

# Hierarchical Visual Agent: Managing Contexts in Joint Image-Text Space for Advanced Chart Reasoning

Qihua Dong<sup>1</sup>, Ruozhen He<sup>2</sup>, Junwen Chen<sup>3</sup>, Yizhou Wang<sup>1</sup>, Xu Ma<sup>3</sup>, Songyao Jiang<sup>3</sup>, Yun Fu<sup>1</sup>

<sup>1</sup>Northeastern University, <sup>2</sup>Rice University, <sup>3</sup>Amazon AGI

## Abstract

Advanced chart question answering requires both precise perception of small visual elements and multi-step reasoning across several subplots. While existing MLLMs are strong at understanding single plots, they often struggle with multi-step reasoning across multiple subplots. We propose **HIERVA**, a **hierarchical visual agent** framework for chart reasoning that iteratively constructs and updates a working context in a joint image-text space. A high-level manager generates plans and maintains a compact context containing only key information, while specialized workers perform reasoning, gather evidence, and return results. In particular, the agent maintains separate visual and textual contexts, using a *zoom-in tool* to restrict the visual context. Experiments on the CHARXIV reasoning subset demonstrate consistent improvements over strong multimodal baselines, and ablation studies verify that hierarchical architecture, scoped visual context, and distilled context contribute complementary gains.

## 1 Introduction

*Chart question answering* is a key capability for scientific assistants, document understanding systems, and accessibility tools. While existing multimodal large language models (MLLMs) are strong at understanding single plots, they often struggle with multi-step reasoning across multiple subplots (Wang et al., 2024). This hinders their application in real-world chart reasoning tasks.

The core challenge is that complex chart reasoning is inherently an *image-text mixed task*: answering a single question often requires aggregating fine-grained visual details—tick labels, axis units, legend entries, dense markers—from multiple scattered regions. This creates two compounding difficulties. First, relevant evidence is often small or cluttered when the chart is viewed at global scale, making accurate recognition unreliable. Second,

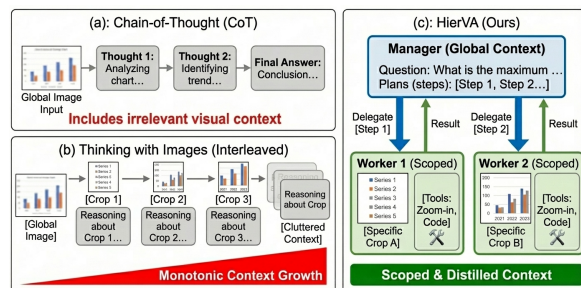
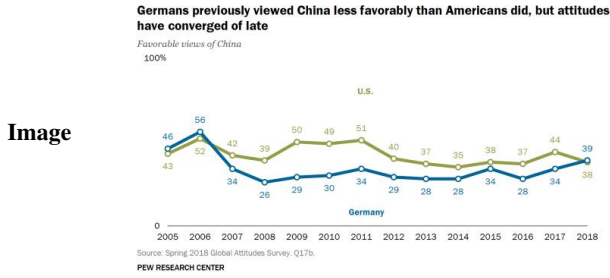


Figure 1: **Comparison of chart reasoning paradigms.** (a) **Chain-of-Thought (CoT)** reasons sequentially over a global image but is often distracted by irrelevant visual content. (b) **Thinking with Images (Interleaved)** iteratively acquires visual crops but appends them to a monotonically growing context, leading to clutter and distraction. (c) **HIERVA (Ours)** uses hierarchical visual agents that maintain a joint image-text working context. Each subtask operates in a scoped local context with task-specific crops and tasks, returning only distilled results to prevent context overload.

multi-step reasoning (e.g., comparing series across coordinates, computing differences, or identifying constrained extrema) demands retaining and coordinating intermediate results across steps. As the reasoning chain grows, the accumulated context becomes long and heterogeneous, mixing task-relevant signals with information that is irrelevant to the current step—causing earlier evidence to be “forgotten” or diluted.

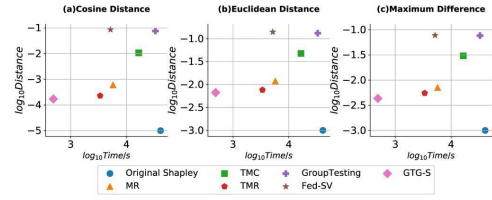
Multiple approaches have been proposed to manage context in chart reasoning tasks. For multi-step reasoning, text-based chain-of-thought (CoT) prompting (Wei et al., 2022; Yao et al., 2022; Wang et al., 2023; Shinn et al., 2023; Zhang et al., 2025) is mainly adopted to elicit explicit reasoning traces. Recent “thinking with images” patterns incorporate visual details into reasoning traces by acquiring additional visual evidence (e.g., zooming into a region) during inference (OpenAI, 2025b; Lai et al., 2025; Zheng et al., 2025), making it easier for the agent to focus on the current step’s region of

## Basic Chart Understanding



Image

## Advanced Chart Reasoning



**Question** What's the color of graph with 56 as the highest value?

In the graphs, how many times does the  $\log_{10}$ -Distance value for TMR fall within the range of  $[-2.5, -2.0]$ ?

Table 1: Comparison of basic vs. advanced chart reasoning tasks. Basic tasks require localizing a single region and one-step retrieval. Advanced tasks require aggregating evidence across multiple regions and multi-step computation.

interest. However, a common failure mode is *monotonic context growth*: since images consume tokens, repeatedly appending crops and intermediate text dilutes the global context and degrades later steps that rely on it—especially those requiring access to the original visual context.

