COUNTLOOP: Training-Free High-Instance Image Generation via Iterative Agent Guidance

Anindya Mondal[1], Ayan Banerjee[2], Sauradip Nag[3], Josep Lladós[2], Xiatian Zhu[1], Anjan Dutta[1]
[1]University of Surrey, [2]Universitat Autònoma de Barcelona, [3]Simon Fraser University
[1]{a.mondal, anjan.dutta, xiatian.zhu}@surrey.ac.uk, [2]{abanerjee, josep}@cvc.uab.es, [3] snag@sfu.ca

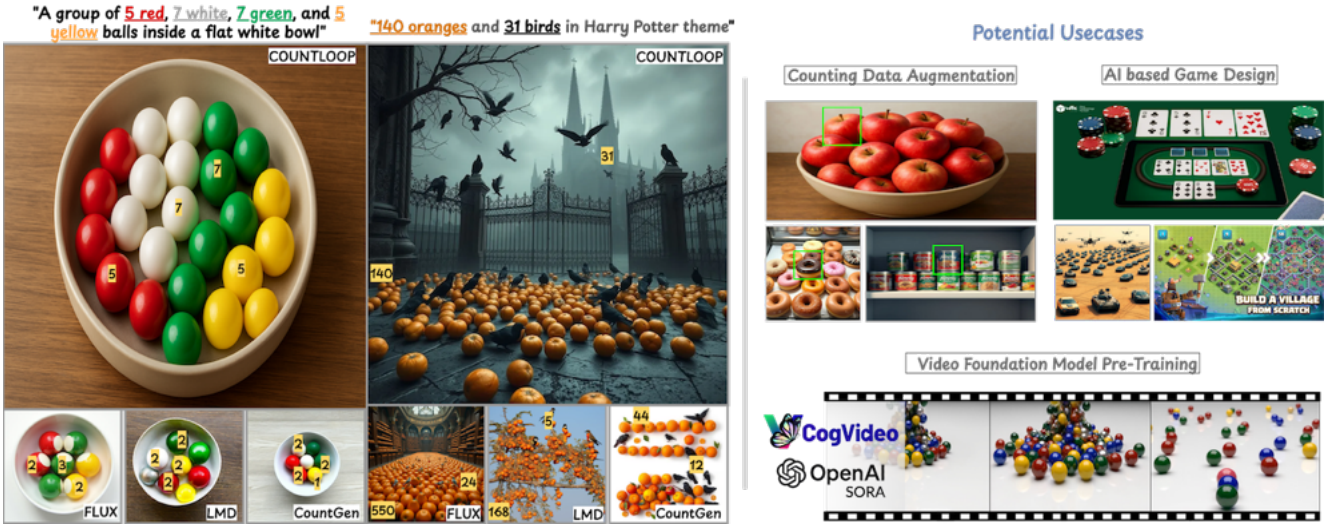

Figure 1. Given prompts with explicit per-class counts, COUNTLOOP (top-left) produces high-instance images whose *detected* counts align with targets. Under identical prompts, recent text/layout-based image generation benchmarks often under- or over-generate at high cardinalities. We further illustrate practical uses of count-specific image generation (right): (a) in object counting [41], for augmenting datasets; (b) in AI-driven games [34], where accurate object counts (*e.g.*, buildings, cards) are crucial for gameplay design; and (c) in video foundation model pre-training [20, 50], where synthetic count images can enhance diversity and generalization compared to scarce real-world counting datasets.

Abstract

Diffusion models excel at photorealistic synthesis but struggle with precise object counts, especially in high-density settings. We introduce COUNTLOOP*, a training-free framework that achieves precise instance control using iterative, structured feedback. Our method alternates between synthesis and evaluation, using a VLM-guided agent as both a layout planner and a critic. This agent provides explicit feedback on object counts, spatial arrangements, and attributes to refine the scene layout iteratively. Instance-driven attention masking and cumulative attention composition further prevent semantic leakage, ensuring clear object separation even in occluded scenes. Evaluations on high-instance benchmarks show* COUNTLOOP *achieves up to 2x higher counting accuracy and significantly improves spatial alignment over strong layout-based, gradient-guided, and agentic approaches, while maintaining photorealism.*

1. Introduction

Digital creators, designers, and artists increasingly use text-to-image diffusion models like DALL-E 3 [6], SDXL [39], and FLUX [8] to produce high-quality visuals. However, these models struggle with scenes containing many distinct yet related object instances [37], limiting their effectiveness in applications where cardinality is crucial, such as game asset generation (*e.g.*, crowds of characters or repeated environmental elements) or augmenting object-counting datasets and even as a pretraining task in video diffusion models [50]. Current image diffusion models typically saturate at around 10 instances per category [7], with precise quantity being a known long-tail compositional failure [61], yielding semantic drift (mixed attributes), spatial collapse (cluttered or overlapping objects), or instance duplication. For instance, a prompt like "140 oranges and 31 birds in Harry Potter theme" might under/over-produce an incoherent pile of either oranges or birds or both (Fig. 1), compromising accuracy and

usability.

Current solutions fall into main three categories: (1) gradient-guided methods [10, 25]; (2) layout-to-image (L2I) pipelines [4, 7, 17, 28, 32, 64]; and (3) agentic diffusion frameworks [51, 55, 59]. However, none scale effectively to high-instance scenes or fully resolve the failure cases illustrated in Fig. 2. Gradient-guided methods inject counting signals during denoising to improve count accuracy but often introduce artifacts or worsen semantic leakage, which is an intrinsic challenge of high-instance generation, especially as object density increases [14, 15] (see Fig. 2(b)). L2I pipelines guide diffusion using bounding boxes or masks, but suffer from autoregressive biases [5, 56] that cause unnatural, grid-like layouts (see Fig. 2(a)), and typically require detailed annotations or carefully engineered prompts [7], limiting scalability. Agentic diffusion methods use LLM-based editing but lack explicit scene structure, leading to poor spatial grounding, overcorrection, or object omission. Their focus on aesthetics over structure makes them unreliable for dense, count-sensitive generation.

To tackle the ongoing challenge of generating visually coherent scenes with accurate object counts, we present CountLoop, a training-free framework that approaches high-instance image generation as an iterative design process rather than a single-pass operation. Inspired by how human designers progressively refine their compositions, CountLoop follows a structured loop: it parses the input prompt into a planning graph that captures both object attributes and spatial relationships. This graph is then used to generate a layout, which guides image synthesis under layout constraints. A vision-language model (VLM) critic offers structured feedback by evaluating two key aspects: (a) spatial coherence and appearance fidelity, which are assessed using a pretrained image encoder [52]. For (b) counting accuracy, the Critic VLM employs an off-the-shelf object detector [32]. This design choice is essential because recent studies show that VLMs alone struggle with accurate counting in dense scenes [18]. The structured feedback from the VLM is then used to update the planning graph and the prompt, repeating the loop until the output meets target quality thresholds. Unlike generative models, which may hallucinate or drift from the intended specification, VLMs excel as discriminative evaluators [24], making them ideal critics in our agentic loop. Their multi-modal understanding enables reliable scoring of both semantic [26, 58] and spatial alignment [11, 58], guiding precise and targeted corrections.

Our CountLoop also introduces a cumulative attention mechanism during the denoising process to mitigate semantic leakage, which is a common issue in high-instance scenes. Rather than generating all subjects simultaneously, it provides per-instance grounding by preventing semantic entanglement and maintaining the identity of individual objects. By imposing attention locality within instance-specific

Figure 2. Issues in High-instance image generation

regions, CountLoop encourages independence across objects and prevents the borrowing of features from nearby or similar instances. Together, this iterative agent-guided loop, the use of per-instance cumulative attention composition, and VLM-based visual feedback form a powerful, training-free pipeline. Unlike prior methods requiring model retraining or suffer from grid-like rigidity, CountLoop acts as a plug-and-play enhancement to standard diffusion backbones, scaling up to 100+ objects while ensuring accurate counts and natural spatial layouts.

We summarize our contributions as follows: **(1)** We present CountLoop, a training-free iterative pipeline for generating high-instance images with precise object counts and strong aesthetic quality; **(2)** We introduce a cumulative attention composition mechanism that sequentially injects each object in the latent space using instance-specific attention masks. This effectively mitigates semantic leakage, ensuring clear boundaries and identity preservation even in densely populated scenes; **(3)** We leverage a VLM as a structured critic to evaluate generated images along two axes: count consistency and appearance fidelity, and provide interpretable feedback to refine the layout and prompt iteratively; **(4)** We conduct extensive evaluations on COCO-Count, T2I-CompBench, and newly introduced high-instance benchmarks. Results show that CountLoop more than doubles the counting accuracy and significantly improves visual coherence compared to all existing methods.

2. Related Work

Count Control in T2I Generation: Modern text-to-image diffusion models such as LDM [42], Imagen [43], SDXL [39], and FLUX [8] achieve remarkable photorealism by denoising a shared latent representation, but they break down when prompts demand structured control, such as "40 red cans on a shelf" or "12 apples in a bowl and 8 on the table". Beyond 10-15 identical objects, they often miscount, exhibit attribute leakage, and suffer spatial collapse [7, 10, 14]. These limitations stem from architectural constraints: cross-attention fails to preserve per-instance identity, and there is no global mechanism enforcing cardinality or spatial coherence. Gradient-guided corrections [10, 25] offer partial remedies but require retraining and still fail in dense scenes. In contrast, CountLoop is a training-free iterative framework that plans, generates, and

critiques images through a vision-grounded loop. By integrating instance-aware composition and a cumulative attention mechanism to prevent attribute leakage, COUNTLOOP achieves high-fidelity, count-accurate generation even at extreme object densities.

L2I Generation: GLIGEN [28] and LMD [29] condition diffusion on boxes/masks (or LLM-derived layouts) to control count and placement, but they do not model rich relations and have not been shown to scale to very dense (20+ instance) scenes. Scene-graph pipelines such as SG2IM [23] encode pairwise relations but depend on expensive graph annotations. LLM layout planners (*e.g.*, LayoutGPT [17]) can draft plausible layouts, yet robustness under high-instance prompts is under-explored. CountGen [7] improves count control by retrieving and adapting layouts from similar images, but its effectiveness depends on retrieval coverage and the downstream generator, with limited evidence at extreme densities or broad attribute variation. Independent studies report cross-attention leakage and identity confusion in multi-object T2I [10, 14], especially as objects crowd together. In contrast, we use a VLM-driven planning graph with iterative, instance-aware composition (cumulative attention) to preserve texture and prevent leakage under occlusion, achieving precise counts without retraining and demonstrating reliable generation at 100+ instances.

Agentic Diffusion Correction: Recent frameworks use LLM agents as planners or critics to improve diffusion generation iteratively. For example, SLD [55] employs an LLM to detect generation errors and suggest prompt revisions. However, it treats the image as a black box, lacks layout control, and often over-corrects, repeating or skipping objects. GenArtist [51] deploys multiple agents to edit color, style, and composition, but focuses on aesthetics rather than object count or spatial precision. RPG-DiffusionMaster [59] uses role-playing agents to draft and review prompts, improving narrative clarity and ignoring issues like overlap or counting in dense, occluded scenes. While all three frameworks improve prompts, they lack an explicit scene representation, making them unreliable in high-instance settings. In contrast, COUNTLOOP introduces a targeted refinement loop designed for dense instance generation. It builds a structured planning graph by encoding objects and relations, uses a VLM guided by an open-vocabulary detector for grounded critique and an aesthetic scorer for quality estimation, and applies a parameter-free textual optimizer to update layouts, achieving accurate, layout-aware, and visually consistent results without retraining the diffusion backbone.

