nThe apps mentioned in the chart are YouTube, Facebook, Instagram, Pinterest, LinkedIn, Snapchat, Twitter (X), Reddit, and BeReal
<answer>9</answer>
SFT Rationale:
<type>line</type>
<table>\n{'columns': ['Year', 'YouTube', 'Facebook', 'Instagram', 'Pinterest', 'LinkedIn', '....</table>
<step-1>: Provide a description of reasoning The chart lists various social media platforms along with their usage percentages over
<step-2>: Gather ALL the appropriate data from the chart There are ten distinct lines representing different social media platforms: YouTube, Facebook, Instagram, Pinterest, LinkedIn, WhatsApp, Snapchat, Twitter (X), Reddit, and BeReal.\n\n
<step-3>: Break down the query into smaller parts and verify each part with the data The question asks for the number of apps mentioned. By counting the unique lines in the chart, we can determine there are 10 distinct platforms.\n\n
<step-4>: Do the final calculation or reasoning to derive the answer Counting the lines confirms that there are indeed 10 apps
<answer>10</answer>
Chart-RVR Rationale:
<type>Line</type>
<table>{"columns": ["Year", "YouTube", "Facebook", "TikTok", "Instagram", "Pinterest", "LinkedIn"....</table>
<step-1>: Count the number of app names on the chart, identifying them along the legend.
<step-2>: Gather information about each app, recognizing it on the graph, which represents YouTube, Facebook, TikTok, Instagram, Pinterest, LinkedIn, WhatsApp, Snapchat, Twitter (X), Reddit, and BeReal.
<step-3>: Verify the chart, counting each app mentioned.
<step-4>: Examine the data, confirming there are 11 apps in total.\n<step-5>: Verify the legend, confirming it lists all the applications.\n<step-6>: Provide a count, answering that there are 11 apps mentioned in the chart.
<answer>11</answer>

CoT Rationale:
<type>scatterplot</type>
<table>\n{\n "columns": [\n "Year",\n "All items",\n "Excluding food and energy",\n "Excluding shelte…</table>
<step-1> The chart shows the year-over-year change in the Consumer Price Index (CPI) for all urban consumers in the U.S. over time.</step-1>
<step-2> From the data provided, we can see that the CPI excluding food and energy was at 2.2% in April 2024.</step-2>
<step-3> This information directly answers the question asked.
<answer>2.2</answer>

SFT Rationale:
<type>line</type>
<table>{'columns': ['Year', 'All items', 'Excluding food and energy', 'Excluding shelter'], 'rows': [['2019', 'None',</table>
<step-1>: Identify the relevant data point for April 2024 under the 'Excluding food and energy' category.
<step-2>: The data shows that the percentage value for 'Excluding food and energy' in April 2024 is 3.4%.
<step-3>: Verify that there are no other conflicting values or additional context needed for this specific question.
<step-4>: Confirm that the answer matches the provided answer of 3.4%.
<step-5>: VERIFY the final answer is correct for no hallucinations.
<answer>3.4</answer>

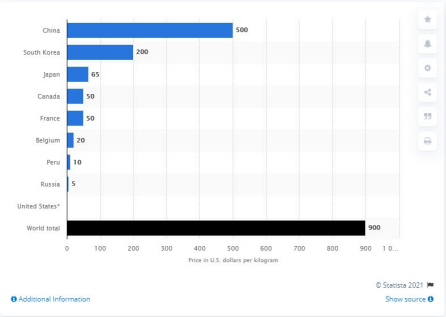
Chart-RVR Rationale:
<type>Line</type>
<table>{"columns": ["Year", "All items", "Excluding food and energy", "Excluding shelter"],"rows": [["2019", 2, 2, 2], …</table>
<step-1>: Examine the chart, identifying the year for April 2024, which is represented as \'24\'.
<step-2>: Gather the value associated with \'Excluding food and energy\' for this year. It represents the data for April 2024.
<step-3>: Verify the percentage value, confirming it is 3.6%.
<step-4>: Compare the chart with the information given, noting the value associated with April 2024.
<step-5>: Confirm that this corresponds to the \'Excluding food and energy\' category, with a value of 3.6%.
<step-6>: Provide a direct answer to confirm the question asks for this percentage value in April 2024.
<answer>3.6

Figure 8: Example failure cases from the EvoChart dataset (OOD) where Chart-RVR outperforms CoT and SFT on 2 challenging examples.



