MoGE-Bench: A Benchmark for On-device Diffusion Models

Anonymous authors
Paper under double-blind review

Abstract

Developing on-device image models requires considering practical user needs, task types and device resource constraints. However, existing benchmarks provide limited guidance on identifying capability gaps between on-device models and server-side models. We therefore introduce **MoGE-Bench**, a benchmark grounded in mobile user needs, comprising 64 subtasks and 1150 samples. It defines 16 ***operations***, such as removal and replacement, and eight visual ***target*** categories covering content such as portraits and text. These form two orthogonal axes, with applicable combinations yielding explicit task coverage. To better guide model improvement, we propose **MoGE-Eval**, which introduces task-specific ***criteria*** covering instruction following, visual quality and detail preservation. It yields overall quality scores and capability profiles along the operation, target and criteria axes. We evaluate and deploy on-device models, and compare their output quality with server-side models. Our results show that this decoupled *Operation* $\times$ *Target* design provides finer-grained guidance for improving compact models. For example, it highlights the need to prioritize specific capability gaps (*e.g.*, text-replacement) over broad application-level goals (*e.g.*, improving text editing). We further construct a *Quality-Deployability* capability map using the deployment results. The map shows that current on-device models can achieve efficient generation or editing, but this deployment efficiency is not consistently accompanied by broad editing capability. Further progress requires improving editing quality and expanding task coverage within resource budgets, or exploring task-aware on-device–cloud routing. We will release the benchmark annotations, inference code and evaluation toolkit.

1 INTRODUCTION

Advances in large diffusion models have improved image generation and editing capabilities, with increasing use in mobile photo-editing applications. Cloud-based services such as GPT Image 2.5 OpenAI (2026) and Seedream 5.0 ByteDance Seed (2026) bring substantial inference costs and raise privacy concerns over personal image uploads. These concerns have motivated the development of compact on-device models Shaker et al. (2026); Duan et al. (2026); Deng et al. (2026). Evaluating these models requires determining which mobile tasks they support, how well their outputs satisfy user requirements, and whether inference fits device resource budgets. However, existing benchmarks such as GenEval Ghosh et al. (2023) and ImgEdit-Bench Ye et al. (2026) primarily assess general generation and editing capabilities. They provide limited guidance on connecting mobile application needs, performance on specific requests and deployment costs on actual devices. This leaves developers without a clear basis for deciding which requests a compact model can handle, and researchers without a detailed account of the capabilities that remain behind larger models.

A request such as removing text from a photograph specifies both an action and the visual content it concerns. We define the requested action, such as removal or replacement, as the *operation*, and the type of visual content to generate or modify as the *target*. Integrating more general application categories such as `Text Editing` can extend coverage of mobile scenarios but does not always make the task coverage explicit. For example, `Text Editing` can include both text removal and text replacement, so a high category score does not establish proficiency in both operations. Conversely, removing text and removing a person share the same operation applied to different targets. Grouping scores only by application category makes it difficult to compare removal capability across targets and between on-device and server-side models. We therefore introduce **MoGE-Bench**, a **Mo**bile

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.

Generation and **E**diting benchmark grounded in a user survey of mobile application needs. We specify operation and target axes separately and use a user survey to identify priority target categories for mobile applications. It defines 16 operations and eight target categories as two orthogonal axes of an *Operation* $\times$ *Target* Task Matrix. Applicable combinations are manually reviewed, yielding 64 subtasks comprising 1150 samples. This organization supports comparisons between operations for the same target and between targets for the same operation, helping developers assess the requests relevant to their applications.

Task performance alone does not explain which output requirements a model fails to satisfy. When asked to remove a specified person from a group portrait, a model may remove the wrong person, alter the remaining people's identities or leave visible artifacts. We propose **MoGE-Eval**, which associates each subtask with applicable criteria (*i.e.*, instruction following, visual quality and detail preservation) in a sparse *Operation* $\times$ *Target* $\times$ *Criteria* Rubric Cube. Based on these criteria, we develop detailed questions for each sample to evaluate models' outputs and aggregate their scores into overall quality Q and capability profiles. Joint *Operation* $\times$ *Target* profiles show performance on specific requests, while criterion profiles help researchers identify whether improvements are needed in instruction following, visual quality or preservation of existing details.

For mobile applications, output quality must also be considered alongside deployment costs. Even high-quality models are unsuitable for on-device use if their memory requirements exceed device capacity, while acceptable waiting times vary across applications. We therefore retain quality Q and deployability D as separate dimensions in a *Quality-Deployability* capability map. The quality dimension considers the overall score Q, task coverage and criterion-group scores, while the deployability dimension considers end-to-end latency, peak memory and deployed model size. Based on these measurements, we define quality tiers and deployability tiers and use them to construct a *Quality-Deployability* capability map. The map positions models by capability and resource requirements, helping developers select suitable models and researchers identify priorities for broader task coverage, higher output quality or lower deployment costs.

We evaluate 19 generative models and measure text-to-image deployment costs on iOS and Android devices. Our results first show that compact models approach stronger server-side models on some requests while remaining far behind on others within the same application category. For example, Ovis-U1 Wang et al. (2025a) approaches FLUX.2 [klein] Black Forest Labs (2026) on text removal but performs substantially worse on text replacement. This result suggests that improving specific tasks (*e.g.*, text replacement) should take priority over text editing in general. This distinction also helps users select models for their intended purposes. Criterion profiles further distinguish on-device models' development priorities: DreamLite Feng et al. (2026) shows weaknesses in visual integrity and coherence, whereas Ovis-U1 has a lower identity-preservation score. Additionally, efficient mobile inference is not consistently accompanied by broad editing capability. Mobile-O Shaker et al. (2026) and DreamLite both reach D4 on the iPhone 17 Pro, but their reference quality tiers are Q1 and Q3, respectively. Further progress therefore requires improving editing quality and expanding task coverage while preserving resource efficiency. The task-dependent capability gaps also motivate exploring on-device–cloud routing, with requests assigned according to model capability and deployment constraints.

In summary, our contributions are:

- We introduce **MoGE-Bench**, an on-device diffusion benchmark built on an *Operation* $\times$ *Target* Task Matrix. Its 1,150 samples span 64 subtasks designed around mobile application needs, covering image generation and four types of editing operations.
- We propose **MoGE-Eval**, which derives overall quality Q and capability profiles along the operation, target, and criteria axes from a sparse *Operation* $\times$ *Target* $\times$ *Criteria* Rubric Cube. It further introduces, to the best of our knowledge, the first unified *Quality-Deployability* capability map for benchmarking models instead of a single score.
- We benchmark representative generative models and further examine *Quality-Deployability* trade-offs for on-device models through measurements on iOS and Android devices. We will release the license-audited benchmark annotations, inference code and evaluation toolkit.

2 Related Work

Mobile diffusion research improves inference efficiency through compression and quantization and increasingly supports image editing Zhao et al. (2024b); Li et al. (2024b); Feng et al. (2026); Deng

Table 1: The Operation × Target Task Matrix of MoGE-Bench. Each cell describes the corresponding task. Merged cells indicate tasks shared across target categories; N/A marks combinations excluded from the benchmark.

Operation \ Target		General Objects		Animals	Portraits				Text & Docs.
		Concrete Obj.	Abstract Obj.		Body & limbs	Face & hair	Clothing & Accs.	Multi-subjects	
Gen.	T2I	Object generation	Abstract visuals / wallpaper	Animal / pet generation	Full-body portraits	Portrait close-ups	Outfit and accessory / fashion generation	Group portraits	Posters, memes, images with text
Local edit.	Add	Add an object	Add lighting, effects, shadows, etc.	Add an animal / pet	N/A	Add makeup, hairstyle, etc.	Add clothing, glasses, masks, earrings, etc.	Position-aware addition within a group	Add text
	Remove	Remove an object	Remove highlights, reflection, shadows	Remove an animal	N/A	Remove makeup, acne, wrinkle, etc.	Remove clothing or accessories	Position-aware removal within a group	Remove text
	Replace	Replace an object	N/A	Replace an animal	N/A	Replace hairstyle, makeup, facial identity	Replace clothing or accessories	Replace within a group	Replace text
	Adjust	Adjust object attributes	Adjust position, lighting, etc.	Adjust fur, pose, etc.	Adjust body shape, pose, etc.	Adjust facial attributes, expression, retouching	Adjust clothing or accessories	Adjust within a group	Adjust font, size, layout, etc.
Global edit.	Style	Apply a visual style across the image							
	BG edit	Modify the background while retaining the foreground							
	Restore	Restore image clarity and detail (denoise, deblur, remove moiré or watermarks, colorize or upscale)			Restore portrait quality (remove shadows, blur, overexposure or other degradation)				Restore document clarity
	View	Change the camera angle or viewpoint							Correct doc perspective / extract page for PDF
Under.	Semantic	Edit from commonsense or causal context			Expression or age editing that requires semantic understanding				Translate text or documents
	Outpaint	Extend the image beyond its original boundaries							
	Inpaint	Reconstruct an image partially obscured by one or more masks							
Ref. Edit.	R1	Recreate a reference object	N/A	Recreate a reference animal	N/A	Identity-based portraits	Extract clothing or accessories	Recreate a reference group	Reference-based document / poster design
	B1+R1	Transfer an object	Transfer visual style	Transfer an animal	Transfer a pose	Transfer identity, hair or makeup	Transfer clothing or accessories	Insert a reference person	Transfer typography or layout
	B1+R2	Combine two role-specific references in one base image							
	B1+R3	Combine three role-specific references in one base image							

et al. (2026). Generation and editing benchmarks evaluate output quality across diverse tasks Ghosh et al. (2023); Ye et al. (2026); Pan et al. (2025). MoGE-Bench organizes tasks by operation and target and, to our knowledge, is the first benchmark to jointly evaluate the quality and deployability of on-device image diffusion models. A detailed review is provided in the supplementary material.

3 MoGE-Bench

MoGE-Bench organizes tasks through an *Operation* × *Target* Task Matrix. Operations specify what the model should do, and targets identify the visual content to generate or modify. Applicable combinations define subtasks of the form "`Operation` on `Target`".

We survey 1,000 participants to guide task scope and target coverage. Portrait retouching, object removal and restoration are the most frequently selected use cases; text/document editing and pet-related tasks are also common (Fig. 1(a)). We therefore prioritize editing and retain generation to assess basic capabilities. Survey details are provided in the supplementary material.

3.1 Operation × Target Task Matrix

Operation taxonomy. Following prior taxonomies Yu et al. (2025); Ye et al. (2026), we organize 16 operation units into five groups (Tab. 1):

- *Generation.* Creating an image from a prompt without an input image. Tasks conditioned on a reference image are assigned to Reference Editing.
- *Local Editing.* Adding, removing, replacing or adjusting specified content while preserving unrelated regions. Replacement substitutes content; adjustment changes properties such as color, size or position.
- *Global Editing.* Changing overall appearance or scene context through stylization, background modification, restoration or camera-view changes.
- *Understanding Editing.* Inferring changes or completing content from semantic and visual context. Semantic edits specify an intended outcome, such as showing ice cream melting in the sun. Inpainting fills masked regions, while outpainting extends image boundaries.
- *Reference Editing.* Using reference images to specify content or attributes such as identity, pose or style. Input configurations comprise one reference without a base image, or a base image with one to three references Oshima et al. (2026).

Target taxonomy. To reflect these mobile application needs, we include Animals, Portraits, and Text & Documents categories alongside General Objects. Targets are categorized by the primary visual content to generate or modify, rather than every entity present in the input image.

- *General Objects.* Concrete objects (*e.g.*, vehicle and furniture), and abstract visual elements, including lighting, shadows and spatial relations.

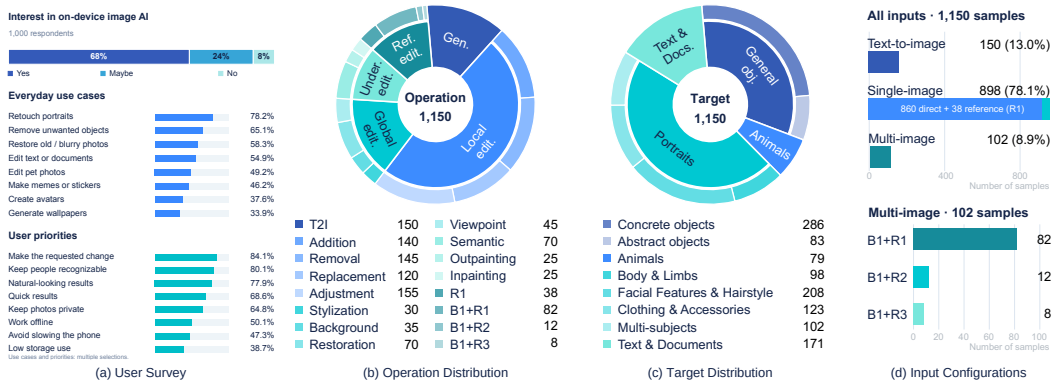

Figure 1: (a) Survey responses on interest in on-device image AI, everyday use cases and user priorities. (b–d) Sample distributions by operation, target and input configuration. Each sample is assigned to one target category according to the primary visual content to generate or modify.

- *Animals.* This category covers animals, including pets, and their accessories.
- *Portraits.* Body and limbs, facial features and hairstyles, clothing and accessories, and multiple subjects. Multiple-subject tasks include edits to selected individuals and group-level edits.
- *Text & Documents.* Text and its visual presentation, including typography and layout in documents and posters.

Task coverage. We manually review operation–target combinations for applicability. For example, `Addition` × `Body & Limbs` is marked N/A because adding anatomical parts falls outside our portrait-editing scope. Operations are adapted to practical target-specific needs (*e.g.*, camera-view editing becomes perspective rectification for documents rather than viewpoint changes in general scenes). We also group target categories under a shared subtask when they follow the same task definition (*e.g.*, stylization of the whole image). Reference editing with two or three reference images also uses shared subtasks to combine information from multiple sources. Each merged cell counts as one subtask, yielding 64 subtasks in total (Tab. 1). Detailed task descriptions and examples are provided in the supplementary material.

3.2 Dataset Construction and Quality Control

Sample construction and quality control. Following ImgEdit-Bench Ye et al. (2026), we allocate samples within each subtask, select images from Pexels and Wikimedia Commons, and draft instructions. A VLM Google (2026) proposes instruction templates, which we manually select and refine for individual samples. Following prior work Ye et al. (2026); Jiang et al. (2026); Zhang et al. (2023), two annotators cross-review each sample for instruction clarity, task feasibility and image–instruction consistency. They also check for duplicates and review within-subtask diversity, adjusting sample counts accordingly. Construction details and review procedures are provided in the supplementary material.

3.3 Benchmark Statistics and Comparison

Fig. 1 summarizes the user survey and sample distributions by operation, target and input configuration. Local editing and portraits form the largest operation and target groups, respectively. For tasks spanning multiple targets, samples are categorized by their primary visual target. Detailed statistics and comparisons with existing benchmarks are provided in the supplementary material.

4 MoGE-Eval

4.1 Quality Rubric Cube

We extend the *Operation* × *Target* Task Matrix with a criteria axis to form a *Quality Rubric Cube*. Each applicable cell specifies the evaluation requirements for an *operation* × *target* combination under one criterion.

