Thinking with Drafting: Optical Decompression via Logical Reconstruction

Anonymous ACL submission

Abstract

Existing multimodal large language models have achieved high-fidelity visual perception and exploratory visual generation. However, a precision paradox persists in complex reasoning tasks: optical perception systems transcribe symbols without capturing logical topology, while pixel-based generative models produce visual artifacts lacking mathematical exactness. To bridge this gap, we propose that reasoning over visual inputs be reconceptualized as *optical decompression*—the process of reconstructing latent logical structures from compressed visual tokens. Guided by the axiom that *Parsing is Reasoning*, we introduce Thinking with Drafting (TwD), which utilizes a minimalist Domain-Specific Language (DSL) as a grounding intermediate representation. Unlike standard approaches that hallucinate answers directly, TwD forces the model to draft its mental model into executable code, rendering deterministic visual proofs for self-verification. To validate this, we present VisAlg, a visual algebra benchmark. Experiments demonstrate that TwD serve as a superior cognitive scaffold. Our work establishes a closed-loop system where visual generation acts not as a creative output but as a logical verifier, offering a generalizable path for visual reasoning.

1 Introduction

Recent advances in multimodal large language models (MLLMs) mark a decisive shift in artificial intelligence from passive perception toward active cognitive interaction (Huang et al., 2023; Alayrac et al., 2022; Xue et al., 2024; Liu et al., 2023). On the input side, **optical character recognition (OCR)** systems have undergone a dramatic evolution. Modern approaches—exemplified by large-scale vision-language models trained for document understanding—are now capable of faithfully transcribing complex visual artifacts (Kim et al., 2022; Wang et al., 2024; Cui et al., 2025; Tang et al., 2023), including dense text, structured layouts, tables, and mathematical formulas. This progress effectively realizes what may be termed contextual optical compression (Wei et al., 2025): rich visual documents are compressed into high-fidelity internal representations, enabling machines to read with unprecedented accuracy.

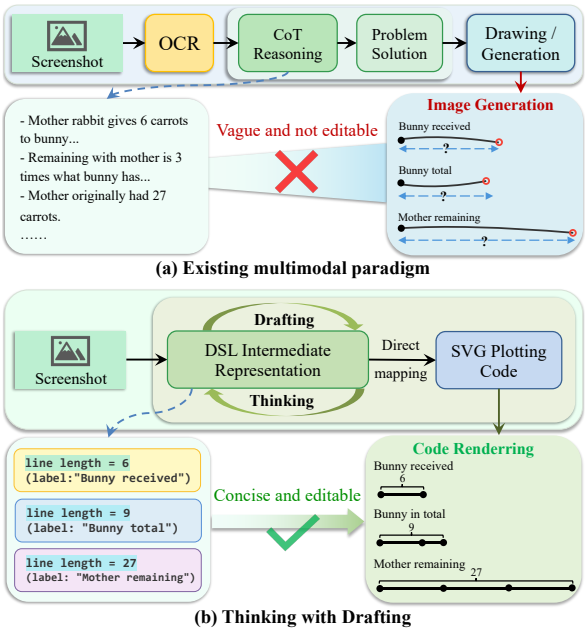


Figure 1: Illustration of paradigms. (a) Existing multimodal paradigms treat image understanding, textual reasoning, and visual generation as disconnected tasks. (b) Thinking with Drafting (TwD) reframes visual reasoning as logical reconstruction into a minimalist DSL.

Concurrently, progress on the output side has given rise to a complementary paradigm often referred to as **Thinking with images** (Su et al., 2025; Chern et al., 2025). Rather than relying solely on a textual chain-of-thought, recent models increasingly generate visual artifacts like diagrams, sketches, or intermediate images as part of the reasoning process. By externalizing cognition into visual form, these methods aim to mirror a fundamental aspect of human problem solving, where drawing and visualization serve as tools

for thought (Zheng et al., 2025; Qiao et al., 2025). Taken together, these trends suggest that modern systems are approaching a read–draw loop (Lu et al., 2023; Shen et al., 2023), with perception supplying faithful inputs and generation enabling visualized intermediate states.

Despite this apparent completeness, a critical gap remains when these systems are applied to tasks requiring strict logical precision. This gap manifests as a precision paradox. On the one hand, OCR systems excel at transcription: they can reliably extract symbols, numbers, and text spans from images. However, transcription alone does not capture logical topology. A numeral such as "123" may represent a total, a difference, or a constraint, depending on context. While the perceptual signal is high-fidelity, the relational semantics remain implicit and unstructured. OCR systems are designed to recognize symbols, represent the logical relations that govern them. On the other hand, visual generation models optimize perceptual plausibility rather than logical validity (Yao et al., 2022). They can generate images that resemble diagrams or mathematical constructions without guaranteeing that the underlying relations are exact. A generated line segment may appear longer than another, yet fail to satisfy a precise quantitative ratio.

To bridge this divide, we argue that reasoning over visual inputs must be reconceptualized as a process of *optical decompression* (Schick et al., 2023; Hsu et al., 2023). If OCR compresses the visual world into perceptual tokens, then reasoning is the act of reconstructing the latent logical structure encoded within those tokens. From this perspective, understanding does not hinge on producing fluent textual explanations, but on recovering an explicit, executable representation of entities, relations, and constraints. This leads to our central axiom: **Parsing is Reasoning**. True comprehension arises only when a model can translate ambiguous natural language and visual cues into a structured form.

We materialize this philosophy through the **Thinking with Drafting (TwD)** paradigm. Taking the Singapore bar model—a canonical representation of visual algebra—as our primary testbed, we introduce a minimalist geometric DSL (Domain-Specific Language). This DSL occupies a unique strategic niche: it serves as an intermediary between the ambiguity of natural language, the syntactic noise of general-purpose code, and the rigidity of geometric axioms. This DSL is designed for interoperability; it can be compiled into GeoGebra scripts for mathematical validation or SVG code for visual rendering. The generated draft serves not merely as a visualization, but as a deterministic visual verifier, enabling the system to detect logical conflicts and self-correct. Within TwD, drafting is not treated as a final output but as a *deterministic visual verifier*, enabling a closed logical–visual loop in which reconstruction, verification, and correction are tightly coupled.

2 Related Work

2.1 Optical Perception

Recent advancements in Optical Character Recognition (OCR) (Wang et al., 2024; Li et al., 2025b; Cui et al., 2025) and Vision Language Models (VLMs) (Hurst et al., 2024; Comanici et al., 2025; Bai et al., 2025c; Yang et al., 2025a) have fundamentally transformed the landscape of document understanding. Traditional approaches have evolved to recover high-fidelity text content while preserving complex contextual structures such as layouts, tables, and formulas (Zhang et al., 2025; Chumachenko et al., 2025). Notably, recent works like DeepSeek-OCR (Wei et al., 2025) demonstrate the feasibility of contexts optical compression, proving that pixels can serve as an efficient compression medium for textual information.

However, optical perception alone is insufficient for tasks that require rigorous logical consistency like mathematical problem solving (Gupta and Kembhavi, 2023; Surís et al., 2023; Lu et al., 2021). Current unstructured outputs may capture the document's visual syntax but neglect its underlying logic, leaving entities and quantitative relations implicit and ungrounded. We argue that reliable reasoning requires a shift from transcription accuracy to logical reconstruction. Unlike standard perception tasks, our approach transforms raw perception into a verifiable intermediate representation, thereby enabling the *Thinking with Drafting* paradigm to operate on grounded logical structures.

2.2 Visual Reasoning

While optical perception digitizes the input, reasoning requires manipulating digitized concepts to derive solutions. The dominant paradigm relies on LLMs to perform reasoning via textual generation, exemplified by Chain-of-Thought (CoT) (Wei et al., 2022; Kojima et al., 2022) and Program-of-Thought (PoT) (Chen et al.; Gao et al., 2023). These methods decompose complex problems into

step-by-step deductions or executable code snippets. Conversely, vision-centric approaches attempt to solve reasoning tasks directly in the pixel space (Chen et al., 2025; Yang et al., 2025b; Zhang et al., 2023). Recent works such as VisionARC (Hu et al., 2025) demonstrate that certain abstract reasoning tasks are more naturally formulated as image-to-image translation problems.

Despite their efficacy, these models often struggle with *semantic grounding*—specifically, translating complex natural language constraints into geometric artifacts. We propose *Thinking with Drafting* to bridge the gap between implicit semantic thought and explicit visual verification. By parsing visual text into a structured intermediate representation, the model drafts its understanding into a rule-constrained canvas. It creates an *optical decompression* loop: implicit logical relations are decompressed into explicit visual structures.

3 Method

3.1 Preliminaries

We consider the problem of multimodal mathematical reasoning where a model is presented with a visual input $\mathcal{I}$ (containing visual text, layout, and geometry) and a natural language query $\mathcal{Q}$. The objective is to derive a correct final answer $a \in \mathcal{A}$. Unlike standard end-to-end approaches that map $(\mathcal{I}, \mathcal{Q}) \rightarrow a$ directly, we formalize *Thinking with Drafting* as a multi-stage iterative generation process involving a structured intermediate representation. Let $\mathcal{T}$ denotes the space of unstructured natural language and $\mathcal{S}$ denotes the space of DSL, which represents geometric and logical constraints. We define the reasoning process P_θ, parameterized by a MLLM, as a probabilistic mapping from perception to logical reconstruction.

To underscore the theoretical distinctiveness of TwD, we contrast our formulation with three dominant paradigms: text-only CoT, thinking with images, and traditional OCR.

Text-only CoT Standard Multimodal CoT approaches rely exclusively on the linguistic space $\mathcal{T}$ to bridge the input and output:

$$\hat{t}_{cot} \sim P_\theta(t, \mathcal{I}, \mathcal{Q},),\ \hat{a} \sim P_\theta(a|\mathcal{I}, \mathcal{Q}, \hat{t}_{cot}), \quad (1)$$

where $\hat{t}_{cot} \in \mathcal{T}$ is a linear sequence of tokens. The fundamental limitation of CoT is that natural language is ambiguous and lacks strict geometric constraints.In contrast, our DSL space $\mathcal{S}$ enforces logical rigidity; a defined entity in $\mathcal{S}$ must satisfy explicit geometric rules, acting as a regularizer for the reasoning process.

Thinking with Images Emerging "Thinking with Images" paradigms utilize a generative model to produce an intermediate image $\hat{\mathcal{I}}_{\text{gen}}$:

$$\hat{a} \sim P_\theta(a \mid \mathcal{I}, \mathcal{Q}, \hat{\mathcal{I}}_{\text{gen}}) \quad (2)$$

While $\hat{\mathcal{I}}_{\text{gen}}$ provides visual feedback, it operates in the pixel space, which suffers from stochastic imprecision. A model may generate a diagram that *perceptually* plausible but *mathematically* inaccurate. TwD, conversely, employs programmatic drafting. Our intermediate representation $\hat{s}$ is symbolic code. The rendered output is mathematically exact, ensuring reliable verification.

OCR focuses on transcription fidelity, mapping the visual input to a sequence of characters:

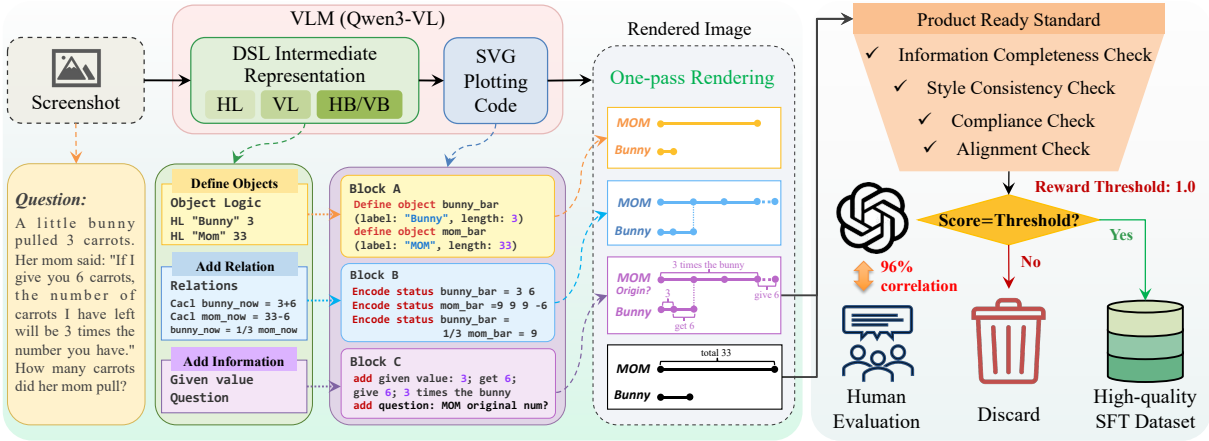
$$\text{Seq} \sim P_\theta(\text{Seq} \mid \mathcal{I}), \quad (3)$$

OCR addresses the question "*What is written?*", whereas TwD addresses "*What does it mean?*". OCR extracts the *syntax* but leaves the *semantics* implicit. TwD performs logical reconstruction, upgrading the task from transcription to parsing. By mapping $\mathcal{I} \rightarrow \mathcal{S}$, we explicitly capture the logical topology that OCR ignores, thereby converting raw pixels into actionable reasoning primitives.