This paper advocates a simple but consequential principle: *multimodal reasoning benefits from managing contexts in joint image-text space with the same discipline used to manage textual reasoning traces*. Concretely, effective chart agents should (i) treat visual context with the same care as textual context, (ii) scope each step's context to only the information necessary for that step, and (iii) distill intermediate traces into compact, high-signal state. In other words, instead of continuously growing the context, the system should *refine and compress* it, keeping the conditioning information focused and stable across steps.

We operationalize this principle with **Hierarchical Visual Agent (HIERVA)**, a training-free manager-worker framework for advanced chart reasoning. The manager performs **planning and zoom-in** to restrict textual and visual contexts to each step's needs, dispatching them as explicit inputs to the appropriate worker. Crucially, our hierarchy enforces **context distillation and encapsulation**: intermediate artifacts (e.g., multiple planning drafts) are refined and only the clean final version is retained in the manager context; each worker's reasoning trace is hidden behind an abstraction barrier so the manager observes only the assigned task and its result; and workers operate with minimal, task-specific contexts, improving efficiency and reducing distraction. This design keeps the global

context compact and high-signal while preserving rich intermediate computation locally where it is needed.

We evaluate on the CHARXIV reasoning subset and show consistent improvements over strong multimodal baselines. To summarize, our contributions are:

- A **manager-worker hierarchical visual agent** (HIERVA) with explicit multimodal working context.
- **Context distillation and encapsulation** to improve efficiency and reduce distraction.
- **Improved performance** on advanced chart reasoning tasks without additional training.

## 2 Method

### 2.1 Problem setup

Given a chart image  $I_0$  and a natural-language question  $q$ , the goal of chart question answering (QA) is to produce an answer  $a$ .

### 2.2 HIERVA: Hierarchical Visual Agent

We implement the principle from §1—*acquire, scope, and distill evidence*—with **HIERVA**, composed of a **manager** and multiple **workers**. Conceptually, both manager and workers are instances of the same underlying multimodal model, but they are used with different *roles, tools, and context*.

**Global vs. local contexts.** The manager maintains a persistent global working context  $C_M$  (a message history) containing: (i) the original question and chart reference, (ii) a refined plan, and (iii) compact, structured summaries of completed subtasks. In contrast, each worker is invoked with

---

**Algorithm 1:** HIERVA for chart task

---

**Input:** Chart image  $I_0$ , question  $q$ , skill library  $\mathcal{S}$ , tools  $\mathcal{T}$   
**Output:** Answer  $a$   
Initialize manager context  $C_M \leftarrow \{q, I_0\}$ ;  
 $P \leftarrow \text{PLAN}(C_M)$  with optional refinement;  
store only refined  $P$  in  $C_M$ ;  
**for**  $t = 1, 2, \dots$  **do**  
    **if**  $t > 1$  and  $\text{SHOULDSTOP}(C_M)$  **then**  
        **return**  $\text{FINALANSWER}(C_M)$ ;  
     $\tau_t \leftarrow \text{CREATENEXTTASK}(C_M)$  using  
        schema-constrained self-correction;  
     $o_t \leftarrow \text{EXECUTEWORKER}(\tau_t)$  with one  
        image and selected skills/tools;  
    **if**  $\text{HASHOOK}(\tau_t)$  **then**  
         $o_t \leftarrow \text{EXECUTEHOOK}(o_t)$ ;  
    Append a distilled summary of  $(\tau_t, o_t)$   
        to  $C_M$ ;

---

an isolated local context  $C_{W_t}$  containing only: (i) the task instruction and minimal required background, (ii) an optional *skill* procedure selected by the manager (see §2.5), and (iii) a *single* image input (either  $I_0$  or a zoomed crop). This separation enforces **encapsulation**: the manager never appends the worker’s deliberation trace into  $C_M$ , preventing irrelevant intermediate text from accumulating across steps.

**Action space.** At each step  $t$ , the manager chooses between (a) requesting additional *visual evidence* via a zoom-in crop, or (b) requesting a *textual fact* (e.g., a read-off value, a comparison result, an intermediate computation). Zoom actions are executed by workers through a dedicated image tool, and the cropped image will be returned to the manager; textual subtasks may additionally use lightweight tools (e.g., code execution) chosen by the manager.

### 2.3 Manager control loop

Algorithm 1 summarizes the orchestration loop. The process begins with a two-stage planning phase, then iterates over task creation, worker execution, optional coordinate normalization, and distilled reporting back to the manager until termination.

**Two-stage planning with refinement.** The manager first prompts for a high-level plan, then prompts for an explicit step list. The plan is critical

for the final results since it lists clearly the context for each worker and we find the two-stage planning is more effective. If we ask for a detailed explicit plan at once, the manager may struggle to separate context for workers in a clean way. Crucially, the manager retains only the *final refined plan* in  $C_M$ , discarding intermediate drafts.

**Iterative execution and termination.** After the first task is executed, the manager periodically checks whether it has enough information to answer the original question. If not, it generates the next subtask, dispatches an appropriate worker, and incorporates the resulting distilled fact into  $C_M$ . When the manager predicts that no further subtasks are needed, it produces the final answer in a standardized format (we use a `\boxed{\}` wrapper for reliable extraction).

### 2.4 Adaptive zoom and worker execution

Fine-grained chart evidence is often not readable from a single global-scale encoding of  $I_0$ . We therefore treat zooming as an explicit action: the manager can create an image-expected task whose worker calls a zoom-in tool. A zoom action specifies a bounding box and gets a cropped, resized image for high-resolution inspection.

Each worker is created on demand with exactly one image input (either the original chart or a previously generated crop), enforcing scoped evidence. Workers produce outputs matching the expected type; for text tasks, we extract only the final sentence to keep manager-visible updates compact (see Appendix B for details on task schema and image indexing).

