REFTREE: SCALING REFERENCE-TO-IMAGE GENERATION VIA HIERARCHICAL IMAGE DECOMPOSITION

Anonymous authors
Paper under double-blind review



Figure 1: Qualitative results and supported reference controls of RefTree. Top: a challenging many-reference example **generated** with FLUX.2 [klein] Base 9B (Black Forest Labs, 2026) fine-tuned on our dataset; green boxes indicate that the entity is from one reference. Bottom: Different categories of reference images supported in our dataset.

ABSTRACT

Although open-weight image generation models increasingly support reference images as inputs, they lag substantially behind proprietary counterparts, particularly as references grow in number and form complex hierarchical dependencies. To address this bottleneck, we introduce RefTree, a scalable and extensible data engine for multi-reference image generation. RefTree recursively decomposes target images into trees of visual concepts, which guide collaborative agents to extract the concepts as reference images, verify their consistency, and compose reference-grounded prompts via bottom-up merging. Operating entirely without closed-source models, the engine yields RefTree-480K, a dataset of 480K verified samples with 3.72M reference images and up to 20 references per sample, spanning characters, objects, scenes, and abstract controls such as poses, attributes, and styles. In addition, we present RefTree-Bench, which evaluates multi-reference generation using tree-derived, fine-grained visual rubrics. Models fine-tuned on

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

RefTree-480K substantially improve reference consistency and prompt following, achieving state-of-the-art open-weight performance across four public benchmarks and RefTree-Bench and even matching Nano Banana Pro on two public benchmarks. Code, datasets, and models will be released.

1 INTRODUCTION

Visual intent is far easier to show with an image than to describe with text. An image can pin down a person's identity, the texture of an intricate pattern, or even an abstract control such as a pose, a layout, or an artistic style, which would take lengthy and still vague text to convey. Frontier image generation models such as GPT-Image-2, Nano Banana Pro, and Seedream 5.0 Pro (OpenAI, 2026; Google DeepMind, 2025; ByteDance Seed, 2026) therefore natively support *reference-to-image* generation, letting users interleave reference images with text to specify the visual concepts.

In contrast, open-weight models, despite rapid progress in text-to-image generation and image editing, remain far less reliable in this domain as the number of references grows (Chen et al., 2026; Huang et al., 2026). In this paper, we target two main obstacles: the scarcity of training data that expose models to long, heterogeneous, and interdependent reference inputs, and the lack of evaluation that can identify which reference, binding, or instruction requirement has failed.

First, open reference-to-image training data remain limited in scale, references per sample, and complexity: the public corpora in Table 1 contain at most 11 references per sample, and most cover only one or two reference types. Current large-scale datasets, such as MICo, UniPic 3.0, and MACRO (Wei et al., 2026b;a; Chen et al., 2026), usually adopt a *bottom-up* pipeline: they sample elements from category pools, such as people, garments, and objects, and ask a proprietary generator to render their composition. Although this can theoretically produce any combination, it is costly to scale, restricted by licenses, and passes the teacher's errors to the student: even GPT-Image-2 misses at least one reference in over 70% of our test cases with 15 or more references (Sec. 5.2). Another, less studied pipeline is *top-down*, which recovers references from completed images, either by extracting whole objects across video frames (Wu et al., 2025b) or by editing synthetic images from predefined templates (Huang et al., 2026). Such simple designs struggle to raise the number and difficulty of references, or to represent complex inputs such as several references controlling one entity or arbitrary combinations of control types.

However, compared with the limited coverage and reference counts of existing data, users may combine references more freely. A user may specify a woman with a portrait, dress her in the jacket from a second image, add the logo from a third image to that jacket, and change her hair color in text. References of different types and levels thus jointly control one entity, and some modify what others show. Such nested inputs are hard to cover with concept pools or fixed templates, and cannot be recovered by extracting whole objects alone.

To this end, we propose RefTree, an agentic workflow for scalable reference-to-image data generation and evaluation. Inspired by the decomposition of visual concepts into hierarchical trees (Vinker et al., 2023), RefTree takes the top-down route but represents each target image as a tree of visual concepts rather than a flat list of objects. Each node is a visual concept, either a concrete entity or an abstract control such as material, pose, or style, and each child node is either physically attached to its parent or a concept that modifies or controls it. Since any node can be given as an image, as text, or within its parent's reference, the tree can express diverse reference combinations, including several references that jointly control one entity, without hand-crafted templates. This tree is the shared context of four agents built on open-source models (Fig. 3). A *Tree Constructor* recursively builds the tree; an *Extractor* analyzes the tree context and precisely extracts each visual concept as a reference image without leaking other nodes; a *Verifier* checks each generated reference image; and a *Prompt Composer* merges the tree bottom-up, forming reference-grounded descriptions for subtrees all the way up to an instruction that binds every reference to its role. Because each agent solves only a local task on its subtree, the whole framework runs on open-source models alone.

The same tree also addresses the second obstacle, evaluation. We collect RefTree-Bench, a set of challenging cases with up to 20 references of mixed types. However, we find that end-to-end VLM judges become unreliable as the visual context grows, especially beyond ten references, a regime that existing benchmarks rarely reach (Wei et al., 2026b; Wu et al., 2025b; Chen et al., 2026;

Table 1: Comparison with existing datasets on scale and reference diversity.

Dataset	#Samples	#Ref.	#Ref. Type	Avg. Ref.	Max. Ref.	Avg. Prompt Length
MICo-150K	148.6K	491.8K	1	3.31	11	45.7
MACRO	100.0K	584.2K	2	5.84	10	51.4
OmniRef	14.6K	59.6K	4	4.08	8	52.5
UniPic3	215.0K	870.8K	1	4.05	6	37.9
Echo-4o	72.8K	195.8K	2	2.69	4	58.4
Ours	**480.2K**	**3,723.0K**	**4**	**7.75**	**20**	**165.0**

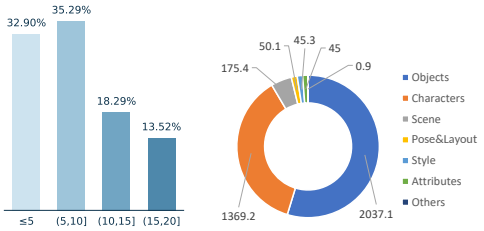


Figure 2: **Statistics of RefTree-480K.** Left: Reference number distribution. Right: References per category (in thousands).

Huang et al., 2026). Based on the reference tree, we therefore apply a divide-and-conquer strategy: the evaluation is decomposed into a rubric for each reference and a checklist for the prompt, so every score can be traced to a specific reference or requirement. Under the same GPT-5.5 judge, this protocol raises agreement with human ratings from 70.3% to 79.2% compared with end-to-end evaluation (Sec. 5.4).

With the RefTree engine, we build RefTree-480K purely from open-source models: a corpus of 480K samples with 3.72M reference images and 7.75 references per sample on average (up to 20), 2.23× and 4.28× the size of the largest public corpora in samples and reference images (Table 1). Fine-tuning two open-weight models, FLUX.2 [klein] and Cosmos3-Nano (Black Forest Labs, 2026; NVIDIA, 2026), on RefTree-480K yields the best open-weight results on four public benchmarks and on our RefTree-Bench. Fig. 1 illustrates how the fine-tuned models compose many heterogeneous references, from characters and scenes to poses and layouts, into one coherent image. Relative to the FLUX.2 [klein] base checkpoint, fine-tuning improves the MacroBench and OmniRef-Bench scores by 64.8% and 30.4%, and both fine-tuned models perform on par with Nano Banana Pro on these two benchmarks. Our contributions are as follows:

- We propose RefTree, an agentic data engine that decomposes images into reference trees and generates multi-reference training data at scale using only open-source models.
- We build RefTree-480K, with up to 20 references of mixed types per sample, and RefTree-Bench, a challenging benchmark with traceable tree-based evaluation.
- Fine-tuning two open-weight models on RefTree-480K yields the best open-weight results on four public benchmarks and RefTree-Bench.

2 Related Work

Reference-to-Image Generation Models. Early methods adapt pretrained text-to-image models to one or a few reference concepts through fine-tuning, visual encoders, or feature injection (Gal et al., 2023; Ruiz et al., 2023; Kumari et al., 2023; Ye et al., 2023; Xiao et al., 2023; Ma et al., 2023; Wang et al., 2025). With the development of MM-DiT and unified multimodal models, recent work merges reference conditioning, instruction following, and editing into a single model that accepts interleaved text and a variable number of images (Xiao et al., 2025; Han et al., 2024; Deng et al., 2025; Black Forest Labs et al., 2025; Wu et al., 2025b; Wei et al., 2026a; Huang et al., 2026).

Data and Evaluation for Multi-Reference Generation. Training data are constructed either bottom-up or top-down. *Bottom-up* pipelines sample references from category pools, write a composition instruction using an LLM or VLM, and render the target with a strong teacher model (Ye et al., 2025; Wei et al., 2026b;a; Chen et al., 2026; Shi et al., 2025), so the targets inherit the teacher's omissions, identity drift, and binding errors. *Top-down* pipelines extract references from completed images or video frames through grounding, segmentation, outpainting, or image editing (Xiao et al., 2025; Wu et al., 2025b; Chen et al., 2025; Huang et al., 2026; Liu et al., 2025), but return flat entity sets or follow fixed task schemas. Concept decomposition (Vinker et al., 2023) and Break-A-Scene (Avrahami et al., 2023) also extract multiple concepts from an image, but through per-image optimization rather than as standalone reference images. For evaluation, most benchmarks use *holistic* judges that inspect all inputs and the output together (Wu et al., 2025b; Chen et al., 2026; Huang et al., 2026), which becomes hard to audit as the visual context grows. Decomposed protocols assess each reference in turn with the help of specialist face and pose models (Wei et al., 2026b; Huang