3.2 The Logic Graphic DSL

To instantiate the principle that *Parsing is Reasoning*, we formally define the structure of our DSL space, $\mathcal{S}$. A statement $s \in \mathcal{S}$ is not a sequence of natural language tokens, but a structured composition of atomic reasoning primitives. Unlike general-purpose plotting languages that prioritize pixel-level control, the grammar of $\mathcal{S}$ is designed to abstract away rendering redundancies and expose the bare logical topology of the problem. The DSL consists of three fundamental operator categories:

Entity Primitives (HL) These represent the physical quantities or objects from the input $\mathcal{I}$ as horizontal line segments. A key innovation in our design is the status-aware segmentation. We define a segment sequence vector $\mathbf{v} = [v_1, v_2, ..., v_n]$ where $|v_i|$ denotes length. Crucially, we utilize the sign of v_i to encode *existential status*: $v_i > 0$ renders a solid line (existing quantity), while $v_i < 0$ renders a dashed line (process quantity, e.g., subtracted part or hypothetical extension). This allows



(a) Thinking with Drafting (Generation) **(b) Data pipeline with DSL Verifier**

Figure 2: Overview of Thinking with Drafting framework. (a) Optical decompression generates a Logic Graphic DSL from visual input and OCR, comprising entity, relational, and aggregation primitives. (b) A verifier scores samples by syntactic validity, visual completeness, and logical consistency, retaining high-quality data for training and discarding the rest to ensure topological and geometric correctness.

the model P_θ to generate a compact representation for complex change models.

Relational Primitives (VL) In bar models, logic is primarily defined by geometric alignment. The Vertical Line (VL) operator explicitly encodes relational equality between horizontal entities. Parameterized by an explicit x-coordinate and row indices, it functions as an equality constraint, asserting that specified segments coincide at a shared value. This compels the model to perform *alignment reasoning*, identifying shared semantic boundaries rather than treating coordinates as independent variables.

Aggregation Primitives (HB/VB) To ground abstract arithmetic operations into geometry, we employ Horizontal (HB) and Vertical (VB) Braces. An HB operator encapsulates a part-whole relationship within a single entity, while a VB represents summation or comparison across multiple entities.

3.3 Topological Abstraction and Rendering

A major bottleneck in generating visual code is the high entropy of continuous coordinate spaces. To mitigate this, we introduce a Topological Abstraction layer that decouples logical reasoning from metric rendering.

Virtual Grid System We map the continuous canvas $\mathbb{R}^2$ to a discrete logic space $\mathbb{Z}^2$. We define a virtual grid where the y-axis is discretized into logical rows and the x-axis is governed by relative offsets rather than absolute pixels. The model generates code relative to this grid. For instance, creating a new entity involves assigning it to a new row_id rather than calculating a pixel offset. It ensures layout invariance: the model focuses solely on the logical ordering and grouping of entities.

Deterministic Rendering The mapping from a syntactically correct DSL statement to a visual verification image $\mathcal{V}$ is executed by a deterministic rendering engine: $\mathcal{V} = \text{Render}(s)$. We introduce common topological patterns into semantic macros. For example, a *comparison pattern* macro automatically generates the difference brace and alignment lines when the model detects a "more than" relation. These macros ensure that correct logical parsing always yields a visually canonical diagram.

3.4 Thinking with Drafting

Building upon the structured space $\mathcal{S}$ and the deterministic renderer, we instantiate the TwD framework in Figure 2 as a sequential generation-verification process.

Optical Decompression via Logical Parsing In the first stage, the model acts as a parser. It perceives the raw input $\mathcal{I}$ and attempts to decompress the implicit logical topology into an explicit structural draft. This yields a preliminary textual explanation $\hat{t}$ and an initial draft $\hat{s}$:

$$(\hat{t}, \hat{s}) \sim P_\theta(t, s|\mathcal{I}, \mathcal{Q}), \tag{4}$$

Crucially, the generation of $\hat{s}$ is not a single step, but a step-by-step decomposition of the problem. It embodies our axiom that *Parsing is Reasoning*: the generation of s forces the model to resolve ambiguities in $\mathcal{I}$ into discrete logical atoms.

Drafting and DSL-Conditioned Inference The generated hypothesis $\hat{s}$ is passed to the rendering engine to produce the verification drafting image $\mathcal{V}$. It provides an explicit visual proof of the model's internal reasoning for human verification. The model derives the solution conditioned on the structured draft $\hat{s}$ it constructed. Unlike standard Chain-of-Thought which relies on ambiguous natural language, the drafting $\hat{s}$ acts as a logical context:

In the second stage, the model utilizes the output from the previous stage as a "drafting context." The initial draft $\hat{s}_1$ acts as an externalized cognitive scaffold, allowing the model to inspect its own reasoning. The model generates a refined explanation $\hat{t}_2$, a completed DSL $\hat{s}_2$, and the final answer $\hat{a}$:

$$\hat{a} \sim P_\theta(t, s, a | \mathcal{I}, \mathcal{Q}, \hat{t}, \hat{s}). \tag{5}$$

By grounding the reasoning in $\hat{s}$, the calculations are guided by the explicit topology defined in the draft. The TwD paradigm thus posits that the *act of constructing* the draft is the reasoning engine itself, ensuring the final answer is a derivative of a verified logical structure.

4 Dataset

We introduce VisAlg, a benchmark for evaluating logic-aware visual reasoning by assessing whether a system can recover the explicit logical topology underlying visual algebra problems through *optical decompression*. Each instance pairs a image of a natural language algebra problem with a structured intermediate representation, an executable bar-model DSL that defines the ground-truth logical parse. VisAlg is constructed through a multi-stage pipeline, as shown in Figure 3.

4.1 Dataset Construction

Drafting data generation We collect 15,000 bar-model word problems from public datasets and websites, covering common visual algebra patterns. For each problem, we prompt Gemini-2.5-Pro (Comanici et al., 2025) to produce a synchronized draft with two components. The first component is a textual analysis that explicitly parses the problem schema, and the second component is a program written in our DSL. The detailed prompt is provided in Appendix A.1.

Data refining Initial drafts frequently fail to meet verifiability requirements. We therefore introduce a checklist refinement stage in which the model revises each draft through three sequential checks: (1) *Syntax check*, ensuring the grammar is correct and executable; (2) *Analysis check*, verifying that all objects, quantities, relations, and targets identified in the analysis are consistently instantiated; (3) *Style check*, enforcing canonical bar-model layout conventions such as boundary placement and cross-row alignment. All corrected instances are stored. The detailed prompt is in Appendix A.2.

Scoring and filtering We employ an LLM-based judge calibrated with expert evaluations to filter the dataset. A domain expert scores 1,000 instances using a fixed rubric, and the judge prompt is iteratively refined until achieving 96% agreement. The calibrated judge is then applied to the full dataset, retaining only full-score instances, resulting in 11,372 product-ready instances. Details are provided in Appendix A.3 and A.4.

Product ready The final filtering enforces four criteria: geometric alignment, semantic completeness, representational compliance, and stylistic consistency. (1) *Alignment*: Horizontal bracket endpoints must coincide with boundary coordinates defined by cumulative segment lengths; vertical links must align with these boundaries across spanned rows. (2) *Completeness*: All stated quantities, relations, and targets must appear explicitly in labels; unknowns may be denoted by "?". (3) *Compliance*: Vertical brackets represent only multi-object aggregates, and vertical links are allowed solely at cross-row shared partition points. (4) *Consistency*: Transfers follow a paired $-t/+t$ pattern across two rows; post-transfer equality is indicated by a shared boundary via a vertical link.

4.2 Dataset Analysis

Category VisAlg focuses on optical decompression in bar-model reasoning, emphasizing recovery of logical topology over surface symbol transcription. We analyze five canonical schemas: proportional distribution, rate & percentage, change & revert, sum & split, and difference analysis. Figure 4 shows the joint composition of difficulty (inner ring) and schema (outer ring). As summarized in Table 1, proportional distribution and rate & percentage form the two dominant schema groups. In terms of difficulty, medium accounts for 72.9% of instances, with easy (13.4%) and hard (13.7%) providing balanced coverage of both basic parsing and constraint-dense cases.

Reasoning depth. Logical complexity is measured by the number of bar-model operations needed for reconstruction. As shown in Table 1, only 17.6% of

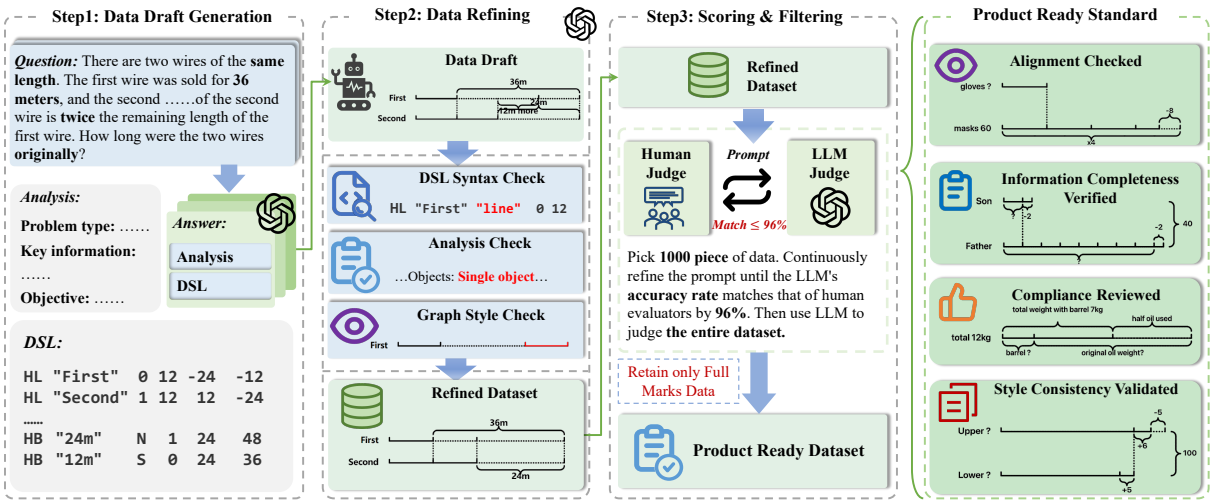


Figure 3: The benchmark data construction pipeline of VisAlg.

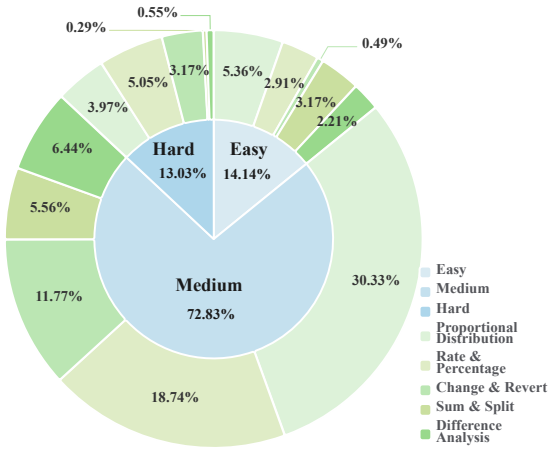


Figure 4: Difficulty and schema composition in VisAlg.

Table 1: Corpus-level statistics of VisAlg.