Query: ["What was The Times Company's net income in the first quarter of 2021?"]
Ground Truth: ['41.11']

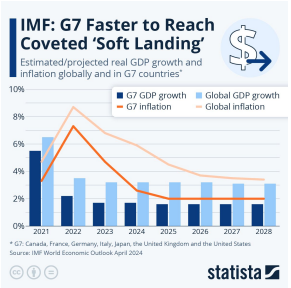
CoT Rationale:
<type>bar</type>
<table>{"columns": ["year", "Net income (in million U.S. dollars)"], "rows": [["Q1 2010", 12.79], ["Q2..,</table>
<step-1>: The chart is a bar chart. It shows the Net income (in million U.S. dollars) over time.
<step-2>: I can see that the x-axis represents the quarters from Q1 2010 to Q4 2021, while y-axis is Net income.
<step-3>: The query asks for The Times Company\'s net income in the first quarter of 2021. Looking at the chart, I can see that the first quarter of 2021 corresponds to Q1 2021 on the x-axis.
<step-4>: In the first quarter of 2021, the bar reaches up to 10.76 million U.S. dollars.
<step-5>: Therefore, The Times Company\'s net income in the first quarter of 2021 was 10.76 million U.S. dollars.
<answer>10.76</answer>
SFT Rationale:
<type>bar</type>
<table>{'columns': ['Quarter', 'Net income in million U.S. dollars'], 'rows': [['Q1 2010', '12.79'], ['Q2 2010', '-4.26'],</table>
<step-1>: Identify the relevant data point for Q1 2021 from the chart.
<step-2>: Locate the value corresponding to Q1 2021, which is labeled as 11.1.
<step-3>: Verify that the value matches the expected answer.
<step-4>: Confirm there are no discrepancies or errors in the data.
<step-5>: VERIFY the final answer is correct for no hallucinations.
<answer>11.1</answer>
Chart-RVR Rationale:
<type>Bar</type>
<table>{"columns": ["Quarter", "Net income in million U.S. dollars"],"rows": [["Q1 2010", 12.79], ["Q2 2010",..</table>
<step-1>: Examine the chart, identifying the data for the first quarter of 2021, which is Q1 2021.
<step-2>: Gather the net income value associated with this quarter. It represents the first quarter of 2021.
<step-3>: Verify the value: the chart shows 41.11 million U.S. dollars for Q1 2021.
<step-4>: Confirm this is the net income for the first quarter of 2021.
<step-5>: Verify it corresponds to Q1 2021, confirming the question asks for this information.
<step-6>: Provide a direct answer, verifying the Times Company\'s net income in the first quarter of 2021 was 41.11 million U.S. dollars.
<answer>41.11</answer>



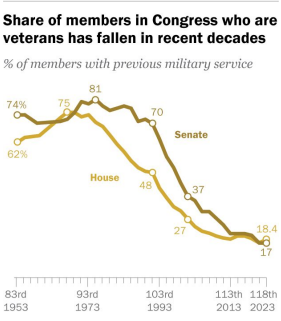
Query: ["What country was the world's largest refinery producer of indium in 2020?"]
Ground Truth: ['China']

CoT Rationale:
<type>bar</type>
<table> {"columns": ["country", "price"], "rows": [["China", 500], ["South Korea", 200],..</table>
<step-1> I observe that chart is a bar chart, which is used to compare different categories across a single dimension.
<step-2>I gather the following data from the chart:\n- China: 500\n- South Korea: 200\n- Japan: 65\n- Canada: 50\n- France: 50\n- Belgium: 20\n- Peru: 10\n- Russia: 5\n- United States*: 900
<step-3>To answer the question, I need to identify the country with the highest value, as it represents the largest refinery producer of indium in 2020.
<step-4>Comparing the values, I see that the United States has the highest value at 900.
<step-5>Therefore, the United States was the world\'s largest refinery producer of indium in 2020.
<answer>United States</answer>
SFT Rationale:
<type>bar</type>
<table>{'columns': ['Country', 'Price in U.S. dollars per kilogram'], 'rows': [['China', '500'], ['South Korea'..</table>
<step-1>: Identify the country with the highest value in the chart which represents the largest refinery producer.
<step-2>: The United States has the highest value at 900 U.S. dollars per kilogram.
<step-3>: Verify that no other country exceeds this value.
<step-4>: Confirm that the United States is indeed the largest refinery producer according to the chart.
<step-5>: VERIFY the final answer is correct for no hallucinations."]
<answer>United States</answer>
Chart-RVR Rationale:
<type>Bar</type>
<table>{"columns": ["Country", "Price in U.S. dollars per kilogram"],\n"rows": [["China", 500], ["South… </table>
<step-1>: Identify the country with the highest value for the price per kilogram, indicating the largest refinery producer of indium. The chart shows China has the highest value at 500 U.S. dollars per kilogram.
<step-2>: Examine the chart, comparing the prices for each country, noting the largest number, which represents the world total at 900 U.S. dollars per kilogram.
<step-3>: Verify the data, confirming it pertains to the world total.
<step-4>: Recognize that this total includes China at 500 U.S. dollars per kilogram, making it the largest producer.
<step-5>: Verify the question asks about the country with the largest refinery production of indium. China leads with this highest value.
<step-6>: Provide a direct answer, confirming China was the world\'s largest refinery producer of indium in 2020.
<answer>China</answer>