Evaluation criteria. Following ImgEdit-Bench Ye et al. (2026), we organize evaluation into three criterion groups: instruction following, visual quality and detail preservation. These groups contain eight criteria that form the third axis of the rubric cube:

- *Instruction Following (IF)* includes *Fidelity* to the requested outcome and constraints, and *Selection* of the intended region or instance.

- *Visual Quality (VQ)* includes *Integrity* of generated or modified content (*i.e.*, shape defects and artifacts), *Coherence* between edited regions and their surroundings (*i.e.*, boundary and lighting consistency), and whole-image *Presentation* (*i.e.*, overall readability, composition and physical plausibility).
- *Detail Preservation (DP)* covers non-target objects and background details under *Context*, and unchanged framing and spatial relations under *Structure*. Given user demand for identity preservation, *Identity* checks human identity and appearance features not targeted by the instruction.

Task-specific rubrics. Each rubric entry (o, t, c) defines requirements for operation o, target t and criterion c. For example, `Removal × Multi-Subjects × Identity` checks whether the remaining people's identities are preserved. We manually label entries as always applicable, sample-dependent or not applicable. DP is inapplicable to text-to-image generation. Identity in object editing applies only when people should remain unchanged. These applicability labels define the sparse rubric cube. We resolve sample-dependent entries using the instruction and input images, and exclude non-applicable entries.

Sample-grounded questions. Following question-based evaluation methods (*e.g.*, TIFA Hu et al. (2023)), we construct a question set $\mathcal{Q}_{s,c}$ for each sample s and applicable criterion c. Gemini-3.1-pro Google (2026) proposes questions based on the task-specific rubric, instruction and input images. Two annotators cross-review rubric applicability and question relevance, removing duplicate requirements. The finalized questions are shared across all evaluated models.

4.2 Quality Scoring and Capability Profiles

We aggregate question-level judgments into overall quality Q and capability profiles by operation, target and evaluation criterion.

Scoring and aggregation. VLM judges (*i.e.*, GPT-4o Hurst et al. (2024) and Gemini-3.1-pro Google (2026)) score each question using the output, instruction and required input images. Following Qwen-Image-Bench Li et al. (2026), scores of $\{0, 0.6, 1\}$ indicate unmet, partially met and fully met requirements. We average question scores into criterion scores $z_{s,c}$, then criterion-group scores $v_{s,k}$ for $k \in \{\mathrm{IF}, \mathrm{VQ}, \mathrm{DP}\}$:

$$z_{s,c} = \frac{1}{|\mathcal{Q}_{s,c}|} \sum_{q \in \mathcal{Q}_{s,c}} r_{s,q}, \qquad v_{s,k} = \frac{1}{|\mathcal{C}_{s,k}|} \sum_{c \in \mathcal{C}_{s,k}} z_{s,c}, \tag{1}$$

where $r_{s,q}$ is the question-level score and $\mathcal{C}_{s,k}$ contains the applicable criteria in group k for sample s. Non-applicable criteria are excluded, and applicable criteria receive equal weight within each group regardless of question counts. We compute sample quality using a weighted geometric mean Ku et al. (2024a); Kamath et al. (2025):

$$u_s = \prod_{k \in \mathcal{K}_s} v_{s,k}^{\widetilde{w}_{s,k}}, \qquad \widetilde{w}_{s,k} = \frac{w_{s,k}}{\sum_{\ell \in \mathcal{K}_s} w_{s,\ell}}, \tag{2}$$

where $\mathcal{K}_s$ contains the applicable criterion groups and $w_{s,k}$ is a predefined weight for criterion group k, set by task operation and fixed across models. Geometric aggregation limits compensation for poor performance in one criterion group by high scores in others. Only IF and VQ contribute to text-to-image generation. Weight settings and analysis are provided in the supplementary material.

Overall quality for model m is $Q_m = |\mathcal{S}_m|^{-1} \sum_{s \in \mathcal{S}_m} u_s$, where $\mathcal{S}_m$ is the retained sample set. Starting from all 1150 samples, we exclude confirmed generation or judge execution errors. Unsupported tasks receive $u_s = 0$ and remain in $\mathcal{S}_m$. Successfully generated and evaluated outputs retain their scores regardless of quality. Each retained sample contributes equally to Q_m, while subtasks contribute in proportion to their retained sample counts.

Capability profiles and Quality tiers. Beyond the overall score Q, we report capability profiles by operation, target and evaluation criterion. Operation and target scores average u_s over samples in each category. Target labels follow the sample-level assignments in Sec. 3. We examine *Operation* × *Target* combinations to compare performance on specific tasks. These profiles connect model performance to mobile application needs and support model selection for different operations and targets. We also average $v_{s,k}$ and $z_{s,c}$ over their applicable samples to obtain profiles for the three criterion groups and eight criteria. This profile highlight areas for model improvement. For example, successful person removal may still involve unintended changes to the remaining people's identities. All profiles follow the same inclusion rules as overall quality.

Table 2: Quality and deployability tiers. All listed requirements must be satisfied, with higher tiers taking precedence. $C_{0.6}$ denotes the proportion of subtasks with mean quality scores of at least 0.6. The group requirement applies to each of IF, VQ and DP. D2–D4 require successful generation on the target device. — indicates no additional constraint.

Quality				Deployability			
Tier	Overall Q	Coverage $C_{0.6}$	Each group	Tier	Size (MiB)	E2E (s)	Memory (MiB)
Q4	≥ 0.80	$\geq 85\%$	≥ 0.75	D4	$\leq 2{,}048$	≤ 5	$\leq 2{,}048$
Q3	≥ 0.60	$\geq 60\%$	—	D3	$\leq 4{,}096$	≤ 30	$\leq 4{,}096$
Q2	≥ 0.40	—	—	D2	—	—	—
Q1	< 0.40	—	—	D1	Deployment or generation failure		

We further use quality tiers $Q1$–$Q4$ to summarize overall quality, task coverage and benchmark-level IF, VQ and DP scores. For subtask mean quality $\bar{u}_j$, task coverage is $C_\tau = |\mathcal{T}|^{-1} \sum_{j \in \mathcal{T}} \mathbf{1}[\bar{u}_j \geq \tau]$, where $\mathcal{T}$ contains the 64 subtasks and $\tau = 0.6$. Unlike the sample-averaged score Q, coverage weights subtasks equally. These joint requirements distinguish high average quality from broad task coverage and performance across quality requirements. Each model receives the highest qualifying tier in Tab. 2, from Q1 to Q4.

4.3 Quality-Deployability Map

The quality score Q measures output quality but does not account for the time and resources required for mobile inference. We assess deployability using end-to-end latency, peak memory and deployed model size. These metrics capture waiting time, runtime memory usage and storage requirements in mobile use. We therefore present quality and deployability on separate axes in the *Quality-Deployability Map*.

Deployability measurement. A deployment configuration specifies model weights, component precisions, inference settings and runtime. Deployment tests use text-to-image generation, the task supported by all on-device models. Within each model, prompts, seeds, resolution and sampling settings remain fixed across configurations. We report the following deployment metrics:

- *Model Size.* Total storage required for standalone inference, including the backbone, decoder and required encoders. Duplicate artifacts and development files are excluded.
- *Inference Latency.* End-to-end time for text encoding, generation steps, image decoding and post-processing. Model loading, warm-up and result-file writing are excluded. We report mean timings over three recorded runs.
- *Peak Memory.* Maximum sampled application memory per run, measured as physical footprint on iOS and process PSS on Android.

Storage and memory are reported in MiB. The main comparison uses FP16 backbones and decoders on iPhone 17 Pro. Quantization configurations and detailed measurement procedures are provided in the supplementary material. Failed and unmeasured configurations are marked separately.

Deployability tiers. Deployment tiers follow the joint requirements in Tab. 2. D1 denotes unsuccessful deployment or failure to produce valid outputs under the tested conditions. D2–D4 require successful execution within progressively tighter resource budgets, with D4 the highest tier. Thresholds are shared across devices, models and precision settings. Both quality and deployability thresholds are manually selected for consistent comparison across models. Sensitivity analyses are provided in the supplementary material.

Q-D Map construction. We construct the Quality-Deployability Map using reference quality tiers from Sec. 4.2 on the horizontal axis and device-specific deployability tiers on the vertical axis. A complementary plot presents continuous quality scores against end-to-end latency, with color indicating task coverage and bubble area indicating total parameter count. Models without successful mobile inference appear separately without latency coordinates. Android results and quantized deployment configurations are reported in the supplementary material.

5 Experiments

5.1 Experimental Setup

Evaluation Models. Given the scarcity of lightweight open-source models, we include widely used and recently released open-source models, as well as closed-source APIs, for quality comparison. In total, we evaluate 16 open-source and 3 closed-source models. The open-weight models are

Table 3: Quality and Deployability on MoGE-Bench. The Q/D-tiers follow Tab. 2. Params. includes all parameterized components loaded, including encoders and image codecs. Res., Infer., E2E and Mem. denote the default square generation resolution, denoising steps, end-to-end inference latency and peak memory, respectively.

Model	Params.	Res.	Infer.	Quality			Deployability			
				$Q\uparrow$	$C_{0.6}\uparrow$	Q-tier	Size (MiB)	E2E (s)	Mem. (MiB)	D-tier
Closed-source Models										
Nano Banana 2	–	–	–	0.951	1.000	Q4	–	–	–	D1
Seedream 5.0 Pro	–	–	–	0.952	1.000	Q4	–	–	–	D1
GPT Image 2	–	–	–	0.967	1.000	Q4	–	–	–	D1
Open-weight Reference Models										
LongCat-Image Team et al. (2025)	14.6B	1024^2	50	0.120	0.125	Q1	–	–	–	D1
Lumina-DiMOO-8B Xin et al. (2025)	8.2B	1024^2	64	0.434	0.234	Q2	–	–	–	D1
InternVL-U-4B Tian et al. (2026)	4.3B	1024^2	20	0.513	0.359	Q2	–	–	–	D1
VIBE-Image-Edit Alekseenko et al. (2026)	4.3B	1024^2	20	0.594	0.562	Q2	–	–	–	D1
OmniGen2 Wu et al. (2026a)	7.8B	1024^2	50	0.617	0.547	Q2	–	–	–	D1
BAGEL-7B-MoT Deng et al. (2025)	14.7B	992^2	50	0.640	0.578	Q2	–	–	–	D1
LongCat-Image-Edit Team et al. (2025)	14.6B	1024^2	50	0.667	0.672	Q3	–	–	–	D1
Lance-3B Fu et al. (2026)	7.6B	768^2	30	0.674	0.672	Q3	–	–	–	D1
DeepGen 1.0 Wang et al. (2026)	6.7B	512^2	50	0.691	0.703	Q3	–	–	–	D1
FLUX.2 [klein] 4B Black Forest Labs (2026)	8.0B	1024^2	4	0.828	0.906	Q4	–	–	–	D1
On-device Models (total params. < 4B)										
Janus-Pro-1B Chen et al. (2025c)	2.1B	384^2	–	0.080	0.078	Q1	2975.4	58.26	1505.4	D2
Mobile-O-0.5B Shaker et al. (2026)	1.7B	512^2	20	0.097	0.109	Q1	1718.3	2.81	426.8	D4
SANA-Sprint-0.6B Chen et al. (2025b)	3.5B	1024^2	2	0.100	0.109	Q1	2860.9	4.49	1727.2	D3
SANA-600M Xie et al. (2024)	3.5B	1024^2	18	0.104	0.109	Q1	2857.2	15.33	1775.2	D3
DreamLite Feng et al. (2026)	2.5B	1024^2	4	0.623	0.641	Q3	1684.6	3.22	1305.7	D4
Ovis-U1-3B Wang et al. (2025a)	3.6B	1024^2	50	0.655	0.594	Q2	6910.4	325.75	4298.7	D2

listed in Tab. 3. We access GPT Image 2, Nano Banana 2 and Seedream 5.0 Pro through their APIs.

Evaluation Setting. We run the open-weight models on NVIDIA A100 GPUs using their default hyper-parameters. We use GPT-4o Hurst et al. (2024) for the main results and independently evaluate the same outputs with Gemini 3.1 Pro Google (2026). Both judges follow the sample-specific questions and scoring protocol in Sec. 4. Inference settings, API and judge versions, and Gemini results are detailed in the supplementary material.

Mobile deployment. We test models below 4B total parameters on an iPhone 17 Pro using MLX for text encoders and Core ML for other components. Models that cannot be deployed on the target device serve as quality references and receive D1. Tests use T2I generation with FP16 backbones and decoders; models, resolutions and other configurations are listed in Tab. 3. Each successful configuration receives one warm-up and three measured runs. We report mean latency and mean per-run peak memory, with physical footprint sampled every 25 ms. Android results on Xiaomi 15 Pro, runtime details and quantization experiments are provided in the supplementary material.

5.2 Overall Quality

Quantitative Evaluation Tab. 3 compares overall quality and task coverage across models, as evaluated by GPT-4o. $C_{0.6}$ denotes task coverage at a quality threshold of 0.6. The three closed-sourced models achieve the highest overall quality and meet the task-quality threshold on all 64 subtasks. FLUX.2 [klein] is the only evaluated open-weight model to reach Q4.

Deployability is measured on an iPhone 17 Pro with FP16 backbones and decoders. Among models below 4B total parameters, Ovis-U1 achieves the highest Q (0.655), while DreamLite alone reaches Q3 with higher task coverage (64.1%). These results highlight the importance of considering both overall quality and task coverage when selecting models for diverse editing needs.

Generation-only models receive Q1 because unsupported editing tasks contribute zero under the benchmark protocol. Their low overall scores reflect limited benchmark coverage rather than uniformly poor generation quality. No evaluated model below 4B reaches Q4, indicating remaining gaps in task coverage and quality.

Qualitative Evaluation Fig. 2 illustrates model outputs across the five operation groups. Results show that on-device models still fall behind large models in reference editing, text translation and identity preservation. Additional qualitative results are provided in the supplementary material.

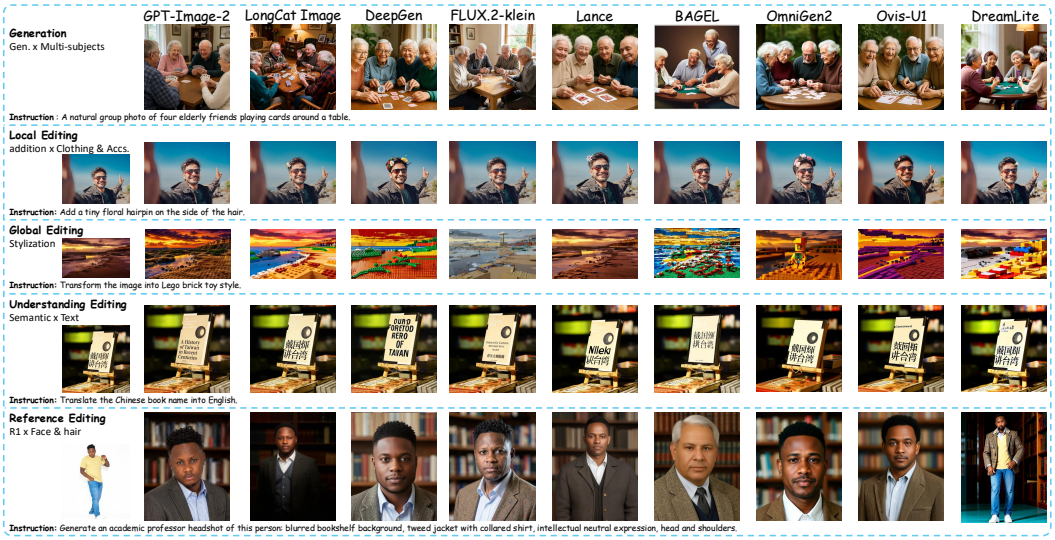

Figure 2: Qualitative comparison of representative image generation and editing models on MoGE-Bench across the five operation groups. Zoom in for better quality.