	Train	Test
Problem schemas		
Proportional Distribution	4,265	245
Rate & Percentage	2,771	265
Change & Revert	1,635	119
Difference Analysis	905	141
Sum & Split	854	172
Difficulty levels		
Easy	1,400	208
Medium	7,602	680
Hard	1,428	54
Operation length		
≤ 3 operations	1,834	163
4 operations	2,490	279
5 operations	2,693	296
6 operations	1,612	115
7 operations	1,002	56
≥ 8 operations	799	33
Total instances	**10,430**	**942**

training instances require three or fewer operations, while most fall in the four–six range. A non-trivial 7.7% require eight or more operations, reflecting long dependency chains and multi-step reasoning.

Scale and split consistency. The benchmark comprises 10,430 training and 942 test instances, with additional curated splits for fine-tuning, preference optimization, and evaluation. The test set mirrors the training distribution in schema and difficulty, ensuring evaluation emphasizes structural generalization rather than distributional shift.

4.3 Evaluation Metrics

Objective metrics Consistency is evaluated at both code and image levels. Code similarity is measured using BLEU, ROUGE-L, and chrF, with chrF as the primary metric due to its robustness to mixed symbols, numbers, and text in the DSL. Image similarity is assessed using PSNR, SSIM, and LPIPS, with SSIM prioritized for its sensitivity to structural topology and edge continuity.

Subjective metrics via LLM-as-judge An LLM-based verifier scores outputs on five dimensions: structural alignment, information coverage, numerical consistency, semantic compliance, and answer leakage. Each is rated in $[0, 1]$, with the final subjective score given by their mean.

Main score Main results are reported using a composite score: $\text{Score} = \frac{1}{3}(\text{chrF} + \text{SSIM} + \text{LLM}_{\text{judge}})$, which jointly reflects code-level consistency, image-level structural fidelity, and semantic normative correctness.

Human evaluation We additionally conduct a human evaluation of DSL quality; the criteria and protocol are provided in Appendix B.

Table 2: Main results on VisAlg. Align: structural alignment; Cover: information coverage; Num: numerical consistency; Norm: semantic compliance; Leak: answer leakage. Detailed descriptions of them are in Appendix B.3.

Model	Code Similarity			Image Similarity			Verification Scores						Overall
	BLEU	ROUGE-L	chrF	LPIPS	SSIM	PSNR	Align	Cover	Num	Norm	Leak	Avg.	
InternVL3-8B (Zhu et al., 2025)	9.93	48.51	37.57	32.64	82.70	24.25	0.31	0.56	0.32	0.15	0.89	44.69	54.99
InternVL2.5-8B (Chen et al., 2024)	9.12	46.38	48.41	51.10	58.97	17.22	0.32	0.44	0.36	0.14	0.68	38.73	48.70
Intern-S1-mini (Bai et al., 2025a)	8.41	36.47	22.68	59.52	49.32	14.65	0.57	0.26	0.87	0.36	0.97	60.39	44.13
Mimo-VL-7B-RL (Li et al., 2025a)	10.36	46.17	33.43	79.47	25.87	7.75	0.38	0.48	0.76	0.23	0.85	54.05	37.78
Qwen3-VL-8B (Bai et al., 2025b)	6.65	39.04	23.94	83.69	20.10	0.00	0.60	0.16	0.84	0.29	**1.00**	57.80	33.95
Gemini-3-Pro (Team et al., 2023)	30.18	59.06	57.53	18.23	90.36	27.32	**0.97**	0.95	0.94	**0.78**	0.96	**91.98**	79.96
Gemini-2.5-Pro (Comanici et al., 2025)	28.94	58.25	57.43	18.12	89.97	26.92	0.95	0.76	**0.99**	0.73	0.32	74.97	74.12
Claude-4 (Anthropic, 2025)	28.66	58.54	57.17	18.16	89.97	26.88	0.94	0.74	0.99	0.72	0.30	73.71	73.62
GPT-5.1 (Achiam et al., 2023)	22.93	56.13	51.23	25.86	86.89	25.70	0.77	0.67	0.95	0.51	0.19	61.69	66.60
GPT-4o (Hurst et al., 2024)	16.24	50.64	35.73	24.49	89.15	26.59	0.50	0.42	0.83	0.31	0.72	55.44	60.11
TwD (Ours)	**48.23**	**72.22**	**68.29**	**11.97**	**93.68**	**30.25**	0.90	**0.96**	0.70	0.73	1.00	85.91	**82.63**

5 Experiment

5.1 Experimental Setup

We evaluate VisAlg against state-of-the-art MLLMs. The proprietary models include GPT-5.1 (Achiam et al., 2023), GPT-4o (Hurst et al., 2024), Claude-4 (Anthropic, 2025), Gemini-3 (Team et al., 2023), and Gemini-2.5-Pro (Comanici et al., 2025), representing the current upper bound of general-purpose multimodal reasoning. For open-weight baselines, we consider InternVL3-8B (Zhu et al., 2025), InternVL2.5-8B (Chen et al., 2024), Intern-S1-mini (Bai et al., 2025a), Mimo-VL-7B-RL (Li et al., 2025a), and Qwen3-VL-8B (Bai et al., 2025b). Our model is initialized from Qwen3-VL-8B and supervised fine-tuned on the training split, enabling parameter-efficient comparison with open-weight peers while treating proprietary models as upper bounds. SFT is conducted on a 8-GPU node with a visual token cap of 2,048 and a maximum sequence length of 5,128. We train for 2 epochs using a learning rate of 5×10^{-6} and a warmup ratio of 0.05.

5.2 Main Results

Table 2 reports the main results on VisAlg across code similarity, image similarity, and verifier-based evaluation. Our model, initialized from Qwen3-VL-8B and supervised on VisAlg, achieves the highest overall score of 82.63, surpassing all open-weight baselines and outperforming the strongest proprietary models, including Gemini-3-Pro (Team et al., 2023) (79.96) and Gemini-2.5-Pro (Comanici et al., 2025) (74.12). This highlights the importance of explicit supervision on logic reconstruction for verifiable bar-model reasoning. A clear performance gap is observed between open-weight and proprietary systems. Open-weight models such as InternVL3-8B (Zhu et al., 2025), InternVL2.5-8B (Chen et al., 2024), Intern-S1-mini (Bai et al., 2025a), Mimo-VL-7B-RL (Li et al., 2025a), and Qwen3-VL-8B (Bai et al., 2025b) score below 55, with weaknesses in code fidelity and diagram reconstruction, indicating difficulty in generating syntactically valid and topologically consistent DSL programs without task-specific alignment.

Our gains primarily arise from improved structural fidelity. The model leads in code and diagram alignment, achieves strong information coverage, and avoids answer leakage. Relative to top proprietary models, the remaining gap is mainly in numerical consistency, while structural legality and semantic completeness are largely preserved.

5.3 Results by Visual Algebra Schema

Figure 5 reports schema-wise performance across five visual algebra types. Our TwD consistently achieve competitive performance compared to both open-weight and proprietary baselines across all schemas. The gains are most pronounced on structure-intensive schemas such as *proportional distribution* and *difference analysis*, where accurate multi-segment decomposition and boundary-aligned comparison are critical. While proprietary models achieve competitive results, their performance varies noticeably across schemas. In contrast, TwD remains uniformly strong across problem types, supporting the claim that optical decompression benefits from explicit, verifiable logic.

5.4 Alignment with Human Expert

Figure 6 shows a strong correlation between expert human ratings and verifier-based VisAlg scores $r = 0.9575$, validating the verifier as a reliable proxy for human judgment in visual algebra reasoning. Model rankings are largely preserved across the

Figure 5: Schema-wise performance comparison across five visual algebra problem types.

full performance range. TwD remains top-ranked under both evaluations, indicating that the reported gains reflect genuine improvements in structural correctness rather than metric artifacts.

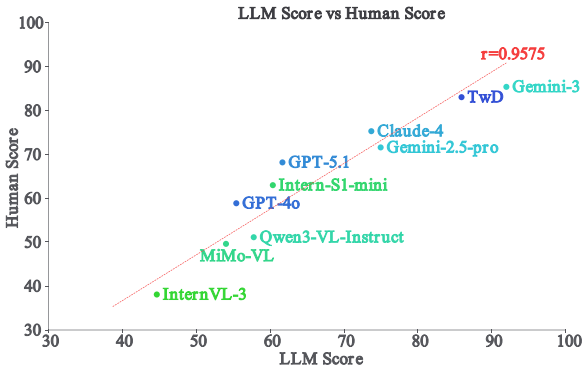


Figure 6: Correlation between verifier-based VisAlg scores and human expert ratings.

5.5 Generalize to Complex Logical Topology

We extend our evaluation to advanced set-theoretic reasoning tasks involving multi-set constraints. As shown in Figure 7, these tasks require the model to manage high-order intersections and nested boolean boundaries. Frontier MLLMs like GPT-5 (Achiam et al., 2023) often exhibit topological hallucination in this regime. While they may attempt to align segments visually, they fail to preserve the strict boolean logic of overlaps. The model cannot distinctively ground intersections $A \cap C$ and $A \cap B \cap C$, violating containment and alignment constraints and rendering the graphic unreadable and unverifiable. This *calculation–construction gap* highlights that correct arithmetic does not guarantee preservation of global structural invariants such as boundary legality and consistency. TwD successfully decomposes the abstract set problem into sequential geometric operations. By explicitly rendering the atomic intersections, TwD effectively visualizes the algebra of sets. Additional case studies are provided in Appendix D.

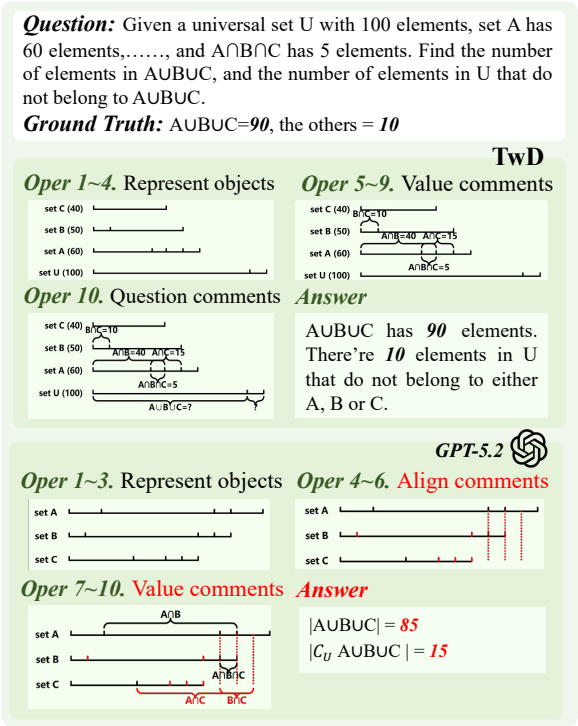


Figure 7: Generalization to set-theoretic reasoning.

6 Conclusion

In this work, we addressed the precision paradox in multimodal reasoning, where systems achieve high perceptual fidelity yet fail to preserve rigorous logical topology. We formalized this challenge through the lens of *optical decompression*, introducing VisAlg benchmark to evaluate whether models can reconstruct latent logical structures into verifiable artifacts. To bridge the gap between perception and reasoning, we established Thinking with Drafting paradigm that enforces structural invariants via a minimalist graphic DSL. Experiments demonstrate that that a compact 8B model, when equipped with the TwD cognitive scaffold, outperforms leading proprietary frontiers on visual algebra problems. By closing this loop, we show that explicit structural drafting acts as a necessary foundation for trustworthy multimodal intelligence.

Limitations

The core limitation lies in the scope of structural representation: the DSL is intentionally designed around bar-model visual algebra, emphasizing linear topological relations to enable intuitive structural supervision. Extending this DSL to support broader classes of scientific diagrams remains an important direction for future research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Anthropic. 2025. Claude sonnet 4.5 system card. Technical report, Anthropic PBC. Official system card describing Claude Sonnet 4.5 capabilities and safety evaluation. Available at: https://assets.anthropic.com/m/12f214efcc2f457a/original/Claude-Sonnet-4-5-System-Card.pdf.