Figure 9: Example failure cases from the ChartQA dataset (ID) where Chart-RVR outperforms CoT and SFT on 2 challenging examples.



Query: ['What is the value of G7 inflation in year 2023?']
Ground Truth: ['5']



Query: ['How many percent of house members have previous military sevice in 2023?']
Ground Truth: ['17']

CoT Rationale:
<type>\nbar\n</type>
<table>{\ "columns": ["year", "G7 GDP growth","Global GDP growth",... "rows": ["2021", 5.6, 6.4, 7.1] </table>
<step-1> The chart shows three bars representing G7 inflation for different years. We need to find the value of G7 inflation in 2023.
<step-2> In the year 2023, the bar representing G7 inflation is located at the intersection of the year 2023 and the orange bar
<step-3> The height of the orange bar in 2023 indicates the value of G7 inflation for that year.
<step-4> The value of G7 inflation in 2023 is 3.2%.
<answer>3.2</answer>

SFT Rationale:
<type>bar</type>
<table>{'columns': ['Year', 'G7 GDP growth', 'Global GDP growth', 'G7 inflation', 'Global inflation'], 'rows': [['2021..</table>
<step-1>: Identify the relevant data point for G7 inflation in 2023.
<step-2>: Locate the year 2023 in the chart and find the corresponding value for G7 inflation.
<step-3>: Verify that the value for G7 inflation in 2023 is indeed 2.5%.
<step-4>: Confirm that there are no other conflicting values or errors in the data.
<answer>2.5</answer>

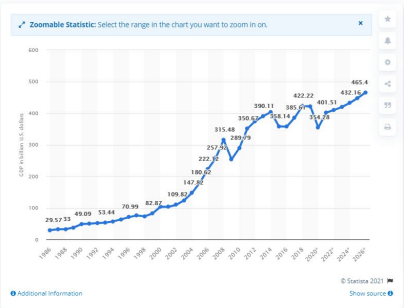
Chart-RVR Rationale:
<type>Line</type>
<table>{"columns": ["Year", "G7 GDP growth", "Global GDP growth", "G…n"rows": [["2021", 5.8, 6.4, 3.8, 6.5],</table>
<step-1>: Examine the chart, identifying the year 2023. This year corresponds to 2023.
<step-2>: Gather the value associated with G7 inflation for this year. It represents the data for 2023.
<step-3>: Verify the graph, confirming it shows the value associated with each year.
<step-4>: Locate the information for 2023, comparing it to the values for G7 inflation.
<step-5>: Verify the value, which is 6.0% for 2023.
<step-6>: Provide a direct answer to confirm the value of G7 inflation in 2023 is 6.0.
<answer>6.0</answer>

CoT Rationale:
<type>line</type>
<table>{"columns": ["decade", "House", "Senate"], "rows": [["83rd", "74%", "75%"], ["93rd", "62%", "81%"],</table>
<step-1>First, I need to identify the relevant data points for the House and Senate in the year 2023. The chart shows that in the 118th decade (which corresponds to 2023), the House has 18.4% of its members who have previous military service, while the Senate has 17%.
<step-2>Next, I should confirm if there is any additional information needed to answer the question. In this case, the data provided directly answers the question without requiring further analysis.
<step-3>Finally, I can conclude that the percentage of House members who have previous military service in 2023 is 18.4%.
<answer>18.4</answer>

SFT Rationale:
<type>line</type>
<table>{'columns': ['Year', 'House', 'Senate'], 'rows': [['83rd 1953', '74', '62'], ['93rd 1973', '75', '81'],</table>
<step-1>: Identify the relevant data point for the House in 2023.
<step-2>: From the chart, the value for the House in 2023 is approximately 18.4%.
<step-3>: Verify that there are no other conflicting values or annotations for 2023.
<step-4>: Confirm that the answer matches the provided answer of 18.4.
<step-5>: VERIFY the final answer is correct for no hallucinations.
<answer>18.4</answer>

