Are Image Generators Zero-Shot Perceivers? A Rigorous Evaluation

Shangzhe Di Zhaokai Wang
Shanghai Jiao Tong University
dishangzhe@sjtu.edu.cn

Abstract

Recent work such as Vision Banana shows that lightweight instruction tuning can enable an image generator to achieve state-of-the-art performance across multiple visual perception tasks. Motivated by this perspective, we ask how far image generators can go on public visual perception benchmarks in a zero-shot setting. We introduce **PROBEGEN***, a benchmark for* ***zero-shot generative perception*** *that casts monocular depth estimation, referring/reasoning segmentation, and object counting as conditional generation tasks specified through text prompts, and compares 20 models in total—including proprietary and open-weight image generators, specialist perception models, and MLLMs—across 11 published benchmarks. We observe that pretrained image generators show measurable zero-shot perceptual competence, but with a clear trade-off: specialist models remain stronger in in-distribution accuracy and efficiency, whereas generative models are often more robust under distribution shift and better at compositional semantic reasoning. We hope this study helps establish zero-shot generative perception as a meaningful research direction and provides a useful foundation for future work at the intersection of visual generation and understanding.*

1. Introduction

Modern image generators must internalize a rich understanding of the visual world, including object structure, geometry, material appearance, and semantic relationships, in order to synthesize realistic scenes [12, 13, 25, 30]. This observation invites a provocative hypothesis: *the same internal representations that enable generation may already encode the information needed for visual perception.* Recent work lends credibility to this idea. Vision Banana demonstrates that instruction-tuned generators can be recast as generalist vision learners across a broad task spectrum [10], and video generation models have been shown to exhibit emergent zero-shot reasoning [34]. Yet a fundamental question remains open: when evaluated rigorously on public benchmarks, **how competitive are mainstream image generators as *zero-shot perceivers* compared to state-of-the-art specialist models?**

The current landscape offers partial answers from opposite ends. On the discriminative side, dedicated architectures set strong standards: the Depth Anything series [20, 37, 38] for monocular depth, the Segment Anything family [5, 17, 28] for segmentation, and CountGD [1, 2] for object counting. On the generative side, methods such as Marigold [16] and Lotus-2 [14] show that diffusion models can be turned into competitive dense predictors, but only after task-specific fine-tuning or architectural modification. Between these two poles lies an unexplored regime that we call *zero-shot generative perception*: the perceptual capability of general-purpose image generators, evaluated without any adaptation, on the same benchmarks used to measure specialist progress. This is the regime we investigate.

We conduct a systematic, controlled study of zero-shot generative perception—which we release as PROBEGEN—on three complementary visual perception tasks: monocular depth estimation, which probes geometric understanding; referring/reasoning segmentation, which probes semantic grounding and compositional reasoning; and object counting, which probes instance-level localization. All three tasks are cast as conditional image generation problems specified entirely through text prompts, yielding a unified inference protocol that accommodates proprietary and open-weight generative models. We evaluate a diverse set of mainstream image generators, *e.g.*, Nano Banana [12, 13], Seedream [29, 30, 32], GPT-Image [22, 23, 25], Flux.2-dev [18], and Qwen-Edit-2511 [35], and benchmark them head-to-head against leading specialist models under identical data and metric conditions.

Our findings reveal that, although specialist models retain a clear advantage in in-distribution accuracy and computational efficiency, general-purpose generators demonstrate markedly greater robustness under distribution shift and superior compositional reasoning on complex queries. The gap is most pronounced in out-of-distribution (OOD) domains, such as manga imagery (Manga109 [36]) and artistic paintings (DRAM [6]) in Figure 1, where the

