[81] Yiyang Zhou, Haoqin Tu, Zijun Wang, Zeyu Wang, Niklas Muennighoff, Fan Nie, Yejin Choi, James Zou, Chaorui Deng, Shen Yan, et al. When visualizing is the first step to reasoning: Mira, a benchmark for visual chain-of-thought. *arXiv preprint arXiv:2511.02779*, 2025. 6

[82] Le Zhuo et al. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. *arXiv preprint arXiv:2504.16080*, 2025. 3

A. Data Synthesis Pipeline

We provide implementation details for our automated data collection pipeline that generates 12K multimodal chain-of-thought training trajectories.

A.1. Pipeline Architecture

The agentic framework coordinates three model roles in an iterative loop (Fig. 6): **Image Gen Model** produces initial images from user prompts, **Vision-language Model** evaluates satisfaction and performs verification with content memory and subgoal decomposition, and **Image Editing Model** applies refinements based on VLM planning. This loop continues until the VLM determines the image satisfies all requirements, producing interleaved text-image chain-of-thought trajectories that are filtered for quality.

A.2. Model Components

Prompt generation: Llama-4-Scout-17B-16E generates 20K diverse prompts covering compositional attributes, spatial relations, and multi-object generation tasks based on T2I-CoReBench.
Image generation: Flux Pro produces initial images from prompts. For complex prompts, Qwen3-VL decomposes prompts into subgoals and executes the first step in initial generation.
Verification: Qwen3-VL evaluates whether images satisfy prompts. If not, it generates explicit chain-of-thought reasoning, identifying deficiencies, planning improvements, and specifying editing instructions.
Editing: Flux Kontext or Qwen-Image-Edit applies editing instructions based on VLM planning.

A.3. Example Trajectory

Fig. 7 shows a concrete bookshelf generation trajectory demonstrating the three cognitive behaviors induced by our framework. The VLM performs **verification** by identifying that books are present when the prompt specifies "no books, only picture frames." It exhibits **subgoal decomposition** by breaking the correction into sequential steps—first removing books, then adding frames. Finally, it demonstrates **content memory** by explicitly referencing and comparing Images #1, #2, and #3 to track cumulative progress across refinement rounds.

A.4. Training Data Statistics

After quality filtering, we obtain 12K trajectories with the following characteristics: training trajectories average **3.6 refinement rounds** (range 1-8 rounds). Training on this data requires 700 H100 GPU hours.

A.5. VLM Prompt Design

The vision-language model uses a structured prompt template (Table 7) to induce cognitive behaviors during data synthesis. The prompt guides the VLM through three steps: (1) detailed image description with explicit object counts and spatial relationships, (2) comparison analysis against user requirements with reflection on previous attempts, and (3) decision making between editing, backtracking, or completion. This structured reasoning naturally produces trajectories exhibiting verification, subgoal decomposition, and content memory.

B. Additional Qualitative Results

The VLMs produce interleaved text and image tokens with explicit thinking tokens. The reflection step generates detailed reasoning about *why* outputs fall short and *how* to improve them, rather than simply issuing new instructions.

Additional qualitative examples are provided in the main paper (Fig. 8), showing representative trajectories across different task types and computational budgets.

C. Generalization Preservation

Fine-tuning on 12K reasoning-heavy trajectories does not cause catastrophic forgetting of the base Bagel model's general capabilities. We compare Bagel before and after fine-tuning on our data (without test-time scaling at inference): the fine-tuned model achieves 0.783 alignment on OneIG-Bench (vs. 0.764 for vanilla Bagel) and 2.26 on ImgEdit (vs. 1.31), indicating that the chain-of-thought training data improves rather than degrades the base model's instruction-following capabilities even without multi-round inference.

D. Scaling Beyond $C{=}10$

We cap evaluation at $C{=}10$ due to GPU memory constraints. We observe that image quality collapses when editing rounds produce minimal visual changes (LPIPS < 0.03 between consecutive images), as accumulated autoregressive noise degrades fidelity. Such non-improving editing steps are infrequent, so scaling remains effective up to $C{=}10$. Beyond this point, we expect TTS performance to saturate or degrade once quality collapse dominates. The exact inflection point depends on the base generation and editing model capabilities. Potential mitigations include: (a) perceptual thresholding to skip rounds with minimal