Figure 3: (a) Mean quality score for Operation × Target subtasks. Merged cells denote shared tasks; † marks unsupported tasks scored as zero. (b) Profiles across eight evaluation criteria, with the radial scale starting at 0.4. (c) Reference quality versus E2E latency, with models lacking successful mobile deployment shown separately without latency coordinates. (d) Joint quality and deployability tiers; positions within each cell are offset for readability. In (c–d), color indicates task coverage and bubble area indicates total parameter count; diamonds denote undisclosed counts.

5.3 Capability Analysis

Capability overview. Tab. 4 summarizes model performance across operation and target categories and the three criterion groups. We show higher-scoring reference models alongside compact models with editing support and complete results are provided in the supplementary material. Among the selected compact models, Ovis-U1 scores higher than DreamLite on local and global editing, whereas DreamLite performs better on understanding editing and detail preservation. Both models remain limited in reference editing, reflecting a combination of unsupported input configurations and quality on supported tasks. Text and documents receive the lowest target scores across the displayed open-weight models, indicating a shared challenge for mobile document and poster applications. These aggregate differences motivate closer examination of specific task combinations and individual quality requirements.

Table 4: Capability profiles of selected models on MoGE-Bench, evaluated by GPT-4o. IF, VQ and DP report scores for the three criterion groups. Among the displayed open-weight models, including the on-device group, red, blue and green indicate the first-, second- and third-ranked scores, respectively. Ties at the reported precision receive the same marking.

Model	Operation					Target				Criteria		
	Gen.	Local	Global	Underst.	Ref.	General	Animals	Portraits	Text.	IF	VQ	DP
Closed-source Models												
GPT Image 2	0.988	0.977	0.931	0.951	0.959	0.958	0.985	0.971	0.963	0.965	0.984	0.978
Seedream 5.0 Pro	0.977	0.964	0.909	0.932	0.949	0.943	0.958	0.960	0.943	0.947	0.973	0.976
Open-weight Reference Models												
LongCat-Image-Edit	0.000	**0.914**	0.823	0.659	0.201	0.687	0.672	0.692	0.542	0.661	0.704	0.857
DeepGen 1.0	**0.906**	0.773	0.719	0.572	0.201	0.717	0.763	0.708	0.549	0.735	0.744	0.696
FLUX.2 [klein] 4B	0.905	0.844	**0.878**	**0.703**	**0.720**	**0.862**	**0.895**	**0.860**	**0.619**	**0.817**	**0.883**	**0.951**
Lance-3B	0.762	0.772	0.722	0.636	0.158	0.729	0.762	0.722	0.363	0.685	0.739	0.761
On-device Models												
Ovis-U1-3B	0.823	0.719	0.749	0.590	0.153	0.734	0.751	0.692	0.325	0.684	0.704	0.716
DreamLite	0.803	0.673	0.697	0.607	0.145	0.711	0.729	0.654	0.285	0.652	0.636	0.758

Operation × Target analysis. Fig. 3(a) shows why low scores in a broad target category require further analysis by operation. DreamLite and Ovis-U1 score 0.656 and 0.712 on text removal, close to FLUX.2 [klein]'s 0.762. On text replacement, they score 0.028 and 0.268, compared with 0.677 for FLUX.2 [klein]. Each subtask contains 20 scored samples per model. *Operation × Target analysis supports model selection for specific requests and identifies concrete development priorities, such as text replacement rather than text editing in general.*

Criterion-level analysis. DreamLite scores higher on Structure (0.828) than Integrity (0.629) and Coherence (0.596) (Fig. 3(b)). Ovis-U1 has higher Coherence (0.684) but lower Identity (0.649) than DreamLite (0.732). Both score approximately 0.60 on Selection, below FLUX.2 [klein]'s 0.803, suggesting a shared weakness in region or instance selection. These profiles include unsupported-task zeros and indicate benchmark-level priorities rather than individual failure causes. *The profiles suggest different improvement priorities: visual integrity and local coherence for DreamLite, identity preservation for Ovis-U1, and precise region or instance selection for both.*

Evaluation reliability. We compare GPT-4o and Gemini 3.1 Pro on 19 models with results from both judges. Overall quality rankings have a Spearman correlation of $\rho = 0.968$. Absolute scores and quality tiers vary between judges. Detailed protocols and agreement with human judgments are provided in the supplementary material.

5.4 Mobile Deployment and Q-D Analysis

Baseline deployment. DreamLite and Mobile-O reach D4 at resolutions of 1024×1024 and 512×512, respectively (Tab. 3). DreamLite has the lowest latency among the tested 1024×1024 configurations. SANA Sprint meets the D4 latency and memory limits but receives D3 because its size exceeds 2,048 MiB. Latency alone therefore does not determine deployability.

Q-D analysis. Fig. 3(c–d) connects reference quality with mobile deployment. Ovis-U1 has higher Q than DreamLite (0.655 versus 0.623), but requires approximately 325 s rather than 3.22 s for generation. DreamLite reaches Q3–D4, whereas Ovis-U1 receives Q2–D2, reflecting their differences in task coverage and deployment costs. Mobile-O reaches D4 but remains Q1 because its generation-only support provides limited coverage of this user-needs benchmark. *Successful deployment does not imply broad editing capability; model suitability depends jointly on deployment cost, task coverage and output quality. Task-specific strengths also motivate exploring on-device–cloud routing, with suitable requests such as stylization handled locally and harder requests such as portrait retouching routed to cloud models.*

6 Conclusion

We introduce MoGE-Bench and MoGE-Eval to connect mobile user needs, task-specific quality and deployment costs. Our results reveal operation-specific gaps within broad application categories and distinct weaknesses in selection, visual coherence and identity preservation. Efficient deployment is not consistently accompanied by broad editing capability. The resulting profiles and Q-D map support model selection and identify development priorities. Deployment tests currently cover text-to-image generation; evaluating editing directly on mobile devices remains future work.

AI USE STATEMENT

Generative AI tools assisted manuscript drafting and language refinement, figure preparation, and benchmark construction. For benchmark construction, a VLM proposed instruction templates and sample-specific evaluation questions, which were manually reviewed and refined. We also used VLM judges for automated evaluation, as described in Sec. 4. The authors reviewed the AI-assisted content and take responsibility for the final manuscript, claims and released artifacts.

ETHICS STATEMENT

MoGE-Bench uses images from Pexels and Wikimedia Commons, and released materials will follow applicable licensing and attribution requirements. Survey participation was voluntary and fully anonymous. The survey platform collected no email addresses, account identifiers, IP addresses or other identifying information. The questionnaire concerned non-sensitive preferences for mobile image generation and editing, and results are reported in aggregate. Ethics review was not required for this fully anonymous, non-sensitive survey under the applicable institutional policy. The benchmark evaluates model capabilities and limitations; its quality and deployability scores do not assess fairness or misuse risks.

REPRODUCIBILITY STATEMENT

Sec. 3 describes benchmark construction and quality control, while Sec. 4 defines the evaluation criteria, scoring rules and deployment metrics. Sec. 5 reports the experimental settings and results. We will release benchmark annotations, evaluation questions, inference code and the evaluation toolkit, together with configuration details needed to reproduce the reported results.

REFERENCES

Grigorii Alekseenko, Aleksandr Gordeev, Irina Tolstykh, Bulat Suleimanov, Vladimir Dokholyan, Georgii Fedorov, Sergey Yakubson, Aleksandra Tsybina, Mikhail Chernyshov, and Maksim Kuprashevich. Vibe: Visual instruction based editor. *arXiv preprint arXiv:2601.02242*, 2026.

Xuehai Bai, Yang Shi, Yi-Fan Zhang, Xuanyu Zhu, Yuran Wang, Yifan Dai, Xinyu Liu, Yiyan Ji, Xiaoling Gu, and Yuanxing Zhang. Edit-compass & editreward-compass: A unified benchmark for image editing and reward modeling. *arXiv preprint arXiv:2605.13062*, 2026.

Eslam Mohamed Bakr, Pengzhan Sun, Xiaoqian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *2023 ieee/cvf international conference on computer vision (iccv)*, pp. 19984–19996. IEEE, 2023.

Samyadeep Basu, Mehrdad Saberi, Shweta Bhardwaj, Atoosa Malemir Chegini, Daniela Massiceti, Maziar Sanjabi, Shell Xu Hu, and Soheil Feizi. Editval: Benchmarking diffusion based text-guided image editing methods. *arXiv preprint arXiv:2310.02426*, 2023.

Black Forest Labs. FLUX.2 [klein] 4B. `https://huggingface.co/black-forest-labs/FLUX.2-klein-4B`, 2026. Hugging Face model card.

ByteDance Seed. Seedream 5.0 Pro. `https://seed.bytedance.com/en/seedream5_0_pro`, 2026. Official model page. Accessed: 2026-09-26.

Thibault Castells, Hyoung-Kyu Song, Tairen Piao, Shinkook Choi, Bo-Kyeong Kim, Hanyoung Yim, Changgwun Lee, Jae Gon Kim, and Tae-Ho Kim. Edgefusion: On-device text-to-image generation. *arXiv preprint arXiv:2404.11925*, 2024.

Ruchika Chavhan, Malcolm Chadwick, Alberto Gil Couto Pimentel Ramos, Luca Morreale, Mehdi Noroozi, and Abhinav Mehrotra. Nanoflux: Distillation-driven compression of large text-to-image generation models for mobile devices. In *Forty-third International Conference on Machine Learning*, 2026.

Jierun Chen, Dongting Hu, Xijie Huang, Huseyin Coskun, Arpit Sahni, Aarush Gupta, Anujraaj Goyal, Dishani Lahiri, Rajesh Singh, Yerlan Idelbayev, et al. Snapgen: Taming high-resolution text-to-image models for mobile devices with efficient architectures and training. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7997–8008. IEEE, 2025a.

Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. Sana-sprint: One-step diffusion with continuous-time consistency distillation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16185–16195. IEEE, 2025b.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025c.

Yu-Hui Chen, Raman Sarokin, Juhyun Lee, Jiuqiang Tang, Chuo-Ling Chang, Andrei Kulik, and Matthias Grundmann. Speed is all you need: On-device acceleration of large diffusion models via gpu-aware optimizations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4651–4655. IEEE, 2023.

Jiwoong Choi, Minkyu Kim, Daehyun Ahn, Taesu Kim, Yulhwa Kim, Dongwon Jo, Hyesung Jeon, Jae-Joon Kim, and Hyungjun Kim. Squeezing large-scale diffusion models for mobile. *arXiv preprint arXiv:2307.01193*, 2023.

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.

Fei Deng, Yanwu Xu, Zhipeng Bao, Zhixing Zhang, Haolin Jia, Karthik Raveendran, and Jianing Wei. Blazeedit: Generalist image editing on mobile devices with image-to-image diffusion models. *arXiv preprint arXiv:2605.28067*, 2026.

Kunpeng Du, Haizhen Xie, Sen Lu, Lei Yu, Binglei Bao, Huaao Tang, Chuntao Liu, Hao Wu, Yang Zhao, Zhicai Huang, et al. Elasticdit: Efficient diffusion transformers via elastic architecture and sparse attention for high-resolution image generation on mobile devices. *arXiv preprint arXiv:2605.15684*, 2026.

Kangsheng Duan, Ziyang Xu, Wenyu Liu, Xiaohu Ruan, Xiaoxin Chen, and Xinggang Wang. Moebius: 0.2 b lightweight image inpainting framework with 10b-level performance. *arXiv preprint arXiv:2606.19195*, 2026.

Kailai Feng, Yuxiang Wei, Bo Chen, Yang Pan, Hu Ye, Songwei Liu, Chenqian Yan, and Yuan Gao. Dreamlite: A lightweight on-device unified model for image generation and editing. *arXiv preprint arXiv:2603.28713*, 2026.

Fengyi Fu, Mengqi Huang, Shaojin Wu, Yunsheng Jiang, Yufei Huo, Hao Li, Yinghang Song, Fei Ding, Jianzhu Guo, Qian He, et al. Lance: Unified multimodal modeling by multi-task synergy. *arXiv preprint arXiv:2605.18678*, 2026.

Sara Ghazanfari, Wei-An Lin, Haitong Tian, and Ersin Yumer. Spotedit: Evaluating visually-guided image editing methods. *arXiv preprint arXiv:2508.18159*, 2025.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Google. Gemini 3.1 pro. Google AI Blog, 2026. URL `https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/`. Version: gemini-3.1-pro-preview.

Rui Gui, Yang Wan, Haochen Han, Dongxing Mao, Fangming Liu, Min Li, and Alex Jinpeng Wang. Texteditbench: Evaluating reasoning-aware text editing beyond rendering. *arXiv preprint arXiv:2512.16270*, 2025.

Dongting Hu, Aarush Gupta, Magzhan Gabidolla, Arpit Sahni, Huseyin Coskun, Yanyu Li, Yerlan Idelbayev, Ahsan Mahmood, Aleksei Lebedev, Dishani Lahiri, et al. Snapgen++: Unleashing diffusion transformers for efficient high-fidelity image generation on edge devices. *arXiv preprint arXiv:2601.08303*, 2026.

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 *2023 ieee/cvf international conference on computer vision (iccv)*, pp. 20349–20360. IEEE, 2023.

Hang Hua, Ziyun Zeng, Yizhi Song, Yunlong Tang, Liu He, Daniel Aliaga, Wei Xiong, and Jiebo Luo. Mmigbench: Towards comprehensive and explainable evaluation of multi-modal image generation models. *arXiv preprint arXiv:2505.19415*, 3:3, 2025.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3563–3579, 2025.

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

Bohan Jia, Wenxuan Huang, Yuntian Tang, Junbo Qiao, Jincheng Liao, Shaosheng Cao, Fei Zhao, Zhaopeng Feng, Zhouhong Gu, Zhenfei Yin, et al. Compbench: Benchmarking complex instruction-guided image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1112–1122, 2026.

Zhangqi Jiang, Zheng Sun, Xianfang Zeng, Yufeng Yang, Xuanyang Zhang, Yongliang Wu, Wei Cheng, Gang Yu, Xu Yang, and Bihan Wen. Geditbench v2: A human-aligned benchmark for general image editing. *arXiv preprint arXiv:2603.28547*, 2026.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2310.01506*, 2023.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. Geneval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025.