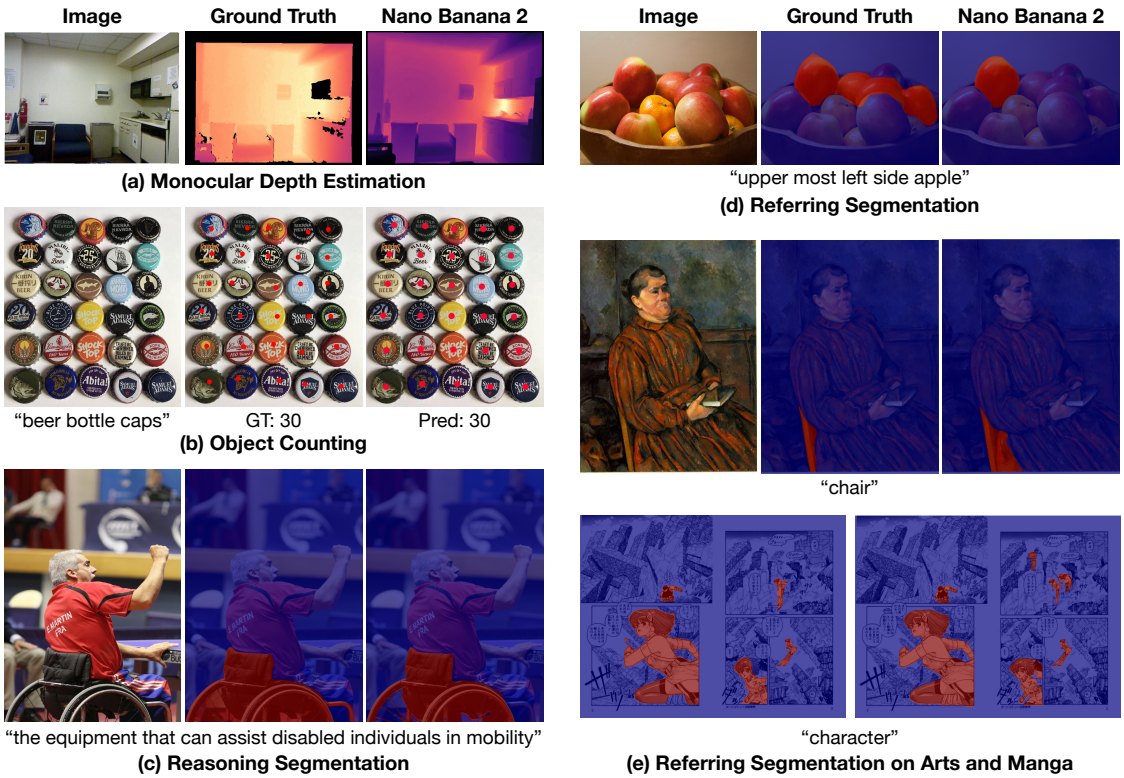

Figure 1. **Zero-shot perception abilities of Nano Banana 2.** (a) Monocular depth estimation. (b) Object counting. (c) Reasoning segmentation. (d) Referring segmentation. Nano Banana 2 manages to segment the desired instance despite the incorrect annotation. (e) Referring segmentation on out-of-distribution domains of artistic and manga-style images. Note that segmentation results are recolored on the input image for visualization.

broad visual prior of a generative model provides semantic grounding that specialist pipelines lack. To better understand this behavior, we further characterize the systematic failure modes of generative perception—chiefly a failure to preserve the input (regenerating the scene instead of transforming it in place) and a failure to follow the task instruction—that distinguish generative errors from those of conventional discriminative models and point toward concrete avenues for improvement.

2. Related Work

Discriminative specialists for visual perception. Visual perception has long been dominated by specialized discriminative models, including Depth Anything for geometry [20, 37, 38], Segment Anything for segmentation and localization [5, 17, 28], and counting specialists [1, 2, 21]. These systems are tailored to specific output spaces and remain strong in accuracy and efficiency, but they raise a complementary question: can perceptual competence also emerge from generative visual pretraining?

Adapting image generators into perception specialists. Recent work connects image generation and dense perception by adapting generators into specialist models. Marigold and Lotus-2 show that diffusion models can become strong dense predictors after task-specific training, architectural changes, or specialized inference [14, 16]. These results reveal useful perceptual priors, but study transformed generators rather than off-the-shelf image generators used directly.

Evaluation gaps in zero-shot generative perception. Other studies suggest that generative models may support visual understanding beyond synthesis, but they leave key gaps: work on video generators lacks comparisons with specialist baselines on public benchmarks [34], while Vision Banana reports instruction-tuned results only for Nano Banana Pro rather than its native zero-shot performance [10]. Thus, the native perceptual ability of off-the-shelf image generators remains unclear.

3. Zero-Shot Generative Perception

We formalize the use of pretrained image generators as zero-shot visual perceivers and describe the unified evaluation protocol behind PROBEGEN.

Disclaimer on "zero-shot". We use *zero-shot* in an op-

erational sense: no task-specific fine-tuning, adaptation, or architectural change, with a text prompt as the only interface to the generator. This does not guarantee that a generator never saw related supervision during its own pretraining or instruction tuning—for proprietary models, this is undisclosed and unverifiable—so our claims concern zero-shot transfer under a fixed, adaptation-free protocol, and we explicitly flag any model whose training data is known to overlap an evaluated task. This concern also motivates our OOD segmentation benchmarks (Figure 1(e)): even if a generator has seen manga images (Manga109 [36]) or paintings (DRAM [6]) during training, dense segmentation supervision in these stylized domains likely constitutes at most a marginal fraction of any plausible training mix—image generators are trained predominantly on image–text pairs for synthesis rather than on segmentation masks—so strong performance there is harder to attribute to data leakage and gives cleaner evidence of genuine zero-shot generalization.

Formulation. Let $x \in \mathbb{R}^{H\times W\times 3}$ denote an input RGB image and let p_t denote a task-describing text prompt. A pretrained image generator G produces a task-conditioned output $\hat{y} = G(x, p_t)$, where $\hat{y}$ is itself an RGB image whose pixel content encodes the desired perceptual quantity (*e.g.*, a depth map or a segmentation mask rendered as an image). A deterministic, task-specific extraction function E_t then projects $\hat{y}$ into the metric-compatible evaluation space:

$$\hat{z} = E_t\big(\hat{y}\big) = E_t\big(G(x,\, p_t)\big). \tag{1}$$

This decomposition isolates the generator's perceptual capacity, encoded entirely in $\hat{y}$, from the mechanical mapping into a benchmark-compatible format performed by E_t. Because E_t is deterministic and fixed identically across all models, the performance differences we report reflect differences in generative perception rather than per-model post-processing design. The remainder of this section describes the prompt protocol, the segmentation extractor, the depth extractor, and the counting extractor.

