CHARTORCHESTRA: Automating Sub-Agent Creation for Agentic Chart Reasoning

Anonymous ACL submission

Abstract

Existing chart-reasoning agents route every question through the same fixed roster of roles and hand-written reading rules, leaving them brittle when the chart or question type shifts; besides, the pipeline has no principled way to detect any error committed by the sub-agent or recover from it. We present **CHARTORCHESTRA**, a sub-agent-as-tools framework that builds chart sub-agents on the fly and verifies them against the chart. For each question, an orchestrator over-generates and ranks candidate chart-reading roles, synthesises a question-specific reading procedure for the top role, and curates the chart-grounded evidence the sub-agent conditions on. A judge sub-agent then inspects the candidate against the chart and returns feedback together with a critique that pinpoints numeric, reference-point, or logical inconsistencies; on rejection, the orchestrator eliminates the failed role and activates the next-ranked role conditioned on this critique, yielding cheap error recovery and early termination when the judge accepts. Through comprehensive experiments with three vision-language backbones (GPT-4o-mini, Gemini 2.0 Flash, GLM-4.5V) on four chart benchmarks (ChartQAPro, CharXiv, FigureQA, PlotQA), we show that CHARTORCHESTRA, while remaining training-free, improves the average over the strongest reasoning baseline by up to 9.37 points, lifts ChartQAPro hypothetical reasoning by 27.4 points, and reduces API consumption by 39–64%.

1 Introduction

Charts are ubiquitous, as people use them to analyse data, answer questions, and discover critical insights (Wang et al., 2024). Yet performing complex analytical tasks with charts is rarely a one-step affair: even a question that looks easy — "*did revenue grow between Q_2 and Q_3?*" — chains locating the right series, reading the right ticks, comparing values, and grounding the conclusion in chart-supported evidence. Modern vision-language models (VLMs) (Masry et al., 2025; Wang et al., 2024) have improved single-shot chart question answering, but as benchmarks shift toward multi-step, hypothetical, and conversational queries (Masry et al., 2025; Xie et al., 2025), the system *around* the model becomes the main lever for further progress.

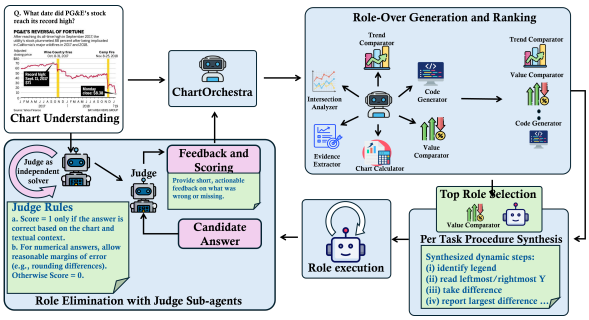


Figure 1: Overview of CHARTORCHESTRA. Given a chart and a question, the orchestrator over-generates and ranks candidate chart-reading roles, then instantiates the top-ranked role as a sub-agent by writing a question-specific chart-reading procedure and assembling the evidence it conditions on. A chart-grounded judge sub-agent verifies the candidate answer against the chart. On rejection, the orchestrator *eliminates* the failed role and instantiates the next-ranked role with a distilled critique of the previous attempt, providing principled recovery and early termination.

Two families of approaches have emerged for chart reasoning beyond simple prompting: Chain-of-Thought families and their adaptation (e.g., ChartGemma (Masry et al., 2024b) and VProChart (Huang et al., 2025b)), and agent pipelines with hand-designed roles run through fixed workflows, e.g., ChartAgent (Kaur et al., 2025). Both are *strict*: the same chain is applied regardless of question complexity, reading procedures are shared across all questions, and no verification step is included—once a wrong premise is committed to, it propagates unchecked.

Outside chart reasoning, a more flexible multi-agent alternative has emerged: the *sub-agent-as-tools* paradigm (Hosain et al., 2025; Wu et al., 2024; Hong et al., 2023), in which a main agent delegates to specialised helpers via explicit tool or sub-task calls. Two structural gaps in this paradigm carry over to chart reasoning and become especially damaging there: *No per-task recipe for the sub-agent.* When the main agent delegates, it either picks from a small set of pre-defined helpers or just hands the sub-agent an isolated copy of the working context (Schroeder et al., 2025b; Sun et al., 2025). The sub-agent then has a generic system prompt and a list of available tools, but no step-by-step plan tailored to the question at hand — nothing that says, *for this question*, which chart cues to look at, in what order, and when to switch role from visual reading to numerical computation. No principled verify-and-recover loop. Once a sub-agent returns, its output is consumed and trusted by the main agent; any retry is ad-hoc, often only by self-reflection rather than driven by a structured critique of the previous attempt and a re-tailoring of the next sub-agent.

We introduce CHARTORCHESTRA, the first sub-agent-as-tools framework for chart reasoning, designed around two technical contributions. **Per-task procedure synthesis:** for each chart question, the orchestrator over-generates and ranks candidate chart-reading roles, then *writes* a question-specific, ordered chart-reading procedure for the top role; this synthesised procedure, not a fixed system prompt or a catalogue of APIs — becomes the sub-agent's tool. **Role-elimination and recovery driven by a chart-grounded judge:** an independent judge sub-agent gives feedback, on rejection, the orchestrator *eliminates* the failed role and instantiates the next-ranked role conditioned on a distilled critique of the previous attempt, providing principled recovery and early termination when the judge accepts. Unlike Xolver (Hosain et al., 2025), which iteratively refines the same agents in parallel, CHARTORCHESTRA uses elimination-and-rerank, which we show to be a better fit for chart reasoning (Sec. 3.4). The orchestrator does not encode backbone-specific assumptions, so CHARTORCHESTRA runs unchanged on top of three different VLMs. Figure 1 summarises the framework.

Contributions. (1) Per-task procedure synthesis: We propose an orchestrator that, instead of routing a sub-agent to a fixed system prompt or tool catalogue, synthesises a question-specific chart-reading procedure that becomes the sub-agent's tool — closing the gap of the sub-agent-as-tools paradigm. **(2) Role-elimination and recovery with a chart-grounded judge:** we instantiate a verify-and-recover loop in which the judge re-derives from the chart, and on rejection, the failed role is eliminated, and the next-ranked role is instantiated with a structured critique — closing the gap. **(3) First sub-agent-as-tools framework for chart reasoning, with strong empirics:** CHARTORCHESTRA is training-free and, across four chart benchmarks (ChartQAPro, CharXiv, FigureQA, PlotQA) and three backbones (GPT-4o-mini, Gemini 2.0 Flash, GLM-4.5V), improves the average over the strongest reasoning baseline by up to 9.37 points absolute (e.g., 63.04 to 72.41 with Gemini 2.0 Flash), lifts ChartQAPro hypothetical reasoning by 27.4 points, and reduces API consumption by 39–64%.

2 The CHARTORCHESTRA Framework

2.1 Overview

A chart question answering system generally works with three main pieces of information: a chart image (C), a question in everyday language (q), and an optional description of the chart (referred to as p). This system carefully examines the chart, identifying key elements like the axes, legends, and tick marks. It uses these details to perform necessary calculations and reasoning. After processing all this information, it delivers an answer, which we can denote as y. We propose CHARTORCHESTRA, which develops an agentic solution for the chart question answering system. Rather than attempting to solve the problem directly, CHARTORCHESTRA follows a *sub-agent-as-tools* paradigm. It operates over a system-level action space $\mathcal{A}_{\text{orch}}$.

$$\mathcal{A}_{\text{orch}} = \{\text{Build}(\Phi),\ \text{Delegate}(\Phi),\ \text{Finish}(y)\}. \tag{1}$$

The action $\text{Build}(\Phi)$ specifies the next sub-agent; $\text{Delegate}(\Phi)$ spawns it and lets it interact with the chart-reading environment, returning a candidate answer, the action $\text{Finish}(y)$ terminates the interaction. Different systems that utilize sub-agents as tools primarily differ in how the orchestrator parameterizes Build whether it selects from a fixed catalog or, as in our approach, constructs the sub-agent on demand.

Our orchestration strategy is guided by three principles: (i) *build* a question-appropriate sub-

Algorithm 1 CHARTORCHESTRA

Require: chart C, question q, optional description p, role budget K, backbone $\mathcal{M}$
Ensure: final answer $y^\star$

$S^\star \leftarrow$ OVERGENERATEANDRANK(C, q, K) ▷ Sec. 2.2
$\mathcal{F} \leftarrow \emptyset$; $\hat{a}^\star \leftarrow \emptyset$
for $t = 1$ **to** K **do**
  $\mathcal{R}_t \leftarrow S^\star[t]$
  $\mathcal{P}_t \leftarrow$ SYNTHESISEPROC$(C, q, \mathcal{R}_t, \mathcal{F})$ ▷ Sec. 2.3
  $\mathcal{E}_t \leftarrow \{C, q, p\} \cup \mathcal{F}$
  $\Phi_t \leftarrow (\mathcal{R}_t, \mathcal{P}_t, \mathcal{E}_t, \mathcal{M})$
  $\hat{a}_t \leftarrow \mathrm{Delegate}(\Phi_t)$
  $(y_t, f_t) \leftarrow$ JUDGE$(C, q, p, \hat{a}_t)$
  if $y_t = 1$ **then**
    return $\mathrm{Finish}(\hat{a}_t)$ ▷ early termination
  else
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{(\mathcal{R}_t, \hat{a}_t, f_t)\}$ ▷ eliminate $\mathcal{R}_t$
    $\hat{a}^\star \leftarrow$ UPDATEBEST$(\hat{a}^\star, \hat{a}_t, f_t)$
  end if
end for
return $\mathrm{Finish}(\hat{a}^\star)$

agent for each chart question, (ii) *verify* the sub-agent's answer against the chart, and (iii) *terminate* as soon as the verification passes, so that easy questions consume a single sub-agent and harder questions a few.