Lei Bai, Zhongrui Cai, Yuhang Cao, Maosong Cao, Weihan Cao, Chiyu Chen, Haojiong Chen, Kai Chen, Pengcheng Chen, Ying Chen, and 1 others. 2025a. Intern-s1: A scientific multimodal foundation model. *arXiv preprint arXiv:2508.15763*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025b. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025c. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Yang Chen, Yufan Shen, Wenxuan Huang, Sheng Zhou, Qunshu Lin, Xinyu Cai, Zhi Yu, Jiajun Bu, Botian Shi, and Yu Qiao. 2025. Learning only with images: Visual reinforcement learning with reasoning, rendering, and visual feedback. *arXiv preprint arXiv:2507.20766*.

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

Ethan Chern, Zhulin Hu, Steffi Chern, Siqi Kou, Jiadi Su, Yan Ma, Zhijie Deng, and Pengfei Liu. 2025. Thinking with generated images. *arXiv preprint arXiv:2505.22525*.

Kateryna Chumachenko, Amala Sanjay Deshmukh, Jarno Seppanen, Ilia Karmanov, Chia-Chih Chen, Lukas Voegtle, Philipp Fischer, Marek Wawrzos, Saeid Motiian, Roman Ageev, and 1 others. 2025. Nvidia nemotron parse 1.1. *arXiv preprint arXiv:2511.20478*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, and 1 others. 2025. Paddleocr-vl: Boosting multilingual document parsing via a 0.9 b ultra-compact vision-language model. *arXiv preprint arXiv:2510.14528*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

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*, pages 14953–14962.

Joy Hsu, Jiayuan Mao, and Jiajun Wu. 2023. Ns3d: Neuro-symbolic grounding of 3d objects and relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2614–2623.

Keya Hu, Ali Cy, Linlu Qiu, Xiaoman Delores Ding, Runqian Wang, Yeyin Eva Zhu, Jacob Andreas, and Kaiming He. 2025. Arc is a vision problem! *arXiv preprint arXiv:2511.14761*.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, and 1 others. 2023. Language is not all you need: Aligning perception with language models. *Advances in*

Neural Information Processing Systems, 36:72096–72109.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Jiaze Li, Jingyang Chen, Yuxun Qu, Jianzhong Ju, Zhenbo Luo, Jian Luan, Shijie Xu, Zhenru Lin, Junyou Zhu, Boshen Xu, and 1 others. 2025a. Xiaomi mimo-vl-miloco technical report. *arXiv preprint arXiv:2512.17436*.

Zhang Li, Yuliang Liu, Qiang Liu, Zhiyin Ma, Ziyang Zhang, Shuo Zhang, Zidun Guo, Jiarui Zhang, Xinyu Wang, and Xiang Bai. 2025b. Monkeyocr: Document parsing with a structure-recognition-relation triplet paradigm. *arXiv preprint arXiv:2506.05218*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36:43447–43478.

Runqi Qiao, Qiuna Tan, Minghan Yang, Guanting Dong, Peiqing Yang, Shiqiang Lang, Enhui Wan, Xiaowan Wang, Yida Xu, Lan Yang, and 1 others. 2025. V-thinker: Interactive thinking with images. *arXiv preprint arXiv:2511.04460*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, and 1 others. 2025. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11888–11898.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19254–19264.

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*.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, and 1 others. 2024. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*.

Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.

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.

Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, and 1 others. 2024. xgen-mm (blip-3): A family of open large multimodal models. *arXiv preprint arXiv:2408.08872*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. 2025b. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Qintong Zhang, Junyuan Zhang, Zhifei Ren, Linke Ouyang, Zichen Wen, Junbo Niu, Yuan Qu, Bin Wang, Ka-Ho Chow, Conghui He, and 1 others. 2025. Docr-inspector: Fine-grained and automated evaluation of document parsing with vlm. *arXiv preprint arXiv:2512.10619*.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

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*.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Additional Details for Dataset Construction

A.1 Prompt for Data Draft Generation

This subsection presents the prompt used in Step 1 (Data Draft Generation) of the VisAlg construction pipeline. The prompt elicits a synchronized draft consisting of structured problem analysis, diagram planning under strict bar-model constraints, and an initial executable DSL program. This stage establishes the logical and visual foundation for subsequent refinement and verification. The complete prompt is provided in Figure 8.

A.2 Prompt for Data Refining

This subsection presents the prompt used in Step 2 of the VisAlg construction pipeline. Given an initial problem analysis and a draft DSL generated in the previous stage, this prompt instructs the model to perform checklist-driven verification and conditional refinement. The objective is to determine whether the draft is *product-ready*, and if not, to apply minimal, targeted corrections. The full checklist-driven refinement prompt is shown in Figure 9.

A.3 Prompt for Scoring and Filtering

This subsection presents the prompt used in the final stage of the VisAlg construction pipeline. Given a refined DSL draft produced after checklist-driven revision, this prompt instructs an LLM-based verifier to perform strict, criteria-based scoring. Only instances receiving a full score are retained as *product-ready* samples in the final dataset. The scoring prompt used for LLM-based verification is shown in Figure 10.

A.4 Human Expert Evaluation Criteria

In addition to automated LLM-based verification, we perform human expert screening on all refined instances. Human evaluators assess each DSL using the same zero-tolerance philosophy, serving as the final gatekeeper for dataset inclusion.

An instance is accepted into the final dataset **only if all of the following conditions are satisfied**:

(1) **Numerical validity**: all bar-segment lengths correspond to valid quantities in the correct solution process, without arbitrary scaling or distortion.

(2) **Information sufficiency**: the rendered diagram alone, based on visible annotations, is sufficient to solve the problem without consulting the original text.

(3) **Alignment accuracy**: all brackets and alignment markers precisely coincide with valid segment boundaries.

(4) **Semantic fidelity**: the diagram correctly encodes object relationships described in the natural language problem.

(5) **Format compliance**: all constructions adhere strictly to the prescribed DSL conventions for reduction, transfer, multiplicative relations, and alignment.

Only instances that satisfy all five criteria are retained as *product-ready* samples in VisAlg.

B Human Evaluation Criteria for Evaluation Metrics

B.1 Evaluation Target and Boundary Conditions

The human assessment is strictly limited to the *given* DSL output.Reviewers must evaluate whether the DSL expresses the problem's key quantities and relationships in a *norm-compliant*, *non-leaking*, and *structurally self-consistent* manner, such that a reader can reliably reconstruct the intended structure and carry out a correct derivation from the diagram.

To minimize subjective preference, prior belief, and post-hoc "mental correction," the evaluation is conducted under the following boundary conditions: (i) reviewers must not introduce any extra information beyond what is explicitly encoded in the DSL; (ii) reviewers must not modify the output in any form, including adding segments, changing numeric values, renaming labels, or reformatting the program; (iii) the assessment does not consider the writing quality, fluency, or style of any accompanying natural-language solution.

B.2 Review Protocol and Evidence-Driven Practice

A structured, evidence-driven expert review protocol is adopted to maximize objectivity and reproducibility. Three domain experts with backgrounds in mathematics education and diagram-oriented coding are recruited. Each sample is rated independently by at least two reviewers; disagreements spanning two or more score levels are resolved by a third reviewer via arbitration.

To discourage intuition-based judgments, each assigned rating must be accompanied by *minimal sufficient evidence* that is directly verifiable from the DSL. Typical evidence includes: an HB end-

Step 1: Data Draft Generation Prompt

You are an expert instructor in bar-model reasoning. You specialize in constructing bar-model diagrams using a structured **Graphic Intermediate Representation (DSL)** to support mathematical problem solving.
Given a math word problem and its associated image, **directly** produce the output in the following format. Do **not** include any preamble, explanation, or commentary.
Step 1: Problem Analysis
Analyze the core structure of the problem using concise language. Do not restate the problem text. Output only the following four items:
(1) **Problem type**: e.g., sum–difference, proportional, transfer, comparison.
(2) **Objects**: specify whether the problem involves a single object or multiple independent objects.
(3) **Key information**: short phrases describing all given quantities and relations.
(4) **Query**: one sentence stating the required unknown.
Step 2: Diagram Planning
List the key considerations required to construct a correct and verifiable bar-model diagram:
(1) Whether a vertical bracket (VB) is permitted (VB is forbidden for single-object problems and allowed only for multi-object aggregation).
(2) Whether all *given* information from the problem is explicitly labeled in the diagram under a **visible-text-only** policy (numerical segment lengths do not count as textual labels).
(3) Whether the queried quantity is explicitly marked in the diagram using interrogative naming (e.g., "total ?", "how many ?").
(4) The core operation type involved (reduction, increase, transfer, comparison) and its corresponding visual encoding.
(5) Whether vertical alignment lines (VL) are required to express cross-row shared boundaries or post-operation equality.
(6) Any special constraints, including strict prohibition of answer leakage in all labels.
Step 3: Initial DSL Draft
Perform the necessary internal reasoning to determine correct segment lengths, then output an initial DSL program. Output **only** a clean DSL code block, without comments, explanations, or blank lines.
"'dsl <DSL code only> **Output Format Requirements**
(1) Begin **directly** with "Step 1: Problem Analysis".
(2) Use the exact step titles shown above.
(3) Output only the three steps; do not generate any additional content.
DSL Syntax
(1) `HL "name" row l1 l2 . . .` : horizontal bar composed of semantic subsegments; positive values denote solid segments, negative values denote dashed segments (absolute value as length).
(2) `VL x row0 row1`: vertical alignment line used only for cross-row shared boundaries.
(3) `HB "label" N|S row x0 x1`: horizontal bracket spanning interval $[x_0, x_1]$, placed above (N) or below (S) the bar.
(4) `VB "label" col row0 row1`: vertical bracket used **only** for aggregating multiple independent objects.
Core Construction Rules (Strict)
(1) All horizontal bars must be decomposed into semantic subsegments; drawing a single undivided bar is not allowed.
(2) Fractions and ratios must be represented via equal-length subsegments; multiplicative relations must be expressed by repeating equal segments, with the base quantity placed on the upper row.
(3) Reduction must be encoded as `(A - t)` followed by `(-t)`, with solid segments on the left and dashed segments on the right; negative segments denote *only* removal, deficit, transfer-out, or unknown placeholders.
(4) Transfer must be represented as paired $-t/+t$ segments across rows; post-transfer equality must be marked with a shared VL at the corresponding boundary.
(5) Vertical alignment lines are permitted only for cross-row shared boundaries and must align exactly with segment boundaries on *all* involved rows.
(6) Bracket endpoints must coincide exactly with segment boundaries; floating or misaligned brackets are invalid.
(7) Final numerical answers must not appear in any labels; computed values may appear only as segment lengths for rendering purposes.
(8) All given information from the problem statement must appear explicitly as textual labels in the diagram.
Silent Self-Check
Before outputting, internally verify alignment correctness, semantic compliance, information completeness, and non-leakage of final answers.

Figure 8: Prompt used in Step 1 for generating structured analysis and the initial DSL draft during VisAlg dataset construction.

point failing to coincide with an HL segment boundary; a VL failing to align with a cross-row critical boundary; a quoted label explicitly containing the final numeric answer to the queried quantity (answer leakage); or arithmetic constraints that cannot hold under the implied sum–difference or transfer relations. Evidence logs enable third-party auditing without reliance on reviewer-specific interpretation.