3.1. Prompt Protocol

Prompting is the sole task interface for all evaluated generators. Figure 2 summarizes the prompt families. For segmentation, prompts request a grayscale mask in which the target region is white and the background is black. For depth estimation, prompts request a grayscale depth map in which brighter pixels indicate points closer to the camera. For counting, prompts request adding red dots on each target object in the image.

To study how strongly each model depends on linguistic specificity, we design prompts at three granularity levels for segmentation (short, mid, long), two for depth (short, mid), and two for counting (short, mid). The short segmentation prompt provides only the target label; the mid prompt additionally specifies the white-foreground/black-background convention; and the long prompt further requests sharp boundaries without anti-aliasing artifacts. For depth, the short prompt requests a grayscale depth map without specifying polarity, while the mid prompt explicitly states the brighter-is-closer convention. For counting, the short prompt asks the model to put a red dot on each object, while the mid version further emphasizes identical dot color and consistency. By default we use the mid versions, and other versions are used only in ablations (Table 6).

Prompt for Segmentation Tasks

Short version:
```
Segment {label}.
```
Mid version (default):
```
Segment {label}.  Make the target object pure
white and everything else pure black.
```
Long version:
```
Segment {label}.  Make the target object
pure white and everything else pure black.
Ensure the boundaries are sharp and accurately
follow the contours of the {label} without
any grayscale gradients, textures, or
anti-aliasing artifacts.
```

Prompt for Monocular Depth Estimation

Short version:
```
Create a grayscale depth map of this image.
```
Mid version (default):
```
Create a grayscale depth map of this image:
brighter = closer, darker = farther.
```

Prompt for Counting

Short version:
```
Put one small solid red dot precisely on each
{label} in this image.
```
Mid version (default):
```
Put one small solid red dot precisely on
each {label} in this image.  Use the same
bright red color for every dot.  Do not change
anything else in the image.
```

Figure 2. **Prompt templates used in our evaluation protocol**.

3.2. Segmentation Output Extraction

For referring segmentation and reasoning segmentation, the generator is asked to produce a high-contrast mask corresponding to the queried target. Because raw outputs are continuous RGB images rather than discrete label maps, we apply a deterministic binarization step before computing segmentation metrics.

Extraction strategy. We benchmark four deterministic binarization strategies, each applied identically across all models, to ensure that our conclusions are not artifacts of a

particular extraction choice:

$$E_{\text{seg}}(\hat{y}) = \mathbf{1}\big[\,\text{gray}(\hat{y}) > \tau\,\big], \tag{2}$$

where $\text{gray}(\cdot)$ converts to single-channel luminance and τ is determined by one of the following thresholds: (i) a fixed intensity threshold at 128, (ii) a fixed threshold at 225, (iii) K-means clustering with k=2, and (iv) Otsu's adaptive thresholding [26], which selects the binarization boundary by maximizing inter-class variance. Based on the ablation study in Table 5, Otsu yields the most robust binarization and consistently suppresses the soft-boundary artifacts characteristic of raw generative outputs across six segmentation benchmarks. We therefore adopt Otsu thresholding as the default extractor in all main segmentation experiments.

3.3. Monocular Depth Extraction

For depth estimation, each generator is prompted to synthesize a grayscale depth map from a single RGB image. Unlike specialist depth estimators that directly predict metrically calibrated depth, general image generators typically output relative depth maps in which pixel intensity reflects proximity but absolute scale and offset are unconstrained.

Affine-invariant evaluation. We therefore evaluate geometric understanding under an affine-invariant protocol following standard practice [16, 27]. Specifically, we request outputs where brighter pixels denote regions closer to the camera, and then align the predicted map $\hat{d} = \text{gray}(\hat{y})$ to ground truth d via optimal scale-and-shift fitting:

$$(s^*, t^*) = \arg\min_{s,\,t} \|s\hat{d} + t\mathbf{1} - d\|_2^2, \tag{3}$$

where the scalar scale s and shift t admit a closed-form least-squares solution, and all metrics are computed on the aligned prediction $\tilde{d} = s^*\hat{d} + t^*$. This post-hoc alignment removes the arbitrary linear ambiguity inherent to generative outputs while preserving the structural and ordinal information encoded in the generated depth map.

3.4. Object Counting Extraction

Unlike depth and segmentation, counting only requires a single integer per image rather than a dense map. We probe this capability through a *mark-then-count* interface: as in Figure 2, the generator is prompted to paint one small solid red dot on every instance of the target category while leaving the rest of the image unchanged. The marked image $\hat{y}$ is then decoded into a count by a deterministic extractor E_{count} that recovers how many valid dots the model deposited.

Three-stage decoder. E_{count} operates at the preprocessed resolution (long side of 1024 pixels) so that the comparison against the original conditioning image is pixel-aligned: (i) *color isolation* keeps only saturated red pixels via two HSV ranges that straddle the hue wraparound at $0/180$; (ii) a *diff gate* retains only red pixels that changed relative to the input, suppressing red structures already present in the scene (*e.g.*, a stop sign or red packaging); and (iii) a *shape-filtered blob detector* returns a set of keypoints whose cardinality is the predicted count $\hat{c}$, where each candidate must satisfy fixed area, circularity, and inertia-ratio constraints. The shape filter separates true markers from elongated red streaks or oversized red regions that survive the color and diff gates. All hyperparameters are fixed once across all generators by a leakage-safe tuning protocol.