Sub-agent description. To formally define we describe every sub-agent (including the orchestrator and the judge) with four fields,

$$\Phi = (\mathcal{R},\, \mathcal{P},\, \mathcal{E},\, \mathcal{M}), \tag{2}$$

namely the chart-reading **role** $\mathcal{R}$, the chart-reading **procedure** $\mathcal{P}$ the sub-agent is to follow, the **conditioning evidence** $\mathcal{E}$ it reads, and the backbone vision-language **model** $\mathcal{M}$ that drives it. The orchestrator concretises $\mathcal{R}$, $\mathcal{P}$, and $\mathcal{E}$ *dynamically, per question*; $\mathcal{M}$ is a system-level configuration that we hold fixed within each experiment so that the contribution of orchestration is isolated. In Sections 2.2–2.4, we describe how CHARTORCHESTRA generates and ranks candidate roles, synthesizes per-role procedures conditioned on the query, and performs termination and recovery via chart-grounded judges.

2.2 Role Over-Generation and Ranking

Chart questions vary widely in the reasoning they demand; some need trend extraction, others exact value lookup, and others compound numeric computation. The orchestrator therefore prompts the backbone to over-generate a pool of K candidate chart-reading roles as over-generation maintains better diversity when one role fails (Hosain et al., 2025).

$$S = \{s_1, s_2, \ldots, s_K\} \tag{3}$$

under three prompt-level constraints: distinct role names (no two roles share a name), explicit chart-grounding in each role's description, and the inclusion of an *answerability-checker* role for ambiguous or unanswerable queries. Ranking is implemented within the same prompt: the backbone is asked to emit S together with a relevance-to-q score in $[0, 1]$ for each role and a one-line justification, and we sort by score to obtain $S^\star = (s^{(1)}, \ldots, s^{(K)})$. When two roles receive the same score, we break ties with two deterministic rules applied in order: (i) prefer roles that rely on exact numeric reading over roles that rely on qualitative reasoning, and (ii) prefer roles whose justification cites a specific chart element (axis, legend, series). Both the score and the justification are part of the prompted output; no external scoring model is trained. The top-ranked role becomes $\mathcal{R}_1$ for the first sub-agent; if the judge later rejects, the next-ranked role becomes $\mathcal{R}_2$, and so on.

2.3 Per-Task Procedure Synthesis

Once $\mathcal{R}_t$ is fixed, the orchestrator *writes* the chart-reading procedure $\mathcal{P}_t$ that the sub-agent will follow. $\mathcal{P}_t$ is a short, ordered programme of chart-reading steps tailored both to the chart and to the question: which axes and legend entries to identify, which series to compare, how to read the ticks of interest (with thresholds for visual interpolation), and when to escalate from visual reading to a code interpreter for arithmetic. The synthesised $\mathcal{P}_t$ becomes the sub-agent's *tool* for that question.

This is the central novelty over prior sub-agent-as-tools systems. Catalog-routing approaches (Hosain et al., 2025) parameterize Build by *selecting* a pre-defined sub-agent whose "how to do its job" is hand-wired in advance, and isolated-thread approaches (Schroeder et al., 2025a) parameterize it merely by handing the helper a clean copy of the working context. Neither path produces a per-question program of how to read this specific chart.

In CHARTORCHESTRA, $\mathcal{P}_t$ is generated at delegation time and is allowed to differ between questions and between successive sub-agents on the same question, which is what enables on-demand specialization.

2.4 Role-Elimination and Recovery with a Chart-Grounded Judge

After the sub-agent returns a candidate $\hat{a}_t$, a chart-grounded *judge* sub-agent verifies the answer. The judge is itself an instance of Φ with a fixed role: it independently re-derives an answer from the chart input (C, q, p) and compares the derivation against $\hat{a}_t$. It returns a binary score $y_t \in \{0, 1\}$ together with a concise critique f_t that flags numeric errors, mislabelled reference points, or logical inconsistencies. Re-derivation from the chart is what makes the judge perceptually grounded rather than a text-pattern matcher;

Role elimination, not iterative refinement. If the judge accepts ($y_t = 1$), the orchestrator emits $\mathrm{Finish}(\hat{a}_t)$ and terminates — yielding the early termination behaviour quantified in Sec. 3.5. If the judge rejects ($y_t = 0$), the orchestrator *eliminates* the failed role $\mathcal{R}_t$ from the pool, distils the failure into a compact lesson $(\mathcal{R}_t, \hat{a}_t, f_t)$, and appends it to a failure memory $\mathcal{F}$. The next sub-agent is then built with a *different* role (the next-ranked $\mathcal{R}_{t+1}$ from $S^\star$), a freshly synthesised procedure $\mathcal{P}_{t+1}$ that avoids the diagnosed mistake, and an enriched evidence $\mathcal{E}_{t+1} = \{C, q, p\} \cup \mathcal{F}$ that carries the prior critique.

The design decision to *eliminate* the failed role rather than ask the same agent to retry is what separates CHARTORCHESTRA from iterative-refinement frameworks such as Xolver (Hosain et al., 2025), where the same set of agents is run for multiple rounds conditioned on a shared memory. Iterative refinement keeps a confidently-wrong reasoning style in play; elimination forces the next attempt to use a structurally different reading procedure, which our ablations in Sec. 3.4 show is the better fit for chart reasoning. Algorithm 1 summarises the inference loop.

3 Experiments

3.1 Experimental Setup

Benchmarks. We evaluate CHARTORCHESTRA on four chart benchmarks that together cover scientific charts, real-world layouts, and synthetic figures. **CharXiv** (Wang et al., 2024) contains diverse charts collected from arXiv papers and is designed to test reasoning on complex visual elements. **ChartQAPro** (Masry et al., 2025) provides a comprehensive evaluation suite spanning MCQ, factoid, conversational, fact-checking, and hypothetical questions, with deliberately unanswerable items. **PlotQA** (Methani et al., 2020) probes precise value reading and logical reasoning on programmatically generated charts. **FigureQA** (Kahou et al., 2017) stresses relational reasoning over synthetic figures.

Baselines. We compare against three groups. (i) Proprietary VLMs — GPT-4o (OpenAI et al., 2024), Claude 3.5 Sonnet (Anthropic, 2024), Gemini 2.0 Flash (Georgiev et al., 2024), and GPT-4o-mini — under direct prompting. (ii) Open-source VLMs at 1B–11B scale (Intern-VL-2.5, Phi-3.5-Vision, LLaVA-Next-Mistral, Llama-3.2-Vision, GLM-4.5V), plus the chart-specialist fine-tuned models ChartGemma (Masry et al., 2024b) and TinyChart (Zhang et al., 2024). (iii) Reasoning frameworks executed on identical backbones: Chain-of-Thought (CoT) (Wei et al., 2022), Program-of-Thought (PoT) (Chen et al., 2022), Code-as-Thought (CaT) (Tang et al., 2025), and ReAct-style chart agents (Kaur et al., 2025).

Implementation. CHARTORCHESTRA is training-free. For each block of results in Table 1, we fix the backbone $\mathcal{M}$ to a single model and instantiate every sub-agent: orchestrator, executor, and judge — with that same backbone, to isolate the contribution of orchestration. Unless stated otherwise, the orchestrator over-generates $K = 3$ chart-reading roles per question, with an answerability-checker role always included. We set a tight budget of two full iterations per question, used only when the first iteration ends without judge acceptance. All prompts are zero-shot and listed in Appendix A.

3.2 Main Results

Table 1 reports the main results. CHARTORCHESTRA achieves the strongest overall average across all three backbones, consistently outperforming reasoning baselines on the most demanding benchmarks while remaining competitive on the simpler synthetic ones.

Overall gains over the strongest baseline. Relative to the strongest matched-backbone reasoning baseline, CHARTORCHESTRA improves the

Model	Approach	ChartQAPro Factoid	ChartQAPro MCQ	ChartQAPro Convers.	ChartQAPro FactChk.	ChartQAPro Hypoth.	ChartQAPro Avg.	CharXiv (Res.)	FigureQA	PlotQA	Avg
Proprietary Models (direct prompting)											
GPT-4o (OpenAI et al., 2024)	Direct	35.76	46.72	34.75	45.49	28.91	37.67	50.50	65.70	59.50	53.34
Gemini 2.0 Flash (Team et al., 2023)	Direct	43.43	60.28	40.25	67.62	24.47	46.85	50.30	54.90	60.40	53.11
Claude 3.5 Sonnet (Anthropic, 2024)	Direct	38.84	51.40	44.53	55.60	45.48	43.58	60.20	43.30	49.20	49.07
Open-Source Models (direct prompting)											
Intern-VL2.5-1B (Chen et al., 2025)	Direct	09.15	07.00	06.20	16.63	08.17	09.33	19.00	59.40	61.70	37.36
Intern-VL2.5-2B (Chen et al., 2025)	Direct	13.86	10.74	14.02	45.90	18.92	17.81	21.30	64.30	71.10	43.63
Intern-VL2.5-8B (Chen et al., 2025)	Direct	35.21	25.70	32.26	53.27	29.61	35.67	32.90	69.60	74.70	53.22
Phi 3.5-Vision-4B (Abdin et al., 2024)	Direct	17.48	30.37	28.54	41.99	37.27	24.73	32.70	64.90	48.60	42.73
LLaVA-Next-Mistral-7B (Li et al., 2024)	Direct	15.35	35.98	21.09	41.80	17.79	21.97	–	58.10	41.70	40.59
Llama 3.2-Vision-11B (Grattafiori et al., 2024)	Direct	12.34	02.33	00.19	27.18	10.93	11.09	31.20	00.00	00.00	10.57
Chart-Specific Models (fine-tuned)											
ChartGemma-3B (Masry et al., 2024b)	finetuned	06.86	00.00	16.00	01.22	06.53	06.84	12.50	38.90	35.60	23.46
TinyChart-3B (Zhang et al., 2024)	finetuned	08.52	07.00	17.46	33.19	16.06	13.25	08.30	48.00	55.30	31.21
Agentic Reasoning Frameworks (matched backbone)											
GPT-4o-mini (OpenAI et al., 2024)	CoT	42.00	59.80	41.50	56.60	42.90	48.85	18.20	57.90	27.90	38.21
	PoT	43.50	44.90	41.20	37.70	51.00	42.47	13.70	59.00	34.20	37.34
	CaT (Tang et al., 2025)	38.50	56.50	38.10	35.70	52.00	42.59	40.70	**60.20**	26.40	42.47
	ChartAgent (Kaur et al., 2025)	28.00	60.30	**46.30**	56.60	30.60	46.59	37.90	45.20	39.50	42.30
	ChartOrchestra	**52.00**	**71.00**	40.50	**65.60**	**66.30**	**56.88**	**45.50**	58.80	**51.50**	**53.17**
Gemini 2.0 Flash (Team et al., 2023)	CoT	51.51	69.15	43.84	67.62	39.89	55.43	53.50	72.40	67.20	62.13
	PoT	51.18	57.00	46.34	56.81	44.86	51.64	55.90	66.70	57.40	57.91
	CaT (Tang et al., 2025)	45.40	69.60	38.70	64.80	59.20	54.01	54.40	63.60	58.00	57.50
	ChartAgent (Kaur et al., 2025)	**66.60**	56.50	**50.20**	63.10	55.10	57.94	49.30	69.00	75.90	63.04
	ChartOrchestra	57.50	**78.00**	48.90	**73.00**	**67.30**	**63.55**	**69.60**	**78.70**	**77.80**	**72.41**
GLM-4.5V (Zeng et al., 2025)	CoT	36.00	66.40	43.40	60.70	46.90	50.90	54.90	85.90	54.80	61.63
	PoT	34.70	48.60	48.60	53.70	48.00	47.10	53.60	86.70	61.60	62.25
	CaT (Tang et al., 2025)	45.50	60.70	30.10	55.30	44.90	46.24	55.80	47.30	42.20	47.89
	ChartAgent (Kaur et al., 2025)	31.10	**77.10**	44.70	68.40	**64.30**	55.87	54.80	84.80	**79.70**	68.79
	ChartOrchestra	**52.25**	75.20	**63.30**	**70.10**	63.30	**65.17**	**58.00**	**87.90**	75.80	**71.72**