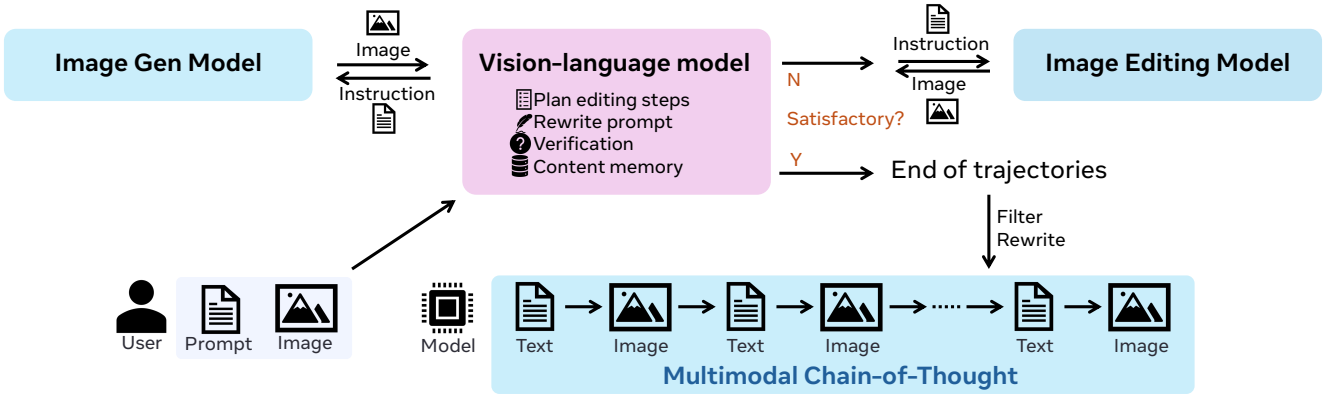

Figure 6. **Data synthesis pipeline architecture.** Three model roles coordinate via information flows: Image Gen Model produces initial images, Vision-language model verifies image and performs planning/prompt rewriting with content memory, Image Editing Model applies refinements. Trajectories loop until satisfied, producing interleaved text-image chain-of-thought data.

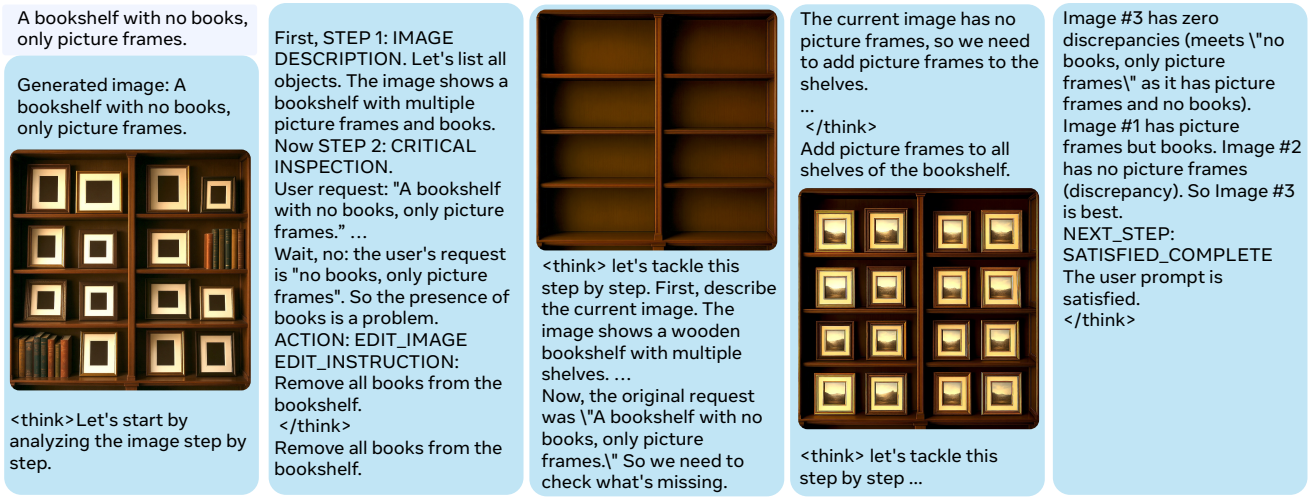

Figure 7. **Detailed chain-of-thought trajectory demonstrating cognitive behaviors.** This bookshelf generation example shows the model's explicit reasoning through `<think>` blocks across three refinement rounds. **Verification:** the model identifies that books are present when the prompt specifies "no books, only picture frames." **Subgoal decomposition:** the model breaks the correction into sequential steps—first removing books, then adding picture frames. **Content memory:** the model explicitly references and compares Images #1, #2, and #3 to track cumulative progress. The reasoning demonstrates how chain-of-thought enables iterative self-correction through explicit evaluation and planning.

changes, (b) "reset" rounds that regenerate from scratch using accumulated reasoning, and (c) adaptive noise scheduling to counteract quality degradation.