4. Experimental Results

4.1. Setup

Benchmarks. We evaluate PROBEGEN on depth, segmentation, and counting across standard and OOD domains. Depth benchmarks include NYUv2 [31], DIODE [33], ScanNet [7], and KITTI [11], evaluated with AbsRel. Segmentation benchmarks include the three RefCOCO variants [15, 39], ReasonSeg [19], Manga109 [36], and DRAM [6], evaluated with gIoU and cIoU. For counting, we use PixMo-Points [9] and report MAE and RMSE. Considering evaluation time and API cost, we sample 100 examples from each benchmark.

Baselines. We compare four families: *Discriminative Specialists* (Depth Anything V2 [38], SAM3-based pipelines [4, 5, 40], and CountGD++ [1]), *Generative Specialists* (Lotus-2 [14] and Marigold [16]), *Generative Generalists* (general-purpose image generators evaluated zero-shot), and *MLLMs* (Gemini-3.5-Flash [8], GPT-5.4 [24], and Claude-Opus-4.7 [3]), which are evaluated only on counting because they output scalar text answers rather than dense maps. Vision Banana is not publicly accessible [10], so we cite its reported numbers only for reference; they use a different protocol and are not directly comparable.

Inference. All generative models are evaluated with a single sample per image under deterministic decoding where possible. Inputs and outputs are standardized to a long side of 1024 pixels while preserving the aspect ratio. Unless otherwise noted, all results follow the default post-processing in Sec. 3: segmentation results with Otsu-based mask extraction, depth results with affine-invariant alignment, and counting results with the three-stage decoder.

Efficiency Evaluation. Because proprietary models are accessed through API calls, their GPU memory consumption cannot be measured and their throughput is not directly comparable to that of locally deployed models; we therefore report their efficiency numbers for reference only. All remaining models are run locally on NVIDIA H800 GPUs.

Notation. Across all result tables, † denotes API response time (not directly comparable to local throughput), $\times$ a failure to produce a valid output for the task, – an unavailable

metric, and ‡ Vision Banana numbers reported under a different protocol (full benchmark, with different prompts and post-processing).

4.2. Referring and Reasoning Segmentation

Generative models excel at reasoning and OOD segmentation. Table 1 shows that general image generators are highly competitive on reasoning-heavy and OOD segmentation. Nano Banana 2 reaches 73.7 gIoU on ReasonSeg, outperforming the SAM3 + Qwen3-VL-8B-Thinking pipeline (63.0 gIoU), while Nano Banana Pro and Nano Banana 2 substantially outperform discriminative specialists on DRAM and Manga109. Figure 1(e) further illustrates this robustness on artistic and manga-style inputs. Vision Banana is not directly comparable because it uses a different protocol, but its strong ReasonSeg result is consistent with the underlying strength of Nano Banana Pro.

Generators remain competitive in standard referring segmentation. On standard referring segmentation benchmarks (Table 2), specialist systems still achieve the best peak performance, but strong generators remain close: Nano Banana 2 obtains 73.1 gIoU on RefCOCO and 76.7 gIoU on RefCOCOg without task-specific fine-tuning. Figure 1 shows that the same generator handles both referring and reasoning segmentation, and even corrects an erroneous ground-truth apple annotation in Figure 1(d). The Vision Banana RefCOCOg result should again be read as non-comparable reference evidence that its gains build on a strong pretrained generator.

Efficiency cost of generative models. The main drawback is efficiency: image generators typically run at only 1–2 images per minute and require more GPU memory than specialist pipelines. Performance also varies widely across generators: Nano Banana models are consistently strong, whereas Seedream-4.0/4.5 often fail to produce usable masks and Flux.2-dev and Qwen-Edit-2511 are weaker on reasoning-heavy segmentation.

4.3. Monocular Depth Estimation

Specialists still lead in depth estimation. Table 3 reports affine-invariant relative depth results. Depth Anything V2-Large remains the strongest efficiency–accuracy reference, running at 828 images per minute, whereas Lotus-2 reaches similar accuracy with lower throughput. The non-comparable Vision Banana numbers are weaker than Qwen-Edit-2511, suggesting that depth benefits more from depth-specific training or supervision than from general perception-oriented tuning alone.

Depth exposure of Qwen-Edit. Qwen-Edit-2511 is the strongest open-weight general generator, achieving 6.1/9.0/7.2/11.0 AbsRel on NYUv2/DIODE/ScanNet/KITTI. However, its technical report states that instruction tuning includes depth-estimation examples [35], so these results are best read as a near-supervised reference rather than evidence of emergent geometry. Our protocol is still adaptation-free and prompt-only, but our zero-shot claim concerns evaluation under a fixed protocol, not guaranteed absence of task exposure in training data.

Comparison among generative models. Nano Banana Pro and Nano Banana 2 are less accurate but still produce coherent relative geometry without depth-specific adaptation, as illustrated in Figure 1. Other general generators degrade or fail outright, including Seedream-5.0 and Flux.2-dev. Overall, relative geometry can transfer to general image generators, but accurate and efficient depth prediction still favors specialist or task-adapted models.