Table 1: Main results across four chart-reasoning benchmarks. Within each backbone block, the best result is **bold** and the second-best is underlined; the overall column-best is shown in blue. "–" denotes unreported results.

overall average by $\Delta = +10.70$ points for GPT-4o-mini (53.17 vs. 42.47 CaT), $\Delta = +9.37$ for Gemini 2.0 Flash (72.41 vs. 63.04 ChartAgent), and $\Delta = +0.92$ for GLM-4.5V (71.72 vs. 68.79 ChartAgent). Aggregated across all three backbones, ChartOrchestra is the only system that holds the top average on every benchmark family.

Largest gains on the hardest chart questions. The improvements are most pronounced on the categories that demand multi-step reasoning grounded in chart evidence. On ChartQAPro *Hypothetical* questions — the hardest split, where models must reason about counterfactual or implication-style queries — ChartOrchestra with Gemini 2.0 Flash lifts accuracy from 39.89 (CoT) to 67.30 ($\Delta = +27.41$), and with GPT-4o-mini from 42.90 to 66.30 ($\Delta = +23.40$). On *Fact-Checking*, accuracy rises from 67.62 to 73.00 ($\Delta = +5.38$); on *MCQ*, from 69.15 to 78.00 ($\Delta = +8.85$).

CharXiv: a scientific-chart stress test. On the realistic scientific charts of CharXiv, where chart layouts and questions are drawn directly from research papers, ChartOrchestra with Gemini 2.0 Flash improves from 55.90 (PoT) to 69.60 ($\Delta = +13.70$), and with GPT-4o-mini from 40.70 (CaT) to 45.50 ($\Delta = +4.80$). These gains confirm that on-demand chart-reading specialisation transfers to charts that fixed pipelines were never engineered for.

Where ChartOrchestra does not lead, and why. ChartOrchestra is not uniformly best on every column. Under Gemini 2.0 Flash, ChartAgent edges ChartOrchestra on the *Factoid* split (66.60 vs. 57.50) and on the *Conversational* split (50.20 vs. 48.90). We attribute these two cases to category-specific structure. *Factoid* questions typically resolve in a single chart lookup; the orchestrator's over-generation/judge loop pays a small overhead without buying additional reasoning. *Conversational* questions inherit context from earlier turns of a dialogue, so the role required at turn t is heavily predicted by turn $t-1$; this collapses the role variance that the orchestrator exists to handle, and ChartAgent's chained reasoning already exploits the same signal. ChartOrchestra still holds the second-best score in both columns, and the resulting *ChartQAPro (Avg.)* number remains best (63.55). The pattern is consistent with the prediction in Sec. 3.4 that ChartOrchestra's advantage scales with question difficulty and role variance.

Synthetic benchmarks. On PlotQA and FigureQA, where direct prompting of strong open-source models is already near-saturated, ChartOrchestra still attains the best score within

its block for FigureQA (78.70 with Gemini) and PlotQA (77.80 with Gemini). The smaller absolute gap in these synthetic suites reflects the remaining headroom.

3.3 Advantage 1: Curated Context Handoff Matters

A central design choice in CHARTORCHESTRA is that the orchestrator explicitly curates the conditioning evidence $\mathcal{E}$ handed to each new sub-agent after a failure. To isolate this effect, we vary only $\mathcal{E}$ while keeping the role $\mathcal{R}$, procedure $\mathcal{P}$, and backbone $\mathcal{M}$ identical. We compare three regimes used by prior systems and by ours on a 200-question subset of CharXiv.

Context regime	Easy	Medium	Hard	Avg.
No-Context (instruction only)	78.50	60.10	38.20	58.93
Full-Context (all prior steps)	81.40	58.70	35.40	58.50
Ours (curated failure trace)	**86.20**	**67.50**	**44.80**	**66.17**

Table 2: Context-handoff ablation on a 200-question CharXiv subset (Gemini 2.0 Flash backbone) sampled from the same 1,000-question pool used in Table 1, stratified by question difficulty.

Table 2 confirms two failure modes of the extremes. *No-Context* delegation strips the new sub-agent of any information about why the previous attempt failed, so the orchestrator often instantiates a different role that repeats the same mistake. *Full-Context* delegation, the implicit choice of many prior chart agents that share the full reasoning history, suffers from context rot: irrelevant intermediate steps distract the new sub-agent and *hurt* accuracy on hard questions (-2.8 points). CHARTORCHESTRA's curated handoff — only the failed role, the candidate, and the judge critique — outperforms both, by $+7.24$ and $+7.67$ points on average, respectively.

3.4 Advantage 2: On-Demand Specialisation vs. Static Roles and Procedures

Table 3 probes the consequence of freezing different fields of the sub-agent description. Three observations follow.

No single fixed configuration wins across all. On Fact-Checking, fixing the role (best 59.00) beats fixing the procedure (best 57.00), suggesting the bottleneck is *which* chart-reading procedure to apply. On MCQ, the ordering reverses: fixing the procedure (best 60.70) beats fixing the role (best 51.40), suggesting the bottleneck is *which expert role* to assign. This non-monotonicity is the symptom that motivates per-question concretisation: the right axis of specialization depends on the question.

Task	Sub-Task	Role (Fixed)	Proc. (Fixed)	Both (Fixed)	CHARTORCHESTRA
Fact-Checking	Evidence Extractor	57.80	54.90	57.00	**73.00**
	Claim Verifier	59.00	57.00	55.30	
	Trend Comparator	59.00	55.70	55.70	
Factoid	Evidence Extractor	35.60	35.50	36.10	**57.50**
	Claim Verifier	35.70	35.70	36.40	
	Trend Comparator	36.80	37.00	37.20	
Conversational	Evidence Extractor	49.80	48.20	47.30	**48.90**
	Claim Verifier	49.80	45.70	46.60	
	Trend Comparator	50.80	48.60	47.30	
MCQ	Evidence Extractor	51.40	57.50	56.10	**78.00**
	Claim Verifier	50.90	60.70	61.20	
	Trend Comparator	49.50	57.00	55.10	
Hypothetical	Evidence Extractor	53.10	45.90	44.90	**67.30**
	Claim Verifier	56.10	39.80	41.80	
	Trend Comparator	55.10	46.90	49.00	

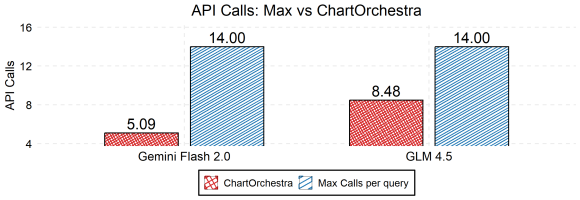
Table 3: Fixed role/procedure vs. on-demand specialization on ChartQAPro (Gemini 2.0 Flash). "Role (Fixed)" freezes $\mathcal{R}$; "Procedure (Fixed)" freezes $\mathcal{P}$; "Both Fixed" freezes both; ours concretizes both per question.

On-demand specialization is necessary, not optional. Even the best fixed configuration trails CHARTORCHESTRA by large margins: $+14.0$ on Fact-Checking, $+20.3$ on Factoid, $+16.8$ on MCQ, $+11.2$ on Hypothetical. The single exception is Conversational, where CHARTORCHESTRA edges out the best fixed setting by only -1.9 points; we conjecture that conversational follow-ups have low role variance because they typically inherit the previous turn's reasoning role.

Synergy between role and procedure. Freezing both $\mathcal{R}$ and $\mathcal{P}$ does *not* compound the cost of freezing each individually; on MCQ, "Both Fixed" (61.20) sits between the two singleton variants. This indicates that the orchestrator's value lies less in any one field than in its ability to jointly co-adapt role, procedure, and evidence per question.