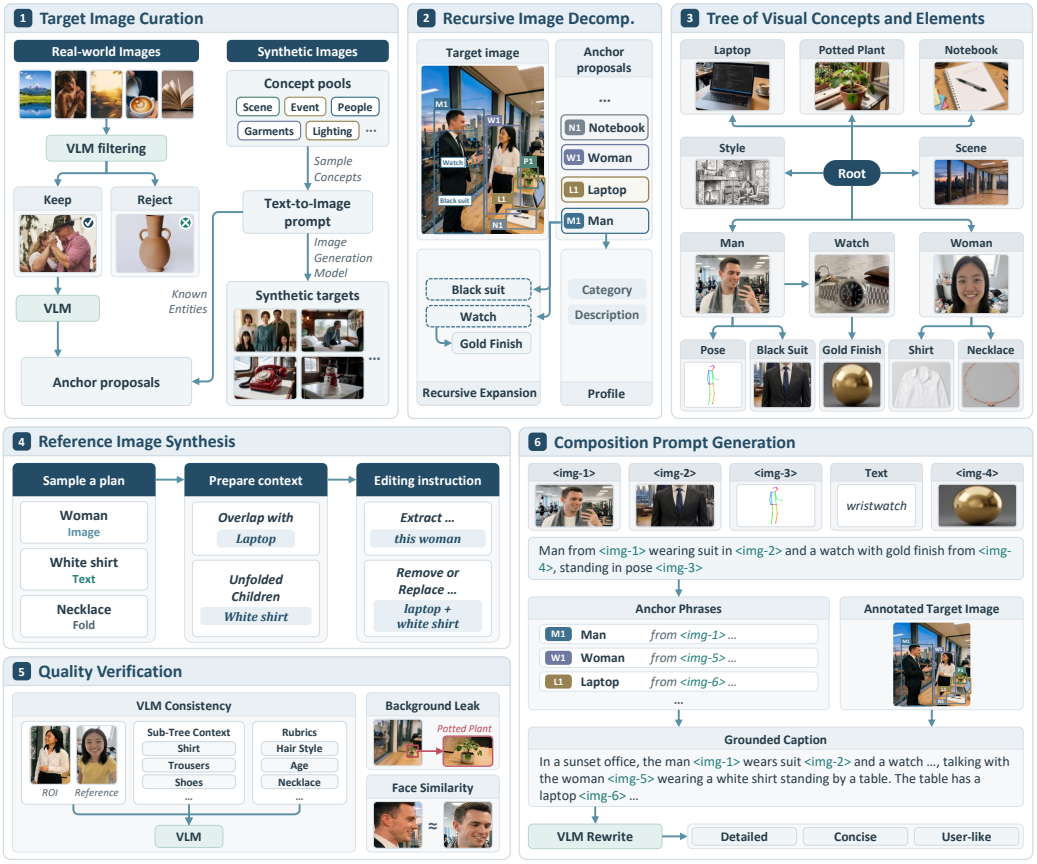


Figure 3: **Overview of the RefTree data engine**. Started from the curated images, a Tree Constructor proposes anchor entities and recursively expands anchor proposals, and organizes all visual concepts into a tree in which each child is physically attached to its parent or modifies it. The Extractor reads the local tree context and compiles editing instructions that extract the concept as a reference image without leaking other nodes. Then, the Verifier checks each reference; unreliable references fall back to text. Finally, the Prompt Composer merges subtrees bottom-up into a reference-grounded instruction, which is rewritten in several styles.

et al., 2026), and other benchmarks target rare concepts, multilingual prompts, or compositional reasoning (Oshima et al., 2025; Wu et al., 2026); none covers cases with as many as 20 references. RefTree instead recursively models attachment and modification relations to construct training data; for evaluation, it follows question-based prompt checklists (Hu et al., 2023; Cho et al., 2024) and reuses the same tree to derive a rubric that assigns each attribute to the reference responsible for it, making every score traceable.

3 RefTree: Data Engine Based on Tree of Visual Concepts

A top-down data engine starts from a completed target image and extracts its visual concepts as reference images. In practice, a user's inputs are more than a flat list of objects: several references often represent a single visual concept together. A person, for example, may be specified jointly by an identity image, a garment image, and a pose skeleton. The inputs thus form a structured, hierarchical mixture of visual concepts, in which some concepts belong to or modify others.

Inspired by hierarchical concept decomposition (Vinker et al., 2023), we decompose each target image into a tree of visual concepts that mirrors this structure. Based on this tree, we propose RefTree, an agentic data engine that organizes a series of agents around it (Fig. 3). Given a curated target (Sec. 3.1), a *Tree Constructor* first builds the tree (Sec. 3.2). An *Extractor* then reads the tree context to extract each visual concept as a reference image while excluding the content of all other nodes, so that no reference leaks information assigned to another, and a *Verifier* checks every extracted reference (Sec. 3.3). Finally, a *Prompt Composer* merges the tree bottom-up into a

reference-grounded instruction (Sec. 3.4). Because each agent reads only the local subtree it needs, every step is a local, verifiable task that open-source VLMs and image editors can handle reliably.

3.1 Target Image Curation

We curate two complementary target pools. *Real* images come from the real-image subset of the open-source Fine-T2I dataset (Ma et al., 2026) and provide natural photographic statistics, interactions, and diverse identities. A two-stage filter removes low-quality images, such as low-resolution, watermarked, or collage images, as well as images with simple compositions or no salient referenceable subjects, such as single-object photos, landscapes, and aerial views. *Synthetic* images extend coverage to long-tail concepts and complex events: we sample a structured scene specification, covering the setting, activity, lighting, characters, and objects, under plausibility constraints, and a VLM converts it into a text-to-image prompt and, after rendering, confirms that every requested entity and the event are present.

3.2 Tree Constructor: Recursive Image Decomposition

The Tree Constructor recursively decomposes a curated target into a tree of visual concepts. Each edge records either *attachment*, where the child is physically attached to its parent or is part of it, or *modification*, where the child modifies or controls how its parent is rendered.

Node types. *Entity nodes* are concrete entities, such as people, animals, and objects, and may have further entities attached to them. *Control nodes* are abstract controls: they either imply a visual pattern of their parent, such as material or texture, or act as high-level controls, such as pose, illumination, or artistic style, and they modify either a local node or the whole image at the root. A *scene node* represents the environment. In Fig. 3 (3), the man has an attached suit and watch and a pose control, the watch has a gold-finish control.

Recursive decomposition. The Constructor first proposes and grounds the salient people and objects in the target, or reuses the sampled concepts of a synthetic target; the verified entities, termed *anchors*, form the first layer of the tree. It then expands each entity node recursively: a VLM inspects the node's crop in the context of the full image, proposes attached objects, parts, and attributes such as material or texture, and assigns each a type. Human poses are extracted with a keypoint detector (He et al., 2017). Recursion stops when no remaining concept is both visually meaningful and independently specifiable.

Profiling and validation. Each node is profiled with its category, distinctive visual cues, visibility, and spatial support. We then remove unreliable nodes, such as those that are too small, heavily occluded, or incorrectly grounded, and keep concepts that cannot be cleanly extracted as text-only nodes. The resulting tree is the shared context of the workflow: any agent can directly retrieve a concept's profile and its relations to other concepts without re-analyzing the image.

3.3 Extractor: Leakage-Aware Reference Synthesis

Modern image editors can readily extract a concept from the target. The difficulty lies in writing an instruction that extracts exactly that concept while excluding every concept assigned to another reference or to text; otherwise, the leaked content lets the trained model bypass the input that actually specifies it. The Extractor therefore needs to compile each editing instruction from the tree context.

Composition plans. Which concepts must be excluded depends on how the user specifies each node. A *composition plan* assigns every node one of three states: *image* (an independent reference), *text* (described in the prompt), or *folded* (kept within the reference of its nearest image ancestor). In Fig. 3 (4), the woman is an image node; her folded necklace must appear in her reference, whereas her white shirt, specified in text, must not. Sampling different plans on the same tree varies reference count, modality mixture, and dependency depth without hand-crafted templates.

Context-aware instruction compilation. For each image node, the Extractor queries the tree for an editing context: the profile to preserve, the non-folded descendants to remove or replace, and other nodes overlapping its region. The node is then routed to a type-specific instruction compiler. For entity nodes, the compiler places the concept on a new background and removes or replaces the excluded concepts; for the woman, the instruction extracts her with her folded necklace while

removing or replacing the overlapping laptop and her white shirt. For attribute controls, it writes two instructions: one transfers the attribute to a new carrier, and the other flips the attribute on the parent while preserving identity. A gold watch thus yields a gold material ball as the control reference and a silver version of the same watch, so the trained model must take the shape from one reference and the finish from the other. The scene reference is obtained by removing all anchors and inpainting the background, while pose references are rendered from detected keypoints and optionally restyled as sketches. Each compiler runs in either a *plain* mode with conservative edits or an *aggressive* mode that also varies viewpoint, lighting, rendering style, and surroundings, discouraging the model from equating reference consistency with pixel copying. In practice, we can add more skills to this workflow to expand more types of visual concepts and corresponding extractor, for example, one can write a light condition extractor that extract the light setting into a standalone control image.

Verification and fallback. Open-source editors may alter identity or fail to remove excluded content. The *Verifier* therefore converts each editing instruction and its local tree context into a short rubric of what must be preserved and what must be absent, and scores the reference against the target region on a 0–3 scale. Human references additionally undergo an ArcFace face-similarity check (Deng et al., 2019), and scene references are checked for leaked foreground content. A node that fails verification falls back to text instead of injecting a noisy reference, and the prompt is composed under the updated plan.

3.4 Prompt Composer: Bottom-Up Instruction Generation

The final instruction is an image-tag-grounded caption: it binds each reference tag to the concept it controls and describes the spatial layout and interactions of the referenced concepts. We observe that a VLM given the full tree at once tends to take a shortcut, enumerating tags while omitting bindings, interactions, and spatial relations. The Prompt Composer therefore writes the prompt bottom-up, reversing the decomposition. It first merges the sampled subtree of each anchor into a local phrase that places every tag right after the concept it controls, e.g., "the man `<img-1>` in the suit `<img-2>`, posed as in `<img-3>`, wearing a wristwatch with the gold finish of `<img-4>`." A VLM then receives these phrases, the scene and global controls, and the target annotated with anchor boxes, and only needs to describe the environment, layout, and interactions among anchors. Finally, the detailed caption is rewritten into *concise* and *user-like* variants that preserve all tags and bindings.