4.4. Object Counting

MLLMs dominate counting. Table 4 shows a different ordering from depth and segmentation: MLLMs perform best because they directly output an integer. Gemini-3.5-Flash [8] achieves MAE 1.6, followed by GPT-5.4 [24] at 3.0 and Claude-Opus-4.7 [3] at 3.4.

Specialists and generators lag behind. CountGD++ [1] trails at MAE 9.8 despite high throughput, likely due to a mismatch with PixMo-Points' long-tail open-vocabulary categories. Among image generators, Nano Banana 2 is strongest (MAE 3.9 / RMSE 8.2), followed by Nano Banana Pro and GPT-Image-2, while Seedream models are inaccurate and Flux.2-dev/Qwen-Edit-2511 often produce invalid outputs.

Interface matters. Counting favors scalar text output: emitting an integer is cheaper and more accurate than rendering marked images. The mark-then-count probe is interpretable but suffers from generation-side errors such as red clutter and merged dots in dense scenes, which post-processing cannot fully recover.

4.5. Failure Modes

Alignment and format failures. Figure 3 shows two common model-level failures. Some generators fail to preserve the input, causing geometric drift or complete re-rendering that breaks pixel alignment. Others fail to follow the requested format, returning stylized or nearly unchanged images instead of masks, depth maps, or marked images. These errors explain why explicit format prompts help weaker models (Table 6).

Surface normals remain unsolved. As shown in Figure 4, none of the evaluated image generators reliably solve surface normal estimation in the zero-shot setting. Some outputs look normal-like, but they do not follow the required RGB-to-normal mapping, likely due to weak format adherence or limited normal supervision.

Model	ReasonSeg		DRAM		Manga109		Speed	GPU Mem
	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU	IMG/min	GB
Discriminative Specialists								
SAM3 + Qwen3-VL-8B-Thinking [5]	63.0	61.6	57.0	59.1	17.5	17.2	2.6	29
SAM3 + Qwen3-VL-32B-Thinking-FP8 [5]	**63.1**	**67.4**	56.0	56.3	**18.0**	**17.6**	2.1	73
Qwen3-VL-8B-SAMTok [40]	31.1	36.8	**73.9**	**73.5**	15.6	14.0	54	19
Generative Generalists								
Vision Banana‡ [10]	79.3	–	–	–	–	–	–	–
Nano Banana Pro [12]	71.2	67.3	**80.7**	**80.3**	**54.8**	55.6	1.2†	–
Nano Banana 2 [13]	**73.7**	**71.8**	79.6	74.0	53.3	**59.9**	2.0†	–
GPT-Image-2 [25]	44.7	50.1	62.6	67.9	30.1	29.6	1.5†	–
Seedream-4.0 [32]	×	×	×	×	×	×	×	×
Seedream-4.5 [29]	×	×	×	×	×	×	×	×
Seedream-5.0 [30]	28.1	29.3	49.1	48.9	16.8	15.6	1.6†	–
Flux.2-dev-NF4 (32B) [18]	20.3	17.2	49.8	44.2	9.2	9.2	1.2	39
Qwen-Edit-2511 (20B) [35]	25.2	25.8	49.4	38.7	13.0	9.8	1.5	61

Table 1. **Reasoning and OOD Segmentation.** We report the gIoU and cIoU metrics (higher is better ↑). GPT-Image 1 and 1.5 [22, 23] are not included because they fail to achieve pixel-level consistency (see Figure 3).

Model	RefCOCO		RefCOCO+		RefCOCOg	
	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU
Discriminative Specialists						
SAM3 + Qwen3-VL-8B-Thinking [5]	64.4	61.4	53.6	50.6	75.2	77.5
SAM3 + Qwen3-VL-32B-Thinking-FP8 [5]	72.0	69.6	64.5	61.8	73.6	72.7
Qwen3-VL-8B-SAMTok* [40]	**78.0**	**72.1**	**76.3**	**67.0**	**80.0**	**82.4**
Generative Generalists						
Vision Banana‡ [10]	–	–	–	–	–	73.8
Nano Banana Pro [12]	**73.1**	65.6	**56.6**	48.7	75.2	74.9
Nano Banana 2 [13]	**73.1**	**66.9**	56.1	**53.6**	**76.7**	**75.5**
GPT-Image-2 [25]	52.4	46.2	36.7	31.2	52.5	49.6
Seedream-4.0 [32]	×	×	×	×	×	×
Seedream-4.5 [29]	×	×	×	×	×	×
Seedream-5.0 [30]	32.1	35.3	31.6	27.1	40.1	36.3
Flux.2-dev-NF4 (32B) [18]	31.4	25.4	22.6	19.6	35.0	31.0
Qwen-Edit-2511 (20B) [35]	26.7	20.7	19.3	14.8	36.3	29.2

Table 2. **Referring Segmentation**. We report the gIoU and cIoU metrics (higher is better ↑). *: Qwen3-VL-8B-SAMTok has been trained on these benchmarks.

4.6. Ablation Studies

Choice of mask extraction method. Table 5 validates the segmentation extraction procedure on Nano Banana 2. Otsu thresholding achieves the best average performance across the six segmentation benchmarks, yielding 68.8 average gIoU and outperforming fixed thresholds and K-means clustering. This supports our choice of Otsu thresholding as the default binarization strategy in the main experiments.