Sravanth Kodavanti, Manjunath Arveti, Sowmya Vajrala, Srinivas Miriyala, et al. Edgedit: Hardware-aware diffusion transformers for efficient on-device image generation. *arXiv preprint arXiv:2603.28405*, 2026.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12268–12290, 2024a.

Max Ku, Tianle Li, Kai Zhang, Yujie Lu, Xingyu Fu, Wenwen Zhuang, and Wenhu Chen. Imagenhub: Standardizing the evaluation of conditional image generation models. In *International Conference on Learning Representations*, volume 2024, pp. 46689–46722, 2024b.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Teufel, Marco Bellagente, et al. Holistic evaluation of text-to-image models. *Advances in Neural Information Processing Systems*, 36:69981–70011, 2023.

Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, et al. Genai-bench: Evaluating and improving compositional text-to-visual generation. *arXiv preprint arXiv:2406.13743*, 2024a.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*, 2024b.

Niantong Li, Guangzheng Hu, Weixu Qiao, Ying Ba, Qichen Hong, Shijun Shen, Jinlin Wang, Fan Zhou, Jianye Kang, Xin Shang, et al. Qwen-image-bench: From generation to creation in text-to-image evaluation. *arXiv preprint arXiv:2605.28091*, 2026.

Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *Advances in Neural Information Processing Systems*, 36:20662–20678, 2023.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. I2ebench: A comprehensive benchmark for instruction-based image editing. *Advances in Neural Information Processing Systems*, 37:41494–41516, 2024.

OpenAI. Introducing ChatGPT Images 2.5. `https://openai.com/index/introducing-chatgpt-images-2-5/`, 2026. Accessed: 2026-09-26.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 448–460, 2026.

Yulin Pan, Xiangteng He, Chaojie Mao, Zhen Han, Zeyinzi Jiang, Jingfeng Zhang, and Yu Liu. Ice-bench: A unified and comprehensive benchmark for image creating and editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16586–16596. IEEE, 2025.

Bimsara Pathiraja, Maitreya Patel, Shivam Singh, Yezhou Yang, and Chitta Baral. Refedit: A benchmark and method for improving instruction-based image editing model on referring expressions. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15646–15656. IEEE, 2025.

Yusu Qian, Jiasen Lu, Tsu-Jui Fu, Xinze Wang, Chen Chen, Yinfei Yang, Wenze Hu, and Zhe Gan. Gie-bench: Towards grounded evaluation for text-guided image editing. *arXiv preprint arXiv:2505.11493*, 2025.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Samin Mahdizadeh Sani, Max Ku, Nima Jamali, Matina Mahdizadeh Sani, Paria Khoshtab, Wei-Chieh Sun, Parnian Fazel, Zhi Rui Tam, Thomas Chong, Edisy Kin Wai Chan, et al. Imagenworld: Stress-testing image generation models with explainable human evaluation on open-ended real-world tasks. *arXiv preprint arXiv:2603.27862*, 2026.

Abdelrahman Shaker, Ahmed Heakl, Jaseel Muhammad, Ritesh Thawkar, Omkar Thawakar, Senmao Li, Hisham Cholakkal, Ian Reid, Eric P Xing, Salman Khan, et al. Mobile-o: Unified multimodal understanding and generation on mobile device. *arXiv preprint arXiv:2602.20161*, 2026.

Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, et al. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

Changyao Tian, Danni Yang, Guanzhou Chen, Erfei Cui, Zhaokai Wang, Yuchen Duan, Penghao Yin, Sitao Chen, Ganlin Yang, Mingxin Liu, et al. Internvl-u: Democratizing unified multimodal models for understanding, reasoning, generation and editing. *arXiv preprint arXiv:2603.09877*, 2026.

Dianyi Wang, Ruihang Li, Feng Han, Chaofan Ma, Wei Song, Siyuan Wang, Yibin Wang, Yi Xin, Hongjian Liu, Zhixiong Zhang, et al. Deepgen 1.0: A lightweight unified multimodal model for advancing image generation and editing. *arXiv preprint arXiv:2602.12205*, 2026.

Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, Lunhao Duan, Shiyin Lu, Minghao Fu, Xiaohao Chen, Jianshan Zhao, et al. Ovis-u1 technical report. *arXiv preprint arXiv:2506.23044*, 2025a.

Yibin Wang, Zhimin Li, Yuhang Zang, Jiazi Bu, Yujie Zhou, Yi Xin, Junjun He, Chunyu Wang, Qinglin Lu, Cheng Jin, et al. Unigenbench++: A unified semantic evaluation benchmark for text-to-image generation. *arXiv preprint arXiv:2510.18701*, 2025b.

Chenyuan Wu, Jiahao Wang, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, et al. Omnigen2: Towards instruction-aligned multimodal generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21964–21975, 2026a.

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*, 2026b.

Xindi Wu, Dingli Yu, Yangsibo Huang, Olga Russakovsky, and Sanjeev Arora. Conceptmix: A compositional image generation benchmark with controllable difficulty. *Advances in Neural Information Processing Systems*, 37:86004–86047, 2024.

Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, Gang Yu, Wenbo Zhu, Bernt Schiele, Ming-Hsuan Yang, and Xu Yang. Kris-bench: Benchmarking next-level intelligent image editing models. *Advances in Neural Information Processing Systems*, 38, 2026c.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Keqi Wang, Yibin Wang, et al. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding. *arXiv preprint arXiv:2510.06308*, 2025.

Yang Ye, Xianyi He, Zongjian Li, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, Li Yuan, et al. Imgedit: A unified image editing dataset and benchmark. *Advances in Neural Information Processing Systems*, 38, 2026.

Hongzhu Yi, Yujia Yang, Yuanxiang Wang, Tong Li, Zhenyu Guan, Tianyu Zong, Jiahuan Chen, Chenxi Bao, Tiankun Yang, Haopeng Jin, et al. Vde bench: Evaluating the capability of image editing models to modify visual documents. *arXiv preprint arXiv:2602.00122*, 2026.

Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26125–26135. IEEE, 2025.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in neural information processing systems*, 36:31428–31449, 2023.

Tianchen Zhao, Xuefei Ning, Tongcheng Fang, Enshu Liu, Guyue Huang, Zinan Lin, Shengen Yan, Guohao Dai, and Yu Wang. Mixdq: Memory-efficient few-step text-to-image diffusion models with metric-decoupled mixed precision quantization. In *European Conference on Computer Vision*, pp. 285–302. Springer, 2024a.

Yang Zhao, Yanwu Xu, Zhisheng Xiao, Haolin Jia, and Tingbo Hou. Mobilediffusion: Instant text-to-image generation on mobile devices. In *European Conference on Computer Vision*, pp. 225–242. Springer, 2024b.

A User Survey and Benchmark Construction

A.1 User Survey

Survey overview. We collected 1,000 anonymous responses on mobile image AI. Fig. 1(a) reports interest, use cases and priorities. Interest in on-device image AI is 68% Yes, 24% Maybe and 8% No. Tab. 5 presents the questionnaire structure and one example scenario. Q1 asks about current usage and interest, and Q2 asks about desired use cases. Q3 contains 25 image-generation and editing scenarios. Each scenario presents 10–15 requirements, including shared usability needs and task-specific requirements. All respondents answer all 25 scenarios and select up to eight requirements per scenario. Selection rates use all 1,000 respondents as the denominator. For shared requirements, a respondent is counted once if they select the requirement in any scenario.

Use in benchmark design. Use cases inform task and target coverage. Usability requirements inform task-specific criteria, such as making the requested change and preserving identity. Selection rates describe the respondents rather than the wider population.

A.2 Task Definitions and Coverage

An *operation* specifies the requested change, and a *target* identifies the content involved. Tables 6 and 7 define both axes.

Operations. The operation axis contains 16 units organized into five groups. Reference-editing units specify the available base and reference images. Replacement substitutes content; adjustment changes its attributes. Outpainting extends image boundaries; inpainting fills masked regions within them. For reference editing, $B1$ denotes one base image and Rk denotes k reference images.

Table 6: Definitions of the 16 operation units.

Group	Operation	Definition
Generation	T2I	Generate an image from text without an input image.
Local editing	Addition	Add specified content while retaining existing, unrelated content.
	Removal	Delete specified content and complete the exposed region.
	Replacement	Replace selected content with a specified alternative.
	Adjustment	Change attributes of existing content, such as color, size or pose.
Global editing	Stylization	Apply a visual style across the image.
	Background	Change the background while retaining the foreground subject.
	Restoration	Recover image clarity or appearance from degradation.
	Viewpoint	Change the camera view, or rectify document perspective.
Understanding editing	Semantic	Infer the required edit from meaning, context or an intended outcome.
	Outpainting	Extend content beyond the original image boundaries.
	Inpainting	Reconstruct missing content within supplied masks.
Reference editing	R1	Generate an image from one reference, without a base image.
	B1+R1	Edit a base image using content or attributes from one reference.
	B1+R2	Edit a base image by combining two role-specific references.
	B1+R3	Edit a base image by combining three role-specific references.

Targets. The eight categories identify the primary content to generate or modify, not every visible entity. For example, changing a jacket targets Clothing & Accessories; removing one person from a group targets Multi-Subjects.

Table 5: Questionnaire structure and one of the 25 Q3 scenarios. Q1 contains three single-choice questions. Q2 allows up to eight selections. Each Q3 scenario presents 10–15 requirements and allows up to eight selections.

Topic	Question and response options
Q1a. Current usage	How often do you use AI tools on your phone, such as chat assistants, image generators or photo editors? *Options:* Daily; Several times a week; About once a week; Less than once a week; Never.
Q1b. Image tools	Would you like to generate or edit images using AI tools on your phone? *Options:* Yes; Maybe; No.
Q1c. On-device interest	Would you be interested in image AI that runs directly on your phone without uploading your photos to a server? *Options:* Yes; Maybe; No.
Q2. Desired use cases	Which of the following would you most like to use AI image tools for on your phone? Select up to eight options. *Options:* (1) Retouch portraits, such as removing blemishes or wrinkles. (2) Remove unwanted people or objects from photos. (3) Restore old, blurry or damaged photos. (4) Edit or translate text in images and documents. (5) Improve or creatively edit pet photos. (6) Make memes or stickers. (7) Create personal avatars or profile pictures. (8) Generate wallpapers. (9) Try different hairstyles, makeup or accessories. (10) Change clothing or preview outfit combinations. (11) Replace backgrounds in personal or product photos. (12) Turn photos into cartoons, paintings or other styles. (13) Create posters, invitations or greeting cards. (14) Improve product photos for online listings. (15) Preview furniture or decorations in a room. (16) Combine people or objects from several photos.
Q3. Usability requirements	Suppose you ask an AI tool to remove a small blemish from your cheek in a portrait and leave the rest of the photo unchanged. What would matter to you when judging the result? Select up to eight options. *Options:* (1) Results are returned quickly. (2) The requested blemish is removed. (3) Photos remain private. (4) Clothing, background and other people remain unchanged. (5) The skin retains natural texture without visible artifacts. (6) The tool requires little storage. (7) The person remains recognisable, with facial features preserved. (8) The tool works offline. (9) Only the intended area is retouched. (10) The overall photo remains clear and visually balanced. (11) The tool does not slow down the phone. (12) The original pose, framing and positions remain unchanged. (13) The retouched area matches the surrounding skin tone and lighting.

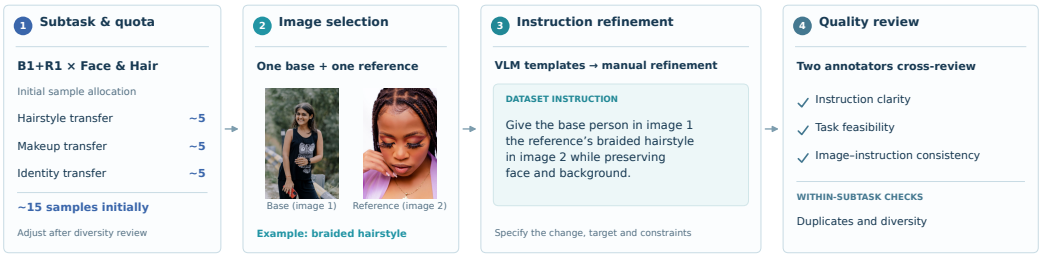

Figure 4: Sample construction and review for a hairstyle-transfer example from $(B1 + R1) \times$ `Face & Hair`. Sample allocation is adjusted after diversity review. The displayed images and instruction are from the benchmark.

Table 7: Definitions of the eight target categories.

Group	Target	Definition
General Objects	Concrete Objects	Discrete entities such as vehicles, furniture and everyday objects.
	Abstract Objects	Visual effects and relations, such as lighting, shadows and reflections.
Animals	Animals	Animals, including pets, their appearance and their accessories.
Portraits	Body & Limbs	Human body shape, limb configuration and pose.
	Face & Hair	Facial features, expressions, makeup, hairstyles and identity.
	Clothing & Accessories	Human garments and accessories, such as coats, glasses and jewelry.
	Multi-Subjects	Groups of people and edits to selected individuals within a group.
Text & Doc.	Text & Documents	Text content, typography, layout and document appearance.

Task coverage and examples. Table 8 provides task definitions and illustrative instructions, not verbatim benchmark annotations. N/A marks combinations excluded from the benchmark scope, such as adding or removing anatomical parts; it does not imply that such requests are impossible. Included tasks unsupported by a model instead receive zero scores. Each merged cell counts as one shared subtask. The benchmark contains 8 generation, 28 local-editing, 7 global-editing, 5 understanding-editing and 16 reference-editing subtasks, totaling 64.

A.3 Sample Construction and Quality Control

Following ImgEdit-Bench Ye et al. (2026), we specify subtasks, select images and draft instructions. Fig. 4 shows the construction and review process.

Subtask-level sample allocation. Sample allocation considers the transformations and target variations within each subtask. For example, $(B1 + R1) \times$ `Face & Hair` includes hairstyle, makeup and facial identity transfer. We review image and instruction diversity and adjust the allocation accordingly. Final counts are shown in Tab. 8.

Image sourcing and instruction drafting. We retrieve input and reference images from Pexels and Wikimedia Commons. Images must provide the information needed for the requested task. Reference tasks are reviewed with all inputs together; text-to-image samples contain prompts only. For inpainting and restoration, predefined code applies masks or degradations to downloaded source images. All models receive the same processed inputs. Gemini 3.1 Pro Google (2026) proposes instruction templates. We refine them to specify the change, target and roles of the input images.

Quality control. Following prior work Ye et al. (2026); Jiang et al. (2026); Zhang et al. (2023), two annotators cross-review each sample. They check instruction clarity, task feasibility and consistency

Table 8: Operation × Target task definitions and illustrative instructions (1/2). Each ordinary or merged cell is one subtask; N/A cells are excluded. n gives the subtask sample count (64 subtasks; 1,150 samples). $B1$ denotes one base image and Rk denotes k reference images.