3.5 Data Statistics and Implementation Details

Data statistics. Table 1 compares our corpus with five public multi-reference datasets. RefTree-480K contains 480.2K samples and 3.72M reference images, $2.23\times$ and $4.28\times$ the largest baseline on each axis. Of these samples, 220K come from real targets and 260K from synthetic ones. The samples also have substantially longer contexts: they contain 7.75 references on average, the highest among all datasets, and up to 20, $1.82\times$ the public maximum, with 31.8% of samples exceeding ten references; their prompts average 165.0 tokens, $2.82\times$ the longest public average, reflecting explicit descriptions of bindings, layouts, and interactions. Beyond objects (54.72%) and characters (36.78%), the corpus covers scene, pose, style, and attribute references (Fig. 2). Notably, this scale is achieved without any proprietary model.

Implementation details. All VLM-based agents use Qwen3-VL-32B-Instruct (Bai et al., 2025). References are synthesized with both FireRed-Image-Edit-1.1 (FireRed Team, 2026) and FLUX.2 [klein] (Black Forest Labs, 2026) to expose the data to different editing priors, and synthetic targets are rendered with Krea-2 (Lee et al., 2026) for its reliable prompt following capabilities.

4 RefTree Benchmark

Like existing training data, public benchmarks contain at most ten references per case and rarely mix reference types (Oshima et al., 2025; Chen et al., 2026; Huang et al., 2026). We therefore introduce RefTree-Bench, a challenging benchmark of 1,006 cases with 10,179 reference images in total. Each case contains 10.1 references on average and up to 20, with 35.3% of cases exceeding ten, and its instruction averages 341.6 tokens, compared with 82.9 in MacroBench. Because end-to-end VLM judges become unreliable for such long and interdependent contexts (Sec. 5.4), we evaluate each case with a divide-and-conquer protocol built on the reference tree. It proceeds in two steps: we

first profile the case by constructing its reference tree and decomposing its prompt into a verifiable checklist, and then score each reference, prompt following, and pixel quality separately.

Test case construction. We collect test cases from two sources. *Real* cases come from photographs and cover realistic photographic effects, such as bokeh effect and occlusion. GPT-5.5 and human annotators select the visual concepts that can serve as references, and Nano Banana Pro (Google DeepMind, 2025) extracts each as a reference image. *Synthetic* cases are designed to be challenging: GPT-5.5 composes each case freely, specifying every entity with its attributes and attachments, as well as the overall prompt. Because evaluation requires no target image, Nano Banana Pro generates only the references, so these cases can include extreme settings that are rare in existing data, such as scenes with ten or more people. To avoid overlap with training, real images are drawn from outside the training pool with near-duplicate filtering, synthetic cases use visual concepts disjoint from training, and neither GPT-5.5 nor Nano Banana Pro is used in our data engine. All cases are reviewed by annotators.

Tree-based case profiling. For each case, we construct the reference tree from the reference images and prompt directly, without using a target image. A VLM first describes each reference and then infers all dependencies among them from the prompt. Once the tree is constructed, we derive each reference's visual rubrics from the tree, focusing on its assigned attributes and excluding those assigned to its sub-references to avoid penalizing intended modifications from other references. We also merge each anchor's subtree into an *in-scene description* of the entity's intended appearance in the output, which the evaluator uses to localize it. Finally, we decompose the prompt into a checklist of verifiable requirements to reduce ambiguity and focus the evaluator on relevant requirements. For each case, all models are evaluated using the same reference tree and checklist, both verified by human annotators.

Reference consistency. For each reference, we first localize the corresponding entity in the output using its in-scene description. A VLM then scores the entity against the reference under its rubric on a 0–3 scale, given the full output with the entity's bounding box and a zoomed crop of the region. Global controls, such as style, and references whose localization fails are scored on the full image, and entities absent from the output receive 0. The 0–3 scores are averaged and linearly rescaled to 0–10 for reporting. For human references, we also report the ArcFace (Deng et al., 2019) cosine similarity between the reference face and the localized output face; for pose controls, we estimate keypoints on the corresponding output person and compute PCK@0.2 against the reference pose.

Prompt following and pixel quality. Prompt following is the fraction of checklist items satisfied per case. Pixel quality is rated separately from reference fidelity with aesthetic and technical-quality scores, plus a photorealism score for photographic cases. Prompt-following and pixel-quality scores are averaged over cases, whereas reference-level metrics are pooled over eligible references.

5 Experiments

We train two models on RefTree data: a specialist model initialized from FLUX.2 [klein] Base 9B (Black Forest Labs, 2026), which already supports reference-to-image generation, and a unified model initialized from Cosmos3-Nano (NVIDIA, 2026), a 16B Mixture-of-Transformers model without task-specific post-training for multi-reference composition. Training details are provided in Appendix B.1.

5.1 Comparison on Public Benchmarks

We evaluate on four public benchmarks: MICo-Bench (Wei et al., 2026b), MacroBench (Chen et al., 2026), OmniContext (Wu et al., 2025b), and OmniRef-Bench (Huang et al., 2026). Hereafter, *MACRO* denotes the MacroBench customization track and *OmniRef* the OmniRef-Bench MLLM average score.

Table 2 shows that the RefTree-trained models establish the best open-weight result on every public benchmark. FLUX.2 [klein]+RefTree leads on MICo and OmniContext, while Cosmos3-Nano+RefTree leads on MACRO and OmniRef, where it matches Nano Banana Pro (8.83 vs. 8.80) and surpasses it (8.65 vs. 8.50), respectively. RefTree supervision thus remains effective across substantially different benchmark formats. Starting from the same FLUX.2 [klein] Base 9B checkpoint, RefTree training raises MACRO from 5.31 to 8.75 ($+64.8\%$) and OmniRef from 6.49 to 8.46 ($+30.4\%$), the two largest relative gains among the public benchmarks.

Table 2: **Quantitative Comparison.** RC, PF, and PQ denote reference consistency, prompt following, and pixel quality, respectively. † indicates reported results from original papers.

Model	Public Benchmarks				RefTree-Bench				
	MICo	MACRO	OmniContext	OmniRef	RC	PF	PQ	Face	Pose
Closed-source models									
GPT-Image-2	70.35	9.63	9.57	9.04	8.56	0.87	8.30	0.58	0.45
Nano Banana Pro	62.83	8.80	9.12	8.50†	7.68	0.85	8.21	0.33	0.45
Open-weight models									
BAGEL	19.30†	3.95†	6.27†	4.64†	3.69	0.45	6.66	0.10	0.32
OmniGen2	31.40†	3.74†	7.18†	3.79†	3.13	0.30	7.35	0.22	0.22
Qwen-Image-Edit-2511	42.47	3.67†	8.22†	4.97†	3.53	0.20	5.95	0.22	0.27
DreamOmni2	29.67	1.93	6.60†	5.18†	2.42	0.24	6.62	0.06	0.22
FLUX.2 [klein] Base 9B	57.41	5.31	8.55	6.49†	5.37	0.65	7.82	0.18	0.43
FLUX.2-dev	59.46	5.29	8.66	6.12	5.60	0.66	7.76	0.27	0.44
HiDream-O1	55.04	4.84	8.04	5.80	6.52	0.62	7.35	0.15	0.47
FireRed-Image-Edit-1.1	49.87	4.04	7.77	5.51	5.27	0.53	7.82	0.27	0.44
Open-weight models post-trained on public multi-reference data									
BAGEL + MICo	34.40†	4.58†	7.74	4.59	4.11	0.48	7.10	0.10	0.37
Qwen-Image + MICo	35.90†	3.65	7.91	5.17	5.09	0.58	7.72	0.06	0.43
BAGEL + MACRO	48.24	8.02†	8.00†	5.63	4.83	0.47	7.54	0.15	0.36
Qwen-Image-Edit-2511 + MACRO	48.29	6.50†	8.45†	5.77	2.62	0.27	7.05	0.06	0.19
Qwen-Image-Edit-2511 + DyRef	49.95	4.72	8.24†	8.38†	4.78	0.40	7.45	0.14	0.47
Ours									
Cosmos3-Nano + RefTree	57.09	**8.83**	8.71	**8.65**	**7.29**	0.72	7.88	0.23	0.45
FLUX.2 [klein] Base 9B + RefTree	**61.91**	8.75	**8.77**	8.46	7.22	**0.75**	**8.14**	**0.35**	**0.48**

Table 3: **All-reference success rate (%) on RefTree-Bench.**

Model	#References		
	≤10	11–14	≥15
GPT-Image-2	86.1	64.3	29.9
Nano Banana Pro	81.8	62.5	24.5
HiDream-O1	51.5	24.5	3.4
FireRed-Image-Edit-1.1	37.6	11.1	0.7
FLUX.2-Klein-9B	54.1	24.0	2.7
Cosmos3-Nano + RefTree (Ours)	68.8	34.6	7.5
FLUX.2 [klein] Base 9B + RefTree (Ours)	**69.1**	**43.3**	**10.2**

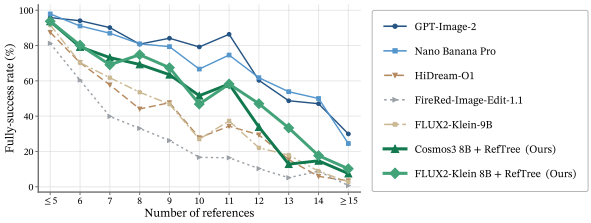


Figure 4: **All-reference success rate curve.** The first and last points pool cases with ≤5 and ≥15 references.

5.2 Results on RefTree-Bench

Both RefTree-trained models achieve the strongest open-weight reference consistency, with scores of 7.29 and 7.22. FLUX.2 [klein]+RefTree additionally achieves the best open-weight prompt following, pixel quality, and face similarity. The strongest prior open-weight model, HiDream-O1, reaches 6.52 in reference consistency but lags in face similarity (0.15) and prompt following (0.62), while most other baselines remain around 5 or below. Closed-source systems remain ahead, with GPT-Image-2 achieving the best score on every metric except pose. Our models nevertheless narrow the gap to Nano Banana Pro: FLUX.2 [klein]+RefTree slightly exceeds it in face similarity (0.35 vs. 0.33) and pose accuracy (0.48 vs. 0.45), while prompt following remains the largest gap.