### 2.5 Skill routing and context management

For certain chart images, code execution can yield more accurate answers (e.g., extracting precise point coordinates). However, workers tend to respond directly rather than invoking code tools. To address this, we introduce task-specific *skills* into the worker’s context. Naively encoding all such procedural knowledge in the manager or worker’s base prompt would bloat  $C_M$  and encourage monotonic context growth. Instead, we maintain a compact **skill library**  $\mathcal{S}$ , where each skill is a short markdown procedure. For each task, the manager selects a relevant subset of skills and injects their text into the worker’s system prompt. Crucially, the manager never reads the skill content, and workers only see skills when explicitly selected. This

**just-in-time** injection guides workers effectively while avoiding context bloat—a direct benefit of the HIERVA design.

Our orchestration explicitly targets monotonic context growth through three mechanisms: **(1) Plan distillation**: only the final refined plan is retained, discarding intermediate drafts. **(2) Worker encapsulation**: workers execute in private contexts; the manager does not inherit worker message histories or tool call traces. **(3) Result distillation**: worker responses are converted to compact summaries (a single sentence for text tasks, a crop handle for image tasks) before being added to  $C_M$ . Together, these mechanisms preserve rich intermediate computation locally while keeping global conditioning focused and stable.

### 3 Basic Chart Understanding Tasks

In this section, we examine why text-based CoT and “thinking with images” alone are insufficient for chart tasks, and argue that explicit visual context management is a critical component of chart reasoning.

#### 3.1 Initial Study on ChartQA

On classic benchmarks such as ChartQA (Masry et al., 2022), current MLLMs with text-based CoT already achieve near-saturated performance (90%+ Bai et al., 2025), while “thinking with images” approaches (Zheng et al., 2025) offer little additional gain. This is unsurprising: ChartQA images are relatively simple, typically containing a single plot with few visual elements, so the visual context remains short and requires no explicit management.

#### 3.2 Task Setup

We consider two chart understanding settings: (i) the CHARTQA validation set (1.92k examples), where each example typically consists of a single plot requiring single-step reasoning (similar to the left sample in Figure 1), and (ii) synthetic composite charts (1k examples), where we systematically control chart complexity by varying the number of subplots (sp#) and data series (see §3.2.1 for generation details). We report accuracy (%) and use the same backbone model across all methods.

##### 3.2.1 Synthetic Charts

In order to fairly study the performance of different methods on chart tasks when the input images

Method	ChartQA	Synthetic Charts			
	sp#1	sp#1	sp#2	sp#4	sp#6
Direct	88.9	88.5	86.5	84.8	83.1
CoT	<b>90.2</b>	<b>90.1</b>	<b>89.2</b>	88.4	87.5
CoT-Plan	90.1	89.8	88.1	86.2	86.0
Thinking w/ Images	89.9	89.7	88.5	87.4	87.5
HIERVA (Ours)	89.9	89.7	<b>89.2</b>	<b>88.5</b>	<b>88.2</b>

Table 2: Comparison of methods on ChartQA and synthetic charts with varying number of subplots (sp#), using Qwen3VL-A22B. The metric is accuracy(%).

are of different complexity, we use GPT5 (OpenAI, 2025a) to synthesize chart data, and control the number of subplots and the number of visual elements in the input image.

**Data Collection** We use few-shot prompting to generate questions that remain ChartQA-like in style and difficulty. For input images, we cover diverse chart types (line, bar, scatter, pie, etc.), varying numbers of data series (2–10), and different subplot counts (1, 2, 4, 6). For each question–plot pair, we generate four input images with 1, 2, 4, and 6 subplots respectively; the target plot is randomly placed in one subplot while the remaining subplots are filled with randomly generated charts. This controlled setup enables systematic study of the impact of visual context complexity.

#### 3.3 Methods

We compare against four baselines using the same backbone model Qwen3VL-A22B (Bai et al., 2025).

**Baselines** **Direct** prompts the model with the question and chart image without any chain-of-thought reasoning. **CoT** follows standard chain-of-thought prompting (Wei et al., 2022), instructing the model to reason step-by-step before producing the final answer. **CoT-Plan** (Wang et al., 2023) uses a “plan then solve” prompt, first generating a multi-step plan decomposing the question, then executing each step sequentially to derive the final answer. Exact prompts for these baselines are provided in Appendix A. **Thinking w/ Images** implements the recently proposed “thinking with images” paradigm (Zheng et al., 2025; Lai et al., 2025), where the model can iteratively acquire additional visual evidence (zoom-in crops) during reasoning. Unlike HIERVA, this baseline appends all intermediate crops and reasoning traces to a single growing context without hierarchical delegation or context distillation.

**HIERVA** HIERVA is our proposed method described in §2. For this study, we use the same backbone model and provide the zoom-in action. No other tools are used, and there are no injected skills. The key differences from “Thinking w/ Images” are: (i) hierarchical delegation where a manager dispatches subtasks to workers with isolated contexts, (ii) context distillation that retains only compact summaries rather than full reasoning traces, and (iii) scoped evidence where each worker receives only the relevant image crop rather than accumulating all crops in a single context.

**Action Space** For HIERVA and **Thinking w/ Images**, we provide a ‘zoom-in’ action.

### 3.4 Results and Analysis

Table 2 summarizes the performance of all methods on ChartQA and synthetic charts with varying subplot counts.