3. COUNTLOOP

Overview: We introduce COUNTLOOP, a training-free, VLM-guided framework for high-instance image generation, producing precise object counts, coherent spatial arrangements, and distinct instance-level attributes from a textual prompt (see Fig. 1). COUNTLOOP operates in three stages. First, a Design VLM interprets the prompt to produce realistic, non-grid layouts (Fig. 2(a)) with natural object placement. Second, these layouts guide style-consistent image generation via a cumulative attention mechanism that mitigates attribute leakage (Fig. 2(b)) and preserves object clarity under overlap. Finally, a Critic VLM assesses the output for counting accuracy and aesthetic quality, providing structured feedback to refine both the layout and prompt. This iterative loop runs until a target quality score is reached, enabling complex, high-instance images without retraining the diffusion model. Fig. 3 shows the full pipeline.

3.1. VLM-Guided Layout Generation

Generating images with precise control over multiple object instances, especially in dense scenes, remains challenging for text-to-image models, often causing unrealistic layouts and object overlaps. While layouts can be extracted from prompts via an LLM and further grounded for accurate counting [29], limited spatial reasoning [40] and autoregressive generation lead LLMs to produce rigid, grid-like structures (see Fig. 2(a)). VLMs offer improved multimodal reasoning [53], but still fall short of the desired flexibility. To overcome this, we introduce spatial reasoning into the VLM to promote more flexible layout arrangements. Inspired by scene graphs [12], we propose planning graphs that augment VLM's Chain-of-Thought with explicit relational and spatial priors. Building on Qwen3-VL [58], our Design VLM produces more consistent object placement, attributes, and relations, reducing grid artifacts and yielding more structured, realistic compositions.

Prompt Parsing: As a precursor to our process, we break down the input prompt into its core components, including object-level quantities, instance-level attributes, and instance-level quantities. For example, the prompt "two cats and a bird in the sky" contains two objects, "cat" and "bird", with desired quantities of two and one, respectively. The object "bird" is associated with an instance-level attribute "in the sky", which has a desired quantity of one, whereas the object "cat" is not associated with any instance-level attributes. We begin by instructing a VLM (Qwen3-VL [58]) to analyze the prompt and the attribute relations and return it in a JSON dictionary. We guide the VLM with specific instructions on how to extract spatial relations from P as shown below.

Prompt Parsing Instruction

You are a scene planner. Given a prompt, return a JSON-based object-attribute relation with:

- `objects`: list of instance nodes with fields—`id`, `category`, `position (x,y)`, `depth`, `color`
- `relations`: list of edges with fields—`source`, `target`, `relation`, `distance`, `angle`
- `context`: background scene type

These object-attribute relations serve as the foundation for the planning graph that injects spatial reasoning into the

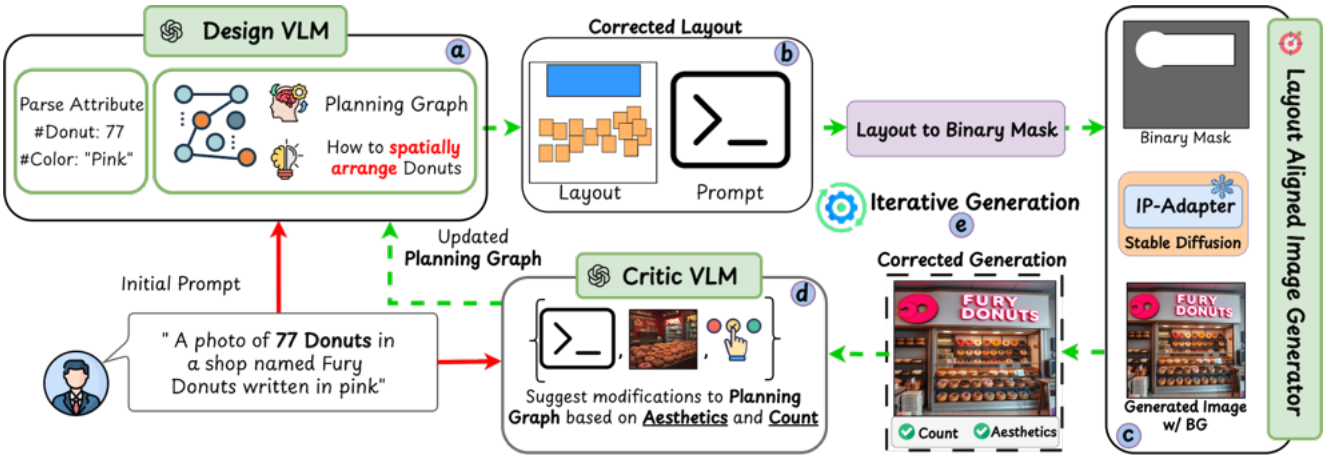

Figure 3. Given a text prompt, ⓐ The Design VLM parses the prompt to construct a planning graph, which is converted into a pixel-aligned layout ⓑ. ⓒ This layout guides an IP-Adapter-enhanced T2I backbone for image generation. ⓓ A Critic VLM evaluates the generated image's count and aesthetics, providing structured feedback to update the planning graph. ⓔ This iterative loop continues until objectives are met.

VLM's chain-of-thought reasoning.

Planning Graph Construction

Prompt: `"A scene with 2 cats and 1 bird in the sky"`
VLM Reasoning (Simplified):
1. Identify objects: 2 cats, 1 bird
2. Assign coarse positions: cats near center, bird above
3. Apply spatial jitter and avoid overlaps

Example: (Prompt: `"A scene with 2 cats and 1 bird in the sky"`) `"objects": ["id": "cat 1", "pos": [0.3, 0.6], "d": 0.4, "color": "gray", "id": "cat 2", "pos": [0.6, 0.6], "d": 0.4, "color": "black", "id": "bird 1", "pos": [0.5, 0.3], "d": 0.2, "color": "white"], "relations": ["from": "cat 1", "to": "bird 1", "r": "below", "dist": 120, "angle": 90], "context": "outdoor, grassy field"`

Planning Graph Construction: The graph construction process begins by using object-attribute relations parsed from the input prompt. Specifically, the planning graph is defined as $G = (V, E, B_{\text{bg}})$, where V denotes object-instance nodes, E represents edges encoding spatial relations, and B_{bg} captures the scene context (*e.g.*, "outdoor environment"). Each node in V includes attributes like category (*e.g.*, cat, bird), a unique identifier (*e.g.*, `cat_1`), normalized position $[x, y] \in [0, 1]^2$, depth prior $d \in [0, 1]$, and color. Edges in E encode spatial relations via directional operators (*e.g.*, "above," "left-of"), normalized distances, and angular orientations. G enforces structured spatial reasoning, nodes specify individual properties while edges ensure relational consistency (*e.g.*, minimum distances to prevent overlaps), enabling realistic multi-object scene construction. To integrate this structured representation into VLM reasoning, we convert the graph into a textual prompt template P_G:

$$P_G = \phi([\text{'Object'}]), [\text{'Relation'}], [\text{'Context'}]) \quad (1)$$

where ϕ denotes a text concatenation operator; `'Object'` $\in V$, `'Relation'` $\in E$, and `'Context'` $\in B_{bg}$ denotes the textual attributes from the planning graph. Full prompt details are provided in the supplementary. The prompt P_G encodes object positions, depth, and sizes in text, enabling spatial reasoning within the VLM. This reasoning is combined with in-context examples for effective grounding: These examples provide a structured format that ensures precise object placement while preserving natural composition. Finally, both the planning graph prompt P_G and the in-context examples (denoted by P_{icl}) are fed into the Design VLM as follows:

$$\mathbb{J} = \text{VLM}(P_G, P_{\text{icl}}) \quad (2)$$

where $\mathbb{J}$ is the VLM's output in JSON format. From this, we extract the object layout coordinates $\mathbb{L}$, the scene description prompt P_d and background prompt P_{bg} respectively. The prompt template is detailed in the supplementary.

3.2. Layout Aligned Image Generation

After obtaining the layouts $\mathbb{L}$, the goal is to generate images that faithfully follow the specified arrangement. However, layout-grounded diffusion models commonly exhibit attribute leakage [14, 15], yielding correct counts but degraded visual quality (Fig. 2(b)). To address this, we take inspiration from multi-turn image generation [13] and avoid generating all instances in a single pass. Instead, we adopt an iterative strategy that synthesizes one object at a time while preserving texture by conditioning on the previously generated content. This sequential process reduces attention leakage and maintains clear separation between objects, even under occlusion.

Layout Aligned Attention Masking: Given the object layouts $\mathbb{L}$ and prompt description P_d, we aim to ground the layout with the text to generate images with accurate instance counts. Since layouts are discrete spatial arrangements, we project them into a continuous space using a layout encoder.

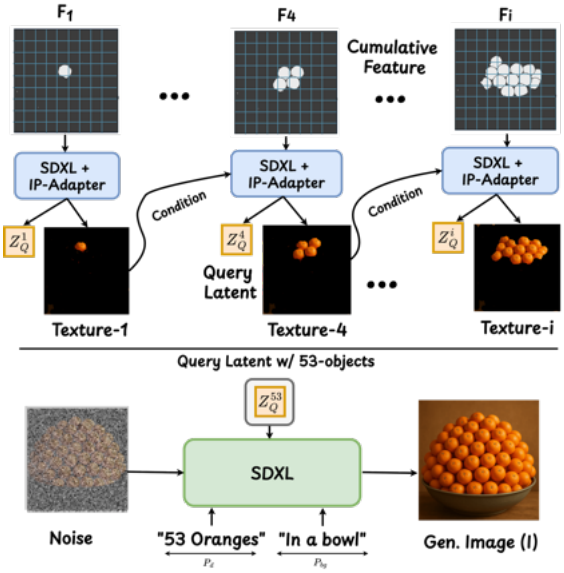

Figure 4. Cumulative latent composition, along with disentangled query feature extraction, mitigates attribute leakage

Specifically, we use the layout encoder of GLIGEN [28], denoted by $\mathbb{E}$, which encodes each per-instance layout $l_i \in \mathbb{L}$ into latent embeddings $Q_i = \mathbb{E}(l_i)$. The full set of embeddings is represented as $Q = \{Q_1, \ldots, Q_N\}$. To ground these layout embeddings with the prompt P_d, we compute cross-attention A_{cross}, where the queries are layout embeddings Q, and the keys and values are derived from the text embedding of P_d. However, directly using A_{cross} for generation introduces semantic leakage because it attempts to generate all instances at once. To mitigate this, we independently process A_{cross} at the instance level. For each object instance i, we apply a binary spatial mask $M_i \in \{0,1\}^{w_i \times h_i}$ (1 inside the bounding box of l_i, 0 elsewhere), derived from the layout $l_i \in \mathbb{L}$. The mask is then reshaped into $\hat{M}_i$ using bilinear interpolation to match the latent dimension of A_{cross}. This mask is further refined via a self-segmentation algorithm [14] to obtain shape-aware masks. The masked layout feature is then computed as:

$$A^i_{\text{mask}} = A^i_{\text{cross}} \odot \hat{M}_i \tag{3}$$

Here, A^i_{mask} denotes the instance-specific masked attention feature, which confines the receptive field of attention to the corresponding object's region in the spatial domain.