Sensitivity to prompt granularity. Table 6 shows that a model's sensitivity to prompt wording is inversely related to its capability. The Nano Banana series is essentially flat across all prompt variants, so its strength is not an artifact of careful prompt engineering; the most verbose prompt even hurts slightly, indicating that the mid prompt is already near-optimal. Weaker generators are far more prompt-dependent. On segmentation, their gains are concentrated almost entirely at the short→mid step: GPT-Image-2 jumps from 24.6 to 52.5 gIoU and Seedream-5.0 from 8.0 to 40.1 once the mid prompt states the white-foreground/black-background convention. The following mid→long step, which only asks for sharper boundaries, adds little and can even hurt. For these models, the real bottleneck is thus specifying the output format, not refining the boundary language. Depth and counting use only short and mid prompts, and there the same short→mid improvement appears but is much smaller. Overall, the mid prompt is a fair, near-optimal default for every model, so our main conclusions do not depend on per-model prompt tuning.

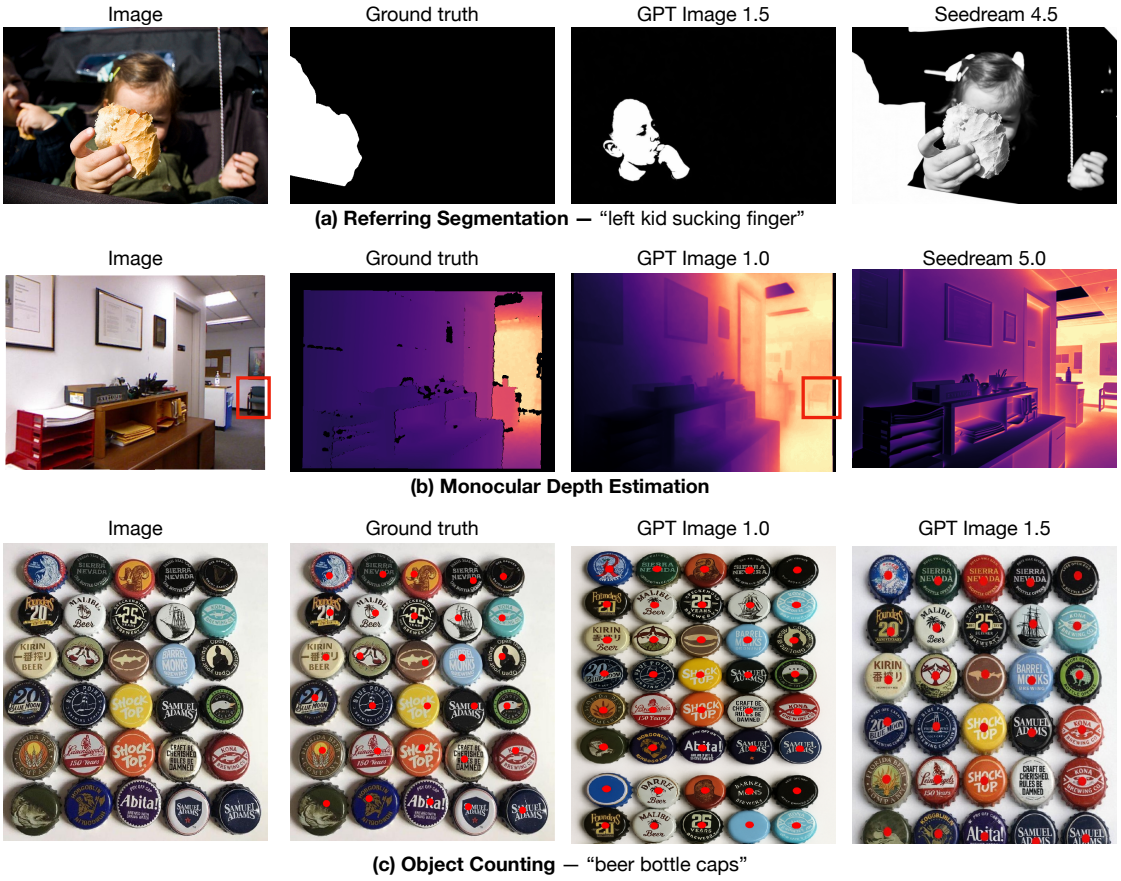

Figure 3. **Systematic failures of GPT-Image-1/1.5 and the Seedream series.** GPT-Image-1/1.5 return outputs in the correct task format but fail to preserve the input, regenerating the scene instead of transforming it in place—from geometric drift that breaks pixel-wise correspondence (b) to wholesale re-rendering that hallucinates new content, such as the redrawn bottle-cap designs (c). The Seedream series, on the other hand, fails to follow the instruction, returning a stylized or near-unchanged copy of the input rather than performing the requested task (a,b).