Scaling with reference count. To examine how performance scales with reference count, we define *all-reference success* as a case in which every reference receives a nonzero rating. Our RefTree models substantially exceed the best open-weight baseline in every reference-count range (Table 3). Nevertheless, all systems degrade as references increase (Fig. 4). Even GPT-Image-2 and Nano Banana Pro fall below 60% at every count above 12; for $\geq$ 15 references, their pooled rates are 29.9% and 24.5%. Under a stricter criterion requiring every reference to receive a rating of at least 60% of the maximum score, these rates fall to 6.1% and 1.4%. This decline suggests that distilling closed systems at high reference counts is challenging and costy.

5.3 Contributions of Different Data Sources

To isolate the contribution of each data source, we train five mixtures with the same Cosmos3-Nano backbone and optimization budget (Table 4). Configuration A uses public multi-reference data from

Table 4: **Data-mixture ablation.**

ID	Training			Public Bench.		RefTree-Bench				
	Pub.	Real	Syn.	MACRO	OmniC.	RC	PF	PQ	Face	Pose
A	✓			8.22	8.44	5.92	0.67	7.44	0.08	0.41
B		✓		5.24	8.29	5.75	0.54	7.71	0.13	0.22
C			✓	8.32	8.58	7.12	0.72	7.85	0.19	0.43
D		✓	✓	8.83	**8.71**	**7.29**	0.72	7.88	0.23	**0.45**
E	✓	✓	✓	**8.93**	8.68	7.25	**0.73**	**7.89**	**0.24**	0.44

Table 5: Agreement (%) and r with human ratings.

#Ref.	GPT-5.5				Open model (Qwen3.8-VL)			
	RefTree		E2E		RefTree		E2E	
	Agr.	r	Agr.	r	Agr.	r	Agr.	r
< 10	**83.7**	**0.86**	79.5	0.84	**75.9**	**0.82**	72.8	0.75
≥ 10	**76.7**	**0.72**	65.4	0.64	**65.0**	**0.63**	53.2	0.48
Overall	**79.2**	**0.76**	70.3	0.63	**68.8**	**0.75**	60.1	0.53

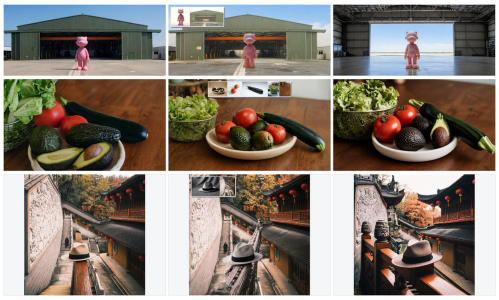


Figure 5: Qualitative data-mixture ablation. From left to right: full (real, and synthetic), public-only, and synthetic-only training. Insets show the shared references.

MICo-150K, MACRO, and OmniRef; B uses only the RefTree real branch; C uses only the RefTree synthetic branch; D combines the real and synthetic branches; and E combines all three sources: public, real, and synthetic data.

RefTree data versus public data. Comparing A with D, the complete RefTree data outperforms public-only training on every displayed metric, especially on reference consistency (+23.1%) and face similarity (0.08→0.23). The public-only model is also prone to a copy-and-paste effect: in the first row of Fig. 5, the toy is inserted without contact or cast shadow. Comparing D with E, adding public data yields only marginal changes (at most 0.10 on any metric), indicating that our data does not rely on public data.

Synthetic data versus real data. Comparing B with C, synthetic data alone are more effective, but real data are complementary: they strengthen basic abilities such as identity preservation and photorealism. Adding real data to synthetic data (from C to D) improves face similarity (+21.1%) and reference consistency (+2.4%). Synthetic-only training also yields oversaturated colors and wrong geometry, such as the implausible railing in the third row of Fig. 5, whereas the full mixture better harmonizes references with scene lighting and geometry.

5.4 Agreement with Human Judgments

Protocol. We compare two judges, GPT-5.5 and Qwen3.8-VL (27B), under two strategies: our tree-based evaluator, which scores one reference at a time with its rubric, and an end-to-end evaluator adapted from MACRO (Chen et al., 2026). We measure their agreement and Pearson correlation with 1,500 human ratings. Human raters were provided with the prompt, generated image, reference images, reference tree, and rubrics to verify whether each reference was correctly generated.

Results. As shown in Table 5, our evaluator outperforms end-to-end evaluation for both judges and all reference-count groups, improving overall agreement by 8.9 and 8.7 points and correlation by 0.13 and 0.22 for GPT-5.5 and Qwen3.8-VL, respectively. The gains are larger for cases with at least ten references. With our protocol, Qwen3.8-VL nearly matches end-to-end GPT-5.5 in agreement (68.8% vs. 70.3%) and exceeds it in correlation (0.75 vs. 0.63). GPT-5.5 with our protocol performs best overall and is used for all main evaluations.

6 Conclusion

We introduced RefTree, a scalable reference-to-image data engine and evaluation benchmark. RefTree decomposes each target image into a reference tree of visual concepts, which serves as the shared context for agents that construct the tree, extract and verify references, and compose reference-grounded prompts, all without any proprietary model. The resulting RefTree-480K dataset covers up to 20 references of mixed types per sample, and the same tree enables traceable, rubric-based evaluation on RefTree-Bench. Two open-weight model trained on RefTree-480K improves reference consistency and prompt following on both public benchmarks and RefTree-Bench.

AI USE STATEMENT

We used AI tools to polish the manuscript and to implement part of the code. The research ideas and the design of the data engine are the authors' own.

ETHICS STATEMENT

RefTree combines real photographs with synthetic images. Our real photographs come from the subset of the open-source Fine-T2I dataset (Ma et al., 2026), which is released under the Apache-2.0 license; these images remain subject to the terms of their source platforms. Our planned release will document real and synthetic provenance and the applicable image, dataset, and model terms. Redistribution will be limited to material whose licenses and permissions allow it; where redistribution is not permitted, we will provide processing code and source-access information instead of image files.

REPRODUCIBILITY STATEMENT

We plan to release the RefTree data-engine code, training and evaluation code, RefTree-480K, RefTree-Bench, and trained model checkpoints.

REFERENCES

Omri Avrahami, Kfir Aberman, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. Break-A-Scene: Extracting multiple concepts from a single image. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–12, 2023. doi: 10.1145/3610548.3618154.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Black Forest Labs. FLUX.2 [klein]: Towards interactive visual intelligence. https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence, 2026. Accessed 2026-09-22.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, et al. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. URL https://arxiv.org/abs/2506.15742.

ByteDance Seed. Beyond generation, it understands design: Introducing Seedream 5.0 Pro. https://seed.bytedance.com/en/blog/beyond-generation-it-understands-design-introducing-seedream-5-0-pro, 2026. Accessed 2026-09-22.

Qi Cai, Jingwen Chen, Chengmin Gao, Zijian Gong, Yehao Li, et al. HiDream-O1-Image: A natively unified image generative foundation model with pixel-level unified transformer. *arXiv preprint arXiv:2605.11061*, 2026. URL https://arxiv.org/abs/2605.11061.

Ruoxi Chen, Dongping Chen, Siyuan Wu, Sinan Wang, Shiyun Lang, Petr Sushko, Gaoyang Jiang, Yao Wan, and Ranjay Krishna. MultiRef: Controllable image generation with multiple visual references. *arXiv preprint arXiv:2508.06905*, 2025. URL https://arxiv.org/abs/2508.06905.

Zhekai Chen, Yuqing Wang, Manyuan Zhang, and Xihui Liu. MACRO: Advancing multi-reference image generation with structured long-context data. *arXiv preprint arXiv:2603.25319*, 2026. URL https://arxiv.org/abs/2603.25319.

Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ITq4ZRUT4a.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. URL https://arxiv.org/abs/2505.14683.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019. URL https://arxiv.org/abs/1801.07698.

FireRed Team. FireRed-Image-Edit-1.1. https://github.com/FireRedTeam/FireRed-Image-Edit, 2026. Model release and code repository; accessed 2026-09-22.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2208.01618.

Google DeepMind. Introducing Nano Banana Pro (Gemini 3 Pro Image). https://blog.google/innovation-and-ai/products/nano-banana-pro/, 2025. Accessed 2026-09-22.

Zhen Han, Zeyinzi Jiang, Yulin Pan, Jingfeng Zhang, Chaojie Mao, Chenwei Xie, Yu Liu, and Jingren Zhou. ACE: All-round creator and editor following instructions via diffusion transformer. *arXiv preprint arXiv:2410.00086*, 2024. URL https://arxiv.org/abs/2410.00086.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, 2017. URL https://arxiv.org/abs/1703.06870.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. TIFA: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20349–20360, 2023. doi: 10.1109/ICCV51070.2023.01866.

Wenwang Huang, Yusen Fu, Junjie Wang, Mengfei Huang, Yulin Li, Gan Liu, Jing Cai, Yancheng He, and Zhuotao Tian. Scaling multi-reference image generation with dynamic reward optimization. *arXiv preprint arXiv:2606.26947*, 2026. URL https://arxiv.org/abs/2606.26947.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2212.04488.

Sangwu Lee, Erwann Millon, Le Zhuo, Matthew Newton, Andrei Filatov, et al. Krea 2 technical report. https://www.krea.ai/blog/krea-2-technical-report, 2026. Accessed 2026-09-22.

Yexin Liu, Manyuan Zhang, Yueze Wang, Hongyu Li, Dian Zheng, Weiming Zhang, Changsheng Lu, Xunliang Cai, Yan Feng, Peng Pei, and Harry Yang. OpenSubject: Leveraging video-derived identity and diversity priors for subject-driven image generation and manipulation. *arXiv preprint arXiv:2512.08294*, 2025. URL https://arxiv.org/abs/2512.08294.

Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion: Open-domain personalized text-to-image generation without test-time fine-tuning. *arXiv preprint arXiv:2307.11410*, 2023. URL https://arxiv.org/abs/2307.11410.

Xu Ma, Yitian Zhang, Qihua Dong, and Yun Fu. Fine-T2I: An open, large-scale, and diverse dataset for high-quality T2I fine-tuning. *arXiv preprint arXiv:2602.09439*, 2026. URL https://arxiv.org/abs/2602.09439.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026. URL https://arxiv.org/abs/2606.02800.

OpenAI. GPT-Image-2 model. https://developers.openai.com/api/docs/models/gpt-image-2, 2026. Accessed 2026-09-22.