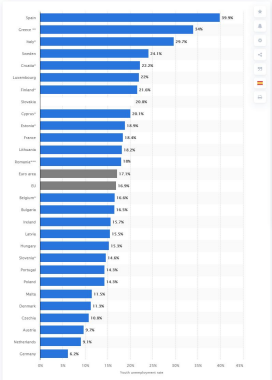
Chart-RVR Rationale:
<type>Line</type>
<table>{"columns": ["Year", "House", "Senate"],"rows": [["83rd 1953", 74, 62], ["93rd 1973", 75, 81],</table>
<step-1>: Examine the chart, identifying the year 2023 for the House members.
<step-2>: Gather the percentage value associated with this year, comparing it to the data for the House.
<step-3>: Verify the information, noting the percentage for 2023 is 18.4%.
<step-4>: Confirm it represents the House members\' percentage of having previous military service in 2023.
<step-5>: Provide a direct answer, verifying the answer is 18.4%.
<answer>18.4</answer>

Figure 10: Example failure case from the EvoChart dataset (OOD) where all Chart-RVR, CoT, and SFT output wrong answers.



Query: ['What was the GDP of the United Arab Emirates in dollars in 2019?']
Ground Truth: ['419.35']

CoT Rationale:
<type>line</type>
<table>{"columns": ["year", "GDP (in billion U.S. dollars)"], "rows": [[1986, 29.5733], [1988, 49.09], [1990, 53.44]...}</table>
<step-1> The chart is a line graph showing the GDP of the United Arab Emirates over time.
<step-2> From the x-axis, we can see that the years range from 1986 to 2028. The y-axis represents the GDP in billions..
<step-3> To find the GDP of the UAE in 2019, I need to locate the year 2019 and read the corresponding value on y-axis.
<step-4> The value for 2019 is approximately 358.14 billion U.S. dollars.
<answer>358.14</answer>
SFT Rationale:
<type>line</type>
<table>{'columns': ['Year', 'GDP (in billion U.S. dollars)'], 'rows': [['1986', '29.5733'], ['1988', '49.09'],..</table>
<step-1>: Identify the year 2019 on the x-axis of the chart.
<step-2>: Locate the corresponding value on the y-axis for the year 2019.
<step-3>: Verify that the value for 2019 is approximately 385.61 billion U.S. dollars.
<step-4>: Confirm that the answer matches the provided answer of 385.61.
<step-5>: VERIFY the final answer is correct for no hallucinations.
<answer>385.61</answer>
Chart-RVR Rationale:
<type>Line</type>
<table>{"columns": ["Year", "GDP in billion U.S. dollars"],"rows": [["1986", 29.57], ["1988", 38.5],</table>
<step-1>: Examine the chart, identifying the year 2019, which corresponds to the data we\'re looking for.
<step-2>: Gather the GDP value associated with this year. It represents the United Arab Emirates\' GDP in billions
<step-3>: Verify the graph, confirming it shows the GDP for each year, including 2019.
<step-4>: Locate the value for 2019, comparing it to the chart.
<step-5>: Verify the GDP amount, which is 422.22 billion U.S. dollars for 2019.
<answer>422.22</answer>



Query: ["What was Spain's youth unemployment rate in January 2021?"]
Ground Truth: ['39.9']

CoT Rationale:
<type>bar</type>
<table> {"columns": ["Country", "Youth unemployment rate (%)"], "rows": [["Spain", 19.9], ["Greece", 54.6],...</table>
<step-1>The chart is a bar graph showing the youth unemployment rates across different countries. The x-axis represents the countries, and the y-axis represents the youth unemployment rate in percentage terms.
<step-2>Gathering data:
To find Spain\'s youth unemployment rate, I need to locate Spain on the x-axis and read the corresponding value on y-axis.
<step-3>Verifying the data:\nI can see that Spain is listed as the first country on the x-axis, and its value on y-axis is 19.9%.
<step-4>Final calculation:\nThe youth unemployment rate for Spain in January 2021 is 19.9%.
<answer>19.9</answer>
SFT Rationale:
<type>bar</type>
<table> {'columns': ['Country', 'Youth unemployment rate'], 'rows': [['Spain', '19.9%'], ['Greece', '5.4%'],</table>
<step-1>: Identify the country in question - Spain.
<step-2>: Locate Spain's corresponding value in the chart - 19.9%.
<step-3>: Verify that the value matches the provided answer - it does.
<step-4>: Confirm there are no discrepancies or errors in the data.
<answer>19.9</answer>
Chart-RVR Rationale:
['<type>Bar</type>
<table>{"columns": ["Country", "Youth unemployment rate"],"rows": [["Spain", 39], [</table>
<step-1>: Examine the chart, identifying the information for Spain. The chart clearly indicates its youth unemployment rate.
<step-2>: Gather the data associated with Spain, noting it has a blue bar representing the country.
<step-3>: Verify the value associated with Spain, confirming it is 39%.
<step-4>: Confirm this represents the youth unemployment rate for January 2021.
<step-5>: Compare the chart with the question, verifying the answer is correct.
<step-6>: Provide a direct answer to confirm Spain\'s youth unemployment rate in January 2021 was 39%.
<answer>39.9</answer>