**Composite charts degrade accuracy.** Across all methods, accuracy consistently drops as the number of subplots increases: from sp#1 to sp#6, we observe a decline of 5.4% for Direct, 2.6% for CoT, 3.8% for CoT-Plan, 2.2% for Thinking w/ Images, and 1.5% for HIERVA. This trend confirms that composite chart images pose a fundamental challenge: the target information occupies a smaller fraction of the visual input, and the other visual context makes it harder for models to answer the question.

**Managing visual context helps.** Methods with zoom capability (**Thinking w/ Images** and HIERVA) degrade more gracefully than **Direct**, **CoT**, and **CoT-Plan**, showing that actively managing visual context—by acquiring high-resolution crops of relevant regions—helps preserve accuracy on composite charts.

**Context management does not yet differentiate.** Interestingly, HIERVA and **Thinking w/ Images** perform similarly on these ChartQA-style questions despite their architectural differences. We attribute this to: most questions here require single-step value retrieval rather than multi-step reasoning, so the benefits of context management do not yet manifest.

**Motivation for complex reasoning tasks.** These results suggest that for simple retrieval questions, *acquiring* evidence suffices—but we hypothesize that *managing* textual and visual context becomes

critical when reasoning chains grow longer and intermediate results must be tracked. This motivates our evaluation on CHARXIV (§4), where questions require multi-step comparisons, aggregations, and computations.

## 4 Advanced Chart Reasoning Tasks

Real-world chart tasks are often far more complex than the basic retrieval questions studied in §3. To evaluate advanced, multi-step chart reasoning in the wild, we turn to the CHARXIV reasoning split (Wang et al., 2024), a challenging benchmark that requires multi-step comparisons, aggregations, and reasoning over composite charts drawn from arXiv papers. Even strong closed-source models struggle on this benchmark, making it an ideal testbed for our approach (Wang et al., 2024).

**Difference between basic and advanced chart tasks.** An example is shown in Table 1. In basic chart tasks (§3), most questions can be answered by performing one-step value retrieval. In contrast, advanced chart questions often require (i) aggregating evidence across multiple regions/series or panels, and (ii) maintaining intermediate results to complete multi-step computations (e.g., deltas, ratios, averages). This shift makes long-horizon *context management* critical: naive approaches that append all intermediate reasoning and visual evidence into one growing transcript are more likely to lose key facts or introduce inconsistencies as the chain length increases.

### 4.1 Task Setup

**CHARXIV reasoning.** We evaluate on the validation split of the CHARXIV reasoning subset, which contains 1000 examples (Wang et al., 2024), and report **accuracy**. For finer-grained analysis, we use an LLM to classify questions into six reasoning types: *read value*, reading or estimating a numeric value; *find extreme*, identifying max/min or ranked extremes; *find the first*, identifying the first crossing, drop, plateau, or threshold event; *reverse read*, selecting the label, method, or subplot matching a described pattern; *compare*, comparing two items; and *freq*, counting occurrences or items meeting a condition.

### 4.2 Methods

**Baselines.** For **Direct**, **CoT**, and **CoT-Plan**, we use the same tool-free baselines as in §3.3.

Method	Image Tools	Skills	CHARXIV							Peak Tok #
			All	Extr	First	Read	RevR	Comp	Freq	
Direct	—	—	45.7	38.5	42.1	75.4	62.3	28.9	27.0	702.48
CoT	—	—	62.1	64.2	53.5	<b>78.9</b>	69.1	54.4	52.5	1926.19
CoT-Plan	—	—	62.4	64.8	54.1	78.2	69.4	55.1	52.8	1947.21
Thinking w/ Images	zoom	—	58.9	63.5	52.1	77.5	<b>70.2</b>	45.1	43.0	2995.02
Thinking w/ Images	zoom, code	—	50.1	48.5	44.2	72.5	63.1	38.5	33.8	2990.23
Thinking w/ Images	zoom, code	fixed	55.3	58.2	48.5	76.8	66.5	42.1	39.7	2837.14
<b>HierVA (Ours)</b>	zoom, code	dynamic	<b>64.2</b>	<b>66.3</b>	<b>54.5</b>	78.4	70.1	<b>59.9</b>	<b>55.0</b>	2568.38

Table 3: Performance on CHARXIV reasoning subset. Accuracy (%) by question type: Extr=Find extreme, First=Find the first, Read=Read value, RevR=Reverse read, Comp=Compare, Freq=Frequency count. Peak Tok #=avg. per-example peak token count.

Variant	Hier.	Dist.	Scop.	Skill	Acc. (↓)
<b>Full HIERVA</b>	✓	✓	✓	✓	<b>64.2</b>
w/o hierarchy	×	✓	✓	✓	53.8 (↓10.4)
w/o distillation	✓	×	✓	✓	55.5 (↓8.7)
w/o scoped context	✓	✓	×	✓	61.1 (↓3.1)
w/o dynamic skills	✓	✓	✓	×	59.6 (↓4.6)

Table 4: Ablation study on CHARXIV reasoning subset. Hier.=hierarchical delegation, Dist.=context distillation, Scop.=scoped context, Skill=just-in-time skill routing.

To fairly isolate the effects of tool access and skill injection, we evaluate three variants of **Thinking w/ Images**: (i) with the *zoom-in tool only*, matching the baseline in §3.3; (ii) with *zoom-in and code interpreter*, matching the tool set available to HIERVA; and (iii) with *zoom-in, code interpreter, and a fixed skill list* prepended to the system prompt. In variant (iii), the skills mirror those used by HIERVA (§2.5), but are provided statically rather than injected dynamically per subtask.

**HIERVA.** Here, in addition to the zoom-in tool, HIERVA can use a code interpreter to programmatically analyze zoomed chart crops (e.g., extracting series/marks by color, locating points/bars, reading values, comparing values, and identifying extrema), and to carry out the corresponding derived computations needed for multi-step questions. HIERVA also uses skills as compact procedural guidance: for each subtask, the manager selects a task-relevant subset from the skill library and injects it into the worker’s system prompt just in time (see §2.5), improving reliability without bloating the global manager context.