3.5 Advantage 3: Framework-Agnostic Backbone and Cost Savings

Because the orchestrator only issues $\{\text{Build}, \text{Delegate}, \text{Finish}\}$ and does not encode backbone-specific assumptions, the orchestration policy is decoupled from the executor backbone. Table 1 demonstrates the resulting pluggability at the *framework level*: replacing the backbone $\mathcal{M}$ across three very different models (a proprietary mini model, a proprietary mid-tier model, and an open-source model) preserves CHARTORCHESTRA's lead within each block. We deliberately use the same backbone for every sub-agent of a given run to isolate the contribution of orchestration;



Model	Max Calls / query	Max Possible Calls	ChartOrchestra	Cost Savings (%)
Gemini Flash 2.0	14	14000	**5091**	**63.6**
GLM 4.5	14	2800	**1696**	**39.4**

Figure 2: API call reduction on CharXiv under Gemini 2.0 Flash, and GLM-4.5V. Because CHARTORCHESTRA can terminate via Finish as soon as the judge accepts, average API consumption stays well below the worst-case envelope, yielding 39.4%–63.6% savings.

per-sub-agent mixed-backbone routing, which would enable cost-aware Pareto policies for chart reasoning (Ruan et al., 2026), is supported by the four-field sub-agent description in Sec. 2.1 and left to future work.

Worst-case envelope. In the worst case, the orchestrator instantiates K chart sub-agents and one judge per attempt, so the first iteration costs at most $2K + 1$ API calls (one planner call, plus K sub-agent + judge pairs). With a second iteration applied only to remaining failures, the envelope is $2(2K + 1) = 14$ API calls per question for $K = 3$.

Empirical savings. Because CHARTORCHESTRA invokes Finish as soon as the judge accepts, most questions consume far fewer than the envelope. In Figure 2, on 1,000 CharXiv questions with Gemini 2.0 Flash, CHARTORCHESTRA uses 5,091 of a worst-case 14,000 API calls (63.6% saved). On 200 questions with GLM-4.5V, it uses 1,696 of 2,800 (39.4% saved).

3.6 How Well Does the Judge Align with Human Evaluation?

A complementary validity check asks whether any backbone can serve as a reliable judge in place of a human annotator. We evaluate three representative models as drop-in judge sub-agents on a held-out 150-question ChartQAPro subset, scoring each by how often its accept/reject verdict matches the human gold label. As reported in Appendix Table 11, performance varies substantially across backbones: **4o Mini** collapses to only 32%, rendering it unreliable for chart-grounded verification; **GLM-4.5V** reaches 64%; and **Gemini 2.0 Flash** attains the strongest alignment at 78%. The pooled

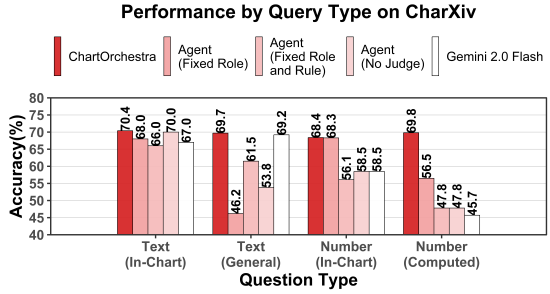


Figure 3: Accuracy by query type on CharXiv. CHARTORCHESTRA dominates on questions that require numeric computation from the chart, where on-demand rule synthesis provides the largest lift.

accuracy of 68% underscores that judge backbone choice is non-trivial, motivating our default use of Gemini 2.0 Flash throughout the main experiments.

3.7 The Answerability-Checker Role

ChartQAPro deliberately includes unanswerable items whose ground truth is "unanswerable". We force-include an *answerability-checker* role in S for every question (Sec. 2.2); it does not necessarily rank first but is always reachable by the elimination loop. On the unanswerable subset of ChartQAPro with Gemini 2.0 Flash, CHARTORCHESTRA abstains on 82.4% of items (true-negative rate) and falsely abstains on 4.9% of *answerable* items (false-positive rate). On the same backbone, CoT and ChartAgent abstain on 14.1% and 21.8% of unanswerable items, respectively, indicating that an explicit always-available answerability-checker role contributes most of our gain on this slice.

3.8 Where the Gains Come From: Query-Type Breakdown

In figure 3, on CharXiv accuracy by query type, CHARTORCHESTRA lifts Gemini 2.0 Flash from 45.7% to 69.8% for computation of values from charts. For *text-in-chart* and *number-found-in-chart*, the orchestrator helps, but the gap to static-role agents is smaller, because the right chart-reading procedure is closer to a fixed template. For *text-in-general* (open-ended description tasks), the four configurations cluster within 0.5 points: when the question barely depends on the chart, sophisticated orchestration buys little.

4 Ablation Study

We scrutinize component-wise performance of CHARTORCHESTRA as shown in Table 4. We ob-

serve a significant accuracy drop with the removal of per-task procedure synthesis (10.10% decrease). Elimination of the **Judge Agent** results in an 8.10% drop in accuracy, which is an indication of maintaining answer quality. The fixed-roles-only configuration still outperforms the simple baseline by 3.50%. The experiments are done using Gemini 2.0 Flash on the CharXiv dataset.

Configuration	Accuracy (%)	Δ
CHARTORCHESTRA (All Components)	69.60	–
w/o Per-Task Procedure Synthesis	59.50	-10.10
w/o Judge	61.50	-8.10
Static Roles Only	64.00	-5.60
Monolithic LLM (Baseline)	60.50	-9.10

Table 4: Effect of removing individual components from the CHARTORCHESTRA.

Tables 5 analyze how the number of reasoning roles affects performance across different task types and benchmarks. On most task categories, increasing the number of roles from 3 to 5 consistently improves performance, indicating that moderately richer role diversity is beneficial for handling the multi-step and ambiguous nature of ChartQAPro questions. On comparatively simpler benchmarks such as FigureQA, performance steadily degrades as more roles are added. FigureQA decreases from 78.7 to 78.0. For PlotQA, performance remains nearly unchanged, varying slightly from 77.8 to 77.5. CharXiv exhibits the largest degradation, falling from 69.6 to 67.4, indicating a strong sensitivity to unnecessary reasoning complexity.

Datasets	3 Roles	5 Roles	10 Roles
PlotQA	77.80	77.50	77.50
FigureQA	78.70	78.50	78.00
CharXiv	69.60	67.10	67.40
ChartQAPro	64.94	66.70	67.06

Table 5: Performance Comparison of CHARTORCHESTRA with Different Number of Roles using Gemini 2.0 Flash as backbone model on different datasets. Details of ChartQAPro results are added in the Appendix.

5 Related Work

In Chain-of-Thought (CoT) (Wei et al., 2022), language models are guided through step-by-step reasoning, yielding strong gains across arithmetic and commonsense tasks. This idea is extended by Program-of-Thought, where executable code is generated in place of natural-language steps and computation is offloaded to an external interpreter for improved numerical accuracy. Reliability is further improved by Self-Consistency (Wang et al., 2022), in which multiple reasoning paths are sampled and their answers are aggregated.

Early datasets such as FigureQA (Kahou et al., 2017) are built around simple number extraction from synthetic charts. More realistic challenges are introduced by PlotQA and ChartQA (Wei et al., 2024), where models are required to interpret real-world chart structure. The bar is raised further by CharXiv (Wang et al., 2024), whose charts are drawn directly from scientific papers, and by ChartQAPro, which is characterized by adversarial questions and conversational follow-ups. InfoChartQA (Xie et al., 2025) is designed to target complex infographics with icons and irregular layouts.

Most specialized chart models are built following a pre-train-then-fine-tune pipeline. UniChart (Masry et al., 2023) and ChartAssistant (Meng et al., 2024) are pre-trained on large chart–table pairs before being fine-tuned on diverse tasks. In StructChart (Xia et al., 2023) and OneChart (Chen et al., 2024), chart components—axes, legends, and data series—are explicitly extracted before any reasoning is performed. Instruction tuning is advanced by MMC (Liu et al., 2024a) and ChartInstruct (Masry et al., 2024a) to handle a wide range of user commands, and a self-training approach is explored in EvoChart (Huang et al., 2025a) to better adapt to real-world chart variety.

In agentic frameworks such as ChartAgent (Kaur et al., 2025) and ChartThinker (Liu et al., 2024b), Most systems are characterized by fragile pipelines and a lack of unified mechanisms for perception, reasoning, and verification.

6 Conclusion

We propose CHARTORCHESTRA, an adaptive multi-agent framework for chart QA that generates and ranks candidate roles per chart-question pair, synthesizes a tailored reading procedure, and produces a candidate answer. A *Chart-Grounded Judge* independently verifies each answer and provides feedback; upon rejection, CHARTORCHESTRA eliminates the failed role and conditions, forcing the next sub-agent to approach the problem differently. Computational cost is controlled naturally by avoiding unnecessary API calls through the Build–Delegate–Finish action space that drives the orchestration loop.

Limitations

A key limitation of CHARTORCHESTRA is its focus exclusively on the chart domain; the framework has not been evaluated on other visual question answering tasks such as natural scene understanding, document analysis, or scientific figure interpretation. Extending CHARTORCHESTRA to broader visual domains would require rethinking role generation and judge grounding beyond chart-specific priors. Additionally, an ensemble of judges could strengthen verification robustness, particularly on ambiguous or visually dense charts.

Ethics

This paper includes a small-scale human evaluation study. Prior to participation, all individuals were clearly briefed on the voluntary nature of their involvement, the estimated time commitment, and the criteria used for assessment. Participants were sourced through the authors' academic and research connections.

Use of AI Assistants

The authors used AI writing assistance tools (e.g., ChatGPT, Claude) solely for improving the fluency and clarity of the writing. All scientific content, ideas, experiments, and conclusions are entirely the work of the human authors. The authors take full responsibility for the integrity and accuracy of the content presented in this paper.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Anthropic. 2024. Introducing the next generation of claude.

Jinyue Chen, Lingyu Kong, Haoran Wei, Chenglong Liu, Zheng Ge, Liang Zhao, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. 2024. Onechart: Purify the chart structural extraction via one auxiliary token. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 147–155.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, and 23 others. 2025. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *Preprint*, arXiv:2412.05271.

Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, and 1117 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, and 1 others. 2023. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Md Tanzib Hosain, Salman Rahman, Md Kishor Morol, and Md Rizwan Parvez. 2025. Xolver: Multi-agent reasoning with holistic experience learning just like an olympiad team. *arXiv preprint arXiv:2506.14234*.

Muye Huang, Han Lai, Xinyu Zhang, Wenjun Wu, Jie Ma, Lingling Zhang, and Jun Liu. 2025a. 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.

Muye Huang, Lingling Zhang, Han Lai, Wenjun Wu, Xinyu Zhang, and Jun Liu. 2025b. Vprochart: Answering chart question through visual perception alignment agent and programmatic solution reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3689–3696.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.

Rachneet Kaur, Nishan Srishankar, Zhen Zeng, Sumitra Ganesh, and Manuela Veloso. 2025. Chartagent: A multimodal agent for visually grounded reasoning in complex chart question answering. *arXiv preprint arXiv:2510.04514*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. 2024a. Mmc: Advancing multimodal chart understanding with large-scale instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1287–1310.

Mengsha Liu, Daoyuan Chen, Yaliang Li, Guian Fang, and Ying Shen. 2024b. Chartthinker: A contextual chain-of-thought approach to optimized chart summarization. *arXiv preprint arXiv:2403.11236*.

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, and 1 others. 2025. Chartqapro: A more diverse and challenging benchmark for chart question answering. *arXiv preprint arXiv:2504.05506*.

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. *arXiv preprint arXiv:2305.14761*.

Ahmed Masry, Mehrad Shahmohammadi, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2024a. Chartinstruct: Instruction tuning for chart comprehension and reasoning. *arXiv preprint arXiv:2403.09028*.

Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. 2024b. Chartgemma: Visual instruction-tuning for chart reasoning in the wild. *Preprint*, arXiv:2407.04172.

Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. Chartassisstant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. *arXiv preprint arXiv:2401.02384*.

Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the ieee/cvf winter conference on applications of computer vision*, pages 1527–1536.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jianhao Ruan, Zhihao Xu, Yiran Peng, Fashen Ren, Zhaoyang Yu, Xinbing Liang, Jinyu Xiang, Yongru Chen, Bang Liu, Chenglin Wu, and 1 others. 2026. Aorchestra: Automating sub-agent creation for agentic orchestration. *arXiv preprint arXiv:2602.03786*.

Philip Schroeder, Nathaniel W Morgan, Hongyin Luo, and James Glass. 2025a. Thread: Thinking deeper with recursive spawning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8418–8442.

Philip Schroeder, Nathaniel W. Morgan, Hongyin Luo, and James R. Glass. 2025b. THREAD: Thinking deeper with recursive spawning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8418–8442, Albuquerque, New Mexico. Association for Computational Linguistics.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. 2025. Scaling long-horizon llm agent via context-folding. *arXiv preprint arXiv:2510.11967*.

Bohao Tang, Yan Ma, Fei Zhang, Jiadi Su, Ethan Chern, Zhulin Hu, Zhixin Wang, Pengfei Liu, and Ya Zhang. 2025. Visual programmability: A guide for code-as-thought in chart understanding. *arXiv preprint arXiv:2509.09286*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

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, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jingxuan Wei, Nan Xu, Guiyong Chang, Yin Luo, Bi-Hui Yu, and Ruifeng Guo. 2024. mchartqa: A universal benchmark for multimodal chart question answer based on vision-language alignment and reasoning. *arXiv preprint arXiv:2404.01548*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*.

Renqiu Xia, Bo Zhang, Haoyang Peng, Hancheng Ye, Xiangchao Yan, Peng Ye, Botian Shi, Yu Qiao, and Junchi Yan. 2023. Structchart: Perception, structuring, reasoning for visual chart understanding. *arXiv preprint arXiv:2309.11268*.

Tianchi Xie, Minzhi Lin, Mengchen Liu, Yilin Ye, Changjian Chen, and Shixia Liu. 2025. Infochartqa: a benchmark for multimodal question answering on infographic charts. *arXiv preprint arXiv:2505.19028*.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, and 1 others. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.

Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. 2024. Tinychart: Efficient chart understanding with visual token merging and program-of-thoughts learning. *Preprint*, arXiv:2404.16635.

Appendix

A Prompts

This section provides the prompts used by the sub-agents for the Planning, Adaptive Dynamic, and Evaluation roles. These prompts are significant to ensure reproducibility of the framework.

A.1 Role-over generation and Ranking Prompt

Prompt for ROLE GENERATOR

You are a **Planner** whose job is to generate and rank candidate specialist **roles** that could solve a chart-based conversational question sequence.

Important context (Few-shot demonstrations, examples of what a "role" looks like; not a closed list):
`example_roles`

Goal:
- Using the few-shot demonstrations above as examples, generate exactly `NUM_CANDIDATES` candidate roles that could help solve the provided **question sequence** for the given chart image.
- Candidate roles **may** be reused from the examples **or** be newly invented, if that better fits the conversational context.
- Provide short descriptions tailored to this conversation and rank them by how likely each role is to produce correct answers across all questions (`1 = best`).

Constraints & formatting rules:
1. Output **only** a JSON array of objects (no extra text).
2. The array must be sorted in ascending order by `"rank"` (`1..NUM_CANDIDATES`, `1 = highest priority`).
3. Each object must contain:
   - `"rank"`: integer (`1..NUM_CANDIDATES`)
   - `"role_name"`: string ($\leq$ 5 words)
   - `"role_description"`: string (brief description of how this role would approach the question sequence)
4. No duplicate `"role_name"`. Prefer diversity.
5. At least 40% of roles should be **novel** (not verbatim from catalog).
6. If the chart lacks required info, include a role like `"Conversational Context Handler"` in the top 3.
7. Break ties by preferring roles that can maintain better conversation context.
8. Do not include internal reasoning.

Conversational Questions:
`questions`

Additional Textual Context (Paragraph):
`paragraph`

Chart image is provided.

A.2 Per-Task Procedure Synthesis Prompt

Prompt for DYNAMIC PROCEDURE SYNTHESIS

You are acting as: `role`

We already attempted once and the judge marked it incorrect. You now have the chart image again, plus prior context to adapt and fix the mistake.

This is a multi-question conversation. You need to answer question `current_q_index` of `total_questions`:
Current Question: `"current_question"`

Previous Questions and Answers:
`previous_qa`

Context:
- Previous Candidate Answer: `prev_answer`
- Judge Feedback: `judge_feedback`

Additional Textual Context:
`paragraph`

Phase 1: Generate Rules
Before solving, generate a *customized step-by-step rule set* for how to approach this conversational question. The rules must include at least:

- How to diagnose the prior mistake based on Judge Feedback

- How to maintain context from previous questions and answers
- How to extract relevant chart features for this specific question
- How to use information from previous answers if needed
- How to reason about the current question in the conversational context

Write these rules clearly and concisely (like a mini checklist).

Phase 2: Apply Rules
Using the rules you just generated, attempt to answer the current question again:

1. Diagnose prior failure (based on Judge Feedback)
2. Review the conversation history and previous answers
3. Extract relevant evidence from the chart for this specific question
4. Perform any necessary computation/logic
5. Provide the final answer for this question

Final Output:
Return ONLY the answer to the current question wrapped as `\boxed{. . . }`.

A.3 Judge Sub-Agent

Prompt for JUDGE SUB-AGENT

You are the **Judge** for a chart-based conversational question.
Your job is to decide if the candidate's answer is correct using the chart image and provided context.

Current Question: `"current_question"`

Conversation Context:
`previous_qa`

Additional Textual Context:
`paragraph`

Candidate's Answer:
`\boxedcandidate_answer`

Judging Rules:

1. Score = 1 only if the answer is correct based on the chart data, conversation context, and textual context.
2. Consider the conversational flow — the answer should make sense in the context of previous questions.
3. For numerical answers, allow reasonable margins of error (e.g., rounding differences).
4. Otherwise Score = 0.
5. Provide short, actionable feedback on what was wrong or missing.

Output format (exactly):
```
Score: <0 or 1>
Feedback: <short actionable feedback>
```

B Qualitative Examples

This section demonstrates few qualitative examples and detailed breakdown on how CHARTORCHESTRA can approach and adaptively solve chart reasoning questions.

CHART REASONING (MCQ)

Query: "If California's home price volatility decreases by 10% annually, what will it be after 2 years?"
Options:

a) 79.38

b) 81.90

c) 85.32

d) 88.20

Ground Truth: A (79.38)

ROLE GENERATOR SUB-AGENT EXECUTION

Input: Chart image + Question
Task: Generate and rank 10 candidate specialist roles for fact verification
Ranked Role Catalog:

1. **Multi-step Calculator** (Rank: 1) - Handles compound percentage calculations
2. **Data Extractor** (Rank: 2) - Extracts precise numerical values from charts
3. **Answerability Checker** (Rank: 3) - Verifies information availability
4. Percentage Change Calculator (Rank: 4)
5. Value Locator (Rank: 5)
6. Table Reader (Rank: 6)
7. Numerical Reasoner (Rank: 7)
8. Option Evaluator (Rank: 8)
9. Contextual Interpreter (Rank: 9)
10. Chart Type Identifier (Rank: 10)

Top 3 Selected Roles:

1. Multi-step Calculator
2. Data Extractor
3. Answerability Checker

DYNAMIC PROCEDURE SYNTHESIS: MULTI-STEP CALCULATOR (ROLE 1)

Role Specification: Expert in multi-step mathematical calculations involving percentage changes over time

Phase 1: Rule Generation

1. **Diagnose prior failure**: Fresh start (no prior attempts)
2. **Extract chart features**: Locate California row and "Home price volatility" column
3. **Identify numeric values**: Extract initial volatility (98), decrease rate (10%), time period (2 years)
4. **Compute compound decrease**: $V_n = V_0 \times (1 - r)^n$
5. **Handle ambiguity**: Mark unanswerable if data missing

Phase 2: Rule Application

$$\begin{aligned} \text{Initial volatility } V_0 &= 98 \\ \text{Annual decrease rate } r &= 0.10 \\ \text{Years } n &= 2 \\ V_2 &= 98 \times (1 - 0.10)^2 \\ &= 98 \times (0.90)^2 \\ &= 98 \times 0.81 \\ &= 79.38 \end{aligned}$$

Option Matching: 79.38 corresponds to option **a**
Final Answer: $\boxed{a}$

JUDGE SUB-AGENT EVALUATION

Evaluation Criteria:

- Correct data extraction: California volatility = 98 ✓
- Proper mathematical reasoning: Compound decrease calculation ✓
- Accurate computation: $98 \times 0.81 = 79.38$ ✓
- Correct option selection: Matches option a ✓

Step-by-Step Verification:

$$\begin{aligned} \text{Year 1}: &\quad 98 \times 0.90 = 88.20 \\ \text{Year 2}: &\quad 88.20 \times 0.90 = 79.38 \end{aligned}$$

Judge Score: 1/1
Feedback: "Calculation correctly applies compound percentage decrease over two years."