Figure 11: Example failure cases from the ChartQA dataset (ID) where Chart-RVR outperforms CoT and SFT on 2 challenging examples.

Query: ['How many patients came from the neighboring state of Mexico?']
Ground Truth: ['63086']

CoT Rationale:
<type>bar</type>
<table>
{"columns": ["State", "Number of patients"], "rows": [["Mexico City", 167189], ["State of Mexico",...
</table>
<step-1> The chart is a bar chart showing the number of patients from different states in Mexico.
<step-2> From the chart, I can see that the state of Mexico has the highest number of patients at 63,085.
<step-3> The question asks specifically about patients coming from the neighboring state of Mexico, which is the State of Mexico itself.
<step-4> Therefore, the answer is the number of patients from the State of Mexico, which is 63,085.']
<answer> 63085 </answer>
SFT Rationale:
<type>bar</type>
<table>
{\'columns\': [\'State\', \'Number of patients\'],
\'rows\': [[\'Mexico City\', \'67085\'], [\'State of Mexico\', \'470\'], [\'Puebla\', \'297\'],
</table>
<step-1>: Identify the relevant data for the neighboring state of Mexico which is "State of Mexico".
<step-2>: From the chart, the number of patients from the State of Mexico is 470.
<step-3>: Verify that there are no other states listed as being part of the neighboring region.
<step-4>: Confirm that the answer matches the data from the chart.
<step-5>: VERIFY the final answer is correct for no hallucinations.
<answer> 470 </answer>
Chart-RVR Rationale:
<type>Bar</type>
<table>
{"columns": ["State", "Number of patients"],
"rows": [["Mexico City", 162189], ["State of Mexico", 63086], ["Puebla", 1470],....
</table>
<step-1>: Identify the information related to the neighboring state of Mexico, which is the State of Mexico. This state has 63,086 patients, indicating it is the neighboring state in question.
<step-2>: Examine the chart, comparing it to the data for Mexico City, the largest state, with 162,189 patients.
<step-3>: Verify the number of patients associated with the State of Mexico, noting it has 63,086 patients.
<step-4>: Confirm this represents the neighboring state of Mexico, as it is the state adjacent to Mexico City.
<step-5>: Verify the question asks for the number of patients coming from this neighboring state. The answer is 63,086.
<step-6>: Provide a final calculation and reasoning, confirming the answer is correct, as it corresponds to the patients from the State of Mexico.
<answer> 63086 </answer>

Figure 12: Example failure case from the ChartQA dataset (ID) where Chart-RVR outperforms CoT and SFT

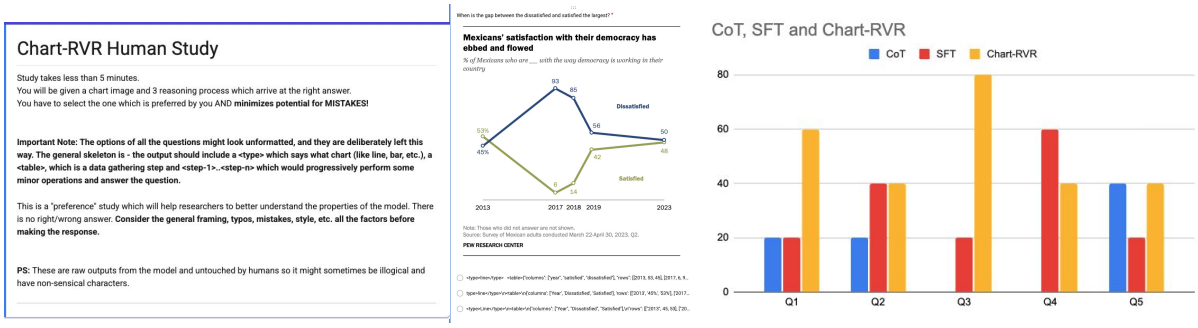


Figure 13: Layout and results of the human study. (LEFT) Instructions to the volunteers, (MIDDLE) Sample Question, and (RIGHT) Percentage of people preferring each of the 3 approaches.