Concretely, we use five injected skills (shown as short markdown documents) including: `ReadValue` (extract and parse a numeric value), `FindExtremum` (argmax/argmin over

extracted values), `CompareValues` (pairwise comparison with unit/format normalization), `ComputeAggregate` (sum/mean/range over extracted values), and `ComputeRatioDelta` (ratios/percent change/differences).

### 4.3 Case Study

Figure 2 presents an example where the model must answer an advanced chart reasoning question: among four closely spaced boxplots (a–d), identify the sub-plot with the highest median. **CoT** attempts to judge medians directly from the global view and incorrectly concludes that (a) and (b) are tied, reflecting brittle estimation when differences are subtle. **Thinking w/ Images** improves perception by zooming, but because it appends every crop and intermediate rationale into a single growing context, it becomes prone to bookkeeping errors—for instance, when attempting to crop subplot (d), it mistakenly crops subplot (c)—and ultimately produces an incorrect answer. In contrast, **HIERVA** decomposes the task into scoped worker subtasks: the manager plans the necessary steps, including cropping the left panel and each sub-plot, and delegates each sub-plot to a dedicated worker with a single relevant view. Each worker localizes the median line and returns a compact scalar measurement (the median line’s vertical pixel coordinate). The manager then compares these returned measurements and correctly identifies (b) as the highest-median sub-plot.

### 4.4 Results and Analysis

**Main results.** Table 3 presents the results on the CHARXIV reasoning subset. HIERVA achieves the highest overall accuracy of **64.2%**, surpassing the strongest baseline (CoT-Plan) by 1.8% and the Thinking w/ Images baseline by 5.3%. Notably,

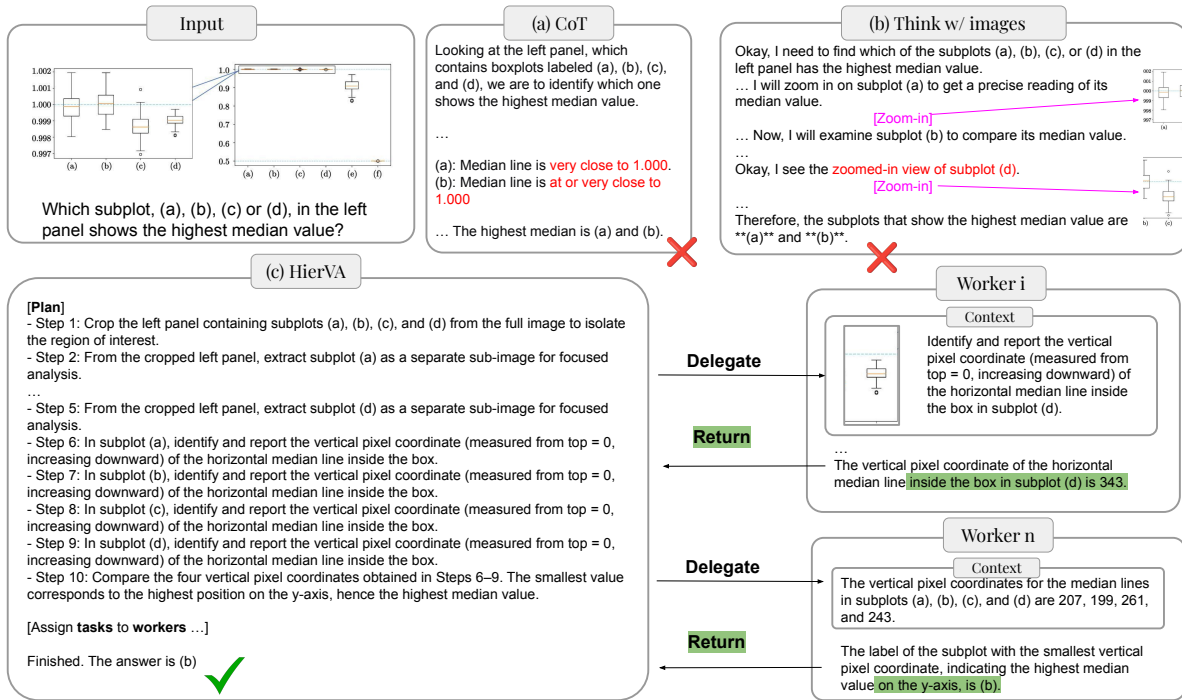


Figure 2: **Case study comparing reasoning approaches.** We illustrate how different methods handle an advanced chart reasoning question. **CoT** reasons sequentially over the global image but struggles with fine-grained details. **Thinking w/ Images** iteratively zooms into regions but accumulates all crops and intermediate reasoning in a growing context, leading to errors. **HIERVA** manages the context effectively by using hierarchical agents and scoped context. Red text highlights incorrect reasoning steps.

our method shows the largest gains on complex reasoning categories such as *Find Extreme* (+1.5%), *Compare* (+4.8%), and *Frequency* (+2.2%), where multi-step evidence management is most critical.

We also compare against tool-augmented variants of Thinking w/ Images. Interestingly, naively enabling code execution degrades performance (50.1%): without procedural guidance, the model often generates incorrect or irrelevant code. Providing a fixed skill list in the system prompt partially recovers accuracy (55.3%), but still falls short of HIERVA. This confirms that *how* tools and skills are managed—via dynamic routing and context distillation—is as important as having access to them.