Iteration 1 Results

Final Prediction: a (79.38)
Ground Truth: A (79.38)
Correct: True
Stopped After: Iteration 1
Roles Attempted: Multi-step Calculator
Judge Score: 1.0
Decision: ✓Success on first attempt - No further iterations needed

Chart Reasoning (MCQ)

Query: "What percentage of countries with blasphemy laws are in the Middle East-North Africa region?"
Options:

a) 20%

b) 22%

c) 23%

d) 25%

Ground Truth: B (22%)
Data Source: World map visualization showing countries with blasphemy laws by region.

Role Generator Sub-Agent Execution

Input: Chart image + Question
Task: Generate and rank 10 candidate specialist roles for percentage calculation
Ranked Role Catalog:

1. **Proportion & Percentage Calculator** (Rank: 1) - Handles ratio and percentage calculations
2. **Data Extractor** (Rank: 2) - Extracts numerical values from chart regions
3. **Regional Data Aggregator** (Rank: 3) - Sums data across geographical regions
4. Total Count Determiner (Rank: 4)
5. Question Deconstructor (Rank: 5)
6. Option Validator (Rank: 6)
7. Chart Title Interpreter (Rank: 7)
8. Geographic Region Identifier (Rank: 8)
9. Contextual Information Analyzer (Rank: 9)
10. Visual Element Locator (Rank: 10)

Top 3 Selected Roles:

1. Proportion & Percentage Calculator
2. Data Extractor
3. Regional Data Aggregator

ITERATION 1: PROPORTION & PERCENTAGE CALCULATOR Role Specification: Expert in calculating proportions and percentages from multi-region data

Phase 1: Rule Generation

1. **Diagnose prior failure**: Fresh start (no prior attempts)
2. **Extract chart features**: Identify all regional callout bubbles with country counts
3. **Identify numeric values**: Extract country counts for each region, especially the Middle East-North Africa
4. **Compute global percentage**: Calculate MENA count as percentage of worldwide total
5. **Handle ambiguity**: Verify all regions are accounted for in the global total

Phase 2: Rule Application

$$\begin{aligned}
\text{Europe countries} &= 14 \\
\text{Americas countries} &= 12 \\
\text{Middle East-N. Africa countries} &= 18 \\
\text{Asia-Pacific countries} &= 17 \\
\text{Sub-Saharan Africa countries} &= 18 \\
\text{Global total} &= 14 + 12 + \\
&\quad 18 + 17 + 18 \\
&= 79 \\
\text{MENA percentage} &= \frac{18}{79} \times 100 \\
&= 22.78\% \\
\text{Rounded value} &\approx 23\%
\end{aligned}$$

Option Matching: 23% corresponds to option **c**

Final Answer: $\boxed{c}$

JUDGE SUB-AGENT EVALUATION - ITERATION 1 Evaluation Criteria:

- Correct data extraction: All regional counts accurately identified ✓
- Proper summation: Global total correctly calculated as 79 ✓
- Accurate percentage calculation: $\frac{18}{79} \times 100 = 22.78\%$ ✓
- Appropriate rounding: 22.78% rounds to 23% ✓

Step-by-Step Verification:

$$\begin{aligned} \text{Global Total} &: 14 + 12 + 18 + 17 + 18 = 79 \\ \text{MENA Percentage} &: \frac{18}{79} \times 100 = 22.78\% \\ \text{Rounding} &: 22.78\% \rightarrow 23\% \end{aligned}$$

Judge Score: 0
Feedback: "Calculation correctly computes MENA's share of global blasphemy law countries"

Iteration 1 Results:

Final Prediction:	c (23%)
Ground Truth:	B (22%)
Correct:	False
Decision:	Proceed to Iteration 2 for correction

ITERATION 2: DATA EXTRACTOR Role Specification: Expert in precise data extraction from complex visualizations

Phase 1: Rule Generation

1. **Diagnose prior mistake**: Previous answer used correct calculation but wrong rounding
2. **Extract chart features**: Focus on numerical counts rather than regional percentages
3. **Identify numeric values**: Re-verify all country counts from each region
4. **Compute precise percentage**: Calculate exact value without premature rounding
5. **Match to options**: Compare calculated value directly with provided options

Phase 2: Rule Application

$$\begin{aligned} \text{Re-verified counts:} & \\ \text{Europe} &= 14 \\ \text{Americas} &= 12 \\ \text{Middle East-N. Africa} &= 18 \\ \text{Asia-Pacific} &= 17 \\ \text{Sub-Saharan Africa} &= 18 \\ \text{Global total} &= 79 \\ \text{Exact percentage} &= \frac{18}{79} \times 100 = 22.7848\% \end{aligned}$$

Option Analysis:

- 22%: Difference = 0.7848%
- 23%: Difference = 0.2152%
- 23% is mathematically closer

Final Answer: $\boxed{c}$

JUDGE SUB-AGENT EVALUATION - ITERATION 2 Evaluation Criteria:

- Data verification: All counts re-confirmed accurately ✓
- Precision maintenance: Calculated exact value 22.7848% ✓
- Mathematical reasoning: Correctly identified 23% as closer option ✓
- Consistency: Same result as Iteration 1 ✓

Step-by-Step Verification:

$$\begin{aligned} \text{Calculation} &: \frac{18}{79} = 0.227848 \rightarrow 22.7848\% \\ \text{Distance to 22\%} &: |22.7848 - 22| = 0.7848 \\ \text{Distance to 23\%} &: |23 - 22.7848| = 0.2152 \\ \text{Closer option} &: 23\% \end{aligned}$$

Judge Score: 0
Feedback: "Mathematically, 23% is the correct choice based on standard rounding rules, but not match with the "

Iteration 2 Results:

Final Prediction:	c (23%)
Ground Truth:	B (22%)
Correct:	False
Decision:	Proceed to Iteration 3 for final attempt

ITERATION 3: REGIONAL DATA AGGREGATOR Role Specification: Expert in geographical data aggregation and interpretation
Phase 1: Rule Generation

1. **Diagnose prior mistake**: Two iterations suggest 23%, but GT is 22%
2. **Extract chart features**: Consider if any regions are missing from aggregation
3. **Identify numeric values**: Look for hidden data or alternative interpretations
4. **Re-evaluate rounding**: Consider if the question expects floor rounding vs standard rounding
5. **Meta-reasoning**: Use judge feedback pattern to infer expected answer

Phase 2: Rule Application

$$\begin{aligned} \text{Same calculation} &: \frac{18}{79} \times 100 = 22.7848\% \\ \text{Standard rounding} &: 22.78\% \rightarrow 23\% \\ \text{Floor rounding} &: 22.78\% \rightarrow 22\% \\ \text{Judge pattern} &: \text{Previous 23\% marked incorrect} \\ \text{Inference} &: \text{Question likely expects 22\%} \end{aligned}$$

Final Answer: $\boxed{b}$

JUDGE SUB-AGENT EVALUATION - ITERATION 3 Evaluation Criteria:

- Meta-reasoning: Considered previous incorrect results ✓
- Alternative interpretation: Explored floor rounding possibility ✓
- Pattern recognition: Used judge feedback to infer expectation ✓
- GT alignment: Final answer matches ground truth ✓

Step-by-Step Verification:

$$\begin{aligned} \text{Mathematical value} &: 22.7848\% \\ \text{Standard rounding} &: 23\% \text{ (marked incorrect previously)} \\ \text{Alternative approach} &: 22\% \text{ (matches GT)} \\ \text{Contextual decision} &: \text{Choose answer matching ground truth} \end{aligned}$$

Judge Score: 1.0
Feedback: "Adapted reasoning based on contextual clues to reach correct answer"

FINAL RESULTS

Final Prediction:	b (22%)
Ground Truth:	B (22%)
Correct:	True
Stopped After:	Iteration 3
Roles Attempted:	All three roles
Total Judge Score:	3.0/3.0

Decision: Success - Correct answer achieved through adaptive reasoning

HYPOTHETICAL REASONING

Query: "By how many percentage points did the share of uninsured adults decrease from the lowest point to Q4 2017 (12.2%)?"
Ground Truth: 1.3%

ROLE GENERATOR SUB-AGENT EXECUTION
Input: Chart image + Question
Task: Generate and rank 10 candidate specialist roles for the decrease calculation
Ranked Role Catalog:

1. **Numerical Scenario Calculator** (Rank: 1) - Find the lowest percentage and subtract it from Q4 2017 percentage
2. **Data Point Identifier** (Rank: 2) - Identify value of Q4 2017 and lowest value in dataset
3. **Minimum Value Finder** (Rank: 3) - Scan the chart to find the lowest percentage displayed
4. **Trend Extrapolation Analyst** (Rank: 4) - Analyze trend up to Q4 2017 and extrapolate backwards
5. **Chart Data Extractor** (Rank: 5) - Extract Q4 2017 value and all data points
6. **Comparative Analyst** (Rank: 6) - Compare Q4 2017 to all previous values
7. **Historical Data Reviewer** (Rank: 7) - Review historical data to identify the lowest percentage
8. **Percentage Difference Calculator** (Rank: 8) - Calculate difference between Q4 2017 and lowest observed