Model	NYUv2	DIODE	ScanNet	KITTI	Speed	Mem
Discriminative Specialists						
Depth Anything V2-Large [38]	**4.7**	11.0	**4.7**	9.1	828	3.5
Generative Specialists						
Marigold v1.1 [16]	6.6	11.7	7.0	12.0	226	10
Lotus-2 [14]	**4.7**	**7.0**	5.2	**8.9**	25	34
Generative Generalists						
Vision Banana‡ [10]	8.1	10.8	–	10.7	–	–
Nano Banana Pro [12]	11.0	35.8	11.8	17.0	1.2†	–
Nano Banana 2 [13]	11.1	31.4	11.7	17.4	1.9†	–
GPT-Image-2 [25]	9.6	32.8	11.2	18.1	1.5†	–
Seedream-4.0 [32]	12.5	14.8	12.3	24.2	4.1†	–
Seedream-4.5 [29]	14.1	18.3	12.1	22.5	3.3†	–
Seedream-5.0 [30]	×	×	×	×	×	×
Flux.2-dev-NF4 (32B) [18]	×	×	×	×	×	×
Qwen-Edit-2511 (20B) [35]	**6.1**	**9.0**	**7.2**	**11.0**	1.5	62

Table 3. **Monocular Depth Estimation**. We report the AbsRel metric (lower is better ↓).

Model	MAE ↓	RMSE ↓	Speed	Mem
Discriminative Specialists				
CountGD++ [1]	9.8	23.4	471	4.1
MLLMs				
Gemini-3.5-Flash [8]	1.6	4.3	9.2†	–
GPT-5.4 [24]	3.0	5.8	22.6†	–
Claude-Opus-4.7 [3]	3.4	6.6	13.0†	–
Generative Generalists				
Nano Banana Pro [12]	4.7	9.2	2.8†	–
Nano Banana 2 [13]	**3.9**	**8.2**	2.4†	–
GPT-Image-2 [25]	5.5	12.4	1.1†	–
Seedream-4.0 [32]	13.1	17.7	3.5†	–
Seedream-4.5 [29]	13.7	18.5	3.1†	–
Seedream-5.0 [30]	14.2	18.8	2.4†	–
Flux.2-dev-NF4 (32B) [18]	×	×	×	×
Qwen-Edit-2511 (20B) [35]	×	×	×	×

Table 4. **Object Counting.** We report MAE and RMSE (lower is better ↓).

Method	RefCOCO	RefCOCO+	RefCOCOg	ReasonSeg	DRAM	Manga109	Average
Threshold (128)	72.7	55.7	**76.7**	72.8	77.5	**53.5**	68.2
Threshold (225)	72.5	55.5	76.4	72.8	77.5	**53.5**	68.0
Otsu	**73.1**	**56.1**	**76.7**	**73.7**	**79.6**	53.3	**68.8**
K-means	**73.1**	52.8	**76.7**	69.2	78.3	53.4	67.3

Table 5. **Ablation Study on Mask Extraction Methods for Segmentation Tasks**. We conduct an ablation study on Nano Banana 2 across six segmentation benchmarks and report the gIoU metric. The default setting is marked in gray.

Model	Seg. (RefCOCOg)			Depth (NYUv2) ↓		Counting (PixMo) ↓	
	short	mid	long	short	mid	short	mid
Nano Banana Pro	74.8	**75.2**	74.8	11.5	**11.0**	5.1	**4.7**
Nano Banana 2	**77.0**	76.7	75.7	11.4	**11.1**	4.0	**3.9**
GPT-Image 2	24.6	52.5	**57.9**	10.9	**9.6**	5.9	**5.5**
Seedream-5.0	8.0	**40.1**	37.5	×	×	×	×
Qwen-Edit-2511	11.6	**36.3**	34.2	6.5	**6.1**	×	×

Table 6. **Ablation Study on Prompt Granularity**, evaluated on RefCOCOg (gIoU), NYUv2 (AbsRel), and PixMo (MAE). Gray shading indicates the default configuration.

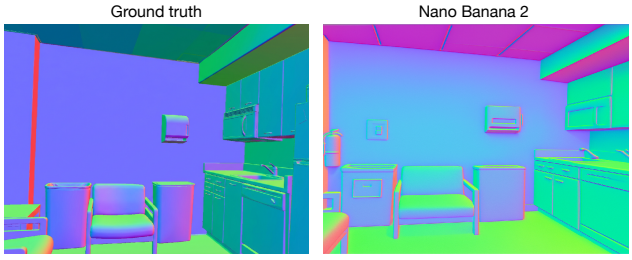

Figure 4. **Failure on Surface Normal Estimation.** Image generators produce normal-like outputs that may look plausible at a glance, but they do not preserve the correct scene geometry or faithfully follow the prescribed RGB encoding of surface normals.

4.7. Discussion

Pixel alignment is the bottleneck. Our results suggest that using image generators for perception is still difficult, because the task demands much stricter output fidelity than ordinary image editing. The main bottleneck is *pixel-wise alignment*: generative visual perception requires outputs to remain geometrically registered to the input, whereas current generators often preserve semantics only coarsely. This is why our post-processing is mainly effective for relative depth, but much less reliable for masks, surface normals, or counting. Progress will likely require stronger perception supervision, explicit spatial-consistency objectives, and reward signals that penalize structural misalignment.

Instruction following matters. Many failures arise not because the model completely misses the scene, but because it produces the wrong type of output, such as a stylized edit instead of a strict mask or depth map.

Efficiency remains a major issue. Even strong generators remain far slower and more memory-intensive than specialist discriminative models. More broadly, the differences across models suggest that zero-shot visual perception depends on three ingredients: semantic grounding, spatial faithfulness, and output-format discipline, where weaknesses in any one of them can cause failure.