Operation	**General Objects**		**Animals**	**Portraits**				**Text & Docs.**
	Concrete Objects	Abstract Objects	Animals	Body & Limbs	Face & Hair	Clothing & Accs.	Multi-Subjects	Text & Docs.
Generation								
T2I	**Object generation** *"A bicycle beside a fence."* $n = 20$	**Abstract visuals** *"A blue abstract wallpaper."* $n = 20$	**Animal generation** *"A dog resting on grass."* $n = 15$	**Full-body portrait** *"A full-body portrait of a runner."* $n = 20$	**Portrait close-up** *"A close-up of a curly-haired woman."* $n = 20$	**Outfit generation** *"A person wearing a blue coat."* $n = 10$	**Group portrait** *"Four friends playing cards."* $n = 15$	**Text-bearing image** *"A poster titled Summer Fair."* $n = 30$
Local Editing								
Addition	**Add an object** *"Add a vase to the table."* $n = 20$	**Add an effect** *"Add a sunbeam through the window."* $n = 15$	**Add an animal** *"Add a parrot to the chair."* $n = 15$	*N/A*	**Add a facial feature** *"Add a grey beard."* $n = 30$	**Add an accessory** *"Add a floral hairpin."* $n = 25$	**Add a group member** *"Add a person on the left."* $n = 15$	**Add text** *"Add OPEN to the sign."* $n = 20$
Removal	**Remove an object** *"Remove the cup."* $n = 25$	**Remove an effect** *"Remove the window glare."* $n = 15$	**Remove an animal** *"Remove the dog."* $n = 10$	*N/A*	**Remove a facial detail** *"Remove the cheek blemish."* $n = 30$	**Remove an accessory** *"Remove the sunglasses."* $n = 20$	**Remove a group member** *"Remove the leftmost person."* $n = 25$	**Remove text** *"Remove the poster title."* $n = 20$
Replacement	**Replace an object** *"Replace the chair with a stool."* $n = 20$	*N/A*	**Replace an animal** *"Replace the dog with a cat."* $n = 10$	*N/A*	**Replace a hairstyle** *"Replace long hair with a bob."* $n = 30$	**Replace clothing** *"Replace the coat with a jacket."* $n = 25$	**Replace a group member** *"Replace the middle person with a woman."* $n = 15$	**Replace text** *"Replace SALE with OPEN."* $n = 20$
Adjustment	**Adjust an attribute** *"Make the car blue."* $n = 20$	**Adjust illumination** *"Make the lighting warmer."* $n = 15$	**Adjust animal appearance** *"Make the fur orange."* $n = 10$	**Adjust body pose** *"Raise the left arm."* $n = 30$	**Adjust facial appearance** *"Widen the smile slightly."* $n = 30$	**Adjust an outfit** *"Make the jacket green."* $n = 20$	**Adjust within a group** *"Raise the rightmost person's hand."* $n = 15$	**Adjust typography** *"Enlarge the title."* $n = 15$
Global Editing								
Stylization	**Apply an image-wide style** *"Render the image in Lego style."* $n = 30$							
Background	**Change scene background** *"Use a beach background; keep the person unchanged."* $n = 35$							

Table 8: Operation × Target task definitions and illustrative instructions (2/2, continued). n denotes the number of benchmark samples in each ordinary or merged task cell.

Operation	**General Objects**		**Animals**	**Portraits**				**Text & Docs.**
	Concrete Objects	Abstract Objects	Animals	Body & Limbs	Face & Hair	Clothing & Accs.	Multi-Subjects	Text & Docs.
Global Editing (continued)								
Restoration	**Restore scene clarity** *"Denoise and sharpen this photo."* $n = 30$			**Restore portrait quality** *"Correct facial overexposure."* $n = 25$				**Restore document clarity** *"Deblur this page."* $n = 15$
Viewpoint	**Change camera view** *"Show the scene from above."* $n = 30$							**Rectify a document** *"Straighten this page."* $n = 15$
Understanding Editing								
Semantic	**Infer a contextual change** *"Show the ice cream after sun exposure."* $n = 30$			**Infer portrait appearance** *"Show the person reacting to an unexpected gift."* $n = 20$				**Translate text** *"Translate this title into English."* $n = 20$
Outpainting	**Extend image boundaries** *"Extend the scene on both sides."* $n = 25$							
Inpainting	**Complete masked regions** *"Reconstruct the masked regions."* $n = 25$							
Reference Editing								
R1	**Recreate an object** *"Show this backpack on a desk."* $n = 8$	*N/A*	**Recreate an animal** *"Show this cat beside a window."* $n = 5$	*N/A*	**Generate an identity-based portrait** *"Create this person's headshot."* $n = 10$	**Recreate an item** *"Create a product photo of this jacket."* $n = 5$	**Recreate a group** *"Show this group in a park."* $n = 4$	**Use a design reference** *"Create a poster in this style."* $n = 6$
B1+R1	**Transfer an object** *"Replace the chair with this one."* $n = 14$	**Transfer a style** *"Apply the reference style."* $n = 8$	**Transfer an animal** *"Add this dog to the garden."* $n = 7$	**Transfer a pose** *"Copy the reference pose."* $n = 8$	**Transfer a hairstyle** *"Copy the reference hairstyle."* $n = 14$	**Transfer an outfit** *"Use the reference outfit."* $n = 15$	**Insert a reference person** *"Add this person to the group."* $n = 8$	**Transfer a layout** *"Copy the reference layout."* $n = 8$
B1+R2	**Combine two references** *"Use the hairstyle from reference 1 and coat from reference 2."* $n = 12$							
B1+R3	**Combine three references** *"Use the hairstyle from reference 1, coat from reference 2 and background from reference 3."* $n = 8$							

Table 9: Task coverage across generation and editing benchmarks. Except for T2I, task columns describe editing tasks and may overlap. ✓: documented coverage; ×: outside the stated scope; —: coverage not confirmed. Loc. denotes local editing, Glob. global editing, and Act. pose or action changes. Pos. denotes object-position or spatial-relation changes; Impl. denotes implicit editing; Max. Refs counts reference images excluding the base image and MT denotes multi-turn editing.

Benchmark	Size	Task coverage								Max. Refs	MT
		T2I	Loc.	Glob.	Act.	Pos.	View	Impl.	Multi-obj.		
Text-to-image benchmarks											
GenEval Ghosh et al. (2023)	553	✓	×	×	×	×	×	×	×	0	×
GenAI-Bench Li et al. (2024a)	1,600	✓	×	×	×	×	×	×	×	0	×
Qwen-Image-Bench Li et al. (2026)	1,000	✓	×	×	×	×	×	×	×	0	×
General image-editing benchmarks											
MagicBrush Zhang et al. (2023)	1,053	×	✓	✓	✓	✓	×	×	✓	0	✓
EditVal Basu et al. (2023)	648	×	✓	✓	✓	✓	✓	×	×	0	×
AnyEdit-Test Yu et al. (2025)	1,250	×	✓	✓	✓	✓	✓	✓	×	0	×
GEdit-Bench Liu et al. (2025)	606	×	✓	✓	✓	×	×	×	×	0	×
KontextBench Labs et al. (2025)	1,026	×	✓	✓	✓	✓	✓	✓	✓	1	×
GEditBench v2 Jiang et al. (2026)	1,200	×	✓	✓	✓	✓	✓	✓	✓	1	×
CompBench Jia et al. (2026)	>3,000	×	✓	✓	✓	✓	✓	✓	✓	0	✓
Edit-Compass Bai et al. (2026)	2,388	×	✓	✓	✓	✓	✓	✓	✓	2	×
Reference-conditioned benchmarks											
OmniContext Wu et al. (2026a)	400	×	✓	✓	✓	✓	×	×	✓	2	×
SpotEdit Ghazanfari et al. (2025)	500	×	✓	×	✓	×	×	×	✓	2	×
MultiBanana Oshima et al. (2026)	3,769	×	✓	✓	✓	✓	✓	✓	✓	8	×
MICON-Bench Wu et al. (2026b)	1,043	×	✓	✓	✓	✓	×	✓	✓	2	×
Specialized editing benchmarks											
RefEdit-Bench Pathiraja et al. (2025)	200	×	✓	✓	✓	✓	×	×	✓	0	×
TextEditBench Gui et al. (2025)	1,196	×	✓	×	×	✓	—	✓	—	0	×
Generation and editing benchmarks											
ICE-Bench Pan et al. (2025)	6,538	✓	✓	✓	✓	×	×	×	✓	1	×
MoGE-Bench	**1,150**	✓	✓	✓	✓	✓	✓	✓	✓	3	×

with the input images. They also check duplicates and visual and instruction diversity within each subtask.

A.4 Dataset Statistics and Benchmark Comparison

Operation and target distributions. MoGE-Bench contains 1,150 samples across 64 subtasks. Local editing is the largest operation group with 560 samples, followed by global editing (180), generation (150), reference editing (140) and understanding editing (120). Portraits account for 531 samples, followed by general objects (369), text and documents (171), and animals (79).

We assign each sample to one category by its primary target, including samples from shared subtasks.

Input configurations. The benchmark contains 150 text-to-image, 898 single-image and 102 multi-image samples. Single-image samples comprise 860 direct-editing and 38 reference-only cases. Multi-image samples use one base image with one, two or three references, comprising 82, 12 and 8 cases, respectively.

Comparison with existing benchmarks. Tab. 9 compares benchmark task coverage and input configurations. MoGE-Bench covers generation and local, global, understanding and reference editing, with up to three reference images. Its Operation × Target organization supports comparisons across specific operations and visual targets. MoGE-Eval complements these quality evaluations with mobile deployment measurements.

B Evaluation Protocol and Experimental Settings

B.1 Task-Specific Rubrics and Criterion Applicability

A *criterion* specifies a quality requirement for an output. MoGE-Eval defines eight criteria under instruction following (IF), visual quality (VQ) and detail preservation (DP) (Tab. 10). Preservation applies only to content or attributes not targeted by the instruction.

Table 10: Definitions of the eight evaluation criteria.

Group	Criterion	Definition
IF	Fidelity	Satisfying the requested outcome and explicit constraints.
	Selection	Applying the change to the correct region or instance.
VQ	Integrity	Well-formed generated or edited content without defects or artifacts.
	Coherence	Consistent boundaries, lighting and appearance within the output and around edited regions.
	Presentation	Overall readability, composition and physical plausibility.
DP	Context	Preserving non-target objects and background details.
	Structure	Preserving framing, positions and spatial relations not targeted by the edit.
	Identity	Preserving human identity and appearance features not targeted by the edit.

Task-specific rubrics. Each entry (o, t, c) defines criterion c for operation o and target t. For example, Fidelity checks text removal or correct replacement text, depending on the operation. For person removal, Identity checks the remaining people. These requirements guide sample-grounded questions.

Criterion applicability. We manually label each rubric entry as:

- *Always applicable.* Required for every sample in the subtask, such as Fidelity in text replacement.
- *Sample-dependent.* Determined by the instruction and input images. For example, Identity applies to object editing when people must remain unchanged.
- *Not applicable.* Excluded from the subtask. For example, text-to-image generation has no source content to preserve, so DP is excluded.

We include always-applicable criteria and resolve sample-dependent criteria for each sample. The selected criteria determine its questions and criterion groups.

Non-applicable criteria are excluded, not scored as zero.

B.2 Sample-Grounded Questions and Judge Prompts

Question construction and review. Gemini 3.1 Pro receives the rubric, instruction and input images. It proposes applicable criteria, questions and score anchors without access to model outputs. Two annotators check applicability, visible evidence and instruction consistency. They remove duplicate requirements and unsupported assumptions. Preservation questions exclude requested changes. All models and judges share the final questions and anchors.

Score anchors. A score of 1 indicates full satisfaction. A score of 0.6 indicates that the core requirement is met with a limited omission or deviation. A score of 0 indicates that it is absent, incorrect or contradicted. Each question specifies full, partial and failed outcomes. The partial condition defines a score of 0.6. For binary requirements, this condition is N/A and only 0 or 1 applies. Fig. 5 illustrates the three score levels for a text-replacement question. The scores assess the requested text change; other requirements are evaluated separately.

Table 11: General scoring anchors. Each question specifies the corresponding task-specific conditions.

Score	Condition
1	The requirement is fully satisfied, including its explicit constraints.
0.6	The core requirement is satisfied, with a minor error or omission.
0	Neither the full nor partial condition is satisfied. This includes an absent change, an incorrect outcome or an unreadable result.



Figure 5: Visual examples of scoring anchors for text replacement. The instruction asks for "Downtown" to be replaced with "Countryside" in the same style. The displayed scores assess Fidelity to the requested replacement text: correct replacement (1), partially correct replacement (0.6), and unchanged original text (0). These are question-level scores, not overall image-quality scores.

Complete sample example. We use sample `local_remove_portrait_multi_010`. The task is Removal × Multi-Subjects. Its instruction is: "Remove the leftmost person from the group photo and reconstruct the occluded background." The input shows a group of people on a street. All eight criteria apply: the edit must remove the specified person, repair the exposed region and preserve the remaining scene and people. Tab. 12 lists all 14 questions and their GPT-4o scores for the Ovis-U1-3B output. Rubric summaries use the paper's criterion names. Some question wording is edited for clarity; scores are reproduced from the original evaluation record.

Table 12: A complete sample-grounded question set. Scores refer to one Ovis-U1-3B output evaluated by GPT-4o.

Criterion	Rubric focus and questions	Scores
Fidelity	*Remove the requested person.* (1) Is the leftmost person completely removed from the image without leaving visible remnants?	0.6
Selection	*Edit the correct person only.* (1) Did the removal target only the leftmost person (man in dark t-shirt), leaving the adjacent man in the white t-shirt untouched?	1
Integrity	*Reconstruct complete shapes and clean textures.* (1) Is the reconstructed background on the far left structurally complete without holes, repeated parts, or malformed shapes? (2) Are the textures in the repaired region on the left free of smearing, ringing, or synthetic artifacts?	1, 0.6
Coherence	*Match the surrounding geometry, colour and lighting.* (1) Does the reconstructed region on the left continue the surrounding street and background geometry without a visible seam? (2) Do the color and lighting in the filled area on the left match the neighboring image area?	1, 0.6
Presentation	*Maintain overall clarity and plausible composition.* (1) Is the final image clear and usable without awkward crop or weak focus? (2) Is the overall scene physically plausible and balanced in scale and composition?	1, 1

Criterion	Rubric focus and questions	Scores
Context	*Preserve the unedited background.* (1) Are the buildings in the background unchanged in appearance? (2) Is the street surface outside the removed region preserved in color, texture, and lighting?	1, 1
Structure	*Preserve framing and spatial relations.* (1) Are the original framing, composition, and camera view of the street scene preserved? (2) Are the positions and spatial relationships of the remaining people in the group preserved?	1, 1
Identity	*Preserve the remaining people.* (1) Does each remaining person retain the same identity as in the input image? (2) Are the remaining people's clothing, poses, and accessories preserved?	1, 1

The criterion means are 0.6 for Fidelity, 1 for Selection, 0.8 for Integrity and Coherence, and 1 for the other criteria. The group scores are $\mathrm{IF} = 0.8$, $\mathrm{VQ} = 0.867$ and $\mathrm{DP} = 1$. Using weights $(0.45, 0.25, 0.30)$ gives a sample quality score of 0.873.

Judge inputs and responses. Each judge call evaluates one question and returns a JSON score. Editing inputs appear in their stored order, followed by the model output. Text-to-image calls contain only the generated image. The accompanying text provides the instruction, question and score conditions. The following examples show judge prompts for base-image editing and text-to-image generation. Braces mark sample-specific fields. These examples do not enumerate all input configurations.