Cumulative Latent Composition: Once instance-level attention maps A^i_{mask} are computed for each object layout $l_i \in \mathbb{L}$, we construct a coherent global latent feature map $\mathbb{F}$ via cumulative composition in the diffusion latent space. Starting from a zero-initialized canvas, we iteratively paste each A^i_{mask} at its designated spatial location, producing intermediate latent maps $\mathbb{F}_i \in \mathbb{R}^{H_F \times W_F \times D}$, where H_F and W_F are spatial dimensions and D is the feature dimension. The composition is defined as:

$$F_{i+1}(x,y) = \mathbb{1}_{(x,y)\in l_i} \cdot \text{Blend}(F_i(x,y), A^i_{\text{mask}}) \tag{4}$$

Here, $\mathbb{1}$ indicates whether pixel (x,y) lies within the bounding box of l_i, and $\text{Blend}(\cdot)$ denotes feature concatenation. This iterative process yields a sequence of cumulative latent feature maps $F = \{F_1, F_2, \cdots, F_N\}$, where each F_i contains an increasing set of composed instances (see Fig. 4). When these disentangled instance-wise latent features are used for image generation independently, the cross-attention mechanism from Eq. 3 ensures per-instance grounding. This prevents semantic entanglement and maintains the identity of individual objects.

Appearance Consistency via IP-Adapter: Generating images independently from disentangled features F reduces semantic leakage but often introduces texture inconsistency, since each latent F_i is denoised separately. To counter this, we condition the diffusion model (*e.g.*, SDXL [39]) on the foreground texture of the previously generated output using IP-Adapter [60]. Because leakage occurs when query tokens attend to different instances during self-attention [14], we further preserve the per-instance query representation (Z_q) before its interaction with keys and values, maintaining instance-level semantics. Formally:

$$I_{i+1}, Z^{i+1}q = \Phi(Fi+1, P_d, \theta(I_i)), \quad i = 1, \ldots, N-1 \tag{5}$$

where I_i is the image generated from F_i, N is the number of objects, and θ is IP-Adapter conditioning. The first image is generated without IP-Adapter due to the absence of prior texture. Iterating over all F_i aligns prompt semantics P_d with accumulated visual cues, reducing hallucinations and preserving object distinctiveness. After extracting all query embeddings $Z_q = \{Z^1_q, \ldots, Z^N_q\}$, we produce a final image with minimal attribute leakage. To generate the final composition, we use the last query latent Z^N_q, which encodes all N objects with consistent appearance. The attention operation is defined as:

$$\mathbb{A}(Z^N_q, K, V), \tag{6}$$

where K and V are the keys and values (see Fig. 4) of the diffusion. Each object-specific feature in Z^N_q attends to a shared key–value set, enforcing semantic coherence across foreground instances while keeping the background disentangled. This operates as an implicit variant of self-attention expansion in video diffusion [2, 54], but the attention is shared across object instances rather than frames. Since using only the foreground prompt P_d may yield a weak background, we concatenate a dedicated background prompt P_{bg} with P_d as the textual condition to the model. The resulting image I (see Fig. 4) preserves the planned layout with semantically separated objects and reduced attribute leakage.

3.3. Layout Refinement via Iterative Feedback

After generating a layout-grounded image I, we ensure that the prompt description P_d, P_{bg} is accurately reflected in terms of object count and aesthetics. We therefore run an

Figure 5. Successive layout refinement using VLM critic. Corresponding layouts in the inset.

iterative refinement loop that (i) evaluates I, (ii) identifies flaws and extracts structural feedback, and (iii) updates both the planning graph and prompt until the output meets the desired quality.

Critic VLM: We employ a VLM agent built on Qwen3-VL [58], reconfigured to serve as a Critic VLM for analyzing generated images and suggesting prompt or layout revisions. LLM behaviour varies sharply with instruction design [33, 47]; the same model can function as either creator or critic based on the prompt, integrating creator/critique signals into its chain-of-thought reasoning. Exploiting this, we supply a critique-style prompt P_{crit} to the VLM which evaluates the generated image I on two aspects: *(a) object count fidelity* and *(b) visual aesthetics*, as shown in Fig. 3. Since VLMs remain unreliable at dense object counting [19], we compute *count accuracy* using an open-vocabulary detector [32] to obtain s_c. Likewise, because VLMs tend to provide overly positive aesthetic judgments [9], we rely on an external aesthetics estimator [52] to evaluate prompt-image alignment, yielding s_a. A composite score S then captures overall quality:

$$S = \alpha \cdot \max\left(0, 1 - \frac{|s_c - s_c^{gt}|}{s_c^{gt}}\right) + \beta s_a \qquad (7)$$

with s_c^{gt} as the prompt-implied count and α=0.6, β=0.4. The score S, together with I, P_d and P_{crit}, is passed to the Critic VLM, which produces textual feedback such as `'cat`$_1$ `overlaps with cat`$_2$`'`, `'2 birds detected but target is 1'`, or `'lighting inconsistent across objects'`. This textual feedback (denoted as P_{feed}) is then utilized for iterative layout refinement to improve count fidelity and visual realism. The prompt P_{crit} is provided in the supplementary.

Parameter-Free Refinement: The Critic VLM's textual feedback must be translated into concrete edits to the planning graph to generate an updated image incorporating the feedback. Instead of fine-tuning model parameters, which is impractical without large annotated datasets, we employ a parameter-free textual refinement operator inspired by [63]. We denote this operator as Ψ, an LLM-based text-editing agent that updates the planning graph through structured natural-language reasoning. Given the current graph G, the critic feedback P_{feed}, and an optimization prompt P_{opt}, the operator produces an updated graph:

$$G' = \Psi(G, P_{\text{feed}}, P_{\text{opt}}).$$

Mirroring how PyTorch's AutoGrad [38] performs gradient updates, $\Psi(\cdot)$ interprets the input feedback and estimates a textual analogue of a *gradient*, using a loss function which is a pre-defined textual prompt template defined in P_{opt}. It then applies gradient-like edits to the planning graph G via textual modifications rather than numerical parameter updates in autograd. Operating entirely on textual representations, Ψ applies targeted structural edits to G. For example: ① For feedback such as `"`*cup*$_7$ `is overlapping with` *cup*$_3$`"`, it increases spatial separation in G. ② For `"only 28 cups detected but target is 30"`, it inserts the missing object nodes. This parameter-free refinement is compatible with any frozen diffusion model and supports precise, semantic-level corrections. After obtaining G', we obtain the prompt $P_{G'}$(Eq. 1) to generate a refined layout $\mathbb{L}$ (Eq. 2), followed by updated image synthesis I (see Fig. 5). The loop terminates once the composite score S exceeds 0.85 and the predicted count s_c matches the ground-truth value s_c^{gt}, ensuring complete count fidelity before finalizing the image.

4. Experiments

4.1. Dataset and Evaluation

Datasets: We evaluate on four sets spanning instance count and compositional difficulty: *COCO-Count* (MS-COCO subset [30]); *T2I-CompBenchCount* (subset of [21]); newly proposed CountLoop-S (single category, 200 prompts, 30–200 instances); and CountLoop-M (multi-category, 200 prompts, 30–100 instances). Benchmark construction details and prompt lists are in Sec. 1.4 of the supplementary.

Evaluation Metrics: We evaluate counting accuracy using *F1* and *MAE* metrics, and assess prompt–image alignment. For counting, we adopt the state-of-the-art open-vocabulary detector OWLv2 [35] inspired by [55], using the number of detected boxes as the estimated count. *Spatial* alignment is measured via CLIP–FlanT5 encoder from VQAScore [27].

Competitors: We compare CountLoop with representative *T2I* (SDXL [39], FLUX [8], SDXL-Turbo [44], SD3.5 [46], Counting Guidance [25], GPT-4o [57]), *Agentic* (GenArtist [51], SLD [55], RPG-DiffusionMaster [59]), and *L2I* (LMD [29], MIGC [64], CountGen [7]) methods. Implementation details are provided in the supplementary.

4.2. Main Results

Quantitative Results: Tab. 1 highlights CountLoop's state-of-the-art count accuracy, especially as instance numbers scale. On standard benchmarks like COCO-Count, CountLoop (95.06 F1) already surpasses strong competitors like SLD (90.34) and GPT-4o (72.00). The key differ-

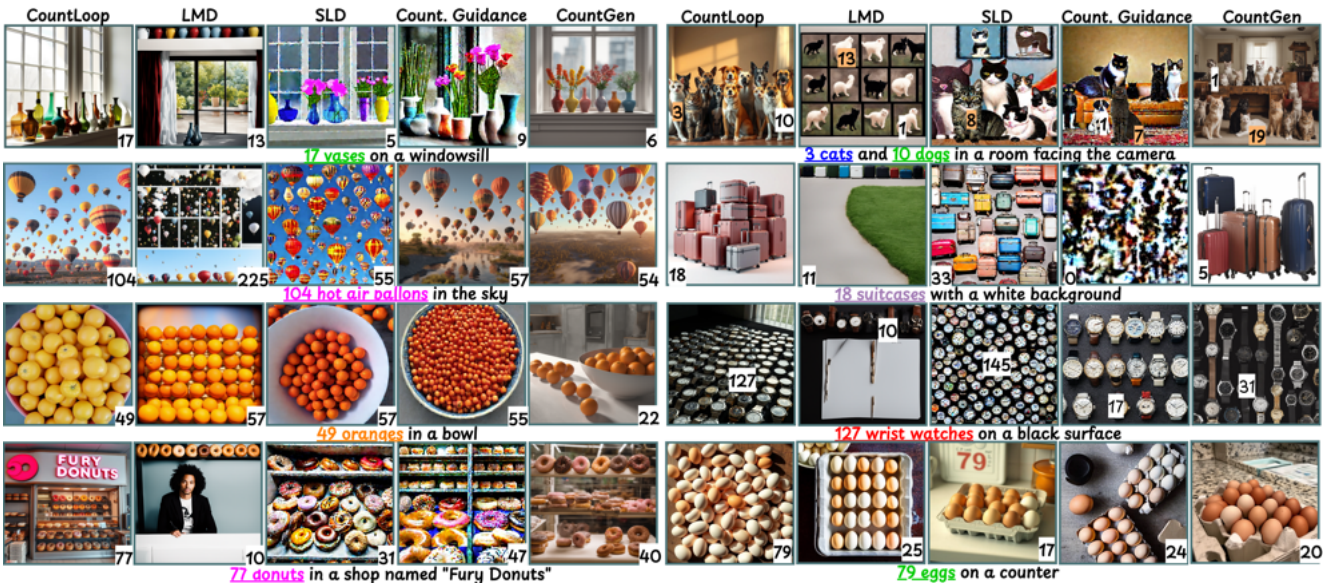

Figure 6. COUNTLOOP maintains precise object counts and natural arrangements in dense scenes, while methods like LMD [29], SLD [55], Counting Guidance [25], and CountGen [7] exhibit abnormal counts, spatial collapse, and grid artifacts. More visuals in the supplementary.

entiators are the high-instance benchmarks: COUNTLOOP-S and COUNTLOOP-M, where COUNTLOOP (87.32 F1) remains robust, while both L2I methods (CountGen: 48.18) and agentic pipelines (GenArtist: 51.00) suffer a clear performance collapse. Crucially, COUNTLOOP also leads in spatial quality (0.93 on COUNTLOOP-S), avoiding the count-quality trade-off that hinders the previous approaches. This showcases the importance of preventing semantic leakage and the role of Critic VLM to generate images without compromising on count accuracy, even for dense scenes for both single and multiple instance scenarios.