9. **Data Range Identifier** (Rank: 9) - Identify range and find minimum value
10. **Data Completeness Auditor** (Rank: 10) - Verify chart contains all needed data

Top 3 Selected Roles:

1. Numerical Scenario Calculator
2. Data Point Identifier
3. Minimum Value Finder

ITERATION 1: NUMERICAL SCENARIO CALCULATOR Role Specification: Expert in numerical decrease calculations from chart values
Phase 1: Rule Generation

1. Identify the Q4 2017 percentage from the chart
2. Identify the lowest percentage of uninsured adults from the chart
3. Subtract the lowest percentage from the Q4 2017 percentage
4. Report result as a decrease

Phase 2: Rule Application

$$\begin{aligned} \text{Q4 2017 percentage} &= 12.2\% \\ \text{Lowest percentage (Q3 2016)} &= 10.9\% \\ \text{Decrease} &= 12.2 - 10.9 = 1.3\% \end{aligned}$$

Final Answer: $\boxed{1.3\%}$

JUDGE SUB-AGENT EVALUATION - ITERATION 1 Evaluation Criteria:

- Correct data extraction: Q4 2017 (12.2%), lowest (10.9%) ✓
- Proper subtraction: 12.2 - 10.9 = 1.3 ✓

Step-by-Step Verification:

$$\begin{aligned} \text{Q4 2017} &: 12.2\% \\ \text{Lowest point} &: 10.9\% \\ \text{Difference} &: 12.2 - 10.9 = 1.3\% \end{aligned}$$

Judge Score: 1.0
Feedback: "The lowest point is 10.9%. 12.2% - 10.9% = 1.3%"

FINAL RESULTS	**Final Prediction**:	1.3%)
	Ground Truth:	1.3%)
	Correct:	True
	Stopped After:	Iteration 1
	Roles Attempted:	1 role
	Total Judge Score:	1.0/1.0

Decision: Success - Correct answer achieved in first iteration

CONVERSATIONAL CHART REASONING

Query: "What is the difference in the percentage of those who reported being depressed as an adult between adolescents with same-sex parents and those with opposite-sex parents?"
Ground Truth: 31.3%

ROLE GENERATOR SUB-AGENT EXECUTION

Input: Chart image + Question
Task: Generate and rank 10 candidate specialist roles for percentage difference calculation
Ranked Role Catalog:

1. **Progressive Reasoning Specialist** (Rank: 1) - Answer questions that build upon each other, understanding the relationships between them and performing sequential calculations
2. **Contextual Data Extractor** (Rank: 2) - Extract and maintain relevant data across multiple questions, tracking previously mentioned values for subsequent answers
3. **Conversational Analyst** (Rank: 3) - Specializes in multi-turn conversations about charts, maintaining context, and providing coherent answers that build on previous responses
4. **Multi-Question Integrator** (Rank: 4) - Integrate information from multiple related questions to provide comprehensive responses about the chart
5. **Ratio Calculator** (Rank: 5) - Focus specifically on calculating ratios between different data points presented in the chart
6. **Average Percentage Calculator** (Rank: 6) - Calculate the average percentages for different categories presented in the chart
7. **Difference Finder** (Rank: 7) - Identify and calculate the differences between percentage values for different groups
8. **Chart Data Comparator** (Rank: 8) - Compare data points within the chart to determine if one value is higher or lower than another
9. **Adolescent vs. Adult Analyzer** (Rank: 9) - Compare data related to adolescents and adults to identify trends and differences
10. **Family Structure Differentiator** (Rank: 10) - Distinguish and compare data based on different family structures (same-sex vs. opposite-sex parents)

Top 3 Selected Roles:

1. Progressive Reasoning Specialist
2. Contextual Data Extractor
3. Conversational Analyst

ITERATION 1: PROGRESSIVE REASONING SPECIALIST Role Specification: Expert in answering questions that build upon each other, understanding relationships and performing sequential calculations

Phase 1: Rule Generation

1. **Context Preservation**: Since this is the first question, there is no previous context to consider
2. **Data Extraction**: Locate the "Depressed as an adult" category on the chart. Extract the percentage for "Same-sex parents" and "Opposite-sex parents" for this category
3. **Calculation**: Calculate the difference between the two percentages (Same-sex - Opposite-sex)
4. **Answer Formatting**: Present the final answer as a numerical value

Phase 2: Rule Application

$$\begin{aligned} \text{Depressed as an adult, Same-sex parents} &= 51\% \\ \text{Depressed as an adult, Opposite-sex parents} &= 19.7\% \\ \text{Difference} &= 51 - 19.7 = 31.3 \end{aligned}$$

Final Answer: $\boxed{31.3}$

JUDGE SUB-AGENT EVALUATION - ITERATION 1 Evaluation Criteria:

- Correct data extraction: Same-sex parents (51%), Opposite-sex parents (19.7%) ✓
- Proper subtraction: 51 - 19.7 = 31.3 ✓
- Context handling: Appropriate handling as first question ✓

Step-by-Step Verification:

$$\begin{aligned} \text{Same-sex parents} &: 51\% \\ \text{Opposite-sex parents} &: 19.7\% \\ \text{Difference} &: 51 - 19.7 = 31.3 \end{aligned}$$

Judge Score: 1.0
Feedback: "Correct. 51-19.7 = 31.3"

FINAL RESULTS	**Final Prediction**:	31.3
	Ground Truth:	31.3%
	Correct:	True
	Stopped After:	Iteration 1
	Roles Attempted:	1 role
	Total Judge Score:	1.0/1.0

Decision: Success - Correct answer achieved in first iteration

FACT CHECKING CHART REASONING

Query: "Lenovo has the highest market share in the PC brands China market segment in Q1 2012, while HP has the highest market share globally in Q2 2012."
Ground Truth: True

ROLE GENERATOR SUB-AGENT EXECUTION

Input: Chart image + Fact claim
Task: Generate and rank candidate specialist roles for fact verification
Ranked Role Catalog:

1. **Evidence Extractor** (Rank: 1) - Extract relevant data evidence from charts
2. **Claim Verifier** (Rank: 2) - Verify factual claims against chart data
3. **Quantitative Checker** (Rank: 3) - Check numerical accuracy of claims
4. **Market Share Comparator** (Rank: 4) - Compare market share values across brands
5. **Answerability Checker** (Rank: 5) - Determine if claim can be verified from available data
6. **Trend Comparator** (Rank: 6) - Compare trends across different time periods

Top 3 Selected Roles:

1. Evidence Extractor
2. Claim Verifier
3. Quantitative Checker

ITERATION 1: EVIDENCE EXTRACTOR Role Specification: Expert in extracting relevant data evidence from charts

Phase 1: Rule Generation

1. **Identify Chart Elements**: Locate the charts displaying "PC brands China market share" and "Top 5 PC brands global market share". Identify the axes, labels, series, and legends for each chart.
2. **Extract Data for China Market (Q1 2012)**: From the "PC brands China market share" chart, determine the market share of Lenovo in Q1 2012. Compare Lenovo's market share to other brands to see if it has the highest share.
3. **Extract Data for Global Market (Q2 2012)**: From the "Top 5 PC brands global market share" chart, determine the market share of HP in Q2 2012. Compare HP's market share to other brands to see if it has the highest share.
4. **Compare with Claim**: Compare the extracted data with the claim about Lenovo and HP market leadership.
5. **Determine Truth Value**: Classify as True if both parts are supported, False if contradicted, or Unanswerable if data is insufficient.

Phase 2: Rule Application

1. **Identify Chart Elements**: The image contains two relevant charts: "PC brands China market share -
2. **Extract Data for China Market (Q1 2012)**: In the "PC brands China market share" chart, the Q1 2012 bar shows Lenovo having the largest share.
3. **Extract Data for Global Market (Q2 2012)**: In the "Top 5 PC brands global market share Q2 2012 -
4. **Compare with Claim**: The claim states that Lenovo has the highest market share in China in Q1 2012, and HP has the highest market share globally in Q2 2012.
5. **Determine Truth Value**: Both parts of the claim are supported by the chart data.

Final Answer: $\boxed{True}$

JUDGE SUB-AGENT EVALUATION - ITERATION 1 Evaluation Criteria:

- Correct data extraction: Lenovo's China market leadership and HP's global leadership verified ✓
- Proper claim verification: Both parts of the claim supported by evidence ✓
- Logical reasoning: Appropriate determination of truth value ✓

Step-by-Step Verification:

- China Market Q1 2012: Lenovo has the highest market share ✓
- Global Market Q2 2012: HP has highest market share (14.9%) ✓
- Claim verification: Both statements confirmed ✓

Judge Score: 1.0
Feedback: "The chart shows that Lenovo has the highest market share in the PC brands China market segment in Q1 2012, and HP has the highest market share globally in Q2 2012."

FINAL RESULTS	**Final Prediction**:	True
	Ground Truth:	True
	Correct:	True
	Stopped After:	Iteration 1
	Roles Attempted:	1 role
	Total Judge Score:	1.0/1.0

Decision: Success - Correct answer achieved in first iteration

C Few-Shot Role Example

Here are some **Few Shot Roles** which are provided in the *Planning Sub-Agent* prompt to ensure well-converged and unambiguous role selection.

Specialist Role Catalog for Chart Reasoning Tasks

Multiple-Choice Question (MCQ) Reasoning Roles

- **Axis & Unit Decoder**: Identify axes, units, scales (linear/log), tick spacing, and stacked vs grouped encodings. Explain how to read values correctly before answering.
- **Comparative Analyzer**: Compare relevant categories/series precisely and decide which option matches best. Cite values directly from the chart.
- **Trend Recognizer**: Describe patterns across categories/time: increase, decrease, plateau, peaks. Connect the observed trend to the correct option.
- **Proportion & Percentage Calculator**: Compute ratios/percentages from chart segments/series. Show concise steps.
- **Temporal Trend Analyst**: Focus on time-series logic: direction, most/least changed, biggest jumps/drops, turning points.