Yuta Oshima, Daiki Miyake, Kohsei Matsutani, Yusuke Iwasawa, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. MultiBanana: A challenging benchmark for multi-reference text-to-image generation. *arXiv preprint arXiv:2511.22989*, 2025. URL https://arxiv.org/abs/2511.22989.

Qwen Team. Qwen-Image-Edit-2511. https://huggingface.co/Qwen/Qwen-Image-Edit-2511, 2025. Model card; accessed 2026-09-22.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2208.12242.

Xuanke Shi, Boxuan Li, Xiaoyang Han, Zhongang Cai, Lei Yang, Dahua Lin, and Quan Wang. ConsistCompose: Unified multimodal layout control for image composition. *arXiv preprint arXiv:2511.18333*, 2025. URL https://arxiv.org/abs/2511.18333.

Yael Vinker, Andrey Voynov, Daniel Cohen-Or, and Ariel Shamir. Concept decomposition for visual exploration and inspiration. *ACM Transactions on Graphics*, 42(6):1–13, 2023. doi: 10.1145/3618315. URL https://arxiv.org/abs/2305.18203.

Xierui Wang, Siming Fu, Qihan Huang, Wanggui He, and Hao Jiang. MS-Diffusion: Multi-subject zero-shot image personalization with layout guidance. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=PJqP0wyQek.

Hongyang Wei, Hongbo Liu, Zidong Wang, Yi Peng, Baixin Xu, Size Wu, Xuying Zhang, Xianglong He, Zexiang Liu, Peiyu Wang, Xuchen Song, Yangguang Li, Yang Liu, and Yahui Zhou. Skywork UniPic 3.0: Unified multi-image composition via sequence modeling. *arXiv preprint arXiv:2601.15664*, 2026a. URL https://arxiv.org/abs/2601.15664.

Xinyu Wei, Kangrui Cen, Hongyang Wei, Zhen Guo, Kai Cui, Bairui Li, Zeqing Wang, Jinrui Zhang, and Lei Zhang. MICo-150K: A comprehensive dataset advancing multi-image composition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b. URL https://arxiv.org/abs/2512.07348.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, et al. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025a. URL https://arxiv.org/abs/2508.02324.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, et al. OmniGen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025b. URL https://arxiv.org/abs/2506.18871.

Mingrui Wu, Hang Liu, Jiayi Ji, Xiaoshuai Sun, and Rongrong Ji. MICON-Bench: Benchmarking and enhancing multi-image context image generation in unified multimodal models. *arXiv preprint arXiv:2602.19497*, 2026. URL https://arxiv.org/abs/2602.19497.

Bin Xia, Bohao Peng, Yuechen Zhang, Junjia Huang, Jiyang Liu, et al. DreamOmni2: Multimodal instruction-based editing and generation. *arXiv preprint arXiv:2510.06679*, 2025. URL https://arxiv.org/abs/2510.06679.

Guangxuan Xiao, Tianwei Yin, William T. Freeman, Frédo Durand, and Song Han. FastComposer: Tuning-free multi-subject image generation with localized attention. *arXiv preprint arXiv:2305.10431*, 2023. URL https://arxiv.org/abs/2305.10431.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. OmniGen: Unified image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2409.11340.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. URL https://arxiv.org/abs/2308.06721.

Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, et al. Echo-4o: Harnessing the power of GPT-4o synthetic images for improved image generation. *arXiv preprint arXiv:2508.09987*, 2025. URL https://arxiv.org/abs/2508.09987.

SUPPLEMENTARY MATERIAL

"REFTREE: SCALING REFERENCE-TO-IMAGE GENERATION VIA HIERARCHICAL IMAGE DECOMPOSITION"

TABLE OF CONTENTS

A Data-Engine Details **15**
A.1 Synthetic Target Generation 15
A.2 Node Proposal, Grounding, and Validation 15
A.3 Reference Verification 15

B Training and Evaluation Details **16**
B.1 Training Details 16
B.2 Evaluation and Baseline Details 16

C RefTree-Bench Details **16**
C.1 Human Annotator UI 16
C.2 Human Agreement Study 16

D Visualization **18**
D.1 Visualizations 18
D.2 Visualization of Baseline Comparison 27
D.3 Failure Cases of Frontier Models on RefTree-Bench 32

A Data-Engine Details

A.1 Synthetic Target Generation

Rather than independently drawing references from a few object-category pools, we factor a candidate image into a setting, scene, lighting, weather, activity, characters, and objects. These factors are sampled conditionally: activities must be plausible in the selected setting, weather and illumination must agree with the scene, and participant types, attributes, and counts are instantiated around the event. A VLM turns the sampled structure into a text-to-image prompt that explicitly describes both the elements and their interactions, after which Krea-2 (Lee et al., 2026) renders one target image. Because the requested concepts are known, a verifier checks that the required entities and event are present before the image enters the common decomposition pipeline, where the sampled concepts also serve as the initial anchor proposals.

A.2 Node Proposal, Grounding, and Validation

Anchor discovery. For a real image target, a VLM proposes the main characters and objects in the image; for a synthetic target, the sampled concepts provide the initial candidates. All proposals are grounded and checked in the image, and the surviving entities become anchor nodes.

Node taxonomy and profiles. Local controls include color, material, pattern, texture, finish, and human pose. Such concepts can still be expressed visually: the gold finish of a watch, for example, may be specified by a gold material sample or by a different gold object. Global controls modify the whole image, are children of the image root, and include artistic style, illumination and weather, sketch or draft guidance, and spatial layouts such as bounding-box canvases. Each node stores a compact profile containing its type, category, visual description, parent relation, and spatial support when applicable.

Recursive expansion. For each entity node, the VLM sees both a tight crop and the node highlighted in the complete target. It proposes visible sub-objects and distinctive attributes, such as material or texture, and assigns each a parent and a node type. Entity candidates are grounded within the parent region, while control candidates retain an explicit link to the entity node or image root they modify. Human pose is estimated with a keypoint detector (He et al., 2017); person-level keypoints define local pose nodes, and keypoints from multiple people can be rendered jointly as a global pose-and-layout control.

Node validation. Recursive proposal improves coverage but also produces regions that are too small, blurred, heavily occluded, incidental, or incorrectly grounded. We therefore validate each candidate using its profile, a tight crop, and the full target with the region highlighted. A VLM checks semantic agreement, visibility, and suitability as a standalone reference, while geometric filters reject undersized or degenerate regions. Human nodes receive additional face-visibility and face-quality checks. Spurious proposals are removed, whereas semantically useful but unextractable concepts are marked ineligible for the image state and retained only in text or in the scene description.

A.3 Reference Verification

The Verifier receives the synthesized reference, the corresponding target region, the editing instruction, and the same plan-conditioned subtree used by the compiler. From this context, it constructs a short rubric covering two questions: whether the identity-bearing features assigned to the node remain consistent, and whether independently controlled or overlapping content has been removed or changed. A VLM scores the result from 0 to 3; references scored 0 or 1 fall back to text, and the prompt is composed under the updated plan. Scene references receive an additional leakage check for residual anchors, and any residual foreground content triggers the same fallback. For eligible human nodes, we additionally compare ArcFace embeddings (Deng et al., 2019) of the target and reference faces.

B Training and Evaluation Details

B.1 Training Details

We train both models for 40K iterations on 64 NVIDIA GB200 GPUs, using Adam with a learning rate of 2×10^{-5} and a packed token budget of 96K per GPU. Cosmos3-Nano has broad omnimodal pretraining but no task-specific post-training for multi-reference composition. During training, we randomly permute the reference images and consistently remap their tags in the prompt, discouraging position-specific shortcuts. FLUX.2 [klein] Base 9B consists of an 8B Qwen3 text embedder and a 9B diffusion-transformer backbone. For Cosmos3-Nano, we freeze the autoregressive reasoner tower and optimize only the diffusion generator tower. The per-GPU packed token budget of 96K counts both text tokens and media tokens after VAE encoding and patchification. During training, we rescale each reference image by a factor sampled from $[0.3, 1.2]$ while preserving its aspect ratio. Training targets and test outputs preserve the source aspect ratio and are capped at a pixel area of 1024^2.

B.2 Evaluation and Baseline Details

Public benchmarks. MICo-Bench contains 897 cases and reports Weighted-Ref-VIEScore, which combines per-source preservation weights with semantic consistency and perceptual quality. MacroBench contains up to ten inputs; we report its customization track. OmniContext contains 400 cases evaluating prompt following and subject consistency. OmniRef-Bench contains up to 7 references drawn from 5 reference types; we report its MLLM average score.

Compared models. Table 2 compares three groups of models. *Closed-source systems*: GPT-Image-2 (OpenAI, 2026) and Nano Banana Pro (Google DeepMind, 2025). *General-purpose open-weight models*: BAGEL (Deng et al., 2025), OmniGen2 (Wu et al., 2025b), Qwen-Image-Edit-2511 (Qwen Team, 2025), DreamOmni2 (Xia et al., 2025), FLUX.2 [klein] Base 9B and FLUX.2-dev (Black Forest Labs, 2026), HiDream-O1 (Cai et al., 2026), and FireRed-Image-Edit-1.1 (FireRed Team, 2026). *Open-weight models post-trained on public multi-reference data*: BAGEL + MICo and Qwen-Image + MICo (Deng et al., 2025; Wu et al., 2025a; Wei et al., 2026b), BAGEL + MACRO and Qwen-Image-Edit-2511 + MACRO (Deng et al., 2025; Qwen Team, 2025; Chen et al., 2026), and Qwen-Image-Edit-2511 + DyRef (Qwen Team, 2025; Huang et al., 2026), where the suffix denotes the public data or recipe used for post-training. Our two models are initialized from Cosmos3-Nano (NVIDIA, 2026) and FLUX.2 [klein] (Black Forest Labs, 2026) and trained on RefTree-480K as described in Appendix B.1.

C RefTree-Bench Details

Statistics. RefTree-Bench contains 246 real, 400 synthetic, and 360 control cases, with 10,179 reference images in total and 10.1 per case on average. Each case has 2 to 20 references; 355 cases (35.3%) have more than 10.

C.1 Human Annotator UI

Annotators review every proposed case in the interface shown in Fig. 6. For each candidate reference, it shows the target crop next to the generated reference and its description; annotators keep or remove the candidate, correct its description, and draw regions for concepts that the pipeline missed.