Qualitative Results: Fig. 6 demonstrates COUNTLOOP's consistent precision across diverse instance counts. For "17 vases", competitors under-generate (LMD: 13, Count Guidance: 9, CountGen: 6), while COUNTLOOP accurately renders all 17 with natural arrangements. In the "104 hot air balloons" scene, COUNTLOOP precisely places all balloons with realistic spacing, unlike Count Guidance (57), CountGen (54), and LMD's artificial clusters (225 overlapping). Crucially, COUNTLOOP consistently avoids semantic drift, grid artifacts, and count inaccuracies that outperforms competitors for high-instance image generation.

4.3. Ablations and Analysis

Key Components: We collectively demonstrate how different architectural components contribute to COUNTLOOP's performance in distinct ways. The main ablation Tab. 2a progressively builds the model from a simple baseline to show how each of our model components provide a significant and cumulative boost to counting accuracy. This study on the COUNTLOOP-S benchmark confirms that while the initial layout and leakage-prevention mechanisms are effective, both Cumulative Attention (CA) and Iterative Refinement (IR) are critical, with each contributing a similar, massive boost of +17-18 F1 points over baseline. Note that we run 3 rounds by default, since Tab 2(a) in Sec. 1.2 of the supplementary shows that three iterations markedly improve both counting and aesthetics, even though a single pass already surpasses all its competitors. We further validate our design choices in Sec. 1.2 of the supplementary, which demonstrate robust performance across various diffusion backbones, VLM models, open-vocabulary detectors, and aesthetic scorers.

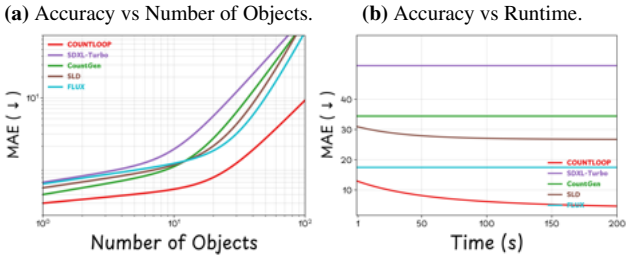

Figure 7. *Left:* Counting difficulty rises with instance count. *Right:* Runtime curves echo the same ordering.

Runtime Analysis: We evaluate end-to-end runtime on the COUNTLOOP-S benchmark by measuring both the anytime MAE trajectory and the total time required to achieve accurate counts. As shown in Fig. 7b, COUNTLOOP continues to improve steadily over time, ultimately reaching a substantially lower error floor. Compared to the agentic SLD [55], COUNTLOOP not only achieves a lower final MAE but also reaches error thresholds faster, with $\sim$1.2$\times$ speedup at 10 instances and up to $\sim$1.4$\times$ at 100. This behaviour mirrors the trends in Fig. 7a, where COUNTLOOP remains robust as object counts grow, while T2I and L2I methods plateau early and fail to recover beyond $\sim$10–20 objects.

Table 1. Comparing counting and aesthetic quality across four benchmarks across T2I , L2I , and Agentic systems. For every dataset we report *Counting* – split into **F1** (higher is better) and **MAE** (lower is better) – and *Spatial* (aesthetic quality).

Model	Single Category									Multi Categories		
	COCO-Count			T2I-CompBench			COUNTLOOP-S			COUNTLOOP-M		
	Counting		Spatial ↑	Counting		Spatial ↑	Counting		Spatial ↑	Counting		Spatial ↑
	F1 ↑(%)	MAE ↓		F1 ↑(%)	MAE ↓		F1 ↑(%)	MAE ↓		F1 ↑(%)	MAE ↓	
SDXL [39]	74.00	2.37	0.38	76.00	2.72	0.75	65.00	29.96	0.63	55.00	9.89	0.55
FLUX [8]	87.00	1.40	0.53	83.00	1.48	0.78	71.00	17.47	0.65	63.00	9.62	0.58
SD 3.5 [46]	49.00	1.10	0.46	84.00	1.58	0.76	70.00	21.81	0.64	69.00	8.40	0.56
SDXL-Turbo [44]	45.20	2.50	0.23	65.45	3.76	0.53	32.25	51.14	0.39	45.21	9.95	0.37
Counting Guidance [25	67.54	1.68	0.63	71.41	3.90	0.56	36.67	42.49	0.47	64.42	8.43	0.41
GPT-4o [57]	72.00	0.58	0.55	91.00	1.71	**0.80**	49.45	33.56	0.69	79.10	4.61	0.60
LMD [29]	58.00	3.09	0.24	74.00	5.56	0.73	66.00	16.62	0.66	71.00	6.34	0.64
MIGC [64]	79.00	1.83	0.36	70.00	2.96	0.65	67.00	17.54	0.65	72.00	6.28	0.62
CountGen [7]	58.99	1.88	0.61	63.75	5.22	0.75	48.18	34.44	0.72	72.00	6.46	0.69
GenArtist [51]	75.40	1.50	0.45	85.33	1.50	0.70	51.00	32.47	0.60	77.87	4.93	0.57
SLD [55]	90.34	1.15	0.70	91.50	1.44	0.77	55.04	29.65	0.75	82.46	3.74	0.65
RPG [59]	84.89	1.28	0.60	91.32	1.47	0.75	51.89	31.85	0.70	80.16	4.34	0.62
COUNTLOOP (ours)	**95.06**	**0.45**	**0.93**	**86.76**	**1.23**	0.79	**87.32**	**7.59**	**0.93**	**86.58**	**2.13**	**0.73**

Table 2. Analysis of COUNTLOOP components and user study. **PG**: Planning Graph, **CA**: Cumulative Attention, **IR**: Iterative Refinement, **OVD**: Open-vocabulary Detector, **AS**: Aesthetic Scorer.

(a) Ablation of design components.

PG	CA	IR	F1 ↑	Spatial ↑
✗	✗	✗	63.44	0.61
✓	✗	✗	68.83	0.68
✗	✓	✗	69.92	0.71
✓	✓	✗	75.77	0.81
✓	✓	✓	**87.32**	**0.93**

(b) Critic VLM configs.

OVD	AS	F1 ↑	Spatial ↑
✗	✗	55.13	0.67
✗	✓	67.22	0.70
✓	✗	81.49	0.83
✓	✓	**87.32**	**0.93**

(c) User Evaluation (5 best, 0 worst).

Metric	COUNTLOOP	LMD	FLUX	SLD	CountGen
Alignment	4.5	3.4	3.7	4.0	3.6
Aesthetics	4.4	3.3	3.5	3.9	3.8
Count	4.6	3.7	4.0	4.2	3.4
Overall	4.5	3.5	3.7	4.0	3.6

Critic Composition: Tab. 2b indicates that a VLM-only critic performs poorly on numeracy. Adding only an aesthetic scorer (**AS**) provides limited improvement. Incorporating an open-vocabulary detector (**OVD**) is decisive, markedly improving counting and layout. The full setting achieves the best overall behavior, suggesting OVD grounds counts while AS stabilizes visual/relational quality.

Human Evaluation: We ran a 30-participant study (20 designers, 10 AI artists) across all the four benchmarks. Each participant rated 15 blinded set of 5 images(COUNTLOOP, FLUX [8], LMD [29], SLD [55], and CountGen [7]) on a 5-point scale for *Prompt Alignment*, *Aesthetic Quality*, *Count Accuracy*, and *Overall Preference*. COUNTLOOP was preferred across all axes (Table 2c), with significant gains over its competitors. Procedure, demographics, and the survey interface are detailed in Sec. 1.4 of the supplementary.

5. Conclusion

We presented COUNTLOOP, a training-free, iterative framework that enables high-instance image generation with precise object counts and strong visual quality. By combining VLM-based planning graphs, instance-driven attention, and cumulative latent composition, COUNTLOOP overcomes key limitations of existing methods, such as count saturation, semantic leakage, and rigid layouts. A critic-in-the-loop further refines generation by updating layout and prompts. Evaluations on COCO-Count, T2I-CompBench, and new high-instance benchmarks show that COUNTLOOP achieves over 2× improvement in counting accuracy while preserving aesthetics and scaling reliably to 100+ instances per image.

Figure 8. Failure cases

Limitations: As a training-free system, COUNTLOOP inherits the limitations of its frozen VLM and detector, allowing their biases to propagate. Dense occlusions, especially in human scenes, can degrade attention quality and spatial consistency. Without explicit 3D priors, COUNTLOOP struggles with generating objects in different poses and complex perspectives. Count fidelity also depends on the diffusion latent dimension, object scale, and canvas resolution; larger objects may merge, limiting achievable counts. Moreover, strong layout guidance can reduce intra-class diversity by biasing toward canonical poses or textures for count accuracy. Some of these limitations are shown in Fig. 8.

Future Work: It would be interesting to extend COUNTLOOP to layout-free generation with weak spatial priors, improve human modeling in dense scenes, and support high object counts through controllable upscaling or multi-canvas fusion. Integrating this approach with ViT-based T2I models may yield valuable insights.

6. Supplementary Material

6.1. Implementation Details

All experiments were conducted on a single NVIDIA A100 GPU (80GB) running Ubuntu 22.04, with Python 3.10, PyTorch 2.1, and CUDA 12.2. For all competitors (LMD [29], SLD [55], CountGen [7], MIGC [64], GenArtist [51], RPG-DiffusionMaster [59], etc.), we used the authors' officially released code and pre-trained checkpoints, following their recommended hyperparameter settings. No modifications were made that would disadvantage the baselines.

Backbone and resolution: Unless otherwise stated, we used Stable Diffusion XL (`sdxl-base-1.0`) as the backbone diffusion model for CountLoop, configured with 50 denoising steps and default classifier-free guidance from the original checkpoint. Layout conditioning was implemented via the GLIGEN [28] layout encoder (box+text mode), and cross-instance texture consistency was enforced using the IP-Adapter (public checkpoint from [60]). Images were generated at a resolution of 1024×1024 for all methods that support this resolution; for baselines whose official code operates at 512×512, we used their native resolution and then bilinearly upsampled to 1024×1024 only for visualization, while all quantitative metrics (F1/MAE/Spatial) were computed at the original resolution to avoid any bias.

L2I baselines: For LMD [29], we keep the authors' full two-stage pipeline: the LLM layout generator and the layout-conditioned diffusion model. All system prompts, layout templates, and scene-decomposition instructions used by their LLM are preserved exactly; only the user-visible prompt (the benchmark prompt) is substituted. MIGC [64] and CountGen [7] are run with their released code, pre-trained diffusion backbones, and unmodified layout encoders. Across all L2I baselines, we preserve the authors' layout formats, conditioning methods, and refinement logic without any tuning.

Agentic baselines: For SLD [55], we use the authors' publicly released self-correction pipeline exactly as implemented: the internal critique prompts, refinement checklists, and corrective rules are kept unchanged. The only substitution is the initial task prompt (our benchmark prompt), while all system- and meta-prompts remain the same. We use the default SD-based backbone, the recommended number of refinement rounds, and the authors' original hyperparameters. For GenArtist [51], we run the official generation pipeline (not the editing pipeline), preserve the original agent roles and inter-agent communication templates, and use the default diffusion backbone. We replace only the user-facing text prompt; all role prompts, decision logic, and the multi-agent controller remain intact. For RPG-DiffusionMaster [59],we use the official role-playing workflow with its recaption-plan-generate sequence, preserving the authors' default refinement schedule, guidance scales, and VLM configuration. No internal prompts or model weights are modified; the only change is substituting the initial prompt with our benchmark prompt. Across all baselines, we avoid tuning hyperparameters or increasing the number of refinement rounds, ensuring a fair comparison with CountLoop .