Editing system prompt example

You are a strict visual judge for an image-editing benchmark. Compare the ordered input image or images with the edited result and score only the stated atomic question. Use the supplied score anchors exactly. If the partial condition is N/A, return only 0 or 1. Do not reward unrelated visual quality and do not speculate beyond visible evidence. Output only one JSON object: {"score": 0}, {"score": 0.6}, or {"score": 1}.

Base-image editing prompt example. For a base-image edit, `image_description` states: "Image 1 is the base input image; Image 2 is the edited result." For multiple inputs, it identifies the ordered reference images and final output.

Editing user prompt

{image_description}

Editing instruction:
"""
{instruction}
"""

Question to answer:
"""
{question}
"""

Score 1 condition:
{success_condition}

Score 0.6 condition:
{partial_condition}

Score 0 condition:
{failure_condition}

Return only {"score": 0}, {"score": 0.6}, or {"score": 1}.

Table 13: Criterion-group weights by operation. A dash denotes an inapplicable group.

Operation	IF	VQ	DP
Text-to-image	0.60	0.40	—
Addition, removal, replacement	0.45	0.25	0.30
Adjustment	0.40	0.25	0.35
Stylization, background editing	0.45	0.35	0.20
Restoration	0.40	0.45	0.15
Viewpoint editing	0.45	0.30	0.25
Semantic editing	0.55	0.25	0.20
Outpainting, inpainting	0.40	0.40	0.20
Reference editing	0.50	0.25	0.25

Text-to-image system prompt

You are a strict visual judge for a text-to-image benchmark. Inspect the generated image against the prompt and score only the stated atomic question. Use the supplied score anchors exactly. If the partial condition is N/A, return only 0 or 1. Do not speculate beyond visible evidence. Output only one JSON object: {"score": 0}, {"score": 0.6}, or {"score": 1}.

Text-to-image user prompt.

Text-to-image user prompt

The attached image is the GENERATED image.

Generation prompt:
"""
{prompt}
"""

Question to answer:
"""
{question}
"""

Score 1 condition:
{success_condition}

Score 0.6 condition:
{partial_condition}

Score 0 condition:
{failure_condition}

Return only {"score": 0}, {"score": 0.6}, or {"score": 1}.

B.3 Aggregation Weights and Sample Inclusion

Aggregation weights. Table 13 lists operation-specific weights, fixed across models. We exclude inapplicable groups and normalize the remaining weights for the sample-level geometric mean. Within each group, applicable criteria receive equal weight regardless of question counts.

Sample inclusion. Evaluation starts from all 1,150 samples. Unsupported tasks receive zero scores and remain in the denominator; confirmed generation or judge execution errors are excluded. Successfully evaluated outputs retain their scores, including zeros for incorrect or low-quality results. Overall quality averages retained samples; capability profiles and subtask scores use the same inclusion rules.

Table 14: Judge configurations. Model identifiers are the deployment names used by the evaluation API.

Setting	GPT-4o	Gemini 3.1 Pro
Model identifier	`gpt-4o-2024-11-20`	`gemini-3.1-p-priority`
Maximum output tokens	64	64
Temperature	Not explicitly set	Not explicitly set
Thinking budget	—	0
Streaming	Disabled	Disabled

B.4 Model and Judge Configurations

Model inference. We use reference implementations and released checkpoints on NVIDIA A100 GPUs. Inference follows the official default parameters unless otherwise specified. Table 3 lists the reference-model configurations. LongCat-Image-Edit uses the standard checkpoint; FLUX.2 [klein] 4B uses the distilled checkpoint. We evaluate one output per sample and use seed 0 where local pipelines support seed control. We access GPT Image 2, Nano Banana 2 and Seedream 5.0 Pro through APIs.

Reference quality and mobile deployment use separate configurations. DreamLite-mobile uses four sampling steps for deployment.

Output resolution. Text-to-image generation uses each model's default resolution; wallpaper prompts use a 9:16 aspect ratio. Editing requests use the base-image dimensions unless another canvas size is specified. Each pipeline retains its native preprocessing and resolution constraints. Editing outputs are not resized after generation, except for judge preprocessing.

Judge settings. GPT-4o provides the main results, and Gemini 3.1 Pro independently evaluates the same outputs with identical questions and score anchors. Table 14 lists configurations; Sec. B.2 provides prompts.

Judge image preprocessing. Both judges use identical preprocessing. We resize input and output images to a maximum side length of 1,024 pixels with bicubic interpolation, preserving aspect ratios without upscaling. Images are converted to RGB and encoded as JPEG at quality 88. This affects only judge inputs, not saved model outputs.

C Complete Quality Results

C.1 Results Evaluated by GPT-4o

Tables 15 and 16 report GPT-4o quality scores, coverage, tiers and capability profiles under the protocol in Sec. B.4. Unsupported tasks remain in the aggregation with zero scores, so results reflect both task support and output quality.

C.2 Results Evaluated by Gemini

Gemini 3.1 Pro (`gemini-3.1-p-priority`) evaluates the same outputs with identical questions and scoring rules. Table 15 compares quality, coverage and tiers across judges; Tab. 17 reports Gemini capability profiles. Model names and ordering match the GPT-4o tables.

C.3 Detailed Capability Profiles

Table 18 reports the eight criterion scores. LongCat-Image-Edit scores higher on Selection (0.809) than Fidelity (0.653); DeepGen shows the reverse (0.678 versus 0.761). DeepGen and Lance have similar VQ scores (0.744 and 0.739), but different Context scores (0.694 and 0.788). These aggregate comparisons indicate different weaknesses; task-matched inspection is needed to identify specific failures.

Table 15: Overall quality and task coverage evaluated by GPT-4o and Gemini 3.1 Pro. $C_{0.6}$ denotes the proportion of subtasks with mean quality scores of at least 0.6. Quality tiers follow Tab. 2; Q4 is the highest tier.

Model	GPT-4o			Gemini 3.1 Pro		
	$Q \uparrow$	$C_{0.6} \uparrow$	Q-tier	$Q \uparrow$	$C_{0.6} \uparrow$	Q-tier
Closed-source Models						
Nano Banana 2	0.951	1.000	Q4	0.932	1.000	Q4
GPT Image 2	0.967	1.000	Q4	0.960	1.000	Q4
Seedream 5.0 Pro	0.952	1.000	Q4	0.934	1.000	Q4
Open-weight Reference Models						
Lumina-DiMOO-8B	0.434	0.234	Q2	0.314	0.141	Q1
BAGEL-7B-MoT	0.640	0.578	Q2	0.597	0.469	Q2
LongCat-Image	0.120	0.125	Q1	0.119	0.125	Q1
LongCat-Image-Edit	0.667	0.672	Q3	0.632	0.609	Q3
DeepGen 1.0	0.691	0.703	Q3	0.568	0.438	Q2
InternVL-U-4B	0.513	0.359	Q2	0.392	0.219	Q1
VIBE-Image-Edit	0.594	0.562	Q2	0.526	0.469	Q2
OmniGen2	0.617	0.547	Q2	0.542	0.359	Q2
FLUX.2 [klein] 4B	0.828	0.906	Q4	0.786	0.781	Q3
Lance-3B	0.674	0.672	Q3	0.595	0.516	Q2
On-device Models						
Ovis-U1-3B	0.655	0.594	Q2	0.542	0.375	Q2
Janus-Pro-1B	0.080	0.078	Q1	0.080	0.078	Q1
SANA-600M	0.104	0.109	Q1	0.103	0.109	Q1
SANA-Sprint-0.6B	0.100	0.109	Q1	0.098	0.109	Q1
DreamLite	0.623	0.641	Q3	0.524	0.422	Q2
Mobile-O-0.5B	0.097	0.109	Q1	0.093	0.094	Q1

C.4 Additional Qualitative Results

Figs. 6 and 7 compare GPT Image 2, Lance and DreamLite across generation and local editing tasks. DreamLite captures the main content in several generation examples, including the animal and two-person scenes. Precise editing remains less consistent. In example 13, DreamLite adds literal wings instead of winged eyeliner; in example 15, both Lance and DreamLite fail to reproduce the requested text correctly.

D Evaluation Reliability and Sensitivity

D.1 Cross-Judge Consistency

Paired results. We compare 19 models with valid results from both judges, using the same outputs, questions and score anchors. Sample-level comparisons retain outputs scored successfully by both judges, including valid zero scores. Excluding preset zeros for unsupported tasks leaves 15,716 paired scores.

Ranking and score agreement. Overall quality rankings have a Spearman correlation of $\rho = 0.968$, including unsupported-task zeros in Q. Paired sample scores have $\rho = 0.790$ and a mean absolute difference of 0.139. Gemini scores are lower by 0.051 on average across the 19 models.

Quality-tier changes. With identical thresholds, 13 models retain their tiers and six move down one tier with Gemini (Tab. 15). FLUX.2 [klein] moves from Q4 to Q3; DeepGen, Lance and DreamLite move from Q3 to Q2; Lumina-DiMOO and InternVL-U move from Q2 to Q1. Rankings are consistent overall, but scores and tiers remain sensitive to judge choice.

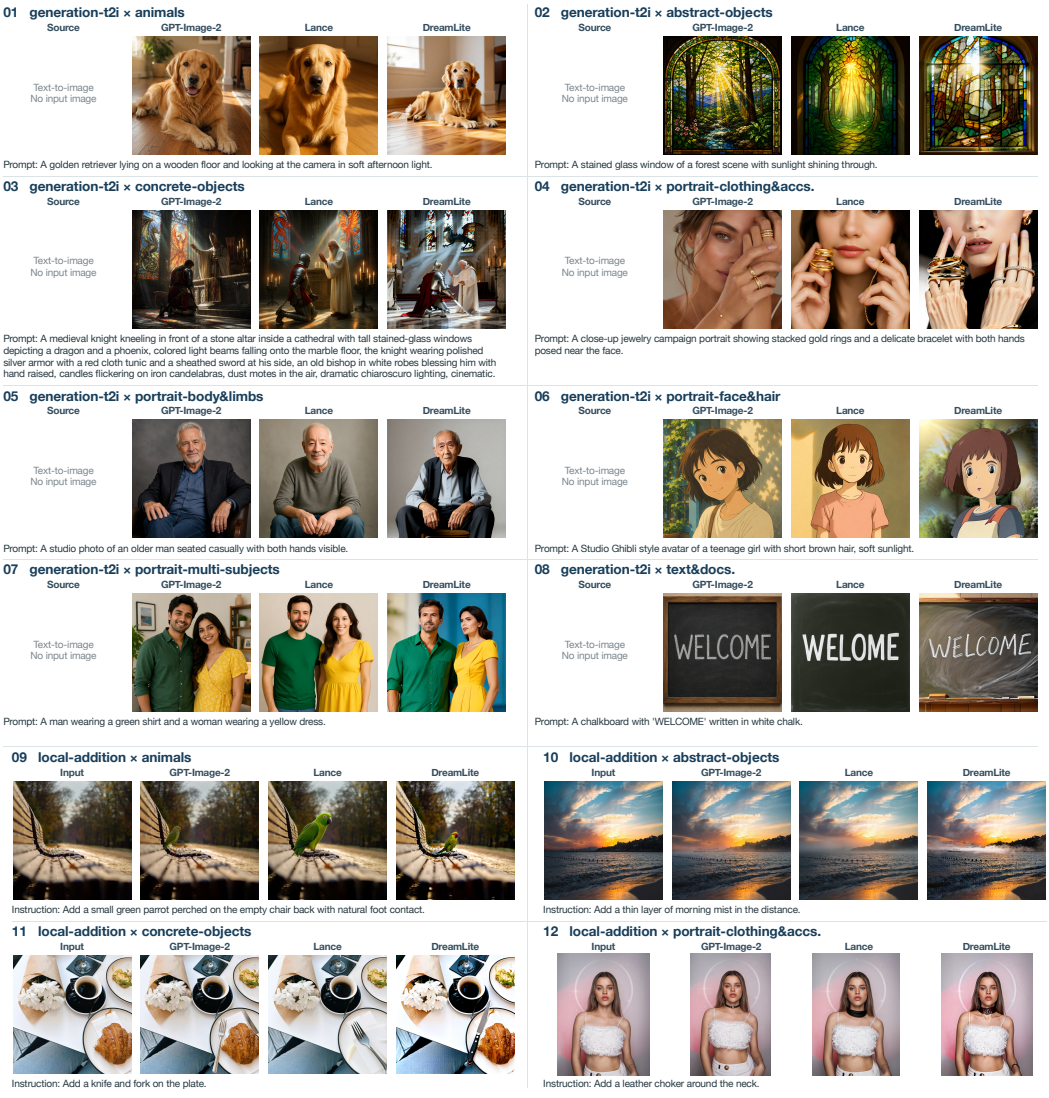

Figure 6: Additional qualitative comparisons for generation and local addition (examples 1–12). Each example shows the prompt or instruction and outputs from GPT Image 2, Lance and DreamLite. Editing examples also show the input image.

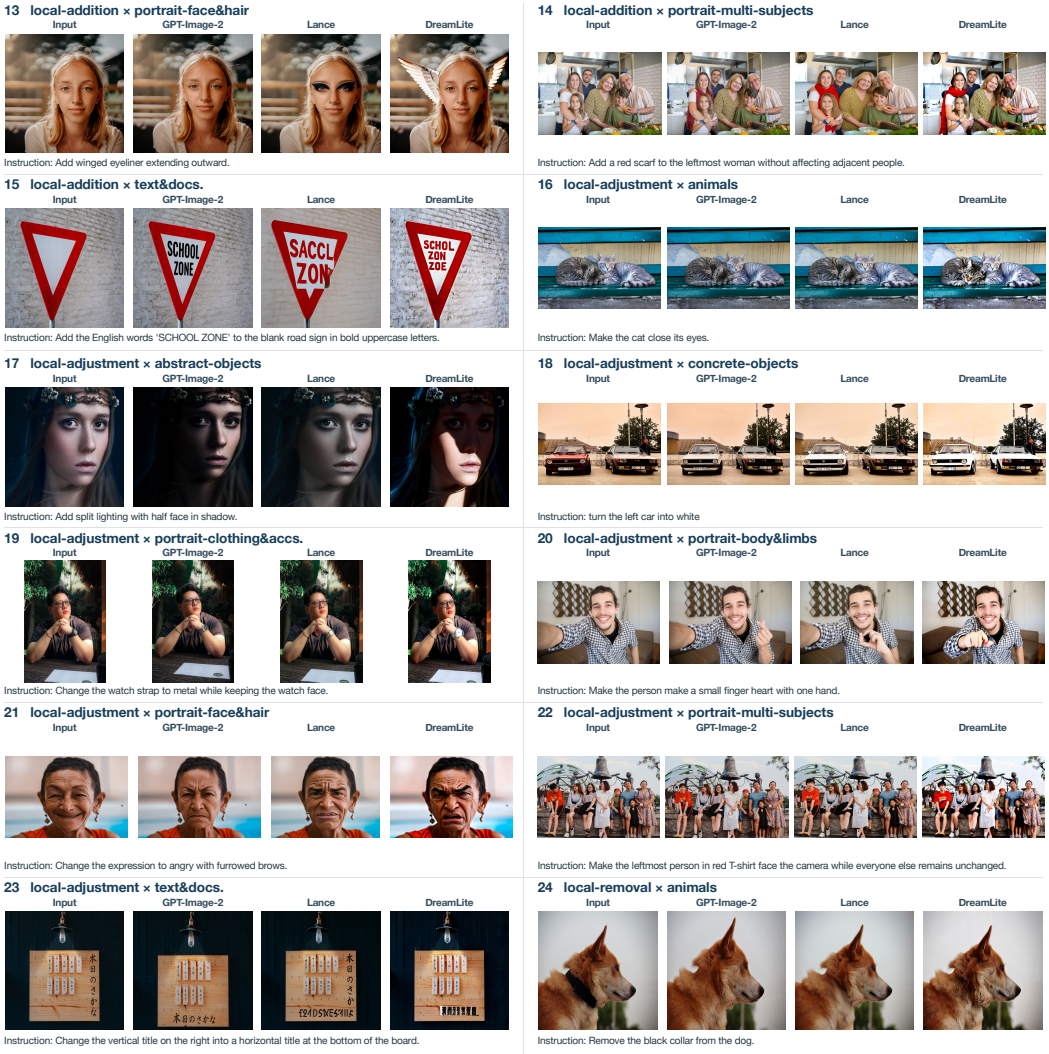

Figure 7: Additional qualitative comparisons for local addition, adjustment and removal (examples 13–24). Each example shows the input image, editing instruction and outputs from GPT Image 2, Lance and DreamLite.