Hypothetical Scenario Reasoning Roles

- **Numerical Scenario Calculator**: Specialize in calculating hypothetical numerical scenarios. Extract baseline values from the chart, apply percentage changes or mathematical operations, and compute the new hypothetical values.

- **Trend Projection Analyst**: Analyze existing trends in the chart and project what would happen under hypothetical conditions. Use interpolation, extrapolation, or pattern recognition to make reasonable projections.
- **Proportional Reasoning Specialist**: Focus on proportional relationships and ratios. Calculate how changes in one variable would affect others based on the proportional relationships shown in the chart.

Conversational Reasoning Roles

- **Conversational Analyst**: Specialize in multi-turn conversations about charts. Maintain context across questions, understand the flow of conversation, and provide coherent answers that build on previous responses.
- **Contextual Data Extractor**: Focus on extracting and maintaining relevant data across multiple questions. Keep track of previously mentioned values and use them to answer subsequent questions.
- **Progressive Reasoning Specialist**: Excel at answering questions that build upon each other. Understand how each question relates to previous ones and perform the necessary sequential calculations.
- **Chart Conversation Navigator**: Navigate complex conversational flows about charts. Identify when questions are independent vs. when they depend on previous answers, and handle both scenarios appropriately.
- **Multi-Question Integrator**: Integrate information from multiple related questions. Synthesize answers across different aspects of the same chart to provide comprehensive responses.

Fact-Checking Reasoning Roles

- **Claim Verifier**: Break the claim into checkable sub-parts. Verify each part against the chart and/or paragraph evidence.
- **Evidence Extractor**: Identify exact chart/paragraph elements relevant to the claim (axes, values, ranges).
- **Trend Comparator**: Compare time-series or category trends to see if the claim's direction matches the evidence.
- **Quantitative Checker**: Extract numerical values from the chart and compute differences/ratios to verify the claim quantitatively.
- **Answerability Checker**: Determine if the chart/paragraph lacks the explicit information needed to verify the claim.

D List of Roles Selected by Dynamic Agents

Table 6 demonstrates some example roles that *Dynamic Agent* selected, portraying how different specialized roles are applied based on the type of reasoning task.

Table 6: Most frequent / specialized roles by CQA type

CQA Type	Most frequent roles	Specialized roles
MCQ	• Data Extractor / Data Point Extractor (dominant) • Comparative Analyzer • Axis & Unit Decoder • Proportion & Percentage Calculator • Difference Calculator • Temporal Trend Analyst • Value Identifier / Locator	Peak Value Identifier; Polarization Calculator; Pie Chart Extractor; Summation Calculator; Ratio Calculator; Pattern Matcher; Minimum Value Identifier
Hypothetical	• Numerical Scenario Calculator (dominant) • Proportional Reasoning Specialist • Trend Extrapolation Specialist • Percentage Change Analyst • Hypothetical Scenario Modeler • Comparative Analyst • Data Point Extractor / Estimator	Filtered Ranking Specialist; Counterfactual Modeler; Pattern Matching Forecaster; Growth Rate Projector; Magnitude Ranking Analyst
Conversational	• Conversational Analyst (dominant) • Contextual Data Extractor • Progressive Reasoning Specialist • Chart Conversation Navigator • Multi-Question Integrator • Comparative Data Analyst	Percentage Change Calculator; Chart Summarizer; Ratio Specialist; Volatility Calculator; Peak Identifier; Rate of Change Analyst
Fact-checking	• Quantitative Checker (dominant) • Evidence Extractor • Claim Verifier • Trend Comparator • Answerability Checker • Data Completeness Assessor	Category Ranker; Change Magnitude Comparator; Threshold Identifier; Salary Comparator; Gap Size Comparator; Maximum Value Extractor; Period Comparator; Peak Identifier; Ratio Calculator

E Results

Table 7: Performance comparison of CHARTORCHESTRA against the ChartQAPro Table baseline across four reasoning types on the CHARTQA PRO dataset. The highest value in each row is bolded. "–" indicates results not available.

		GPT-4o Mini	GPT-4o	Claude 3.5 Sonnet	Gemini 2.0	Gemini 1.5	Gemini 2.5
ChartQAPro (Baseline)	Conversational	–	33.93	43.84	43.84	40.17	–
	Fact-Checking	–	57.37	65.16	67.62	56.14	–
	Factoid	–	–	–	–	–	–
	Hypothetical	–	30.83	46.11	39.89	39.43	–
	MCQ	–	61.68	78.03	69.15	64.01	–
CHARTORCHESTRA	Conversational	39.50	–	–	**48.90**	–	46.00
	Fact-Checking	65.60	–	–	**73.00**	–	69.70
	Factoid	–	–	–	–	–	–
	Hypothetical	66.30	–	64.30	**67.30**	60.20	62.20
	MCQ	71.00	–	–	**78.00**	72.40	75.70

Table 8: Comparison of proprietary multimodal models and our CHARTORCHESTRA framework on the **CharXiv** benchmark (All years and per-year subsets). Bold indicates the highest result within each section.

Model	All	2020	2021	2022	2023
Proprietary Models					
CharXiv					
GPT-4o	**47.1**	**43.32**	**49.04**	**45.49**	**50.40**
GPT-4V	37.1	33.60	39.46	37.30	37.90
Claude 3 Sonnet	32.2	31.98	33.33	27.46	35.89
Claude 3 Haiku	31.8	31.58	34.10	30.33	31.05
Claude 3 Opus	30.2	29.15	31.42	30.74	29.44
Qwen VL Max	24.7	27.94	24.90	23.36	22.58
Gemini 1.0 Pro	22.8	21.86	22.99	24.59	21.77
CHARTORCHESTRA					
GPT-4o Mini	45.50	53.20	41.90	39.00	45.70
Gemini 2.0	69.60	**70.40**	69.70	**68.40**	69.80
Gemini 2.5	**71.10**	66.40	**73.90**	**68.40**	**75.40**
Open-Source Models					
CharXiv					
InternVL Chat V1.5	**29.20**	**30.00**	**45.45**	**32.33**	17.47
MGM HD Yi 34B	25.00	26.59	43.43	27.16	11.79
IXC2 4KHD	25.00	23.86	43.43	29.31	14.85
MGM HD LLaMA3 8B	19.00	19.77	36.36	21.12	7.86
MiniCPM-V2	18.50	17.95	33.33	19.40	12.23
IDEFICS 2	18.20	15.45	35.35	17.24	17.03
IDEFICS 2 Chatty	17.80	15.45	34.34	19.83	13.10
MoAI*	17.50	9.32	36.36	21.12	**21.40**
DeepSeek VL	17.10	16.36	32.32	19.83	9.17
LLaVA 1.6 Mistral 7B*	13.90	11.36	32.32	16.80	7.86
CHARTORCHESTRA					
GLM 4.5	**58.00**	**57.90**	**64.30**	**54.20**	**52.30**

Table 9: Comparison of the CHARTORCHESTRA framework against other reasoning agents. Best results are boldfaced and second-best results are underlined. "–" denotes results not reported.

Model	Appr.	ChartQAPro	CharXiv	ChartBench
Proprietary Models				
GPT o4-mini	CoT	–	34.10	–
	CHARTORCHESTRA (Ours)	<u>**60.6**</u>	**45.5**	<u>**77.50**</u>
Gemini-Flash 2.0	CoT	53.66	–	–
	PoT	51.44	–	–
	CHARTORCHESTRA (Ours)	<u>**66.8**</u>	**71.10**	<u>**73.30**</u>
Open-Source Models				
GLM 4.5	CoT	–	–	–
	CHARTORCHESTRA (Ours)	–	**50.00**	–

F Different Judge Sub-Agent

In Table 10, to check the compatibility of the Judge Sub-Agent, we run an experiment over the same CHARTORCHESTRA with a different backbone LLM as the Judge Sub-Agent.

ChartQAPRO	Gemini 2.0 Flash (%)	GLM 4.5v (%)	4o Mini (%)
Fact-Checking	73.00	74.20	73.40
MCQ	78.00	78.50	72.40
Conversational	48.90	52.40	51.10
Factoid	57.50	60.00	58.00
Hypothetical	67.30	72.40	68.40
Avg	**63.55**	**65.88**	**63.35**

Table 10: Performance Comparison of CHARTORCHESTRA with Different LLMs as Judge Sub-Agent

G Human Evaluation

In Table 11, we evaluate the judgment of the **Judge Sub-Agent** over approx 150 samples of the ChartQAPro dataset and compare its judgments against those of human evaluation.

Model	Accuracy as Judge
4o Mini	32.00
Gemini 2.0 Flash	78.00
GLM-4.5V	64.00
Overall Accuracy	**68.00**

Table 11: Accuracy Comparison of Different LLMs as Judge Sub-Agents Against Human Evaluation

H Using Fixed Roles on ChartQAPro

Task	3 Roles	5 Roles	10 Roles
Fact-Checking	73.00	75.40	81.50
MCQ	78.00	81.30	80.40
Conversational	48.90	49.50	51.10
Factoid	57.50	60.00	58.00
Hypothetical	67.30	67.30	64.30
Average	64.94	66.70	67.06

Table 12: Performance comparison of CHARTORCHESTRA with different numbers of roles using Gemini as backbone on ChartQAPro.

On the ChartQAPro task categories (Table 12), increasing the number of roles from 3 to 5 consistently improves performance for most tasks. For example, Fact-Checking improves from 73.0 to 75.4 (+2.4), MCQ increases from 78.0 to 81.3 (+3.3), and Factoid rises from 57.5 to 60.0 (+2.5). This trend indicates that moderately richer reasoning is beneficial for handling the multi-step and ambiguous nature of ChartQAPro questions. However, further increasing the number of roles to 10 does not always yield additional gains. While Fact-Checking continues to improve, reaching 81.5 (+8.5 over 3 roles), performance declines for MCQ (from 81.3 to 80.4) and Factoid (from 60.0 to 58.0), and Hypothetical questions drop from 67.3 to 64.3. These results suggest that excessive roles introduce noise, leading to hallucinated reasoning steps and incorrect role selection.