T2I baselines: For FLUX [8], we use the publicly released FLUX.1-dev checkpoint (not FLUX-schnell or FLUX-pro), with the authors' default VAE and classifier-free guidance schedule. For GPT-4o [22], we use the standard image-generation endpoint at a fixed resolution of 1024×1024, with high-detail mode disabled and no multi-image conditioning. SDXL [39], SD 3.5 [46], and SDXL-Turbo [48] are all run using their official pipelines with default guidance scales, VAE settings, and sampling schedules. Across all T2I baselines, only the text prompt is changed; all model-specific system prompts and hyperparameters remain untouched.

CountLoop configuration: Both the Design and Critic VLMs in CountLoop were instantiated from the Qwen3-8B [58] VLM variant. We used the base variant of GroundingDINO [32] as the detector guide for the Critic and the pretrained image encoder Q-Align [52] as the aesthetic guide. The composite score weights were set to $\alpha = 0.6$ for count accuracy and $\beta = 0.4$ for aesthetic quality, with a GroundingDINO confidence threshold of 0.3, and the loop terminated when the composite score $S \geq 0.85$ or after three refinement rounds, whichever came first. A fixed random seed of 42 was used for all runs, and all third-party models and detectors were loaded from publicly released checkpoints. The overall workflow of CountLoop is provided in Algorithm 1. Note that for clarity, the prompt examples in the main paper (Sec. 3.1) are simplified snippets; the full, executable system prompts used in our experiments are provided in Fig. 15 (Design VLM) and Fig. 16 (Critic VLM).

6.2. Additional Analyses

Performance with different Design-Critic variants: We evaluate the impact of various Design–Critic configurations on CountLoop-S, pairing three open-source Design VLMs with three Critic VLMs. Results are in Tab. 3.

Table 3. Designer–Critic on CountLoop-S. Counting uses a fixed detector; alignment is the Critic's score. Best per Designer is underlined; overall best is **bold**.

Designer (VLM)	Critic (VLM)	F1 (%)↑	Spatial↑	Iters↓
Qwen3-VL [58]	Qwen3-VL	**87.32**	**0.93**	**3**
Qwen3-VL [58]	LLaVA 1.5B	78.0	0.70	5
Qwen3-VL [58]	Pixtral	79.2	0.72	4
LLaVA 1.5B [31]	Qwen3-VL	<u>80.3</u>	<u>0.74</u>	<u>4</u>
LLaVA 1.5B [31]	LLaVA 1.5B	82.1	0.73	4
LLaVA 1.5B [31]	Pixtral	83.0	0.75	3
Pixtral [1]	Qwen3-VL	83.5	0.76	3
Pixtral [1]	LLaVA 1.5B	81.1	0.72	4
Pixtral [1]	Pixtral	82.0	0.73	3

Performance across different T2I backbones: To assess

Figure 9. Spatial reasoning in image generation. Vanilla LLM (LMD [29]) fails to identify directions.

the generality of CountLoop across diffusion backbones, we replaced the default SDXL model with two additional Stable Diffusion checkpoints: *SD v1.5* and *SD 3.5*. We kept all other components (planning graph, cumulative attention, IP-Adapter, critic loop) and hyperparameters identical. Tab. 4b reports counting F1, MAE, and spatial scores on the CountLoop-S benchmark. While all backbones benefit substantially from CountLoop's structured refinement, we observe that higher-capacity models yield marginally better spatial coherence, with SDXL at the top. Importantly, counting performance remains robust (F1 $\geq 85\%$) across backbones, indicating that CountLoop's instance-control mechanism is largely model-agnostic.

Choice of OV-Detector: In all experiments, we use the base GroundingDINO [32] checkpoint as the open-vocabulary detector for the Critic VLM. We found that modest changes to the confidence threshold mainly trade off between strict count enforcement and tolerance to small-scale or partially occluded instances, but do not qualitatively change the overall trend of CountLoop's gains. For clarity and reproducibility, we therefore fix the detector to this configuration and leave a broader detector sweep to future work.

Choice of Aesthetic Scorer: For aesthetic guidance, we use Q-Align [52] as a frozen image encoder, mapping each generated image to a scalar aesthetic score $s_a \in [0, 1]$ that is combined with the count term. We experimented with replacing Q-Align with purely VLM-based aesthetic judgments and observed higher variance and occasional misalignment with human preferences, consistent with recent findings on VLM aesthetics [9]. Since our qualitative and user-study results already capture aesthetic effects, we keep Q-Align as the sole aesthetic scorer in all reported quantitative experiments.

Table 4

(a) Number of iterations.

Iters	F1 (↑)	MAE (↓)	Spatial (↑)
1	70.31	11.68	0.76
2	78.47	9.72	0.88
3	**87.32**	**7.59**	**0.93**

(b) Backbone swap.

Backbone	F1 (↑)	MAE(↓)	Spatial
SD v1.5	85.32	8.05	0.88
SD 3.5	87.91	7.44	0.90
SDXL	**87.32**	**7.59**	**0.93**

6.3. Textual Refinement Operator Ψ

The main paper introduces a parameter-free textual refinement operator Ψ (Sec. 3.3) that updates the planning graph using feedback from the Critic VLM. Here we spell out its behavior in more detail, without introducing any additional trainable components.

Input and output: At iteration t, Ψ operates on the current planning graph G_t, the critic feedback P_{feed}, and the optimization prompt P_{opt}:

$$G_{t+1} \;=\; \Psi(G_t,\; P_{\text{feed}},\; P_{\text{opt}}).$$

The graph G_t is represented as JSON (nodes with categories, positions, depth, size, color; edges with relations). P_{feed} is the natural-language feedback produced by the Critic VLM (e.g., "`cup_7 overlaps with cup_3`"). P_{opt} is the system prompt that defines the allowable edit operations and constrains the VLM's output format.

Objective signal: The Critic VLM is guided by the composite score

$$S \;=\; \alpha \cdot \max\!\left(0,\; 1 - \frac{|s_c - s_c^{gt}|}{s_c^{gt}}\right) \;+\; \beta \cdot s_a, \qquad (8)$$

where s_c is the predicted count from the open-vocabulary detector, s_c^{gt} is the target count in the prompt, and $s_a \in [0, 1]$ is the aesthetic score. The weights (α, β) and stopping threshold 0.85 are as in the main paper. The role of Ψ is to edit G_t in a way that is expected to increase S and move s_c toward s_c^{gt}.

Edit space (P_{opt} definition): The optimization prompt P_{opt} restricts Ψ to a small vocabulary of graph edits expressed in text/JSON, such as:

- *Local position updates:* nudging a node to reduce overlaps or break grid patterns (small $\Delta x, \Delta y$ in normalized coordinates).
- *Count corrections:* adding a few new nodes when $s_c < s_c^{gt}$ in free regions, or slightly shrinking/moving nodes when heavy overlap causes under-detection.
- *Mild attribute adjustments:* adjusting depth, size, or color when the critic explicitly flags unrealistic layering or inconsistent appearance.

All edits are constrained so that positions remain in $[0, 1]^2$, displacements per iteration are small, and the overall graph structure (object identities, background context) is preserved.

LLM-based implementation: We instantiate Ψ as an LLM-based text-editing agent. Given (G_t, P_{feed}) and the instructions in P_{opt}, it is prompted to (i) summarize the main failure modes (count error, overlap, grid artefacts), and (ii) emit an

updated JSON graph G_{t+1} that fixes those issues. Crucially, Ψ does *not* change any diffusion or VLM weights; it only rewrites the textual/JSON representation of the scene. This makes the refinement procedure compatible with any frozen backbone and keeps COUNTLOOP fully training-free.

Termination: At each iteration, we recompute (s_c, s_a, S) from the new image. The loop stops once $s_c = s_c^{gt}$ and $S \geq 0.85$. In practice, we find that 3 iterations are sufficient to reach high count fidelity and spatial quality on COUNTLOOP-S and COUNTLOOP-M, as reported in the main paper and in Sec. 6.2.

6.4. Benchmarks and Evaluation Details

Here we provide the details of the evaluation metric and the benchmark dataset used to judge the performance of our COUNTLOOP model.

COUNTLOOP-S & COUNTLOOP-M Benchmarks: Existing text-to-image (T2I) counting benchmarks, including T2I-Compbench [21] and COCO-Count [7], suffer from several key limitations: (i) *Limited class diversity*: COCO-Count, for example, samples only 20 classes from MS-COCO, excluding many real-world object types; (ii) *Restricted count range*: Most benchmarks evaluate generation only for low-count scenes (typically <10 objects), failing to challenge models on dense or high-instance compositions; and (iii) *Lack of complex multi-category prompts*: Existing datasets rarely assess the ability to control multiple object types and their relationships within a scene. These constraints make it difficult to assess compositional and numeracy capabilities in state-of-the-art T2I systems rigorously.

To address these gaps, we introduce 2 new benchmarks: **COUNTLOOP-S** and **COUNTLOOP-M**. Both are constructed from 92 diverse classes curated from the OmniCount-191 dataset [36]. COUNTLOOP-S is designed for single-category, high-count evaluation (*e.g.*, *"A photo of 127 watches"*), while COUNTLOOP-M targets multi-category control (*e.g.*, *"A photo of 148 birds and 6 dogs"*), enabling assessment of compositional fidelity at scale. Representative generations are shown in Fig. 13; further qualitative examples are provided below.

Key Features:

- **High class diversity:** 92 categories, including *airplanes, apples, balloons, bananas, bears, birds, bowls, buttons, butterflies, cars, cats, dogs, donuts, elephants, fish, hot air balloons, laptops, monkeys, oranges, pineapples, rabbits, roses, sheep, suitcases, swans, teacups, tigers, trucks, turtles, vases, watches, wine glasses,* and more.
- **Broad count range:** Instance counts from 1 up to 100 and select very large counts (*e.g.*, 107, 140, 148), supporting rigorous evaluation in both sparse and dense settings.
- **Diverse backgrounds:** Prompts encompass a wide array of real-world contexts, such as *in a kitchen cabinet, on a picnic table, on a pantry shelf, on a couch armrest, in the sky, in the water, over a valley, on a refrigerator, on a lunch tray*, etc.
- **Composite categories:** Multi-category prompts combine classes (*e.g.*, cats and dogs, balloons and pineapples, bears and mice, cats and suitcases, candles and donuts, cars and helicopters), enabling compositional reasoning beyond single-object scenes.

A brief statistics of our benchmark is shown in Fig. 10.

Details on Human Evaluation Setup: We designed our human evaluation survey using Google Forms. Raters were

ALGORITHM 1: COUNTLOOP: High-level agentic loop for count-faithful high-instance generation

Input : Prompt p (class c, target count s_c^{gt}); Design VLM V_{design}; Critic VLM V_{crit}; frozen T2I backbone $\mathcal{G}$; layout encoder $\mathbb{E}$; IP-Adapter θ; open-vocabulary detector (OVD); aesthetic scorer (AS); weights (α, β) and threshold 0.85.

Output : Image $I^\star$ with count s_c^{gt} and score $S \geq 0.85$.