Table 16: Capability profiles scored by GPT-4o (`gpt-4o-2024-11-20`). General and Text. denote general objects and text and documents. IF, VQ and DP denote instruction following, visual quality and detail preservation. Scores range from 0 to 1, with higher values indicating better performance.

Model	Operation					Target				Criteria		
	Gen.	Local	Global	Underst.	Ref.	General	Animals	Portraits	Text.	IF	VQ	DP
Closed-source Models												
Nano Banana 2	0.986	0.959	0.912	0.954	0.925	0.953	0.986	0.950	0.931	0.946	0.972	0.974
GPT Image 2	0.988	0.977	0.931	0.951	0.959	0.958	0.985	0.971	0.963	0.965	0.984	0.978
Seedream 5.0 Pro	0.977	0.964	0.909	0.932	0.949	0.943	0.958	0.960	0.943	0.947	0.973	0.976
Open-weight Reference Models												
Lumina-DiMOO-8B	0.858	0.458	0.355	0.327	0.073	0.506	0.532	0.421	0.273	0.478	0.472	0.615
BAGEL-7B-MoT	0.830	0.732	0.607	0.574	0.166	0.646	0.774	0.681	0.436	0.685	0.682	0.771
LongCat-Image	0.920	0.000	0.000	0.000	0.000	0.107	0.180	0.113	0.142	0.120	0.121	0.000
LongCat-Image-Edit	0.000	0.914	0.823	0.659	0.201	0.687	0.672	0.692	0.542	0.661	0.704	0.857
DeepGen 1.0	0.906	0.773	0.719	0.572	0.201	0.717	0.763	0.708	0.549	0.735	0.744	0.696
InternVL-U-4B	0.874	0.502	0.512	0.523	0.164	0.527	0.562	0.540	0.379	0.610	0.550	0.480
VIBE-Image-Edit	0.000	0.818	0.710	0.641	0.150	0.636	0.610	0.645	0.341	0.605	0.634	0.807
OmniGen2	0.828	0.651	0.557	0.460	0.464	0.653	0.758	0.661	0.335	0.658	0.698	0.730
FLUX.2 [klein] 4B	0.905	0.844	0.878	0.703	0.720	0.862	0.895	0.860	0.619	0.817	0.883	0.951
Lance-3B	0.762	0.772	0.722	0.636	0.158	0.729	0.762	0.722	0.363	0.685	0.739	0.761
On-device Models												
Ovis-U1-3B	0.823	0.719	0.749	0.590	0.153	0.734	0.751	0.692	0.325	0.684	0.704	0.716
Janus-Pro-1B	0.616	0.000	0.000	0.000	0.000	0.091	0.151	0.082	0.018	0.088	0.079	0.000
SANA-600M	0.800	0.000	0.000	0.000	0.000	0.102	0.181	0.105	0.071	0.106	0.106	0.000
SANA-Sprint-0.6B	0.768	0.000	0.000	0.000	0.000	0.102	0.172	0.102	0.057	0.104	0.102	0.000
DreamLite	0.803	0.673	0.697	0.607	0.145	0.711	0.729	0.654	0.285	0.652	0.636	0.758
Mobile-O-0.5B	0.744	0.000	0.000	0.000	0.000	0.099	0.153	0.104	0.046	0.097	0.103	0.000

D.2 Agreement with Human Judgments

Human evaluation. We sample 500 model outputs from each model listed in Tab. 4. Unsupported tasks and execution failures are excluded. Two annotators independently evaluate all selected outputs. They did not participate in benchmark construction or evaluation-question review. Model names and VLM judge scores are hidden from both annotators. They inspect the instructions, input images and model outputs. They answer the same questions as the VLM judges using the same 0, 0.6 and 1 anchors. Sample scores follow the aggregation procedure in Sec. B.3.

Agreement. The annotators' sample scores have a Spearman correlation of $\rho = 0.904$. We average their sample scores to form a human reference.

GPT-4o and Gemini 3.1 Pro achieve correlations of $\rho = 0.779$ and $\rho = 0.852$ with this reference on the same outputs. This comparison concerns rubric-based scores rather than user preference.

D.3 Sensitivity to Criterion-Group Weights

We reaggregate the existing GPT-4o judgments for 19 models under alternative criterion-group weights. We test equal weights and increase or decrease each default group weight by 25%, one group at a time. Weights are renormalized over the applicable groups for each sample. Question scores, criterion applicability, sample inclusion and tier thresholds remain fixed. We recompute overall quality Q, task coverage and Q-tiers for each setting.

Tab. 19 shows that all settings preserve the overall quality ranking ($\rho = 1.000$). The largest absolute change in Q is 0.0052. All models retain their tiers under equal weights. Increasing the DP weight or decreasing the VQ weight moves Ovis-U1-3B from Q2 to Q3. Its coverage rises from $38/64$ to $39/64$ subtasks and crosses the 60% threshold. All other tier assignments remain unchanged.

Table 17: Capability profiles evaluated by Gemini 3.1 Pro (`gemini-3.1-p-priority`). General and Text. denote general objects and text and documents. IF, VQ and DP denote instruction following, visual quality and detail preservation. Target scores use the same sample categories as the GPT-4o results. Scores range from 0 to 1, with higher values indicating better performance.

Model	Operation					Target				Criteria		
	Gen.	Local	Global	Underst.	Ref.	General	Animals	Portraits	Text.	IF	VQ	DP
Closed-source Models												
Nano Banana 2	0.969	0.943	0.889	0.885	0.938	0.925	0.971	0.941	0.897	0.939	0.957	0.962
GPT Image 2	0.981	0.968	0.927	0.935	0.967	0.954	0.988	0.967	0.934	0.962	0.981	0.964
Seedream 5.0 Pro	0.966	0.948	0.864	0.907	0.952	0.919	0.954	0.951	0.903	0.934	0.969	0.959
Open-weight Reference Models												
Lumina-DiMOO-8B	0.874	0.288	0.232	0.163	0.052	0.332	0.462	0.301	0.246	0.423	0.358	0.447
BAGEL-7B-MoT	0.838	0.696	0.533	0.467	0.137	0.578	0.725	0.641	0.444	0.671	0.642	0.772
LongCat-Image	0.911	0.000	0.000	0.000	0.000	0.106	0.182	0.109	0.147	0.121	0.118	0.000
LongCat-Image-Edit	0.000	0.879	0.721	0.654	0.191	0.646	0.638	0.670	0.484	0.640	0.695	0.832
DeepGen 1.0	0.888	0.605	0.570	0.453	0.176	0.580	0.700	0.583	0.435	0.685	0.661	0.517
InternVL-U-4B	0.869	0.333	0.396	0.377	0.123	0.388	0.427	0.413	0.318	0.568	0.478	0.326
VIBE-Image-Edit	0.000	0.741	0.588	0.558	0.123	0.562	0.563	0.569	0.300	0.579	0.548	0.729
OmniGen2	0.859	0.569	0.475	0.337	0.352	0.538	0.738	0.586	0.321	0.615	0.658	0.652
FLUX.2 [klein] 4B	0.900	0.795	0.805	0.669	0.700	0.812	0.902	0.811	0.598	0.791	0.847	0.931
Lance-3B	0.792	0.678	0.593	0.489	0.147	0.614	0.741	0.643	0.338	0.640	0.693	0.651
On-device Models												
Ovis-U1-3B	0.821	0.562	0.657	0.415	0.123	0.589	0.644	0.558	0.340	0.619	0.637	0.513
Janus-Pro-1B	0.610	0.000	0.000	0.000	0.000	0.088	0.149	0.078	0.036	0.089	0.076	0.000
SANA-600M	0.789	0.000	0.000	0.000	0.000	0.100	0.176	0.101	0.082	0.108	0.101	0.000
SANA-Sprint-0.6B	0.748	0.000	0.000	0.000	0.000	0.099	0.167	0.095	0.070	0.103	0.097	0.000
DreamLite	0.771	0.561	0.528	0.478	0.145	0.593	0.685	0.538	0.255	0.621	0.522	0.617
Mobile-O-0.5B	0.713	0.000	0.000	0.000	0.000	0.097	0.160	0.092	0.059	0.097	0.098	0.000

Model rankings are stable under these settings, while tier assignments near a coverage threshold can change.

D.4 Sensitivity to Tier Thresholds

Quality thresholds. The tiers use manually selected comparison thresholds. We shift all minimum requirements for Q, $C_{0.6}$ and criterion-group scores by $\delta \in \{-0.05, -0.02, 0.02, 0.05\}$.

The subtask quality cutoff used to compute $C_{0.6}$ remains fixed at 0.6. Measured scores remain unchanged. Depending on the setting, 16–19 of the 19 models retain their tiers (Tab. 20). All changes are limited to one tier. At $\delta = 0.05$, FLUX.2 [klein] moves from Q4 to Q3 and DreamLite from Q3 to Q2.

Deployment thresholds. We scale all D3 and D4 resource limits by a common factor. We test factors of 0.8, 0.9, 1.1 and 1.2 with execution status fixed. All six on-device models retain their iPhone 17 Pro FP16 tiers when budgets decrease by 10% or increase by 10% or 20%. A 20% reduction moves DreamLite and Mobile-O from D4 to D3 because their sizes exceed the reduced storage limit. Across settings, Mobile-O remains Q1 and DreamLite remains Q2 or Q3, while both have the same deployment tier.

Table 18: Scores for the eight evaluation criteria, evaluated by GPT-4o (`gpt-4o-2024-11-20`). IF, VQ and DP denote instruction following, visual quality and detail preservation. Scores range from 0 to 1, with higher values indicating better performance.

Model	IF		VQ			DP		
	Fidelity	Selection	Integrity	Coherence	Presentation	Context	Structure	Identity
Closed-source Models								
Nano Banana 2	0.941	0.952	0.980	0.956	0.978	0.976	0.964	0.986
GPT Image 2	0.963	0.969	0.989	0.975	0.987	0.976	0.979	0.984
Seedream 5.0 Pro	0.945	0.961	0.983	0.959	0.978	0.970	0.979	0.984
Open-weight Reference Models								
Lumina-DiMOO-8B Xin et al. (2025)	0.479	0.419	0.460	0.431	0.525	0.622	0.645	0.528
BAGEL-7B-MoT Deng et al. (2025)	0.688	0.670	0.683	0.635	0.729	0.789	0.782	0.774
LongCat-Image Team et al. (2025)	0.120	0.000	0.121	0.123	0.117	0.000	0.000	0.000
LongCat-Image-Edit Team et al. (2025)	0.653	0.809	0.712	0.683	0.718	0.867	0.848	0.848
DeepGen 1.0 Wang et al. (2026)	0.761	0.678	0.743	0.715	0.774	0.694	0.725	0.652
InternVL-U-4B Tian et al. (2026)	0.667	0.457	0.552	0.528	0.571	0.415	0.515	0.538
VIBE-Image-Edit Alekseenko et al. (2026)	0.597	0.725	0.637	0.607	0.658	0.820	0.822	0.790
OmniGen2 Wu et al. (2026a)	0.684	0.580	0.722	0.645	0.726	0.711	0.748	0.743
FLUX.2 [klein] 4B Black Forest Labs (2026)	0.813	0.803	0.884	0.865	0.900	0.946	0.964	0.940
Lance-3B Fu et al. (2026)	0.681	0.703	0.744	0.726	0.745	0.788	0.742	0.772
On-device Models								
Ovis-U1-3B Wang et al. (2025a)	0.708	0.609	0.695	0.684	0.733	0.698	0.783	0.649
Janus-Pro-1B Chen et al. (2025c)	0.088	0.000	0.072	0.082	0.082	0.000	0.000	0.000
SANA-600M Xie et al. (2024)	0.106	0.000	0.103	0.112	0.102	0.000	0.000	0.000
SANA-Sprint-0.6B Chen et al. (2025b)	0.104	0.000	0.098	0.108	0.098	0.000	0.000	0.000
DreamLite Feng et al. (2026)	0.664	0.604	0.629	0.596	0.682	0.709	0.828	0.732
Mobile-O-0.5B Shaker et al. (2026)	0.097	0.000	0.104	0.108	0.098	0.000	0.000	0.000

Table 19: Sensitivity to criterion-group weights under GPT-4o. Changes are relative to the default weights. Mean and maximum $|\Delta Q|$ are computed across 19 models. Spearman ρ measures model-ranking agreement.

Setting	Mean $\lvert\Delta Q\rvert$	Max. $\lvert\Delta Q\rvert$	Spearman ρ	Same tier
Default	0.0000	0.0000	1.000	19/19
Equal weights	0.0016	0.0052	1.000	19/19
IF $+25\%$	0.0009	0.0038	1.000	19/19
IF -25%	0.0011	0.0045	1.000	19/19
VQ $+25\%$	0.0007	0.0029	1.000	19/19
VQ -25%	0.0009	0.0035	1.000	18/19
DP $+25\%$	0.0008	0.0023	1.000	18/19
DP -25%	0.0009	0.0030	1.000	19/19

E Mobile Deployment Details and Results

E.1 Deployment Configurations and Measurement Protocol

Devices and runtimes. We evaluate mobile deployment on an iPhone 17 Pro and a Xiaomi 15 Pro. The iOS implementation uses Core ML for image-generation components and MLX for external text encoders. The Android implementation uses ONNX Runtime with the Qualcomm QNN HTP backend for image-generation components and CPU execution for text encoders. The baseline uses FP16 backbones and decoders. Text encoders retain their deployment-specific precision. Reference quality evaluation is performed separately from mobile deployment.

Generation settings. Deployment tests use text-to-image generation with the prompt: "A cinematic portrait of a young woman surrounded by flowers, soft natural light." Within each model, the prompt, seed, resolution and sampling settings remain fixed across precision configurations. Output resolution is 1024×1024, except for Mobile-O (512×512) and Janus-Pro-1B (384×384).