**Efficiency and token usage.** We compare the *average peak token count*—the maximum number of tokens in the context window during an inference episode, averaged across all examples. As shown in the rightmost column of Table 3, **Thinking w/ Images** suffers from rapid context growth (approx. 3000 tokens) as it blindly appends visual and textual history. In contrast, HIERVA maintains a more compact working context (2568 tokens, ~14% reduction) despite executing complex tool-use chains.

This reduction confirms that our context distillation and encapsulation mechanisms effectively prune irrelevant context.

**Ablations.** We ablate the key components of HIERVA to quantify their contributions on advanced reasoning (Table 4). Specifically, we ablate four elements: **Hierarchical delegation** (manager-worker decomposition; §2.2), **context distillation** (keeping only refined plans and compact summaries; §2.3), **scoped context** (each worker sees a single relevant image view; §2.4), and **just-in-time skill routing** (injecting only task-relevant procedural skills; §2.5). The results show that **Hierarchical Delegation** is the most critical factor, with a 10.4% drop when removed, confirming that decomposing long-horizon tasks into manager-worker interactions is essential. **Context Distillation** also proved vital (8.7% drop), as appending full reasoning traces without summarization severely degrades performance on multi-step chains. **Just-in-time Skill Routing** (4.6% drop) and **Scoped Context** (3.1% drop) provide further complementary gains by keeping procedural guidance relevant and visual evidence focused.

## 5 Related Work

**Chart understanding and chart reasoning.** Chart QA couples fine-grained perception (ticks, legends, dense marks) with multi-step numeric reasoning. Recent work improves chart QA by converting plots into structured text for downstream language-model reasoning (Liu et al., 2023a), or by training chart-specialized vision–language models via derendering and chart-aware pretraining (Liu et al., 2023b; Masry et al., 2023). Large-scale chart instruction tuning further broadens coverage of real-world questions (Masry et al., 2024a; Meng et al., 2024; Masry et al., 2024b), and reasoning-oriented objectives target harder multi-step inference (Chen et al., 2025). Realistic evaluations still reveal substantial gaps for multimodal LLMs on scientific charts (Wang et al., 2024), motivating agentic approaches that decompose queries into visually grounded subtasks (Kaur et al., 2025). Our work complements these directions by focusing on inference-time orchestration—adaptive zoom, skill routing, and context distillation/encapsulation to control multimodal working context.

**Thinking with Images: Tool-Augmented Multimodal Reasoning.** Beyond charts, a broad line of work treats multimodal problem solving as an *agentic* loop in which an LLM plans, calls external vision modules/tools, and integrates observations (Wu et al., 2023a; Gupta and Kembhavi, 2023; Suris et al., 2023; Yang et al., 2023). Recent “thinking with images” paradigms emphasize multi-turn visual search behaviors—e.g., iteratively zooming into relevant regions to obtain high-resolution evidence when a global view is insufficient (Zheng et al., 2025; Lai et al., 2025; OpenAI, 2025b). In parallel, stronger multimodal backbones increasingly support higher-resolution understanding and longer multimodal contexts, partially reducing (but not eliminating) the need for explicit perception tooling (Liu et al., 2024; Bai et al., 2025). A practical challenge across tool-augmented multimodal agents is that observations (crops, OCR strings, intermediate tool outputs) are often appended into a single ever-growing transcript, which can reduce signal-to-noise over long trajectories. Our hierarchical visual agent targets this bottleneck directly: workers operate with *scoped* local contexts (including a single image view), while the manager only retains distilled, structured results—preventing monotonic growth of irrelevant intermediate text and redundant visual evidence.

**Planning and context management for LLM agents.** Our design also builds on progress in planning and context control for language agents. Chain-of-thought prompting improves multi-step reasoning by eliciting intermediate rationales (Wei et al., 2022), and ReAct-style frameworks interleave reasoning with actions to support tool use and environment interaction (Yao et al., 2022). Subsequent work explores explicit plan–execute decomposition (Wang et al., 2023), iterative self-improvement via reflection (Shinn et al., 2023), and search over candidate reasoning trajectories (Yao et al., 2023). Efficiency-oriented designs such as ReWOO decouple reasoning from observation interleaving to reduce prompt length and repeated computation (Xu et al., 2023), which is closely aligned with our goal of limiting unnecessary context growth during multi-step tool use. A further line studies long-horizon memory and working-context management, including externalized memory hierarchies and hierarchical working-memory mechanisms (Packer et al., 2023; Hu et al., 2024; Zhang et al., 2025), as well as decomposition via multi-agent/role-based collaboration (Wu et al., 2023b; Li et al., 2023). Prior methods are predominantly text-centric and treat observations as cheap to append. In contrast, chart reasoning requires controlling *joint image–text* context under tight budgets. Our approach extends hierarchical planning and memory ideas to multimodal settings by (i) enforcing worker encapsulation, (ii) treating zoomed visual evidence as a scoped resource, and (iii) distilling intermediate artifacts into compact manager-visible state.

## 6 Discussion: Beyond Charts

We choose chart reasoning as a testbed because it exposes the context-management bottleneck sharply, but the underlying principle of *scoped, distilled joint image–text context* is not chart-specific. The same manager–worker decomposition, zoom-as-action, and result distillation apply to any fine-grained visual reasoning setting where evidence is small, spatially scattered, and must be aggregated across steps: dense document VQA, multi-image reasoning over dashboards or slide decks, and scientific imagery (microscopy, medical, remote-sensing) all share this pathology. What would change across domains is mainly the concrete skill instances and tool hooks. The architectural ingredients, namely hierarchical delegation, plan/result

distillation, and scoped per-worker context, are domain-agnostic, and we view validating them in these broader settings as a natural next step.