Plan (Design VLM → Planning Graph).
Parse p into objects and relations; build planning prompt P_G with in-context examples P_{icl}.
$\mathbb{J} \leftarrow V_{\text{design}}(P_G, P_{\text{icl}})$ `// JSON: objects, relations, context`
Extract layouts $\mathbb{L}$ and prompts P_d, P_{bg}; build planning graph $G_0 = (V, E, B_{\text{bg}})$ with basic spatial constraints and a fixed instance order.
Set $G \leftarrow G_0$.
Iterative Synthesize–Critique–Refine.
repeat
  Synthesize (cumulative, instance-aware generation).
  Encode per-instance layouts $l_i \in \mathbb{L}$ with $\mathbb{E}$ to obtain Q_i and cumulative features F_i.
  Generate instances with IP-Adapter using: $I_{i+1}, Z_q^{i+1} = \Phi(F_{i+1}, P_d, \theta(I_i))$; compose the final image I with background inpainting using P_{bg}.
  Critique (count and aesthetics).
  Run OVD on I to obtain count s_c; compute $s_a = \text{AS}(I)$.
  Compute composite score

$$S = \alpha \cdot \max\left(0, 1 - \frac{|s_c - s_c^{gt}|}{s_c^{gt}}\right) + \beta\, s_a$$

  Obtain textual feedback $P_{\text{feed}} = V_{\text{crit}}(I, P_d, P_{\text{crit}}, S, s_c, s_a)$.
  Refine (textual operator Ψ on the planning graph).
  Update the planning graph with the parameter-free refinement operator:

$$G' = \Psi(G, P_{\text{feed}}, P_{\text{opt}}).$$

  Rebuild P_G and layouts $\mathbb{L}$ from G'; set $G \leftarrow G'$.
until $s_c = s_c^{gt}$ ***and*** $S \geq 0.85$
Return. $I^\star \leftarrow I$.

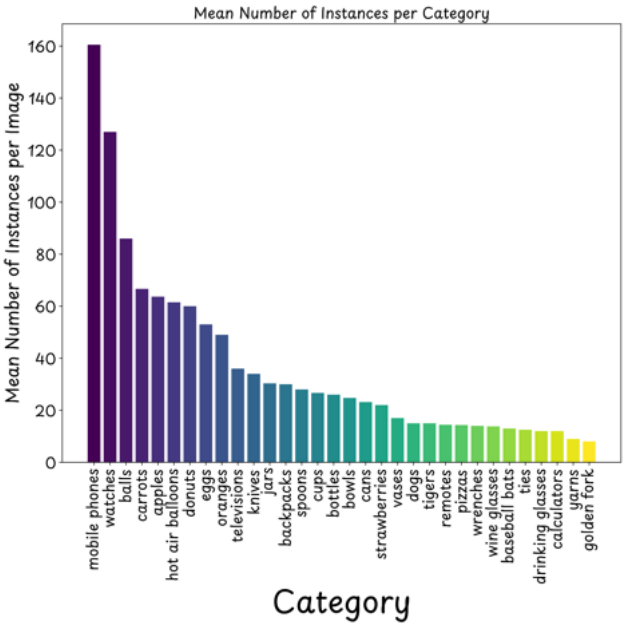

Figure 10. Statistics (instance per image vs category) for the COUNTLOOP-S benchmark.

asked to evaluate five images per set in terms of prompt alignment, aesthetic quality, count accuracy, and overall preference. A total of 15 image sets were selected across all four benchmarks, covering diverse prompts, object categories, and scene complexities, to ensure representative assessment. All images were blinded to method identity and randomized per rater. Participants (N=30) had an average age of 31 (range 22–45), and came from professional backgrounds in graphic design (20), AI art and research (10). Approximately 10 participants had prior experience or domain expertise in tasks requiring precise object counting (*e.g.*, data annotation, inventory management, or computer vision evaluation).

6.5. Potential Usecases

COUNTLOOP allows for high-instance, count-faithful scene generation while adhering to explicit numeric constraints. This feature is particularly valuable in modern interactive systems, ranging from warehouse manipulation simulators to survival and defense games, which often require scenes populated with a large number of distinct entities (*e.g.*, "spawn 120 crates and 6 forklifts in zone A" or "spawn 45 hostile drones and 10 civilian robots"). Manually authoring these scenes is time-consuming, and unconstrained generative models generally overlook exact cardinality, producing either too few instances or visually collapsed duplicates when the requested count exceeds approximately 10-15 instances. This mismatch can be problematic, as many downstream controllers rely on the assumption that the world state (such as inventory or enemy wave size) accurately reflects the specifications. We will highlight three representative use cases. Representative figures are in **Fig. 1** of the main paper.

Data Augmentation for object counting models: Object counting [16, 36, 41, 62] supports applications from crowd monitoring to ecological surveying, yet fully supervised pipelines remain expensive because they require dense point or box annotations. Unsupervised methods remove the labeling cost but are fragile to train and still trail strong supervised baselines [36, 45]. A natural alternative is to use text-to-image generators to create photorealistic, self-labeled data by embedding the category and the desired cardinality in the prompt. In practice, however, diffusion backbones such as SDXL or FLUX drift once the requested count becomes moderately large: instances collapse, merge, or vanish, causing the "self-labels" to no longer match the generated content.

COUNTLOOP circumvents this failure mode. Its layout-driven, agent-guided loop produces *count-faithful* high-instance scenes with realistic spacing, non-grid layouts, and controlled occlusion. This makes the synthetic data not only visually diverse but also numerically reliable. To quantify downstream impact, we augment the FSC-147 [41] training set with COUNTLOOP images covering 1–150 instances per class. Each image comes with exact instance counts, planning-graph boxes/points, and 1–3 exemplar crops. Training follows a simple low→high-count curriculum using mixed real and synthetic batches.

We fine-tune CountGD [3], an open-world counting model that leverages an open-vocabulary detector and supports both *text* and *exemplar* prompts, starting from the authors' FSC-147 checkpoint. We keep the original loss, evaluation protocol, and metrics (MAE/RMSE), and further report performance across count bins (1–5, 6–20, 21–50, 51–150). While SDXL or FLUX-based synthetic augmentation yields only modest gains, COUNTLOOP substantially reduces both MAE and RMSE and collapses the high-count error tail. These results highlight that *count-faithful* synthesis, not generic T2I augmentation, is the key driver of improved counting performance in real-world benchmarks.

Wave composition for games: Wave-based survival modes and large-scale battle games like Call of Duty™ often script difficulty via explicit per-class spawn counts: for example, "spawn 20 light vehicles, 10 heavy tanks, and 5 elite units" in a combat arena, or "spawn 30 cavalry, 10 chariots, and 5 war elephants" in a medieval battle wave. Players are scored on clearing these entities, and designers tune game balance by altering those counts. COUNTLOOP can generate high-entity battlefields that satisfy those numeric quotas across multiple classes, while still varying appearance within each class (*e.g.*, tanks with different turret orientations, horses with varying colors of coat). This is useful both for rapid wave prototyping and for producing training/evaluation frames for AI agents that must estimate threat level from the current mix of enemy types on screen.

Count-supervised synthetic data for T2V models: Recent controllable video generators improve numerosity by *curat-*

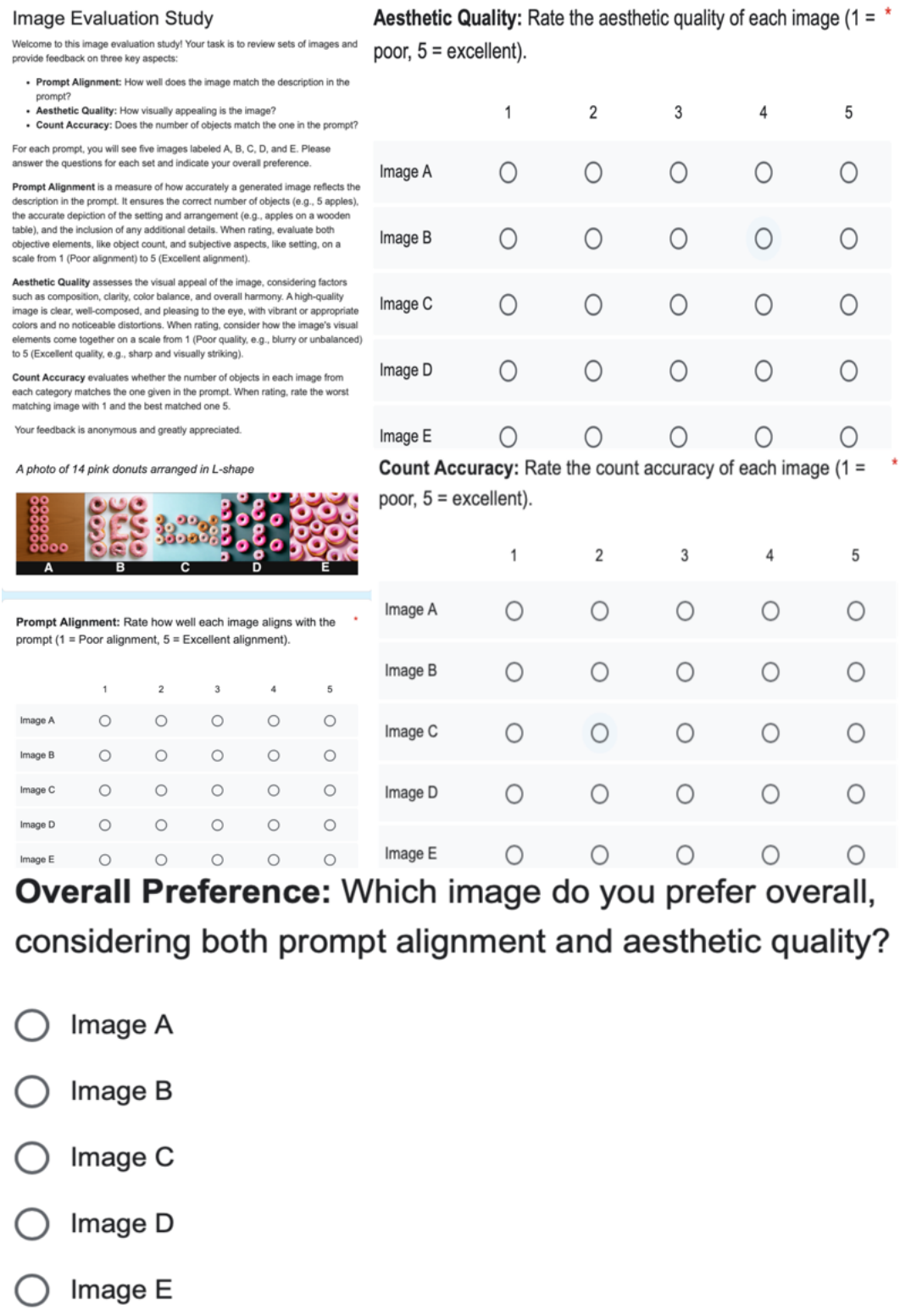

Figure 11. Human evaluation platform interface

ing web images: they mine captions like "three dogs" or "ten cars," then filter those images using an open-vocabulary detector so that the captioned count matches the detected count.[50] This yields approximate number awareness but still depends on finding scenes that already satisfy the requested cardinality. COUNTLOOP inverts that pipeline. In-

Table 5. FSC-147 comparison (MAE/RMSE ↓). Baselines from CountGD; augmentation rows add synthetic training splits.