Table 20: Sensitivity of tier assignments to threshold changes. Quality thresholds are shifted by δ; deployment limits are multiplied by α. Unchanged counts are relative to the default tiers.

Setting	Unchanged	Tier changes
Q-tier: 19 models, GPT-4o		
$\delta = -0.05$	16/19	BAGEL-7B-MoT: $Q2 \rightarrow Q3$; VIBE-Image-Edit: $Q2 \rightarrow Q3$; Ovis-U1-3B: $Q2 \rightarrow Q3$
$\delta = -0.02$	18/19	Ovis-U1-3B: $Q2 \rightarrow Q3$
$\delta = +0.02$	19/19	None
$\delta = +0.05$	16/19	Lumina-DiMOO-8B: $Q2 \rightarrow Q1$; FLUX.2 [klein] 4B: $Q4 \rightarrow Q3$; DreamLite: $Q3 \rightarrow Q2$
D-tier: six on-device models, iPhone 17 Pro		
$\alpha = 0.80$	4/6	DreamLite: $D4 \rightarrow D3$; Mobile-O-0.5B: $D4 \rightarrow D3$
$\alpha = 0.90$	6/6	None
$\alpha = 1.10$	6/6	None
$\alpha = 1.20$	6/6	None

Table 21: Deployment results on the Xiaomi 15 Pro with FP16 backbones and decoders. Res. denotes output resolution; Mem. is peak process PSS. E2E excludes initial loading, warm-up and file saving. D-tiers follow Tab. 2.

Model	Res.	Size (MiB)	E2E (s)	Mem. (MiB)	D-tier
DreamLite	1024^2	1804.5	18.19	2390.5	D3
Mobile-O-0.5B	512^2	1770.2	13.30	1757.1	D3
SANA-Sprint-0.6B	1024^2	3069.1	25.24	3896.7	D3
SANA-600M	1024^2	3065.4	139.68	3897.6	D2
Janus-Pro-1B[‡]	384^2	2974.1	76.01	3253.8	D2

[‡]Janus uses precomputed prompt embeddings, so prompt preparation is excluded.

DreamLite-mobile uses four sampling steps. SANA-600M uses 18 denoising steps in the deployment tests. Mobile sampling steps are reported in Tab. 23.

Repeated measurements. The first five models use one warm-up and three measured runs. We report mean inference latency and mean per-run peak memory. Memory is sampled at a nominal interval of 25 ms. We use application physical footprint on iOS and process proportional set size (PSS) on Android. The two memory metrics are reported separately by platform.

Measurement boundaries. Inference latency covers prompt processing, text encoding, all generation steps, image decoding and post-processing. Initial loading, warm-up and result-file saving are excluded from baseline timing. Loading repeated during generation remains part of measured E2E. Tab. 23 separates totals from measured stages. Peak memory is the highest sampled value within each measured run. Model size includes the backbone, decoder, required encoders and supporting inference files. FP16 refers to the backbone and decoder; text encoders can use lower precision. File size therefore depends on component precision and compiled representation, not parameter count alone. Peak process memory measures runtime usage rather than the total size of model files. Shared files are counted once, and duplicate artifacts and development files are excluded. Storage and memory are reported in MiB. Failed configurations are recorded separately and excluded from successful-run averages.

E.2 Android Deployment Results

Tab. 21 reports latency and resource costs. Baseline rows describe individual deployments.

E.3 Quantization Experiments

Table 22: Backbone–decoder precision matrix with runtime. BB denotes cumulative backbone latency over all denoising steps, or autoregressive generation for Janus-Pro-1B; Dec. denotes image decoding. Mem. is measured in MiB using iOS physical footprint or Android process PSS. C: completed measurement under the standard execution setup; R: completed measurement with resource-adapted execution; F: execution failed; NT: not tested. R configurations use execution adjustments to accommodate device resource limits, such as component offloading. C and R indicate execution completion, not acceptable output quality. Timings exclude initial loading and warm-up, but include repeated context operations within backbone execution. These component measurements are not complete E2E latency.

Precision		iPhone 17 Pro				Xiaomi 15 Pro			
Backbone	Decoder	BB (s)	Dec. (s)	Mem.	Status	BB (s)	Dec. (s)	Mem.	Status
DreamLite: 1024^2, 4 denoising steps									
FP16	FP16	3.06	0.04	1305.7	C	15.19	0.15	2390.5	R
FP16	W8A16	3.42	0.04	1306.5	C	15.22	0.07	2392.5	R
FP16	W8A8	3.06	0.02	1305.0	C	14.86	0.03	2406.1	R
W8A16	FP16	3.26	0.04	1297.9	C	14.18	0.13	2406.2	C
W8A16	W8A16	3.24	0.05	1298.3	C	14.03	0.10	2390.3	C
W8A16	W8A8	3.16	0.03	1307.7	C	13.94	0.05	2377.0	C
W8A8	FP16	–	–	–	F	11.04	0.15	2388.7	C
W8A8	W8A16	–	–	–	NT	10.92	0.10	2392.3	C
W8A8	W8A8	–	–	–	NT	10.85	0.04	2395.5	C
Mobile-O-0.5B: 512^2, 20 denoising steps									
FP16	FP16	2.27	0.23	426.8	C	11.26	1.10	1757.1	R
FP16	W8A16	2.30	0.23	449.2	C	11.30	0.88	1767.0	R
FP16	W8A8	2.26	0.15	423.1	C	11.33	0.45	1768.7	R
W8A16	FP16	2.12	0.22	423.5	C	6.66	1.21	3149.2	C
W8A16	W8A16	2.12	0.22	424.2	C	6.16	0.64	3127.6	C
W8A16	W8A8	2.12	0.16	436.5	C	6.17	0.33	3179.4	C
W8A8	FP16	–	–	–	F	–	–	–	F
W8A8	W8A16	–	–	–	F	–	–	–	F
W8A8	W8A8	–	–	–	F	–	–	–	F
SANA-Sprint-0.6B: 1024^2, 2 denoising steps									
FP16	FP16	0.78	3.15	1727.2	C	9.71	8.29	3896.7	R
FP16	W8A16	0.68	2.68	1851.1	C	8.27	7.83	3893.1	R
FP16	W8A8	0.51	1.49	1759.1	C	8.49	4.62	3892.2	R
W8A16	FP16	0.51	2.29	1744.3	C	1.27	10.74	4031.9	C
W8A16	W8A16	0.60	2.54	1862.5	C	1.38	6.63	4072.1	C
W8A16	W8A8	0.83	2.33	1854.7	C	1.38	3.60	4051.3	C
W8A8	FP16	–	–	–	F	–	–	–	F
W8A8	W8A16	–	–	–	F	–	–	–	F
W8A8	W8A8	–	–	–	F	–	–	–	F
SANA-600M: 1024^2, 18 denoising steps									
FP16	FP16	11.48	3.03	1775.2	C	125.44	7.24	3897.6	R
FP16	W8A16	8.09	2.15	1880.3	C	119.56	6.58	3901.7	R
FP16	W8A8	7.95	1.53	1738.0	C	–	–	–	F
W8A16	FP16	8.10	2.21	1738.4	C	18.77	9.80	4255.4	C
W8A16	W8A16	8.14	2.23	1853.7	C	19.39	6.13	4307.6	C
W8A16	W8A8	8.19	1.56	1854.0	C	20.17	3.63	4316.1	C
W8A8	FP16	–	–	–	F	–	–	–	F
W8A8	W8A16	–	–	–	F	–	–	–	F
W8A8	W8A8	–	–	–	F	–	–	–	F
Janus-Pro-1B: 384^2, 576 image tokens									
FP16	FP16	55.92	0.08	1505.4	C	74.33	0.23	3253.8	R
FP16	W8A16	56.60	0.08	1502.2	C	74.33	0.98	3253.8	R

Continued on the next page.

Table 22 (continued)

Precision		iPhone 17 Pro				Xiaomi 15 Pro			
Backbone	Decoder	BB (s)	Dec. (s)	Mem.	Status	BB (s)	Dec. (s)	Mem.	Status
FP16	W8A8	59.36	0.06	1503.8	C	74.33	0.63	3253.8	R
W8A16	FP16	52.33	0.08	1500.8	C	–	–	–	NT
W8A16	W8A16	52.08	0.08	1503.6	C	–	–	–	NT
W8A16	W8A8	51.50	0.05	1501.7	C	–	–	–	NT
W8A8	FP16	–	–	–	NT	–	–	–	NT
W8A8	W8A16	–	–	–	NT	–	–	–	NT
W8A8	W8A8	–	–	–	NT	–	–	–	NT

Settings. We test FP16, W8A16 and W8A8 for the backbone and image decoder independently. FP16 uses 16-bit floating-point weights and activations. W8A16 and W8A8 use 8-bit weights with 16-bit and 8-bit activations, respectively. These labels apply to the image networks; supporting operators follow the runtime implementation.

Text encoders and other shared resources retain their deployment settings. We consider nine precision combinations per model on each device. The prompt, seed, resolution and generation settings remain fixed within each model's device-level comparison. Tab. 22 reports backbone latency, decoder latency and peak memory. R entries use execution adjustments to accommodate device resource limits. These adjustments may change component placement, loading or offloading. R entries should therefore not be treated as comparisons that vary precision alone.

Execution and resource costs. On Android, 29 configurations complete measurement, 10 fail and six remain untested. The table distinguishes standard execution from configurations adapted to device resource limits. On iOS, 30 configurations complete measurement, 10 fail and five remain untested. On iOS, W8A16 reduces Mobile-O's compiled backbone size from 1129.4 to 566.9 MiB. However, lower precision does not consistently reduce latency or peak memory. With an FP16 decoder, Mobile-O's Android W8A16 backbone takes 6.66 s, with peak PSS of 3149.2 MiB. On iOS, DreamLite's W8A16 backbone takes 3.26 s, compared with 3.06 s for FP16 under the same decoder precision. The results reflect the deployed configurations, including graph partitioning, context reuse and runtime conditions; timing differences cannot be attributed to bit width alone.

Scope of the results. Execution completion does not establish acceptable output quality. The Android W8A8 DreamLite backbone produces mosaic artifacts with all three decoder precisions. Quantized SANA-Sprint backbones exhibit texture collapse in the Android tests. These timings remain useful for runtime analysis, but do not demonstrate usable generation. We do not repeat the full MoGE-Bench quality evaluation for each precision combination or infer quality retention from these deployment measurements.

E.4 Stage-Level Latency

Deployment settings. All models use the prompt in Sec. E.1. Native prompt templates and sequence lengths are retained. Backbones and image decoders use FP16 through Core ML with CPU and Neural Engine execution allowed. DreamLite, Mobile-O and both SANA variants use 4-bit affine text encoders through MLX on the GPU; Ovis-U1 uses an FP16 MLX text encoder. Ovis-U1's vision encoder and conditioning refiner, and Mobile-O's connector, use FP16 Core ML. Janus-Pro-1B performs prompt prefill within its FP16 backbone. Tokenization, scheduling and output processing use host-side execution.

Caching and repeated measurements. DreamLite, Mobile-O, both SANA variants and Janus retain loaded resources and recompute prompt features for each run. Janus-Pro-1B resets its KV cache before each run and reuses it during autoregressive generation. Ovis-U1 uses streaming loads to limit memory usage. Its measured inference latency includes loading and offloading performed during generation. We report means and sample standard deviations (SD).

Table 23: Stage-level latency on the iPhone 17 Pro, in milliseconds. Text/Cond. includes Mobile-O's conditioning connector. Generation includes scheduling; for Ovis-U1, this entry also includes the conditioning refiner. – indicates a stage that is not applicable or not separately timed. Totals also include pipeline overhead not separately listed.

Stage	DreamLite	Mobile-O-0.5B	SANA-Sprint-0.6B	SANA-600M	Janus-Pro-1B	Ovis-U1-3B
Resolution	1024^2	512^2	1024^2	1024^2	384^2	1024^2
Steps / image tokens	4	20	2	18	576	50
Sequence length†	64	77	300	300	23	303/268
Runs used for statistics	3	3	3	3	3	3
Tokenization	–	–	–	–	7.29	–
Vision encoding	–	–	–	–	–	621.36
Text/Cond.	65.03	95.27	280.76	297.56	–	1398.90
Prompt prefill	–	–	–	–	2206.59	–
Denoising / autoregressive	3058.17	2267.73	783.94	11480.65	55916.85	306368.47
Codebook lookup	–	–	–	–	0.18	–
Image decoding	36.37	230.14	3153.85	3034.79	83.06	14991.60
Output processing	4.63	20.60	133.22	151.39	11.76	2108.02
Total (mean)‡	3164.47	2808.67	4488.40	15327.80	58256.19	325739.36
Total (SD)	5.92	32.41	60.95	379.11	1086.01	1902.89

†The first four entries are fixed conditioning lengths, including padding, rather than raw prompt-token counts. Janus reports prefill tokens. Ovis reports positive/negative text-encoder sequence lengths after visual-token merging. ‡Ovis-U1 uses 50 denoising steps and 100 CFG forwards. Its measured mean E2E latency is 325.74 s, excluding initial loading and file writing. DreamLite excludes its separately timed saving block.

F Extended Related Work

F.1 On-Device Image Generation and Editing Models

Mobile image generation is constrained by model storage, activation memory, iterative denoising and device-specific operator support. Early studies enabled Stable Diffusion on phones through GPU-aware graph optimization and model compression Chen et al. (2023); Choi et al. (2023). Subsequent work reduced the cost of UNet and DiT pipelines through pruning, distillation and architecture optimization Li et al. (2023); Zhao et al. (2024b); Castells et al. (2024); Chen et al. (2025a); Chavhan et al. (2026); Hu et al. (2026); Du et al. (2026). Low-bit quantization targets storage and computation costs Zhao et al. (2024a); Li et al. (2024b). Recent mobile systems pursue efficient generation and broader capabilities, including image editing Shaker et al. (2026); Feng et al. (2026); Deng et al. (2026); Kodavanti et al. (2026); Duan et al. (2026).

MoGE-Bench evaluates generation and editing capability alongside device-specific deployment costs.

F.2 Generation and Editing Benchmarks

Image-generation benchmarks have expanded from broad prompt collections, such as DrawBench and PartiPrompts, to evaluations of compositional and semantic correctness Saharia et al. (2022); Yu et al. (2023); Hu et al. (2023); Huang et al. (2025); Ghosh et al. (2023); Bakr et al. (2023); Lee et al. (2023); Li et al. (2024a); Wu et al. (2024); Wang et al. (2025b). Image-editing benchmarks evaluate instruction following, edit fidelity and content preservation. They also cover real-user instructions, reasoning, multi-turn interaction and document editing Zhang et al. (2023); Ju et al. (2023); Ma et al. (2024); Qian et al. (2025); Liu et al. (2025); Jia et al. (2026); Wu et al. (2026c); Ye et al. (2026); Yi et al. (2026); Jiang et al. (2026). Unified suites cover generation, editing and reference-conditioned tasks Ku et al. (2024b); Pan et al. (2025); Hua et al. (2025); Sani et al. (2026).