C.2 Human Agreement Study

GPT-5.5 is the closed-source judge used in our main experiments, and Qwen3.8-VL (27B) is an open-weight alternative. The end-to-end evaluator gives the judge the complete visual context in one prompt and asks it to predict a score for each reference, whereas our evaluator scores one reference at a time using its local subtree and rubric. Five annotators provide 1,500 ratings on a 0–3 scale. For each rating, an annotator sees the same evidence as the automatic evaluator: the relevant subtree, the output with the region of interest marked, and the reference image. We report agreement (Agr.),

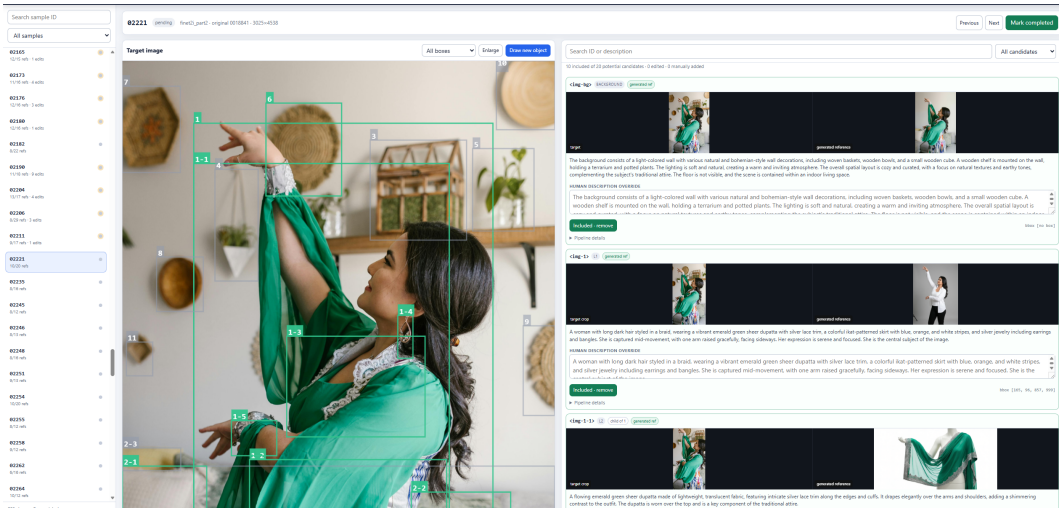


Figure 6: Annotation interface for reviewing RefTree-Bench cases, shown on a real case. The target image (middle) is overlaid with the proposed regions; each candidate reference (right) is shown with its target crop, generated reference, and editable description. Human annotators can create new reference candidates, modify descriptions, and delete existing ones.

the fraction of ratings on which the automatic score exactly matches the human score, and Pearson correlation (r) between automatic and human scores, both overall and stratified by reference count.

Fig. 7 shows the rating interface.

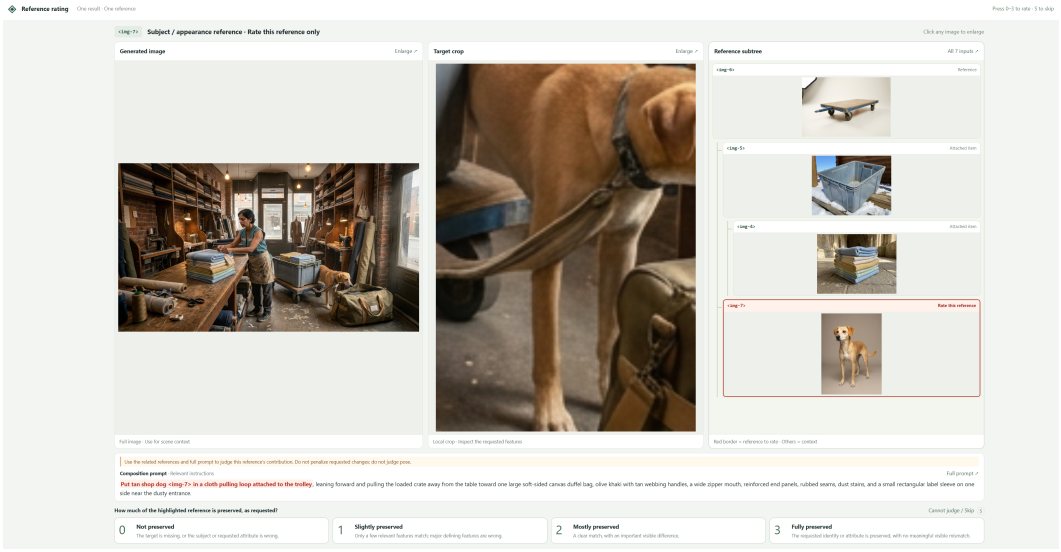


Figure 7: Rating interface for the human agreement study. Annotators see the generated image, a local crop, and the reference subtree with the rated reference highlighted in red, together with the relevant part of the prompt, and rate on a 0–3 scale how well the reference is preserved.

D VISUALIZATION

D.1 VISUALIZATIONS

Figures 8–15 show examples from RefTree-Bench generated by FLUX.2 [klein]+RefTree. Each figure shows the reference inputs in their original tag order, the generated image, and a condensed summary of the generation prompt.

Reference inputs





















Generated image



Prompt summary. Stage a shearing contest in a damp timber barn. The judge <img-1> wears the vest <img-2> and scarf <img-3>, holds the clipboard <img-7>, and reaches toward the hand bell <img-8>. The contestant <img-4> kneels beside the shorn sheep <img-5> and lifts the fleece bundle <img-9>; the shearing handpiece <img-10> rests nearby. The waiting sheep <img-6> stands behind a rope barrier.

Figure 8: **Shearing contest.** Reference inputs and generated output (10 references).

Reference inputs



<img-1>
Sample collector



<img-2>
Knit cap



<img-3>
Waterproof jacket



<img-4>
Field cataloger



<img-5>
Sample vial



<img-6>
Plankton net



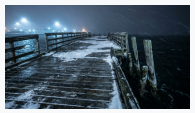
<img-7>
Sample cooler



<img-8>
Field notebook



<img-9>
Sampling bucket



<img-10>
Snowy pier

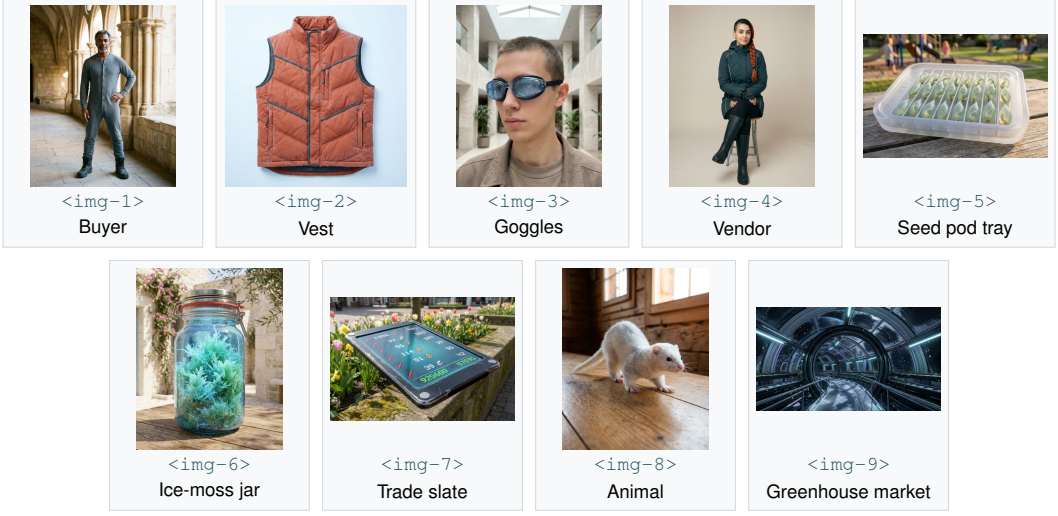
Generated image



Prompt summary. On the snowy pier <img-10>, the collector <img-1> wears the knit cap <img-2> and waterproof jacket <img-3> while passing the sample vial <img-5> to the cataloger <img-4>. Arrange the sampling bucket <img-9>, plankton net <img-6>, open cooler <img-7>, and field notebook <img-8> around the handoff under cold harbor lighting.

Figure 9: **Harbor sampling.** Reference inputs and generated output (10 references).

Reference inputs



Generated image



Prompt summary. In the orbital greenhouse market <img-9>, the buyer <img-1> wears the vest <img-2> and goggles <img-3> while bargaining over the seed pod tray <img-5>. The vendor <img-4> lifts the ice-moss jar <img-6> and taps the trade slate <img-7>; the animal <img-8> paws at the jar. An abacus, levitating scale, snowfall, and cool lighting complete the scene.

Figure 10: **Orbital market.** Reference inputs and generated output (9 references).

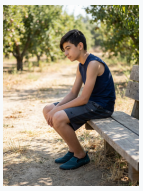
Reference inputs





















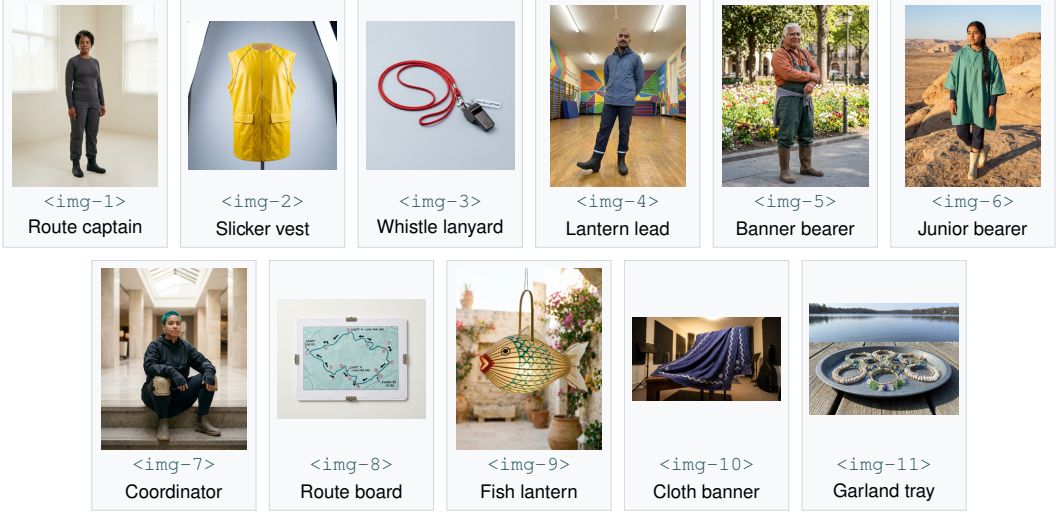
Generated image



Prompt summary. In the flooded ballroom <img-10>, the quartermaster <img-1> wears the relay vest <img-2> and utility belt <img-3> while coordinating three racers. The sprint racer <img-4> reaches for the baton <img-7>; the goggle runner <img-5> exchanges goggles while receiving the flotation sash <img-9>; and the fin racer <img-6> raises a foot for the swim fins <img-8>. Rain and golden reflections surround the exchange.