E. Failure Analysis

E.1. Failure Cases

Despite strong performance, our approach exhibits limitations in specific scenarios. First, tasks requiring precise physical reasoning or fine-grained spatial relationships occasionally fail, as iterative refinement may struggle to correct fundamental physics violations or attribute binding errors inherited from the base generation/editing models (e.g., incorrect leash-dog assignment or wrong helmet placement and sizes in Fig. 3). Second, we observe occasional degradation loops where reflection incorrectly identifies nonexistent issues, leading to unnecessary edits that harm quality rather than improve it—a verification hallucination bottleneck particularly evident when the VLM's verification capabilities are insufficient to accurately assess subtle visual attributes. Third, extremely complex compositional prompts with many interacting constraints can lead to sub-

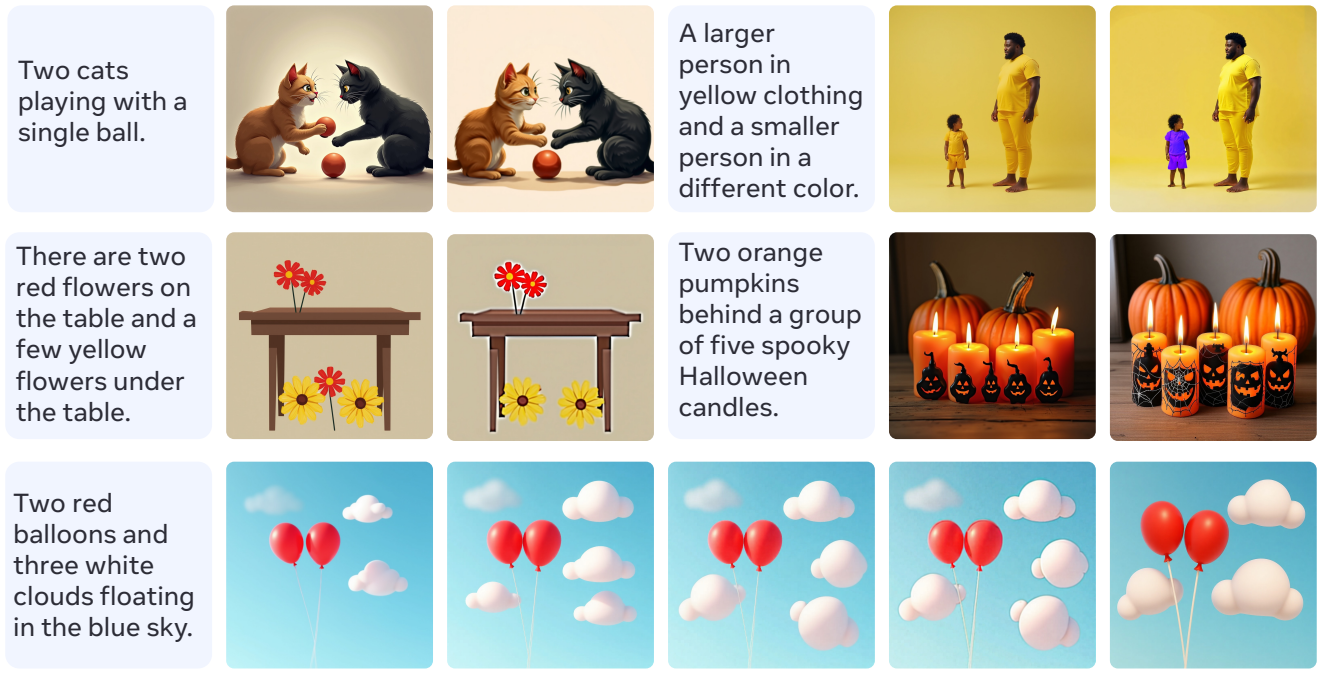

Figure 8. **Qualitative examples of chain-of-thought test-time scaling.** Representative trajectories showing progressive refinement across different tasks and computational budgets. Examples demonstrate how explicit chain-of-thought reasoning enables the model to iteratively improve compositional generation.

goal conflicts during decomposition, where satisfying one constraint inadvertently violates another. Finally, test-time scaling cannot overcome fundamental capability gaps in the base model; if the underlying diffusion or VLM components lack certain semantic understanding, additional inference compute provides diminishing returns. These failure modes suggest directions for future work, including more robust verification mechanisms, physics-aware refinement strategies, and constraint satisfaction planning. We further discuss scaling limits beyond $C{=}10$ in Sec. D. Failure visualizations are presented in Sec. E.