Method	Prompt	Augmentation	Val MAE	Val RMSE	Test MAE	Test RMSE	MAE@51–150
GroundingDINO [32]	Text	None	54.45	137.12	54.16	157.87	–
LOCA [49]	Exemplar	None	10.24	32.56	10.79	56.97	–
CountGD [3]	Text + Exemplar	None (baseline)	7.10	26.08	5.74	24.09	18.3
CountGD	Text + Exemplar	SDXL	6.45	23.90	5.32	21.85	15.6
CountGD	Text + Exemplar	FLUX	6.28	23.10	5.21	21.40	14.8
CountGD	**Text + Exemplar**	**COUNTLOOP (ours)**	**5.62**	**21.05**	**4.68**	**19.72**	**12.1**

Augmentation protocol. SDXL/FLUX rows use the same prompt set and instance ranges as COUNTLOOP for controlled comparison. All models start from the official FSC-147 checkpoint and are fine-tuned with mixed real + synthetic batches using a low→high-count curriculum.

stead of searching for a scene with exactly N instances, it *constructs* one: given a specification (*e.g.*, "100 boxes on shelf A, 20 boxes on shelf B"), COUNTLOOP generates the scene, verifies it with an open-vocabulary detector, and iteratively corrects it until the per-class counts match exactly. The result is both a high-density image and a machine-readable instance list whose counts are guaranteed by construction. This allows data engines to request arbitrary cardinality mixes (*e.g.*, "5 boss units and 40 grunt units") and obtain perfectly count-labeled supervision pairs on demand.

6.6. Additional Qualitative Results

Here we provide some additional results of the VLM and the Image generation pipeline, along with an application of COUNTLOOP.

Qualitative Comparison Analysis: In addition to the qualitative results presented in the main paper, we have also provided a qualitative comparison (Fig. 12) and a generation gallery (Fig. 13). The visual results provide compelling evidence of COUNTLOOP's effectiveness in high-instance generation against SoTA models, under both single and multiple category scenarios.

6.7. Style-Aligned Image Generation

A pretrained diffusion U-Net model fine-tuned with LoRA (Low-Rank Adaptation) can produce vastly different visual styles from the same base concept. For example, the "13 cats" in Fig. 14 maintain the subject's constant while each panel applies a distinct style (photorealistic, semi-realistic 3D, anime, oil painting, sci-fi concept art, and storybook illustration), altering the lighting and rendering approach without altering the core content. Under the hood, LoRA fine-tuning freezes the original diffusion model's weights and inserts a small set of trainable low-rank matrices into the network. These low-rank weight updates capture the new style's visual patterns (*e.g.*, realistic fur vs. flat cartoon shading) without having to modify all of the model's parameters. This parameter-efficient approach enables fast, memory-light adaptation to each style, essentially a learned style transfer inside the diffusion process, while preserving the model's base knowledge (how to depict cats). Crucially, only a few additional parameters (on the order of megabytes) are required for each style, allowing each stylistic variation to be achieved without retraining or duplicating the entire multi-gigabyte models.

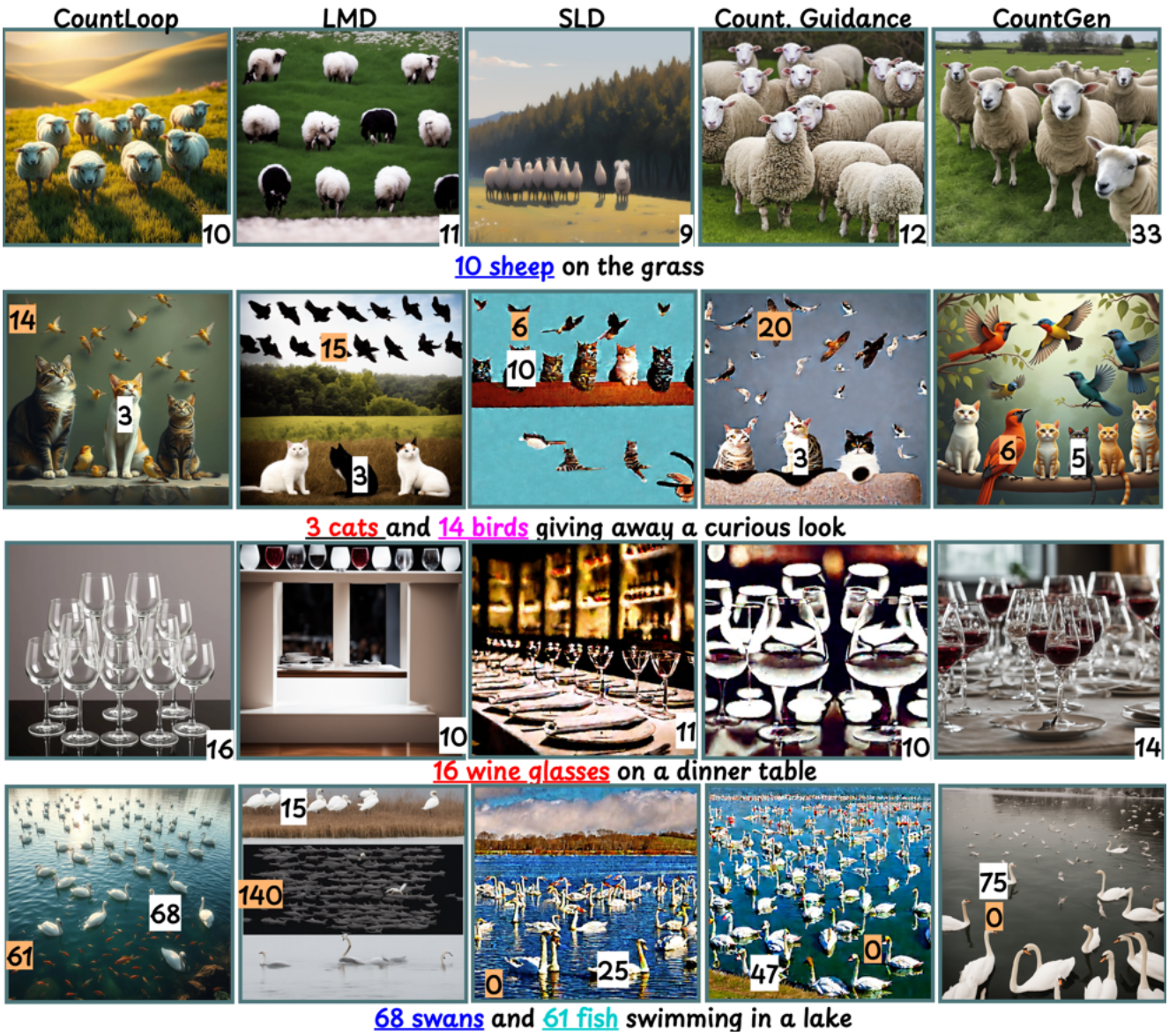

Figure 12. Comparison with SoTA

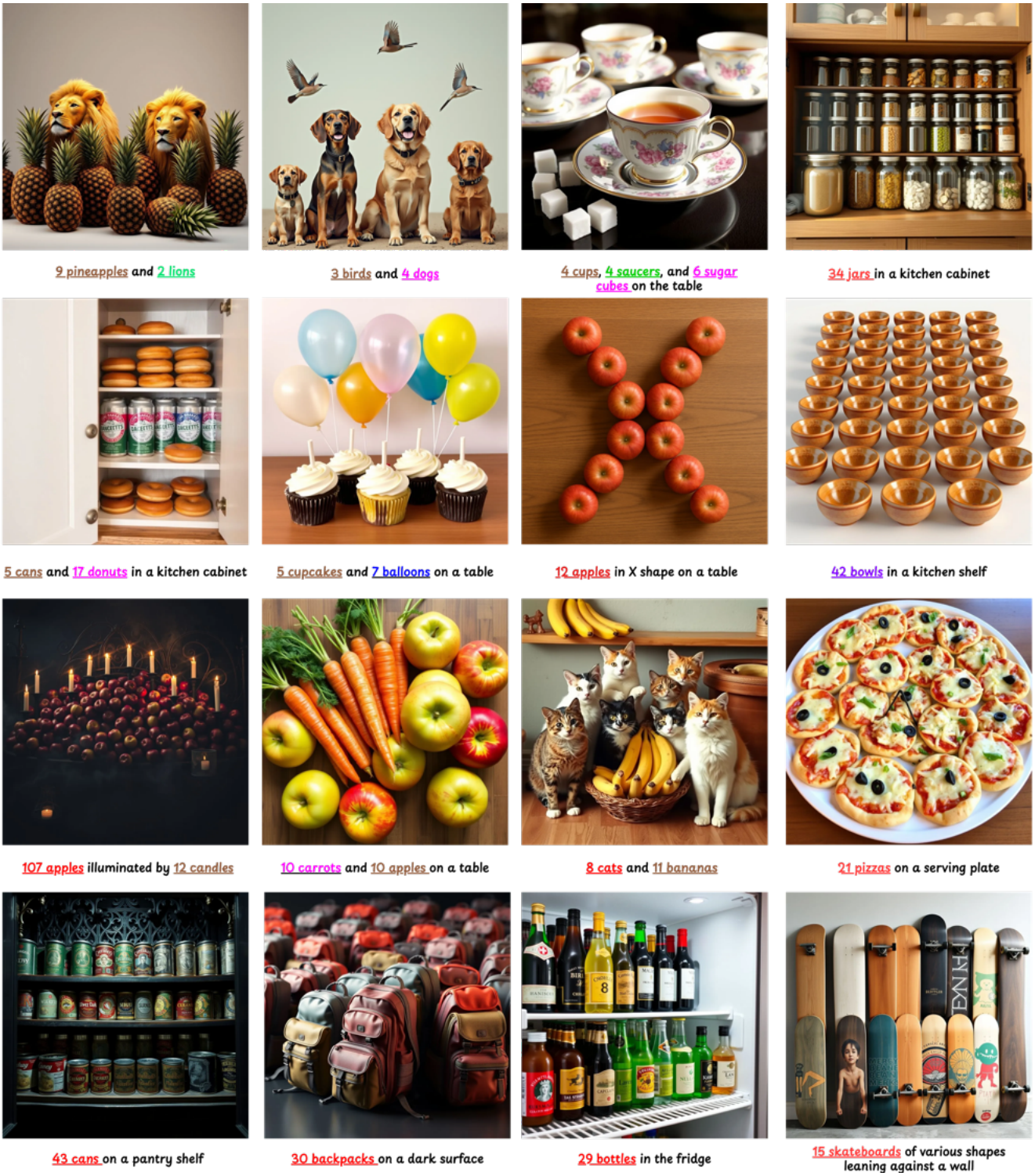

Figure 13. Visuals from our COUNTLOOP-M & COUNTLOOP-S benchmarks using COUNTLOOP .