B.3 Evaluation Dimensions

DSL quality is characterized along five dimensions that jointly determine usability:

Structural Alignment. Whether HB endpoints and VL coordinates strictly coincide with HL seg-

Checklist-Driven Refinement Prompt

You are an expert instructor in bar-model reasoning. You specialize in auditing and refining bar-model diagrams expressed in a structured **Graphic Intermediate Representation (DSL)**.
You are given a math word problem with its image, together with an existing analysis and an initial DSL draft. Your task is to **verify the draft using a checklist** and **revise it only if violations are found**. Do **not** include any preamble or commentary.
Checklist-Based Verification
Select **2–4** verification items that are most relevant to the given problem. The following two checks are **mandatory** and must always be included. Each check must be marked as either `[PASS]` or `[FAIL]`.
(1) **Alignment**: verify that all bracket endpoints coincide with bar-segment boundaries or shared alignment positions, and that any vertical alignment markers lie on valid bar segments.
(2) **Information completeness**: verify that all original quantities and relations stated in the problem are explicitly labeled in the diagram, and that the queried quantity is clearly indicated.
In addition, select applicable checks from the following categories.
(3) **Norm compliance**: verify that reductions follow the left-solid/right-dashed convention; comparisons use right-side dashed segments; and transfers are encoded as paired subtraction and addition across rows.
(4) **Style consistency**: verify that multiplicative structures place the base quantity on the upper row; and that bracket placement prioritizes upper positioning, resolving overlaps by span length.
Refinement Decision
State in one sentence whether the DSL draft satisfies all selected checks.
If any check is marked `[FAIL]`, briefly describe the violations and output a corrected DSL program. If all checks pass, state that the draft can be used without modification.
Final Answer Generation
After the refinement decision, continue *directly* with the final solution generation. The solution must include, in order: (1) a one-sentence identification of the problem type; (2) a concise reasoning outline with formulas written in `$...$`; (3) a clean DSL code block representing the final diagram; (4) step-by-step computations; and (5) the final numerical answer.
Output Constraints
(1) Begin output directly with the checklist-based verification section. (2) Use bold text only for section headers; do not use numbered steps or section markers. (3) Do not repeat the problem statement. (4) Do not include any content outside the specified structure.
DSL Semantics
All DSL syntax and construction rules strictly follow those defined in the draft-generation stage, including semantic subsegment decomposition, VB usage restrictions, reduction and transfer encoding, alignment constraints, and the prohibition of answer leakage.

Figure 9: Prompt used for checklist-driven verification and conditional refinement of initial DSL drafts during VisAlg dataset construction.

ment boundaries, reflecting geometric legality and representational precision.

Information Coverage. Whether all key givens and the queried unknown are explicitly marked or clearly represented based *only* on visible textual labels (i.e., quoted strings). Numeric segment lengths alone do not count as textual labels. This dimension measures whether the intended problem structure is recoverable from the diagram content.

Numerical Consistency. Whether the segment lengths satisfy the intended arithmetic constraints (sum, difference, increase/decrease, transfer amount). Systematic errors such as uniform scaling artifacts or the use of numerous uninterpretable numbers are treated as violations.

Semantic Conformity. Whether the DSL follows task-specific construction conventions, including: reduction encoded as solid-left and dashed-right segments; transfer encoded as paired $-t/+t$ segments with equal magnitude across rows; multiplicative relations expressed via repeated equal-length subsegments with the base quantity emphasized; non-abusive use of VL/VB; and semantically motivated HL decomposition by problem type.

Answer Leakage. A hard constraint assessed solely from visible textual labels: if the final numeric answer to the queried quantity appears explicitly in quoted labels, the output is considered leaking.

B.4 Overall Rating Scale

An overall five-level score is assigned based on the five dimensions above:

- **5 (Excellent):** No leakage; strict structural alignment; complete information coverage; numerically consistent relations; and clear, norm-compliant semantic decomposition.
- **4 (Good):** Overall correct and readable; minor non-critical imperfections (e.g., slight alignment or labeling issues) that do not hinder understanding or derivation.

Scoring Prompt for DSL Verification

You are an expert instructor in bar-model reasoning. You are familiar with evaluating bar-model diagrams expressed in a structured **Graphic Intermediate Representation (DSL)**.
Your task is to **score the given DSL code** based solely on the **problem statement** and the **provided solution (including DSL code)**. You must **not** modify, complete, or reinterpret the DSL code. Do not introduce any information beyond what is explicitly provided.
Evaluation Scope
All judgments must follow the *visible-text-only* principle: only quoted strings in `HL` names and `HB`/`VB` labels are considered visible annotations. Numeric segment lengths and coordinates are not treated as textual information.
Scoring Criteria
Evaluate the DSL code according to the following checklist. For each item, output either `[PASS]` or `[FAIL]`, together with a brief justification.
Critical Criteria (Fail Any ⇒ Score = 0.0)
(1) **Alignment correctness**: all bracket endpoints align with bar-segment boundaries; all vertical alignment markers lie on valid shared boundaries.
(2) **Information completeness**: all given quantities and required unknowns from the problem statement are explicitly annotated using visible text.
(3) **Numerical consistency**: all segment lengths are numerically self-consistent with the problem logic, without arbitrary scaling or unexplained values.
(4) **Transfer correctness**: transfer operations must be represented by paired subtraction and addition across rows, with appropriate alignment markers when required.
(5) **Answer leakage**: no final answer values appear in any visible annotations.
(6) **VB/VL usage**: vertical brackets are used only for multi-object aggregation; vertical alignment markers appear only at shared cross-row boundaries.
Non-Critical Criteria
The following criteria affect the score only if all critical criteria pass:
(7) **Reduction conventions**: reduction and deficit relations follow the left-solid/right-dashed convention.
(8) **Multiplicative structure**: multiplicative relations are expressed using repeated equal-length segments, with the base quantity placed on the upper row.
(9) **Semantic decomposition**: horizontal bars are decomposed into semantically meaningful subsegments rather than drawn as undivided totals.
(10) **Label conciseness**: visible annotations are concise, non-redundant, and free of embedded calculations.
Scoring Rule
If any critical criterion is marked `[FAIL]`, the final score is **0.0**. Otherwise, the base score is **1.0**, with a penalty of **0.1** deducted for each failed non-critical criterion.
Output Format
First output a section titled `[Scoring Rationale]`, listing each criterion with its pass/fail status and justification. Then output a single line:

```
[Final Score]: <float between 0.0 and 1.0>
```

Do not output JSON or any additional formatting.

Figure 10: Prompt used for strict LLM-based scoring and filtering of refined DSL drafts in VisAlg.

- **3 (Acceptable):** Still usable for problem solving but requires frequent reference to the problem statement to resolve ambiguities; partial missing labels, coarse decomposition, or localized norm violations may be present.

- **2 (Poor):** High risk of misinterpretation due to unreliable alignment, missing critical information, multiple semantic violations, or strained numerical relations, making stable derivation difficult.

- **1 (Unacceptable):** Fatal violations, including answer leakage, uniform scaling artifacts, fundamentally invalid sum–difference or transfer relations, large-scale alignment failures, or incorrect core semantic structure.

B.5 Leakage as a Dominant Violation

Answer leakage is treated as the most destructive violation because it breaks the boundary that the diagram should encode structure rather than disclose the solution. Accordingly, once leakage is confirmed (from quoted labels), the output is rated as unacceptable and the evidence must be recorded explicitly.

C Case Studies on Visual Algebra Schemas

We present one representative example for each of the five visual algebra schemas in VisAlg. These cases illustrate how *Thinking with Drafting* (TwD) operationalizes *optical decompression*: it converts abstract textual constraints into an explicit and spatially aligned *DSL*, so that the problem can be

solved directly from the rendered structure without relying on implicit, unverified reasoning.

Alignment-centric schemas. Difference Analysis (Figure 19) and **Proportional Distribution** (Figure 11) both require cross-object alignment to make relational constraints executable. In Figure 19, the DSL anchors one entity as a reference and encodes "more than" / "fewer than" relations as explicit offset segments, turning comparative language into a geometry-consistent subtraction layout. In Figure 11, the DSL realizes the multiplicative constraint by repeating equal-length unit segments and aligning boundaries across rows, so the "$\times 12$" relation is enforced by topology rather than inferred implicitly; the final query is then represented as a single unknown bracket on the composed total.

Decomposition-centric schemas. Sum & Split (Figure 16) and **Rate & Percentage** (Figure 13) emphasize part–whole partition and unit grounding. Figure 16 isolates the known remainder as a dedicated segment and marks the target as the complementary part, making the computation a direct completion on the bar. Figure 13 grounds fractional change by first fixing the base quantity as the unit reference and then attaching the fractional increment as an explicit subsegment, reducing ambiguity about the comparison base.

State-transition schema. Change & Revert (Figure 12) involves a counterfactual transfer and a post-transfer relation. The DSL externalizes the hypothetical "give" operation with paired decrease/increase segments and then imposes the after-state constraint on the aligned configuration, enabling reverse deduction while keeping all visible labels faithful to the original statement (i.e., without leaking computed answers).

D Additional Error Analysis

We summarize a **Taxonomy of Structural Degeneration** observed in baseline diagrams, where the output may remain arithmetically compatible yet loses the structural invariants required for verification.

D.1 Semantic Erasure: Multiplicative Topology Collapsed

In Figure 21, the baseline collapses the given $\times 3$ constraint into an additive "difference" layout. This erases the repeated-unit structure, so the multiplier is no longer visually provable even if the final arithmetic is correct.

D.2 Label Injection: Numbers without Geometric Support

Figure 22 shows a label–structure mismatch: the model writes the computed difference as text, but does not allocate a corresponding sub-segment. The diagram therefore contains *claims* without geometric evidence, and downstream reasoning can mistakenly treat labels as quantities.

D.3 Alignment Conflict: Incompatible Global Boundaries

In Figure 23, the baseline mixes incompatible alignment cues: dashed completion implies one shared endpoint, while vertical guides declare another boundary. This breaks global boundary consistency, so the "less by 35" relation is not stably encoded in the diagram.

E Potential Risks

We identify two primary risks that stem from the formalization of reasoning introduced by the Thinking with Drafting (TwD) paradigm, particularly in educational contexts.

First, the use of a structured DSL may amplify *automation bias*. Because the generated diagrams resemble formal proofs, users may conflate structural validity with semantic correctness, implicitly assuming that a well-formed intermediate representation guarantees a correct solution.

Second, TwD introduces a risk of *cognitive offloading* that may lead to *skill atrophy* in diagrammatic reasoning. By externalizing key steps of problem decomposition and visualization, the system may reduce the learner's engagement in constructing and maintaining structural invariants. Over time, excessive reliance on automated drafting can weaken the user's ability to independently translate textual constraints into spatial representations, undermining the development of foundational visual reasoning skills.

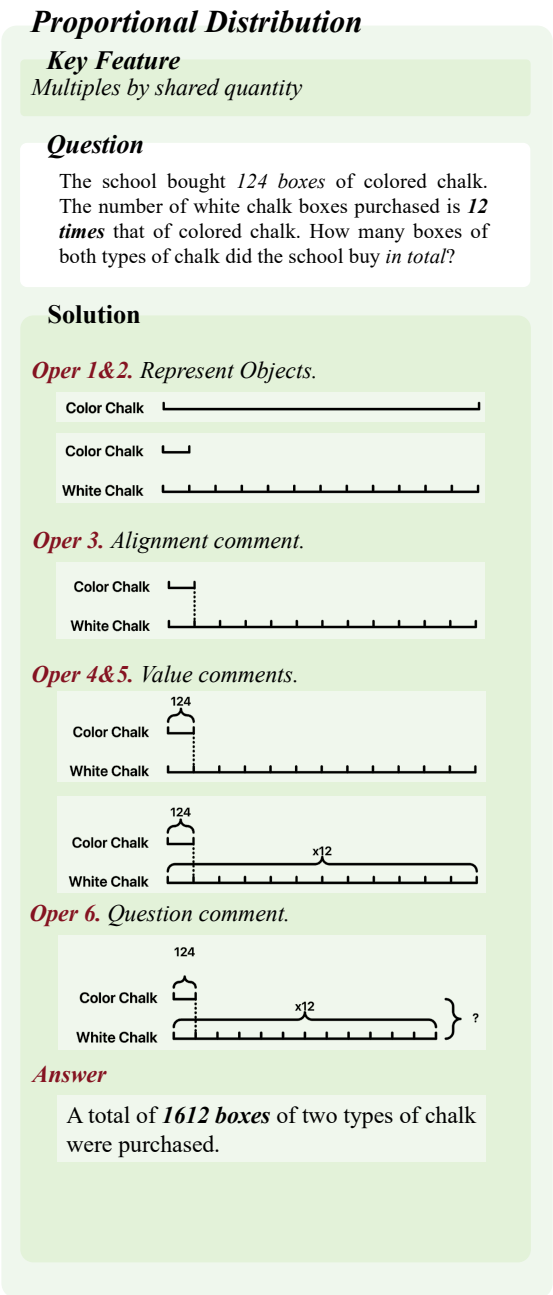


Figure 11: **Proportional Distribution.** The diagram enforces the multiplicative relation via repeated equal-length units and boundary alignment, and marks the queried total as an explicit unknown on the composed bar. TwD does not merely calculate 12×124; instead, it enforces the multiplicative constraint via topological repetition. By rendering the 'White Chalk' bar as a composite of 12 equal-length units aligned with the 'Color Chalk' reference unit, the model transforms an abstract arithmetic operation into a concrete unit-repetition task, making the total sum visually deducible.