While our approach achieves strong performance, we observe limitations in specific scenarios. Fig. 9 visualizes representative failure modes where chain-of-thought test-time scaling struggles to produce satisfactory outputs even with extended computational budget. These cases reveal fundamental challenges in precise compositional reasoning, complex spatial arrangements, and fine-grained attribute control that warrant future investigation.

Limitations. While our approach demonstrates strong results, test-time scaling inherently requires additional computational resources at inference. Future work should explore more efficient reflection mechanisms and adaptive budget allocation strategies that minimize computational overhead while preserving quality gains.

Future directions. Promising directions include extending our approach to additional modalities (audio, video), augmenting reflection with explicit physical reasoning to enforce implicit constraints (e.g., object sizing, perspective, occlusion), investigating reinforcement learning from human feedback to further improve reflection quality, and exploring how test-time scaling interacts with other inference-time techniques such as self-consistency and verifier-guided generation.

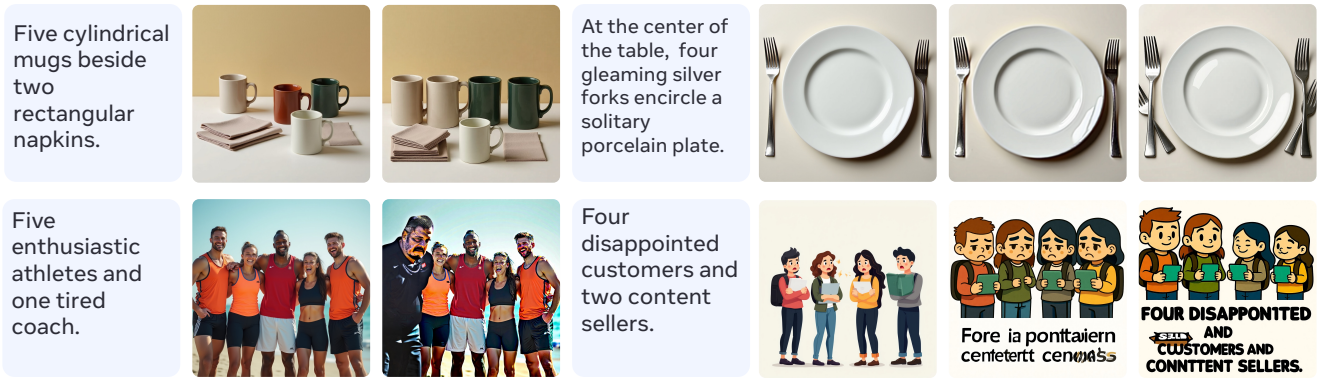

Figure 9. **Representative failure modes. Example 1:** Compositional constraints with precise object counts and spatial arrangements (napkin count). **Example 2:** Complex spatial relationships requiring specific geometric configurations (forks encircling plate). **Examples 3,4:** Layout change from the intermediate images (people count).

```
You are an intelligent and honest
image evaluation agent.

ORIGINAL USER REQUEST: {user_prompt}
[Previous images information with
satisfied/TODO features]

STEP 1 - IMAGE DESCRIPTION:
First, describe what you see in the
current image in detail:
- List ALL objects present with
exact counts
- Describe their positions and
spatial relationships
- Note colors, materials, lighting,
and style
- Describe the overall scene
composition

STEP 2 - COMPARISON ANALYSIS:
Compare your image description with
the user request:
1.  COUNT all objects explicitly
2.  CHECK spatial relationships
3.  VERIFY colors, materials, and
other specific details
4.  IDENTIFY correct objects to
retain and wrong objects to remove
5.  REFLECT on previous attempts -
making progress or stuck?
6.  If instruction failed multiple
times, try simpler language

STEP 3 - DECISION:
Choose ONE action:

ACTION: EDIT_IMAGE
EDIT_INSTRUCTION: [5-18 word
instruction, focus on ONE change]
SATISFIED: [features matching
request with counts]
TODO: [features still needed with
counts]

ACTION: BACKTRACK_TO_IMAGE
BACKTRACK_TO: [image number, e.g.,
"Image #2"]

ACTION: SATISFIED_COMPLETE
SATISFIED: [all requirements met
with verification]
```

Table 7. **VLM verification and planning prompt.** Structured template guiding the vision-language model through image description, comparison analysis, and action decision. This three-step reasoning naturally induces verification, subgoal decomposition, and content memory behaviors during trajectory generation.