Figure 11: **Flooded-ballroom relay.** Reference inputs and generated output (10 references).

Reference inputs



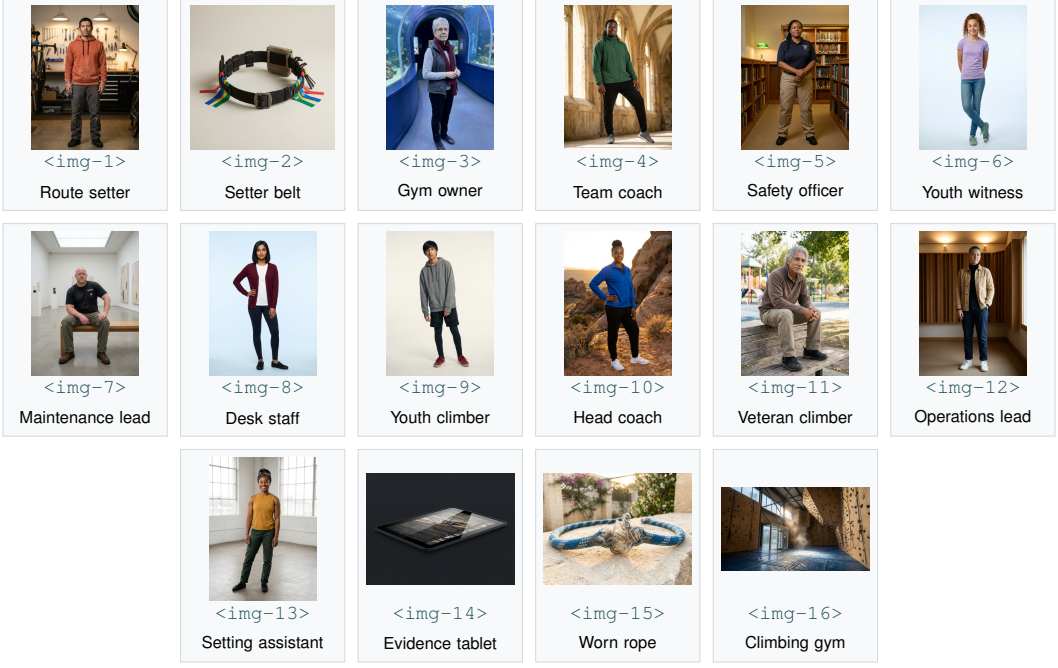
Generated image



Prompt summary. At a coastal field station, the captain <img-1>, wearing the slicker vest <img-2> and whistle lanyard <img-3>, points at the route board <img-8>. The lantern lead <img-4> holds the fish lantern <img-9>. The older bearer <img-5> and junior bearer <img-6> lift the cloth banner <img-10>, while the coordinator <img-7> ties a garland from the tray <img-11> onto the junior bearer's wrist.

Figure 12: **Coastal procession.** Reference inputs and generated output (11 references).

Reference inputs



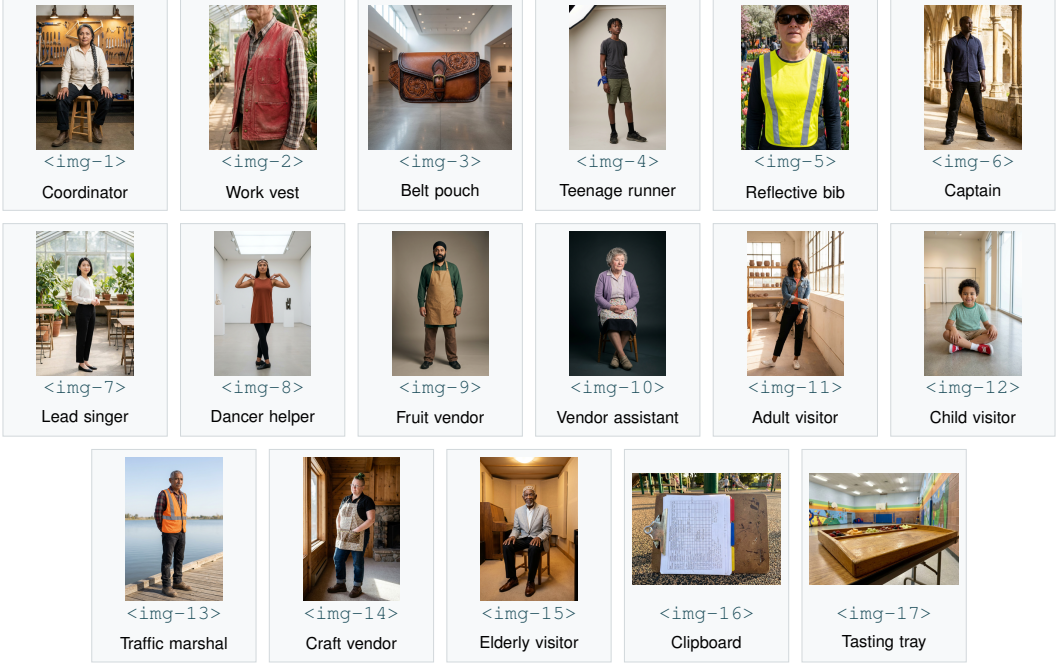
Generated image



Prompt summary. Gather twelve people beneath the overhang in the climbing gym <img-16>. The route setter <img-1> wears the belt <img-2> and presents the evidence tablet <img-14> beside the worn rope <img-15>. Arrange the staff, coaches, and climbers around him in distinct standing, seated, and kneeling poses, with a silver carabiner on the mat and daylight through rain-streaked windows.

Figure 13: **Climbing-gym evidence review.** Reference inputs and generated output (16 references).

Reference inputs



Generated image



Prompt summary. Form a communal handoff chain in an orchard packing yard at blue hour. The coordinator <img-1> passes the clipboard <img-16> through the runner <img-4> to the marshal <img-13>. The captain <img-6>, singer <img-7>, and dancer <img-8> relay a blue sash and apple crate toward the vendors. Share samples from the tasting tray <img-17> while collecting a donation in a small red cash box.

Figure 14: **Orchard handoff chain.** Reference inputs and generated output (17 references).

Reference inputs































Generated image



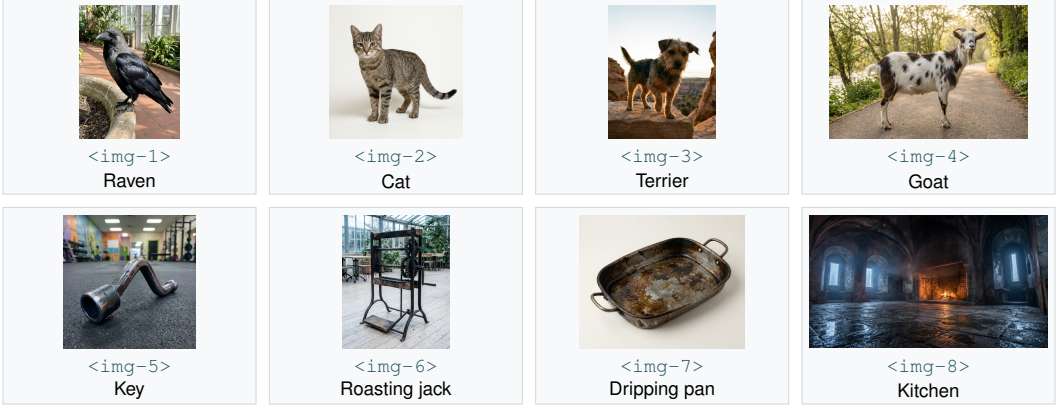
Prompt summary. Coordinate delayed boarding at a mountain cable-car station at blue hour. The supervisor <img-1> checks the manifest <img-14>, while the radio clerk <img-2> relays instructions and the operator <img-4> points to the delay display <img-15>. Keep the family together, guide the elderly traveler <img-7> through the lane, and place the translator <img-11> between the supervisor and rebooking passenger <img-13>.

Figure 15: **Mountain-station boarding groups.** Reference inputs and generated output (15 references).

D.2 Visualization of Baseline Comparison

Figures 16–19 show comparisons with other models, including HiDream-O1, Nano Banana Pro, and GPT-Image-2.

Reference inputs



Ours



HiDream-O1



Nano Banana Pro



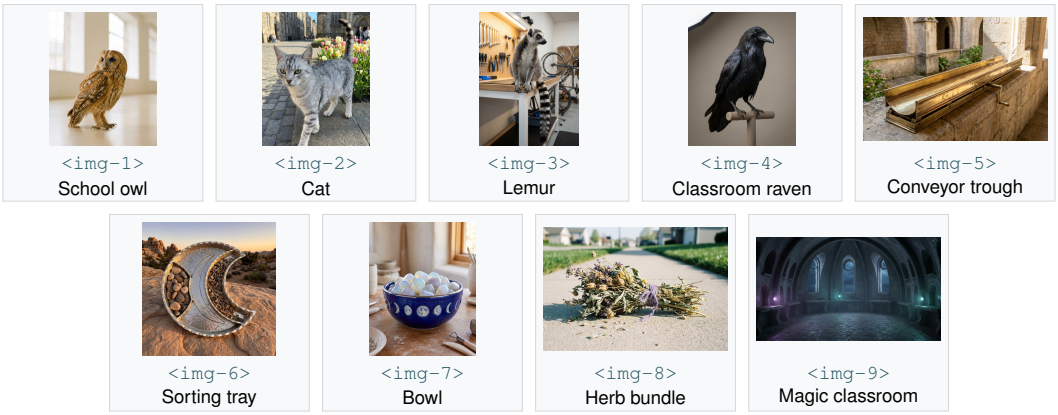
GPT Image 2



Prompt summary. In kitchen <img-8>, arrange raven <img-1>, cat <img-2>, terrier <img-3>, and goat <img-4> beside the hearth. Relay key <img-5> from the raven to the cat and terrier toward the goat, which presses the treadle of roasting jack <img-6>. Place dripping pan <img-7> below the spit in cold, damp pre-dawn firelight.