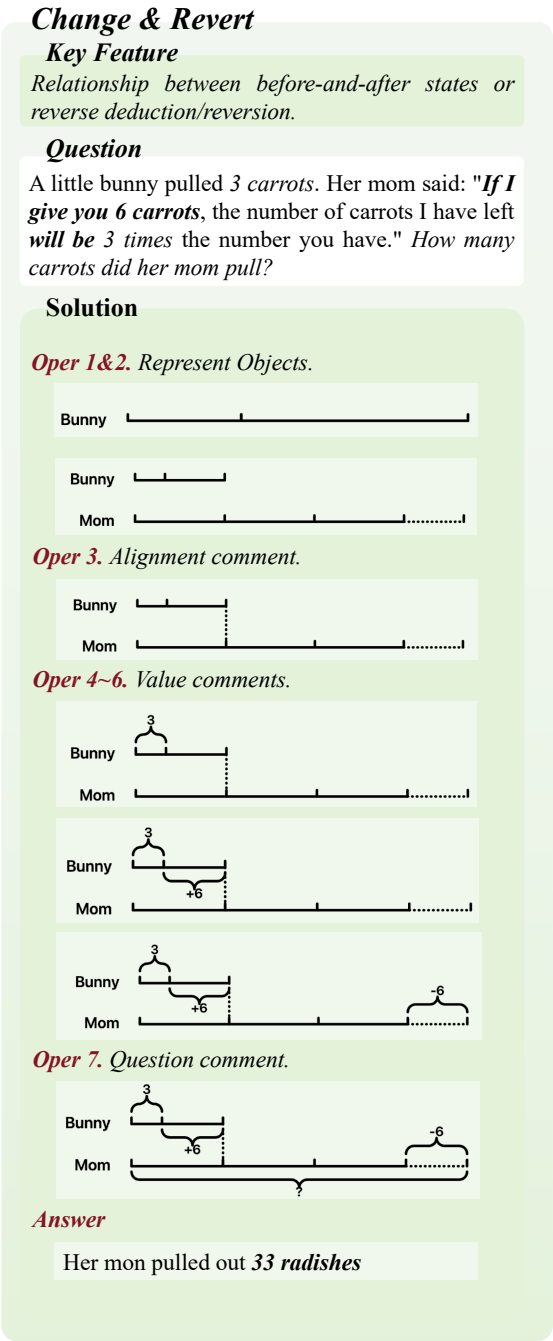


Figure 12: **Change & Revert.** A counterfactual transfer is rendered as paired decrease/increase segments, after which the post-transfer constraint is imposed on the aligned after-state topology. This example illustrates how TwD handles hypothetical state transitions. The model employs a dual-segment representation where transfers are rendered as paired decrease/increase segments. Crucially, the "post-transfer" multiplicative constraint $\times 3$ is imposed not on the initial state, but on the aligned after-state topology. This proves the model's ability to reason about dynamic temporal states within a static spatial diagram.

Rate & Percentage

Key Feature

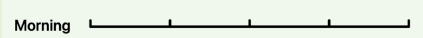
Expressed as rates; primarily about unit "1" and percentage

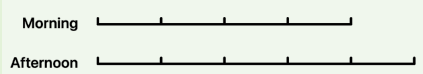
Question

Lele drank *4/5 liters* of water in the morning. In the afternoon, he drank ***1/4 more*** than what he drank in the morning. How many liters more water did Lele drink in the afternoon than in the morning?

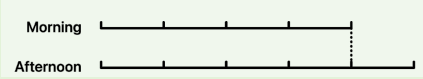
Solution

Oper 1&2. *Represent Objects.*

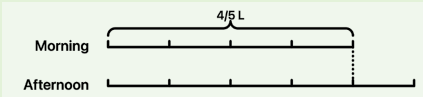




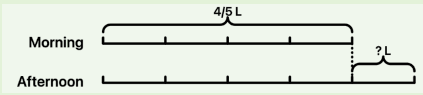
Oper 3. *Alignment comment.*



Oper 4. *Value comment.*



Oper 5. *Question comment.*



Answer

He drinks ***1/5 L*** more

Figure 13: **Rate & Percentage.** Fractional change is grounded by fixing the base as a unit reference and attaching the fractional increment as a dedicated subsegment aligned to the shared boundary. The model fixes the morning consumption as the holistic unit "1", and attaches the fractional increment $(1/4)$ as a dedicated sub-segment aligned to the unit boundary. This explicit segmentation allows the model to visually isolate the Δ from the whole, preventing unit confusion.

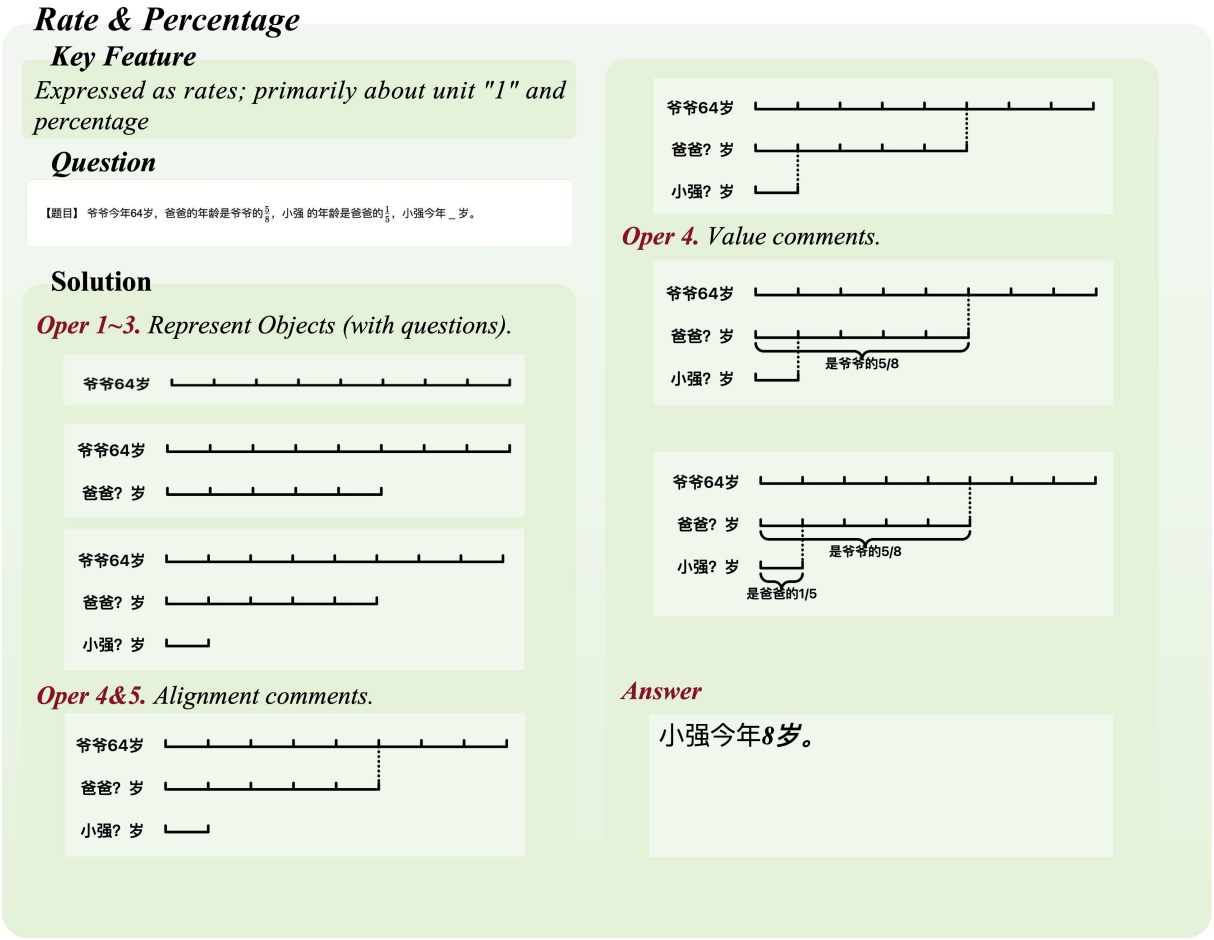


Figure 14: **Rate & Percentage.** A complex multi-step ratio problem involving chain dependencies. TwD manages this hierarchy through cascading alignment: each subsequent row's length is topologically anchored to the specific fraction of the preceding row. The vertical dashed lines serve as transitive logic gates, ensuring that the final quantity is derived from a rigorously valid chain of geometric proportions, minimizing error propagation.

Rate & Percentage

Key Feature

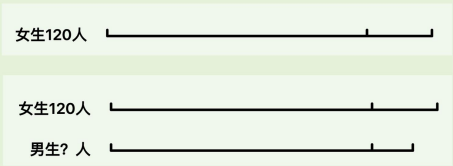
Expressed as rates; primarily about unit "1" and percentage

Question

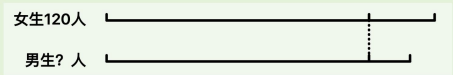
【题目】六年级有女生120人，男生的人数是女生 的$\frac{4}{5}$多15人，六年级有男生多少人？

Solution

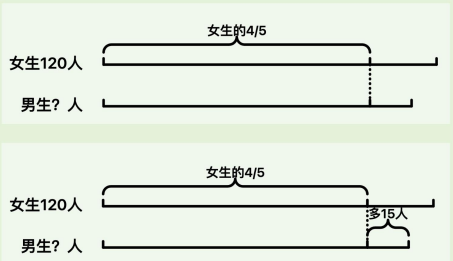
Oper 1&2. *Represent Objects (with question).*



Oper 3. *Alignment comment.*



Oper 4&5. *Value comments.*



Answer

六年级有男生***111***人。

Figure 15: **Rate & Percentage.** Fractional relationships are represented by fixing the base quantity as a unit reference and aligning proportional segments to this shared unit. The fractional change is visualized as a dedicated subsegment, supporting reasoning based on rates and percentages.

Sum & Split

Key Feature

Finding total/remaining amount; whole-part; merging multiple parts.about "sum, remainder, composition/splitting"

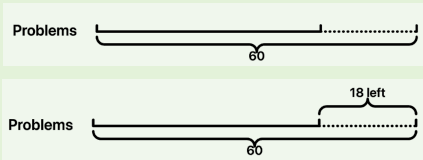
Question

Xia Ya needs to do 60 mental arithmetic problems. 18 problems are still left unfinished. How many problems has Xia Ya already finished?

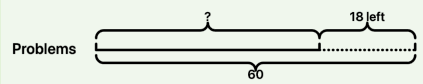
Solution

Oper 1. *Represent Object.*

Oper 2&3. *Value comment.*



Oper 4. *Question comment.*



Answer

She has finished ***42*** problems.

Figure 16: **Sum & Split.** The whole–part structure is made explicit by isolating the known remainder segment and marking the target as the complementary unknown, directly supporting completion-by-subtraction. The unknown remainder is highlighted as the target, supporting solution by subtraction.