Generative perception remains promising. Despite these limitations, general image generation models remain attractive for perception because a single interface can already support segmentation, depth estimation, and other structured tasks without task-specific heads. This flexibility is especially promising for open-world concepts, long-tail queries, and natural-language interaction. Although current models are still inefficient and imperfect, their zero-shot transfer and cross-domain robustness suggest substantial long-term potential if alignment, faithfulness, and efficiency can be improved together.

5. Conclusion

In this paper, we systematically evaluated general-purpose generative models as zero-shot visual perceivers. By repurposing image synthesis for dense perception via text prompting, we showed that these models already encode non-trivial geometric and semantic understanding without task-specific fine-tuning. Our benchmarking reveals a clear trade-off: specialized models remain stronger in metric calibration and efficiency, whereas zero-shot generative prompting is more competitive in OOD scenarios and reasoning-heavy tasks. Despite persistent challenges in alignment, output fidelity, and speed, these results point to a promising path toward unified architectures that bridge synthesis and perception.

References

[1] Niki Amini-Naieni and Andrew Zisserman. Countgd++: Generalized prompting for open-world counting. In *CVPR*, 2026. 1, 2, 4, 5, 7

[2] Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. Countgd: Multi-modal open-world counting. In *NeurIPS*, 2024. 1, 2

[3] Anthropic. Claude opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, 2026. 4, 5, 7

[4] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. 4

[5] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. 1, 2, 4, 6

[6] Nadav Cohen, Yael Newman, and Ariel Shamir. Semantic Segmentation in Art Paintings. *Computer Graphics Forum*, 2022. 1, 3, 4

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 4

[8] Deepmind. Gemini 3.5 flash. https://deepmind.google/models/gemini/flash/, 2026. 4, 5, 7

[9] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *CVPR*, 2025. 4

[10] Valentin Gabeur, Shangbang Long, Songyou Peng, Paul Voigtlaender, Shuyang Sun, Yanan Bao, Karen Truong, Zhicheng Wang, Wenlei Zhou, Jonathan T Barron, et al. Image generators are generalist vision learners. *arXiv preprint arXiv:2604.20329*, 2026. 1, 2, 4, 6, 7

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 4

[12] Google DeepMind. Nano banana pro. https://deepmind.google/models/gemini-image/pro/, 2025. 1, 6, 7

[13] Google DeepMind. Nano banana 2. https://deepmind.google/models/gemini-image/flash/, 2026. 1, 6, 7

[14] Jing He, Haodong Li, Mingzhi Sheng, and Ying-Cong Chen. Lotus-2: Advancing geometric dense prediction with powerful image generative model. *arXiv preprint arXiv:2512.01030*, 2025. 1, 2, 4, 7

[15] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 4

[16] Bingxin Ke, Kevin Qu, Tianfu Wang, Nando Metzger, Shengyu Huang, Bo Li, Anton Obukhov, and Konrad Schindler. Marigold: Affordable adaptation of diffusion-based image generators for image analysis. *IEEE TPAMI*, 2025. 1, 2, 4, 7

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 1, 2

[18] Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025. 1, 6, 7

[19] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, 2024. 4

[20] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *ICLR*, 2026. 1, 2

[21] Chang Liu, Haoning Wu, and Weidi Xie. Count anything at any granularity. *arXiv preprint arXiv:2605.10887*, 2026. 2

[22] OpenAI. Gpt-image-1. *https://openai.com/index/image-generation-api/*, 2025. 1, 6

[23] OpenAI. Gpt-image-1.5. https://openai.com/index/new-chatgpt-images-is-here/, 2025. 1, 6

[24] OpenAI. Gpt 5.4. https://openai.com/index/introducing-gpt-5-4/, 2026. 4, 5, 7

[25] OpenAI. Gpt-image-2. https://openai.com/index/introducing-chatgpt-images-2-0/, 2026. 1, 6, 7

[26] Nobuyuki Otsu et al. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296), 1979. 4

[27] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE TPAMI*, 2020. 4

[28] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *ICLR*, 2025. 1, 2

[29] Seedream Team. Seedream 4.5. https://seed.bytedance.com/en/seedream4_5, 2025. 1, 6, 7

[30] Seedream Team. Seedream 5.0 lite. https://seed.bytedance.com/en/seedream5_0_lite, 2026. 1, 6, 7

[31] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 4

[32] Seedream Team. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025. 1, 6, 7

[33] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 4

[34] Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank

Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025. 1, 2

[35] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 1, 5, 6, 7

[36] Minshan Xie, Jian Lin, Hanyuan Liu, Chengze Li, and Tien-Tsin Wong. Advancing manga analysis: Comprehensive segmentation annotations for the manga109 dataset. In *CVPR*, 2025. 1, 3, 4

[37] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 1, 2

[38] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024. 1, 2, 4, 7

[39] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016. 4

[40] Yikang Zhou, Tao Zhang, Dengxian Gong, Yuanzheng Wu, Ye Tian, Haochen Wang, Haobo Yuan, Jiacong Wang, Lu Qi, Hao Fei, et al. Samtok: Representing any mask with two words. *arXiv preprint arXiv:2601.16093*, 2026. 4, 6