Figure 16: **Hearthside key relay.** RefTree-Bench (8 references).

Reference inputs



Ours



HiDream-O1



Nano Banana Pro



GPT Image 2



Prompt summary. In magic classroom <img-9>, owl <img-1> drops moonstones from bowl <img-7> into conveyor trough <img-5>, while lemur <img-3> feeds herbs from bundle <img-8>. At the downstream end, cat <img-2> flicks moonstones into the left compartment of sorting tray <img-6>, and raven <img-4> places herbs into the right compartment under violet, teal, and moonlit highlights.

Figure 17: **Moonstone sorting line.** RefTree-Bench (9 references).

Reference inputs



<img-1>
Near-side raven



<img-2>
Far-side raven



<img-3>
Game board



<img-4>
Near-side tokens



<img-5>
Far-side tokens



<img-6>
Crate



<img-7>
Border trail

Ours



HiDream-O1



Nano Banana Pro



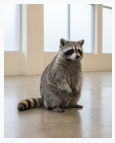
GPT Image 2



Prompt summary. On border trail <img-7>, place game board <img-3> on crate <img-6>. Near-side raven <img-1> nudges a token from <img-4> onto a central square with its beak; opposite it, raven <img-2> slides a token from <img-5> into a blocking square with one claw. Frame the birds and board tightly from trail height in gusty, overcast afternoon light.

Figure 18: **Ravens at the game board.** RefTree-Bench (7 references).

Reference inputs



<img-1>
Near-left raccoon



<img-2>
Center raccoon



<img-3>
Far-right raccoon



<img-4>
Sorting trough



<img-5>
Connectors



<img-6>
Spacers



<img-7>
Sorting tray



<img-8>
Lantern



<img-9>
Jar

Ours



HiDream-O1



Nano Banana Pro



GPT Image 2



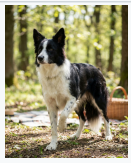
Prompt summary. In a damp glass-walled winter garden, arrange raccoons <img-1>, <img-2>, and <img-3> along sorting trough <img-4>. Pass connectors <img-5> paw to paw down the line into compartments of tray <img-7>. Place spacers <img-6> and jar <img-9> near the output end, with lantern <img-8> casting warm highlights across the paws, parts, and wet tiles.

Figure 19: **Raccoon assembly line.** RefTree-Bench (9 references).

D.3 Failure Cases of Frontier Models on RefTree-Bench

Figures 20–22 show three failure cases for Nano Banana Pro and GPT-Image-2 on RefTree-Bench. The examples highlight changes to referenced object structure, incorrect clothing and handoff roles, duplicated identities, and garments assigned to the wrong person.

Reference inputs







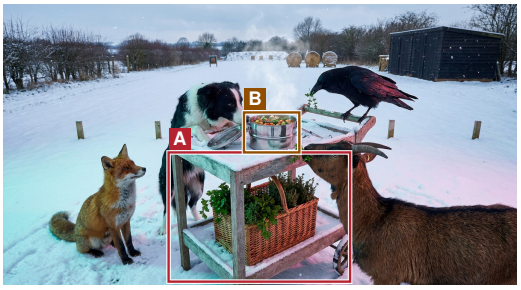








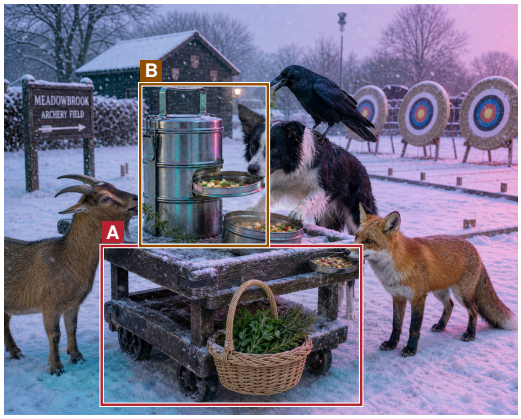
Nano Banana Pro



A Cart <img-5>: the low, flat platform becomes a tall cart with a lower shelf.

B Carrier <img-6>: the tall multi-tier carrier becomes a squat two-tier pot.

GPT Image 2



A Cart <img-5>: a raised upper deck is added to the low, flat platform.

B Carrier <img-6>: one stacked tier slides out sideways like a drawer instead of lifting off.

Requested scene. Use cart <img-5> and an opened tiffin carrier <img-6> for an animal breakfast station: the collie serves the goat and fox while the raven adds herbs.

Figure 20: **Object structure changes.** RefTree-Bench (7 references). Colored boxes and letters are annotated errors.

Reference inputs



<img-1>
Older volunteer



<img-2>
Red parka



<img-3>
Coordinator



<img-4>
Station runner



<img-5>
Passport



<img-6>
Lantern bundle

Nano Banana Pro



A Wrong parka: the older volunteer wears green instead of the red coat <img-2>.

B Wrong recipient: the passport goes directly to runner <img-4>, bypassing coordinator <img-3>.

GPT Image 2



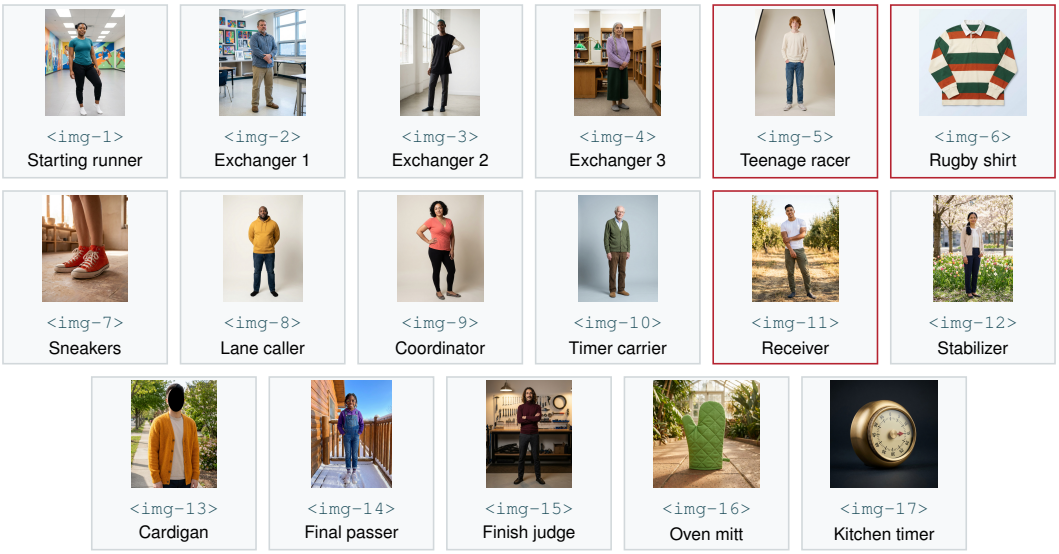
A Wrong parka: the older volunteer keeps a dark coat instead of wearing <img-2>.

B Swapped roles: runner <img-4> occupies the center and coordinator <img-3> stands at right.

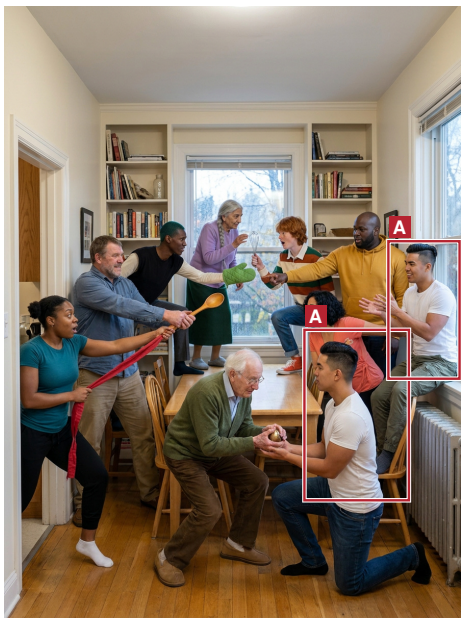
Requested scene. Dress volunteer <img-1> in parka <img-2>. Pass the passport from the left volunteer to coordinator <img-3> at center, then the lanterns to runner <img-4> at right; the runner offers the thermos back to the volunteer.

Figure 21: **Clothing and handoff roles are misbound.** RefTree-Bench (6 references). Colored boxes and letters are author annotations of selected errors.

Reference inputs

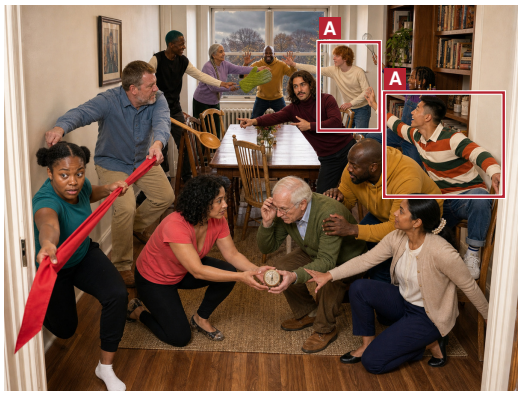


Nano Banana Pro



A Duplicated receiver: two men match <img-11>, one kneeling with the timer and another seated beside the right window.

GPT Image 2



A Wrong garment owner: rugby shirt <img-6> is worn by receiver <img-11>; teenage racer <img-5> remains in a plain light top.

Requested scene. Stage a twelve-person kitchen relay. Teenage racer <img-5> wears rugby shirt <img-6> and sneakers <img-7>; receiver <img-11> kneels to take the timer from carrier <img-10>.

Figure 22: **Identity duplication and garment misassignment.** RefTree-Bench (17 references). Colored boxes and letters are author annotations of selected errors.