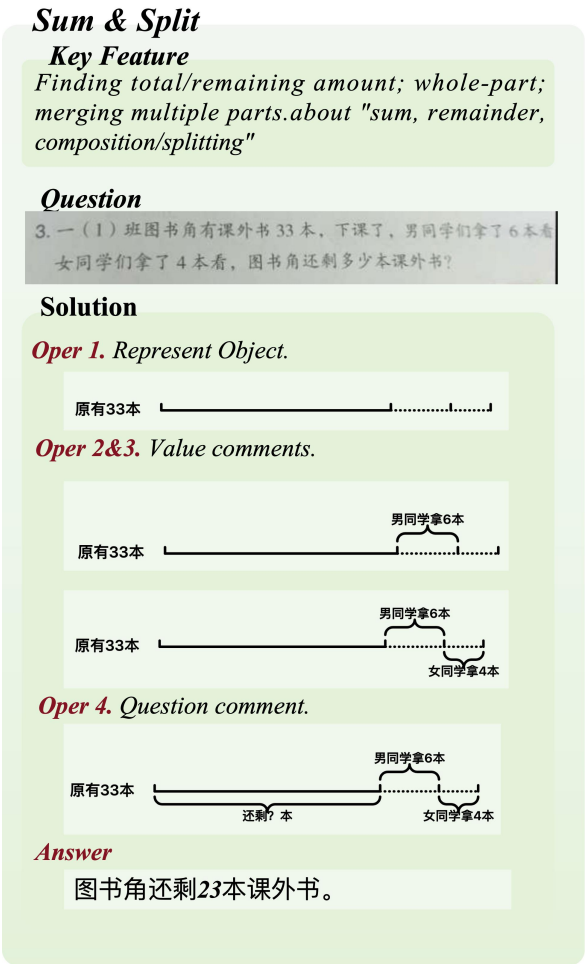


Figure 17: **Sum & Split.** The whole–part structure is made explicit by isolating the known remainder segment and marking the target as the complementary unknown, directly supporting completion-by-subtraction. The unknown remainder is highlighted as the target, supporting solution by subtraction.

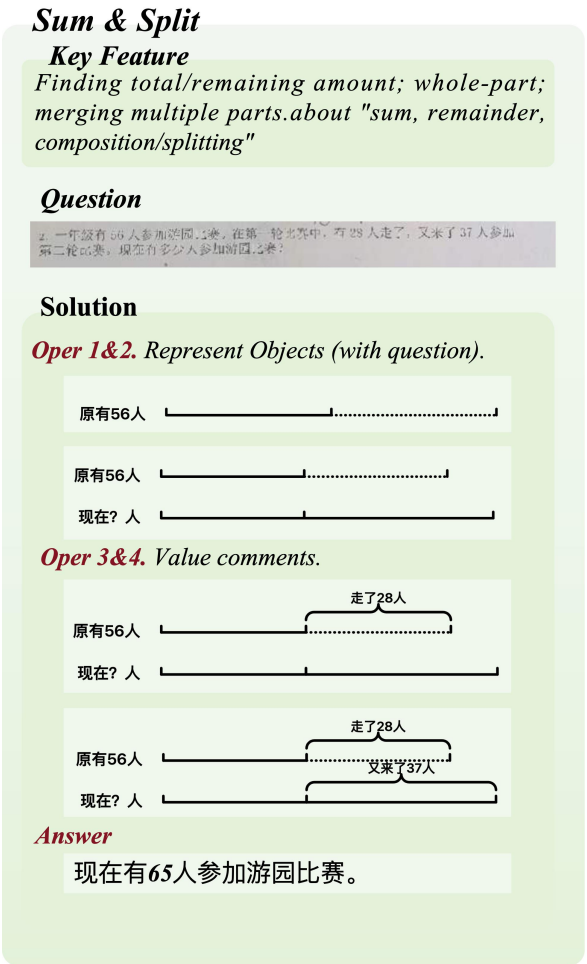


Figure 18: **Sum & Split.** The whole–part structure is made explicit by isolating the known remainder segment and marking the target as the complementary unknown, directly supporting completion-by-subtraction. The final total is identified as the complementary unknown, supporting solution by subtraction and addition.

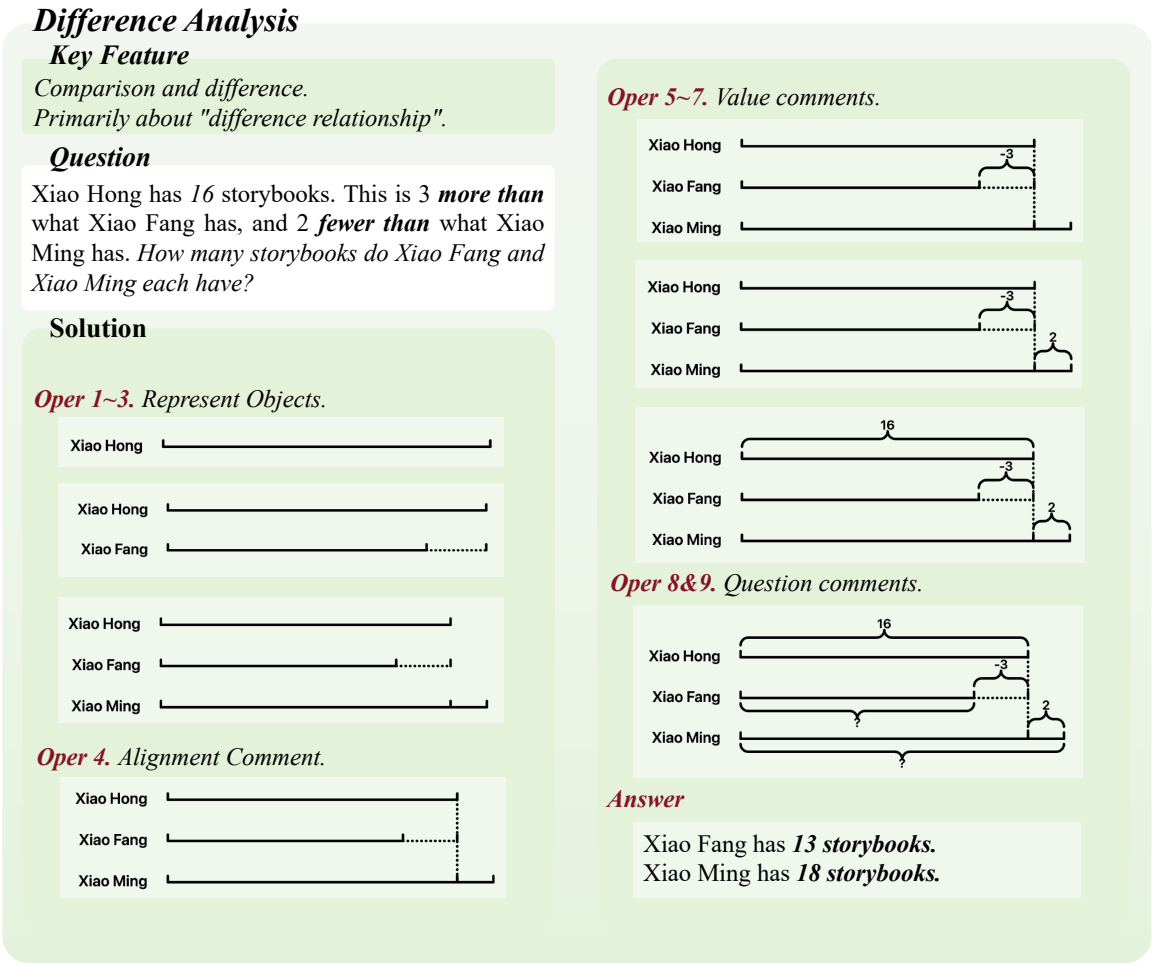


Figure 19: **Difference Analysis.** This example illustrates the Thinking with Drafting process on a multi-entity comparison problem. The model does not hallucinate the answer directly; instead, it performs logical reconstruction in steps: (1) instantiating objects (Oper 1-3), (2) enforcing topological alignment via vertical anchors (Oper 4), and (3) encoding "more than/fewer than" relations as explicit offset segments (Oper 5-7). This step-by-step grounding ensures that the final arithmetic inference is derived from a verified geometric structure.

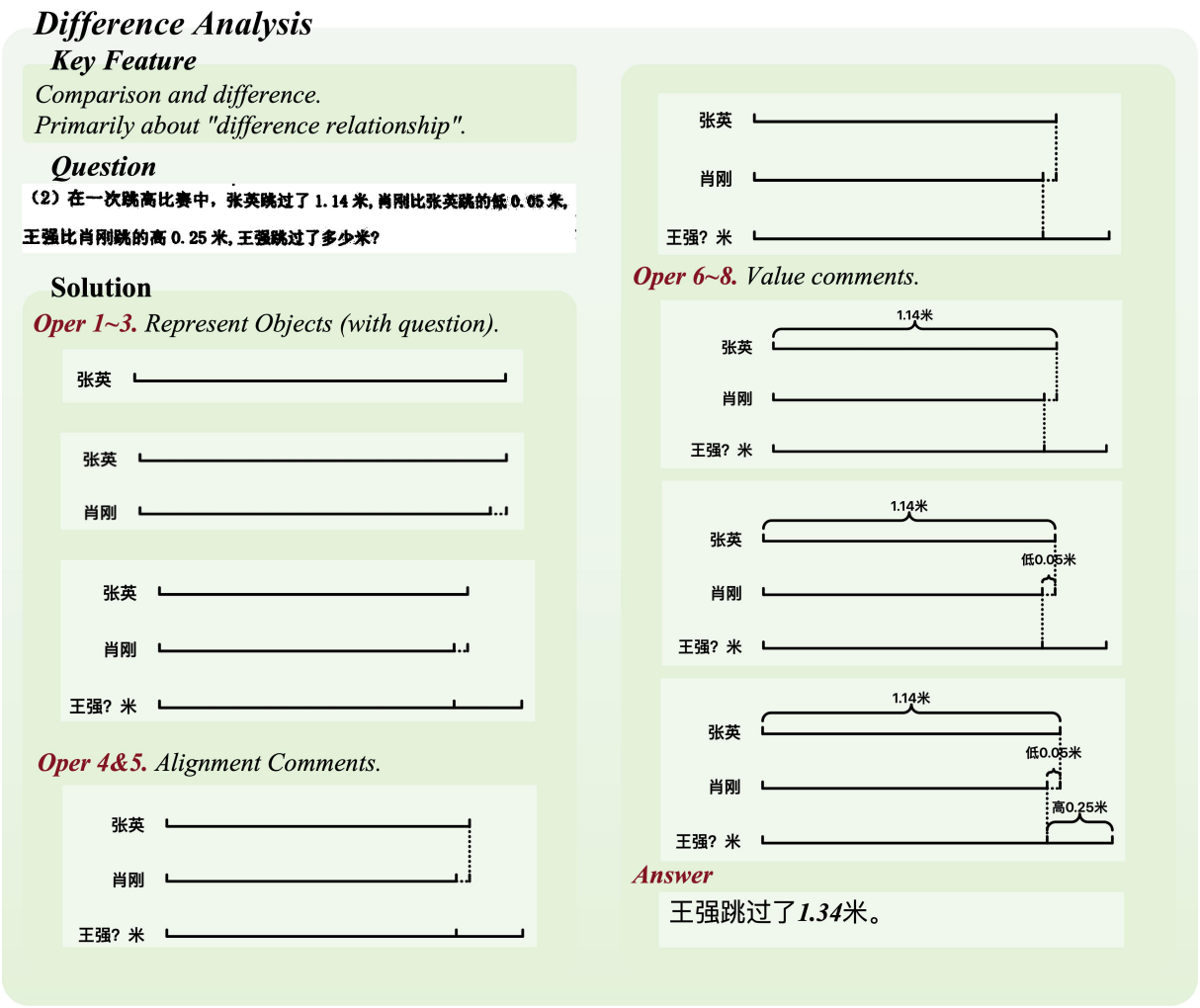


Figure 20: **Difference Analysis.** Application of TwD to a continuous-value scenario involving bidirectional differences ("lower than" vs. "higher than"). The system successfully decodes the textual constraints into precise spatial alignments. Note how the vertical dashed lines act as logical anchors, physically locking the relative positions of the reference entity and derived entities. This transforms an abstract arithmetic word problem into a concrete visual subtraction and addition task, mitigating logical errors in multi-step calculation.