## 7 Conclusion

We presented Hierarchical Visual Agent (HIERVA), a manager–worker framework for chart question answering that treats inference as *multimodal context management*, combining adaptive zoom-in, just-in-time skill routing, and strict distillation/encapsulation to keep the working context compact and high-signal. On CHARXIV, HIERVA achieves 64.2% accuracy, outperforming strong baselines by 1.8–5.3 points, with the largest gains on complex multi-step questions. We expect the principle of *scoped, distilled multimodal context* to extend beyond charts to other fine-grained visual reasoning domains where global views obscure critical details.

## 8 Limitations

Our approach inherits limitations from both the underlying multimodal model and the tool-augmented agent setting. If the backbone (or tool pipeline) cannot reliably perceive fine-grained chart evidence (ticks, legends, units, dense marks), hierarchical orchestration cannot recover missing facts, and zoom choices may amplify errors by focusing on the wrong region or omitting needed context. Decomposition can also be brittle for questions requiring global constraints or cross-panel alignment that do not cleanly factor into independent subtasks. Skill routing is limited by an incomplete library and imperfect selection, and scaling to broader skill sets with reliable tool use remains challenging. Finally, multi-step manager–worker execution increases latency and engineering complexity relative to single-pass prompting, and our evaluation (ChartQA, synthetic composites, CHARXIV) does not cover all real-world chart styles, languages, or document layouts.

## 9 Ethics Statement

This work focuses on inference-time orchestration for chart question answering and does not involve collection of new data or human subjects research. We evaluate on existing public benchmarks (CHARTQA, CHARXIV) that contain charts from publicly available sources. Potential positive impacts include improved accessibility for visually impaired users and more reliable scientific

document understanding. We acknowledge that, like all capable multimodal systems, our method could in principle be misused for extracting information from proprietary charts; however, the techniques we propose do not introduce new capabilities beyond those of the underlying vision–language model. Computational costs are moderate: our hierarchical design reduces peak context length relative to baselines with comparable tool access.

## 10 Use of Large Language Models in Writing

Large language models were used in a limited manner to assist with language polishing and improving clarity in the writing of this paper. All technical content, experimental design, analysis, and conclusions were conceived, verified, and finalized by the authors. The use of LLMs did not affect the scientific claims or results of the work.

## References

- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhi-fang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 2 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.
- Lei Chen, Xuanle Zhao, Zhixiong Zeng, Jing Huang, Yufeng Zhong, and Lin Ma. 2025. Chart-r1: Chain-of-thought supervision and reinforcement for advanced chart reasoner. *arXiv preprint arXiv:2507.15509*.
- Tanmay Gupta and Aniruddha Kembhavi. 2023. *Visual programming: Compositional visual reasoning without training*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14953–14962.
- Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. 2024. *Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model*. *arXiv preprint arXiv:2408.09559*.
- Amanpreet Kaur, Ahmed Masry, Enamul Hoque, and Shafiq Joty. 2025. *Chartagent: A multimodal agent for visually grounded reasoning in complex chart question answering*. *arXiv preprint arXiv:2501.09007*.
- Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. 2025. *Mini-o3: Scaling up reasoning patterns and interaction turns for visual search*. *arXiv preprint arXiv:2509.07969*.

- Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. *Camel: Communicative agents for 'mind' exploration of large language model society*. *arXiv preprint arXiv:2303.17760*. Accepted at NeurIPS 2023.
- Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhui Chen, Nigel Collier, and Yasemin Altun. 2023a. *DePlot: One-shot visual language reasoning by plot-to-table translation*. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10381–10399, Toronto, Canada. Association for Computational Linguistics.
- Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. 2023b. *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, Toronto, Canada. Association for Computational Linguistics.
- Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. *LlaVa-NeXT: Improved reasoning, ocr, and world knowledge*. LLaVA Blog.
- 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, Singapore. Association for Computational Linguistics.
- Ahmed Masry, Do Long, Jia Tan, Shafiq Joty, and Enamul Hoque. 2022. *Chartqa: A benchmark for question answering about charts with visual and logical reasoning*. In *International Conference on Learning Representations (ICLR)*.
- Ahmed Masry, Mehrad Shahmohammadi, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2024a. *ChartInstruct: Instruction tuning for chart comprehension and reasoning*. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10387–10409, Bangkok, Thailand. Association for Computational Linguistics.
- Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. 2024b. *Chartgemma: Visual instruction-tuning for chart reasoning in the wild*. *arXiv preprint arXiv:2407.04172*.
- 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, Bangkok, Thailand. Association for Computational Linguistics.
- OpenAI. 2025a. GPT-5 system card. <https://openai.com/index/gpt-5-system-card/>. Accessed: 2026-01-05.
- OpenAI. 2025b. *Openai o3 and o4-mini system card*. Technical report, OpenAI.
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. *Memgpt: Towards llms as operating systems*. *arXiv preprint arXiv:2310.08560*.
- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. *Reflexion: Language agents with verbal reinforcement learning*. *arXiv preprint arXiv:2303.11366*.
- Didac Suris, Sachit Menon, and Carl Vondrick. 2023. *Vipergpt: Visual inference via python execution for reasoning*. *arXiv preprint arXiv:2303.08128*.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. *Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models*. *arXiv preprint arXiv:2305.04091*.
- Zihan Wang, Ahmed Masry, Enamul Hoque, and Shafiq Joty. 2024. *Charxiv: Charting gaps in realistic chart understanding in multimodal llms*. *arXiv preprint arXiv:2406.18521*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. *Chain-of-thought prompting elicits reasoning in large language models*. *arXiv preprint arXiv:2201.11903*.
- Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023a. *Visual chatgpt: Talking, drawing and editing with visual foundation models*. *arXiv preprint arXiv:2303.04671*.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2023b. *Autogen: Enabling next-gen llm applications via multi-agent conversation*. *arXiv preprint arXiv:2308.08155*.
- Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. 2023. *Rewoo: Decoupling reasoning from observations for efficient augmented language models*. *arXiv preprint arXiv:2305.18323*.
- Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. *Mm-react: Prompting chatgpt for multimodal reasoning and action*. *arXiv preprint arXiv:2303.11381*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. [Tree of thoughts: Deliberate problem solving with large language models](#). *arXiv preprint arXiv:2305.10601*. NeurIPS 2023 camera-ready version.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. [React: Synergizing reasoning and acting in language models](#). *arXiv preprint arXiv:2210.03629*.