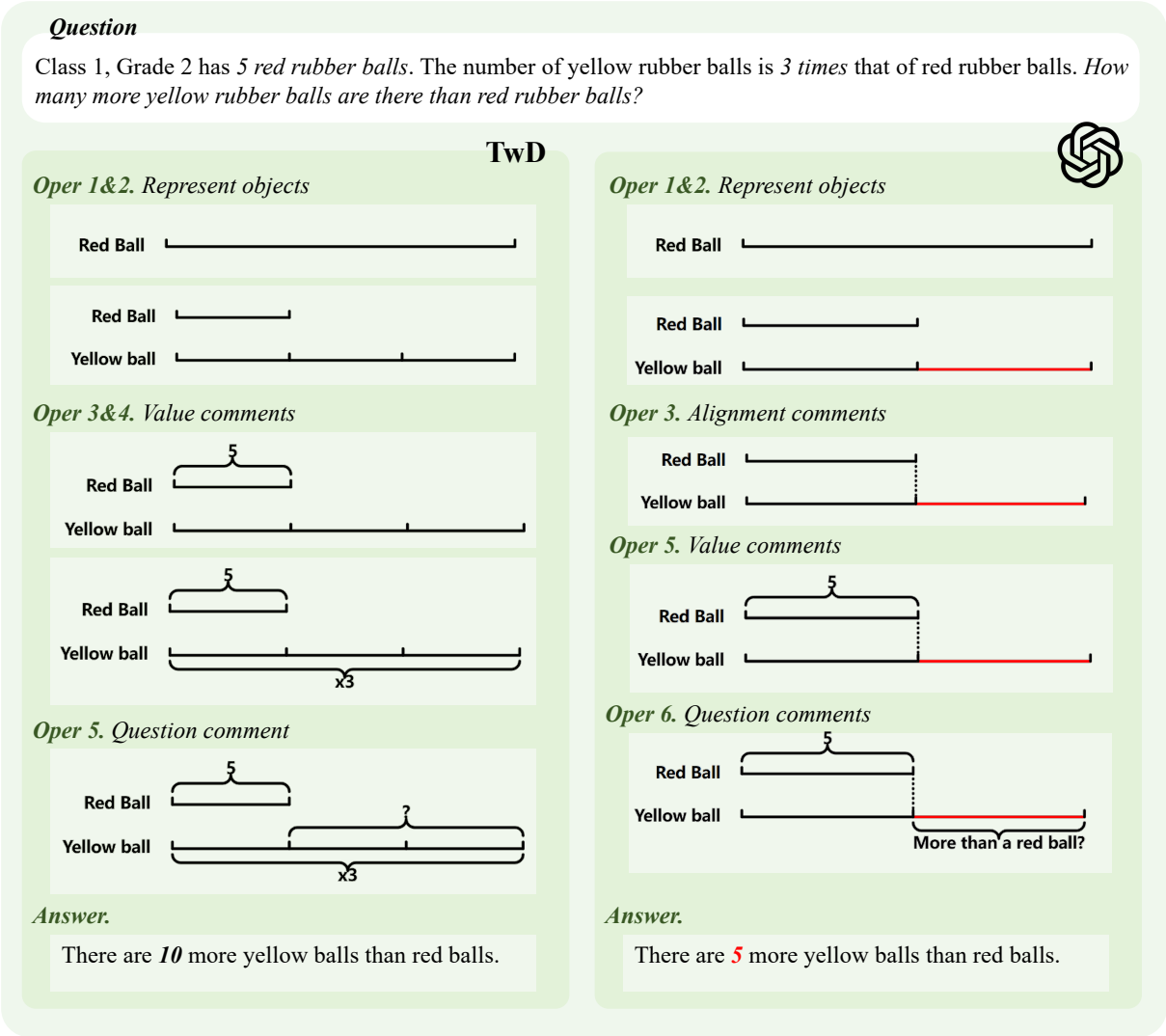


Figure 21: **Semantic Erasure.** The $\times 3$ constraint is collapsed into an additive layout, removing repeated-unit evidence. The baseline model suffers from Semantic Erasure: it collapses the multiplicative constraint into a generic additive layout, failing to render the repeated unit segments. TwD explicitly preserves the unit topology, rendering three distinct segments for the yellow ball row. This structural fidelity enforces the correct arithmetic operation.

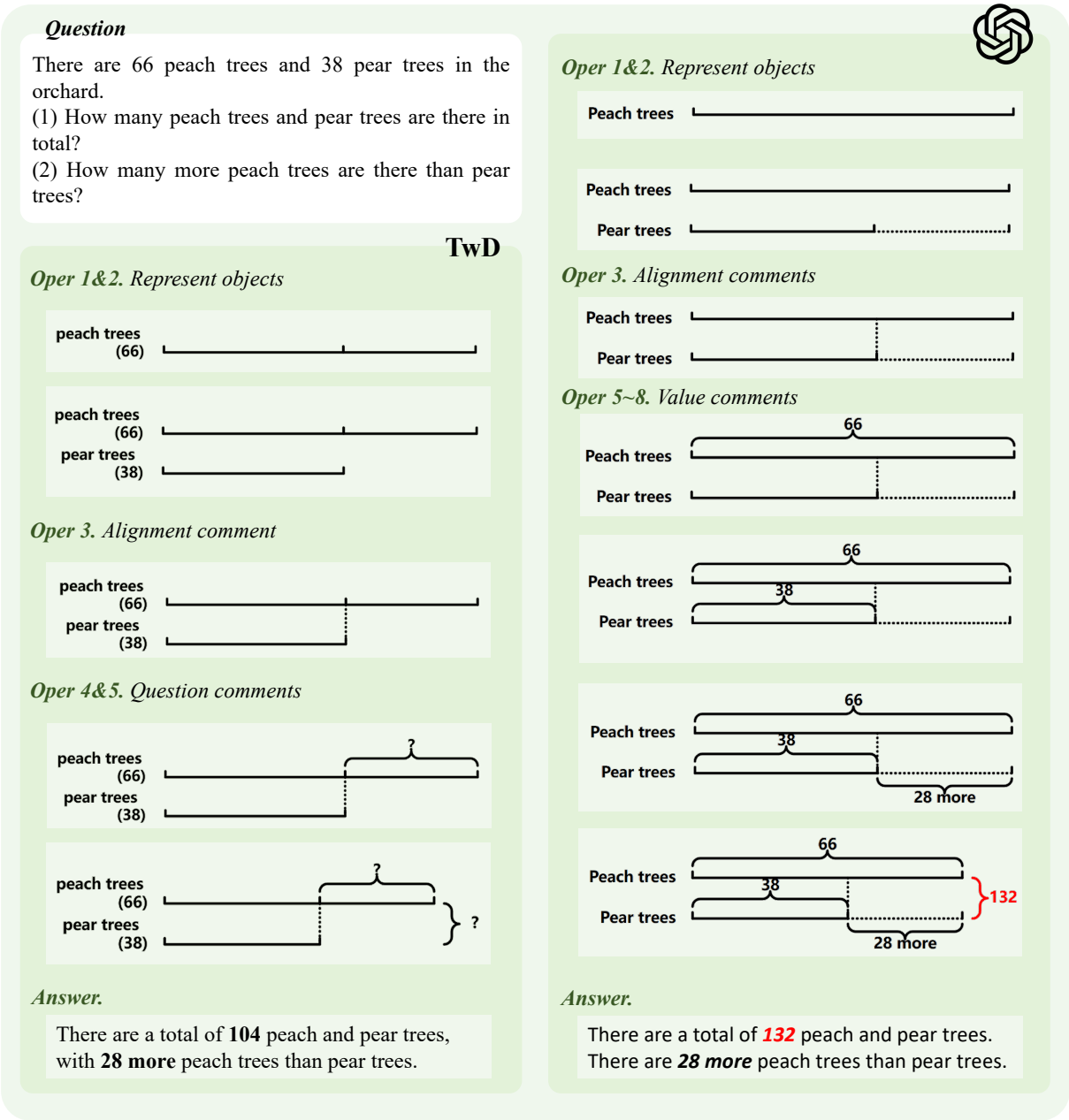


Figure 22: **Label Injection.** A computed value is written as text without a supporting sub-segment, yielding an ungrounded claim. The baseline model exhibits Label Injection: it hallucinates a computed value ("132") and injects it as a text label without generating the supporting geometric sub-segments. The visual diagram thus becomes a deceptive artifact that does not physically represent the sum. TwD constructs the result bottom-up. By strictly aligning the start and end points of the 'Peach' and 'Pear' segments, it creates a valid geometric aggregation, ensuring the final answer is visually deducible.

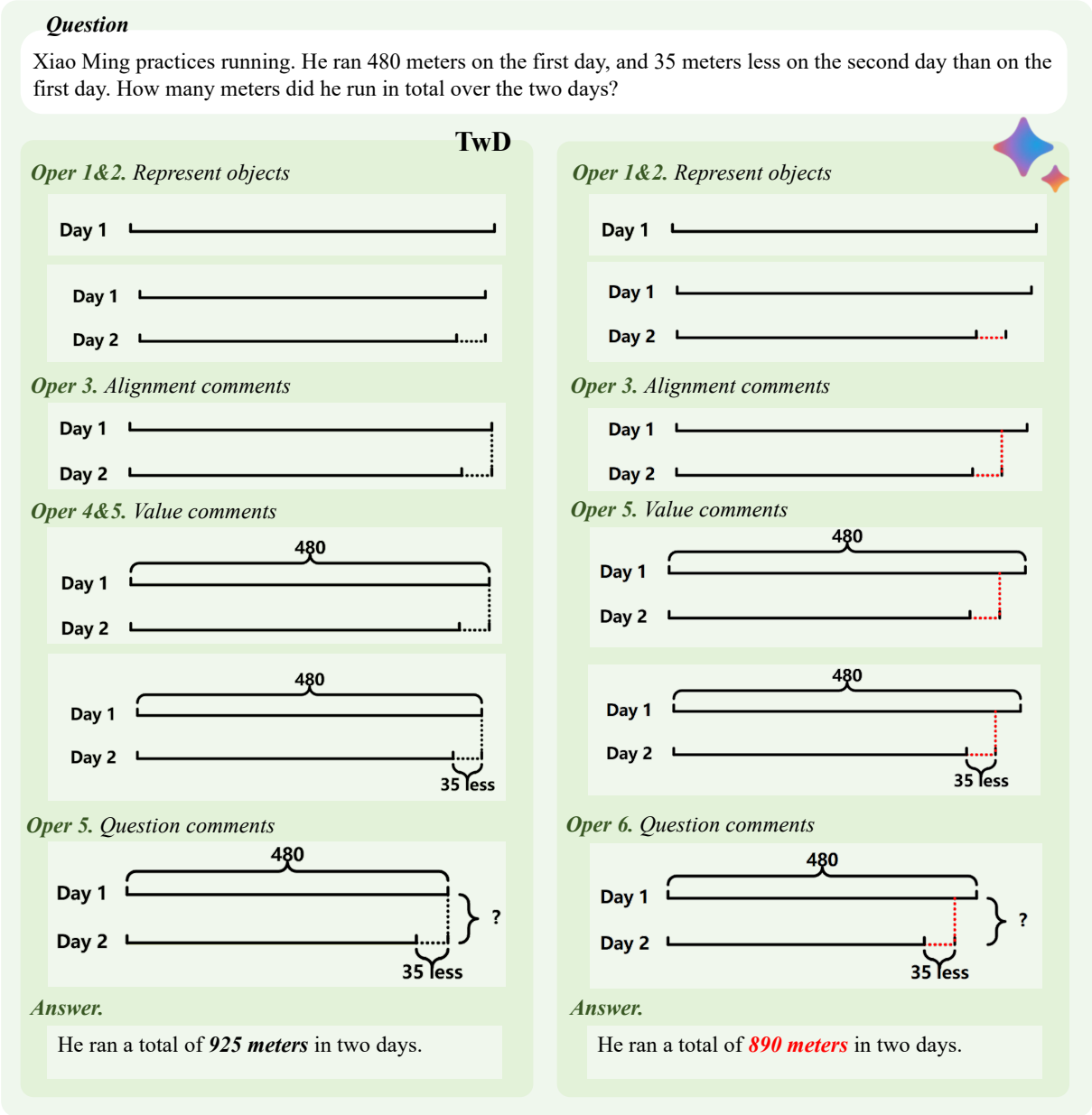


Figure 23: **Alignment Conflict.** Conflicting global boundaries break the stability of cross-row relations. The baseline model generates an Alignment Conflict: the vertical dashed line (alignment anchor) is misplaced, visually suggesting that Day 2 is longer than Day 1 despite the label "35 less". This topological contradiction breaks the logical chain, leading to an erroneous calculation. TwD correctly places the subtractive anchor. The dashed line precisely demarcates the difference segment, enforcing a consistent spatial logic where the length of Day 2 is physically constrained to be shorter, guiding the correct subtraction.