Alex L. Zhang, Tim Kraska, and Omar Khattab. 2025. [Recursive language models](#). *arXiv preprint arXiv:2512.24601*.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. 2025. [Deepeyes: Incentivizing “thinking with images” via reinforcement learning](#). *arXiv preprint arXiv:2505.14362*.

## A Detailed Prompting

This appendix provides the exact prompt suffixes used for each baseline method.

**Direct.** We append the instruction: “You cannot reason or think. You must ONLY answer directly in `\boxed{}`.”

**CoT (Chain-of-Thought).** We append the instruction: “Let’s reason step by step.”

**CoT-Plan (Plan-and-Solve).** We append the instruction: “Let’s first plan how to solve this problem, then answer step by step.”

### A.1 HIERVA Prompt Excerpts

We present shortened, paper-friendly excerpts of the manager prompts used by HIERVA. Full verbatim templates (including low-level formatting constraints) are omitted for readability.

**Listing 1: Planning (manager).** The manager requests a two-stage plan that decomposes the question into atomic, image-scoped sub-tasks, and optionally refines the plan with planning skills.

Listing 1: Manager planning prompt excerpt

```
For the question '{question}', we need to
develop a plan.
We can decompose the image and the question
into sub-images + sub-tasks pairs.
Since the question can be complex, we need
to plan twice.
The sub-image-task pair should be atomic.
Only plan now.
```

**Listing 2: Task creation (manager).** The manager specifies the next worker task by selecting the image input, writing a focused instruction, and declaring the expected output type. A review pass enforces a structured schema.

Listing 2: Task creation prompt excerpts

```
What is the next task? Provide for the
sub-agent: the image to use,
the instruction, and the expected output
type.

After reviewing, output in the following
format:
- Sub-agent ID: [ID]
- Image: [image index]
- Instruction: [instruction]
- Known Information: [necessary information
to finish the task]
- Tools: [image_zoom_in_tool,
code_interpreter, or None]
- Skills: [choose from the skill allowlist
or None]
- Expected Output: [image or text]
```

**Listing 3: Dispatch + termination.** Worker dispatch carries forward only distilled known information. Termination is decided by a yes/no check; final answers are standardized with a boxed wrapper.

Listing 3: Dispatch and termination prompt excerpts

```
We know the following information:
{known_information}.

Now let's assign the next task to the
sub-agent. First check if there is
next task. Answer only "Yes" or "No".
Put your final answer in \boxed{}
```

**Listing 4: Tool schemas (injected in system prompt).** The tool definitions are injected into the system prompt by the tool framework when a tool is enabled. Below we show the registered schema for the two tools used by HIERVA.

Listing 4: Tool prompt excerpts injected in the system prompt

```
Tool: image_zoom_in_tool
Description: Zoom in on a specific region of
an image by cropping it based on
a bounding box (bbox) and an
optional object label.
Parameters:
- bbox_2d: array[4] of numbers, [x1, y1,
x2, y2] with top-left and bottom-right
- label: string, name or label of the
object in the specified bbox
- img_idx: number, index of the input
image (starting from 0)

Tool: code_interpreter
```

Description: Python code sandbox, which can be used to execute Python code.  
Parameters:  
- code: string, the python code

## B Implementation Details

This appendix provides additional engineering details needed to reproduce our system. We specify the task schema used to communicate between manager and workers, our image indexing convention for chaining crops across steps, and the output contracts enforced for text and image tasks.

### B.1 Task Schema

Each subtask at step  $t$  is represented as

$$\tau_t = (k_t, i_t, u_t, m_t, \mathcal{T}_t, \mathcal{S}_t, y_t),$$

where  $k_t$  is the worker index,  $i_t$  selects the input image (original or a crop),  $u_t$  is the natural-language instruction,  $m_t$  is a compact memory summary (known information),  $\mathcal{T}_t$  is the set of enabled tools,  $\mathcal{S}_t$  is the set of injected skills, and  $y_t \in \{\text{TEXT}, \text{IMAGE}\}$  is the expected output type.

Manager-generated task specifications are validated against: (i) required fields (non-empty instruction, declared expected output), (ii) tool and skill allowlists, and (iii) image reference validity. If validation fails, we append a structured error message and re-prompt until a valid task is produced.

### B.2 Image Indexing Convention

To enable multi-step visual reasoning without accumulating redundant images:

- The original chart is referenced as image index 0.
- When a worker with id  $k$  produces a crop, that crop is stored as the *latest image of worker  $k$*  and is referenced by setting `image_index=  $k$`  in later tasks.

This mechanism allows chaining zooms and reusing regions across subtasks.

### B.3 Output Contracts

Workers produce outputs matching the expected type. For text tasks, we enforce a contract: the worker’s final answer must appear in the last sentence, which is extracted and stored. For image tasks, the output is a crop handle. Skills may optionally declare lightweight *hooks* for post-processing (e.g., coordinate normalization).