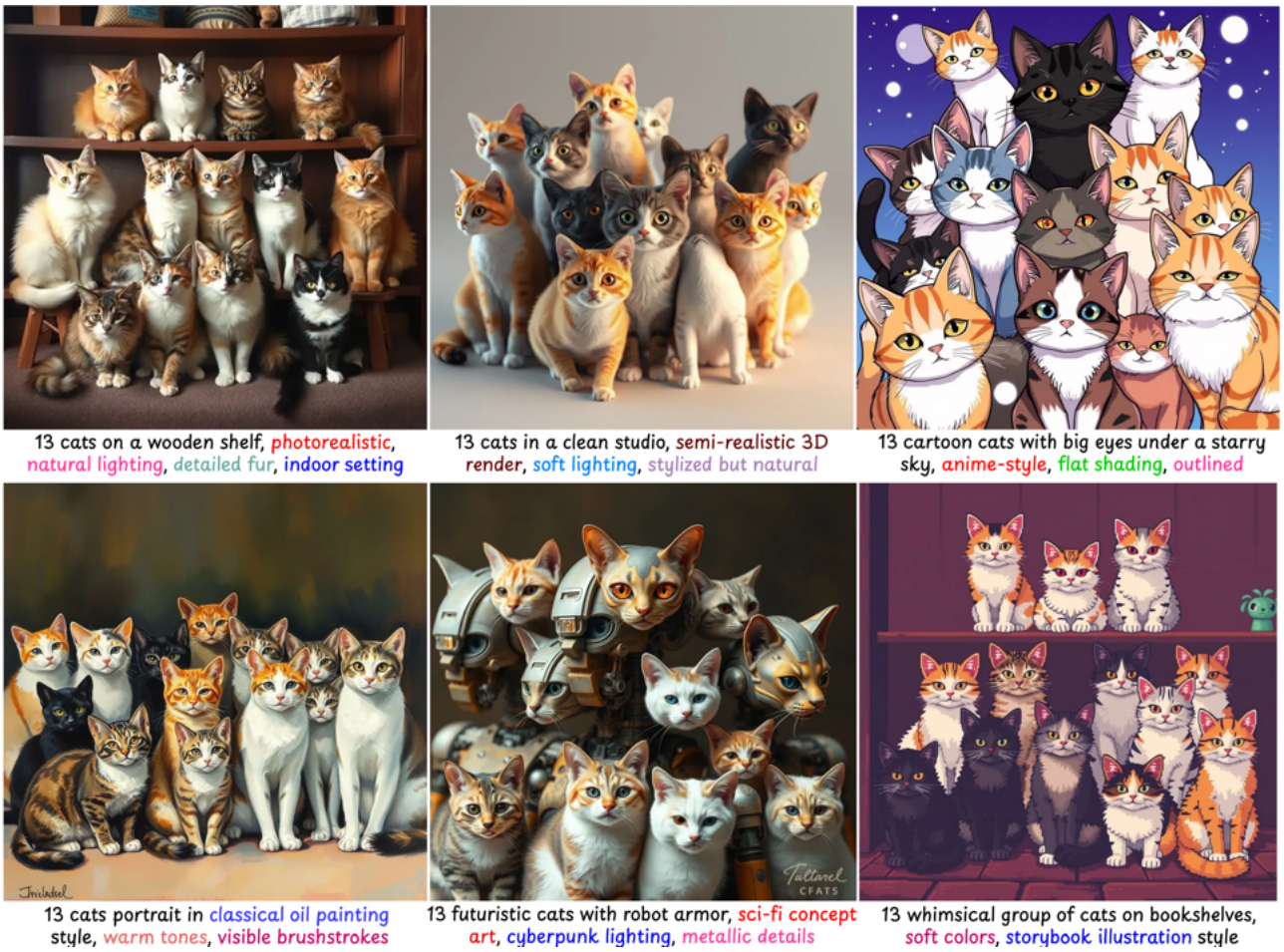

Figure 14. COUNTLOOP's style control capability

Design VLM Prompt (Planning Graph + Anti-Grid)

```
SYSTEM
You are the Design VLM for a high-instance T2I pipeline.
Given a text prompt P, produce:
  (a) a planning graph with object instances + relations, and
  (b) foreground/background prompts Pd and Pbg.
Return ONLY valid JSON. No prose.

GOALS
- Natural, non-grid layouts (no rigid rows/columns).
- Enforce minimum separation between instances.
- Provide instance attributes and global scene context.

CONSTRAINTS
- Positions [x,y] and sizes [w,h] normalized to [0,1].
- Minimum L2 distance >= 0.03 of image diagonal.
- Avoid straight rows/columns of length >= 6.
- Use light jitter in position/orientation to break grids.

SCHEMA
{
  "objects":[
    {
      "id": "string",
      "category": "string",
      "pos": [x, y],
      "d": float,       // depth in [0,1], larger = closer
      "size": [w, h],
      "color": "string",
      "attrs": ["optional attributes"]
    }
  ],
  "relations":[
    {
      "from": "id",
      "to": "id",
      "relation": "above|below|left-of|right-of|near",
      "dist": float,
      "angle": float
    }
  ],
  "context": "background description",
  "prompts": {
    "Pd": "foreground description for text-to-image",
    "Pbg": "background description for inpainting"
  }
}

EXAMPLE (abbreviated)
PROMPT: "20 oranges in a wooden crate"

{
  "objects":[
    {"id":"orange_01","category":"orange","pos":[0.32,0.58],
     "d":0.60,"size":[0.08,0.08],"color":"orange",
     "attrs":["on top layer"]},
    {"id":"orange_02","category":"orange","pos":[0.47,0.59],
     "d":0.62,"size":[0.08,0.08],"color":"orange",
     "attrs":["slightly shadowed"]}
    // ... remaining oranges, respecting anti-grid rules
  ],
  "relations":[
    {"from":"orange_01","to":"orange_02","relation":"left-of","dist":0.12,"angle":0.0}
    // ... a sparse set of relations to anchor local structure
  ],
  "context": "wooden fruit crate on a rustic table, soft daylight",
  "prompts":{
    "Pd": "a crate filled with many ripe oranges on a rustic table, soft daylight",
    "Pbg": "wooden table and crate background, soft daylight, no extra objects"
  }
}

CURRENT PROMPT: "<P>"
OUTPUT: JSON exactly following SCHEMA only.
```

Figure 15. Full executable Design VLM prompt. This detailed JSON schema replaces the simplified summary shown in the main paper. The Design VLM converts a text prompt into a planning graph and foreground/background prompts with anti-grid spatial constraints.

Critic VLM Prompt (Textual Refinement Signal Ψ)

```
SYSTEM
You are the Critic VLM. You receive:
  - foreground prompt Pd,
  - generated image I,
  - target count N for the main object class,
  - predicted count s_c from a detector,
  - aesthetic score s_a in [0,1] from an external scorer,
  - current planning graph G (objects + relations).

Your job:
  1) report updated scores and composite scalar S,
  2) decide whether to stop or continue refinement,
  3) provide structured, local suggestions that the
     textual refinement operator Psi will use to update G.

SCORING
- Count term: C_acc = max(0, 1 - |s_c - N| / max(1, N)).
- Composite: S = alpha * C_acc + beta * s_a
  with fixed alpha = 0.6, beta = 0.4.

OUTPUT FORMAT (JSON only)
{
  "scores": {
    "s_c": int,
    "N":   int,
    "s_a": float,
    "C_acc": float,
    "S":   float,
    "alpha": 0.6,
    "beta": 0.4
  },
  "decision": {
    "continue": boolean,
    "reason": "short explanation"
  },
  "feedback": {
    "summary": "1-2 sentences about layout and style",
    "count": "1-2 sentences about count and visibility",
    "edits": [
      {
        "type": "move|add|remove|resize|degrid",
        "targets": ["object_id_1", "object_id_2"],
        "hint": "concise instruction describing the change"
      }
      // at most 10 items
    ]
  }
}

GUIDELINES
- Prefer small, local edits to G: move a few instances,
  add or remove a few, or gently break grids.
- Do not propose major scene changes (no "redesign scene").
- Hints must be specific enough that Psi can apply them,
  but remain short and unambiguous.

CURRENT STATE:
  Pd = "<Pd>", N = <N>, s_c = <s_c>, s_a = <s_a>, G = <G>.
OUTPUT: JSON exactly following OUTPUT FORMAT.
```

Figure 16. Full executable Critic VLM prompt. This detailed scoring rubric replaces the simplified summary shown in the main paper. The Critic VLM consumes the generated image, detector/aesthetic scores, and the current planning graph, and outputs a scalar score and structured textual feedback that the textual refinement operator Ψ uses to update the planning graph.

References

[1] Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, et al. Pixtral 12b. *arXiv preprint arXiv:2410.07073*, 2024. 9

[2] Amirhossein Alimohammadi, Sauradip Nag, Saeid Asgari, Andrea Tagliasacchi, Ghassan Hamarneh, and Ali Mahdavi Amiri. Smite: Segment me in time. In *ICLR*, 2025. 5

[3] Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. Countgd: Multi-modal open-world counting. *NeurIPS*, 37: 48810–48837, 2024. 12, 14

[4] Ayan Banerjee, Nityanand Mathur, Josep Lladós, Umapada Pal, and Anjan Dutta. Svgcraft: Beyond single object text-to-svg synthesis with comprehensive canvas layout. *arXiv e-prints*, pages arXiv–2404, 2024. 2

[5] Jon Barron. Tweet about bias in gpt-4o image generation. https://x.com/jon_barron/status/1915828262326178145, 2025. "....4o has a bias towards putting things in the upper left of the image....." (Accessed: 2025-07-31). 2

[6] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*, 2(3):8, 2023. https://cdn.openai.com/papers/dall-e-3.pdf. 1

[7] Lital Binyamin, Yoad Tewel, et al. Make it count: Text-to-image generation with an accurate number of objects, 2024. arXiv:2406.10210. 1, 2, 3, 6, 7, 8, 9, 11

[8] Black-Forest-Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1, 2, 6, 8, 9

[9] Shuo Cao, Nan Ma, Jiayang Li, Xiaohui Li, Lihao Shao, Kaiwen Zhu, Yu Zhou, Yuandong Pu, Jiarui Wu, Jiaquan Wang, et al. Artimuse: Fine-grained image aesthetics assessment with joint scoring and expert-level understanding. *arXiv preprint arXiv:2507.14533*, 2025. 6, 10

[10] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. pages 1–10. ACM New York, NY, USA, 2023. 2, 3

[11] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *CVPR*, pages 14455–14465, 2024. 2

[12] Dongping Chen, Ruoxi Chen, Shu Pu, Zhaoyi Liu, Yanru Wu, Caixi Chen, Benlin Liu, Yue Huang, Yao Wan, Pan Zhou, et al. Interleaved scene graphs for interleaved text-and-image generation assessment. *arXiv preprint arXiv:2411.17188*, 2024. 3

[13] Junhao Cheng, Baiqiao Yin, Kaixin Cai, Minbin Huang, Hanhui Li, Yuxin He, Xi Lu, Yue Li, Yifei Li, Yuhao Cheng, et al. Theatergen: Character management with llm for consistent multi-turn image generation. *arXiv preprint arXiv:2404.18919*, 2024. 4

[14] Omer Dahary, Or Patashnik, Kfir Aberman, and Daniel Cohen-Or. Be yourself: Bounded attention for multi-subject text-to-image generation. In *ECCV*, pages 432–448, Berlin, Heidelberg, 2024. Springer, Springer-Verlag. 2, 3, 4, 5

[15] Omer Dahary, Yehonathan Cohen, Or Patashnik, Kfir Aberman, and Daniel Cohen-Or. Be decisive: Noise-induced layouts for multi-subject generation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025. 2, 4

[16] Adriano D'Alessandro, Ali Mahdavi-Amiri, and Ghassan Hamarneh. Afreeca: Annotation-free counting for all. In *ECCV*, pages 75–91. Springer, 2024. 12

[17] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *NeurIPS*, 36:18225–18250, 2023. 2, 3

[18] Paul Gavrikov, Wei Lin, M Jehanzeb Mirza, Soumya Jahagirdar, Muhammad Huzaifa, Sivan Doveh, Serena Yeung-Levy, James Glass, and Hilde Kuehne. Visualoverload: Probing visual understanding of vlms in really dense scenes. *arXiv preprint arXiv:2509.25339*, 2025. 2

[19] Xuyang Guo, Zekai Huang, Zhenmei Shi, Zhao Song, and Jiahao Zhang. Your vision-language model can't even count to 20: Exposing the failures of vlms in compositional counting, 2025. 6

[20] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 1

[21] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *NeurIPS*, 36: 78723–78747, 2023. 6, 11

[22] 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. 9

[23] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, pages 1219–1228, Salt Lake City, UT, USA, 2018. IEEE. 3

[24] Hyeonsu B Kang, Yuwei Bao, and Anjan Goswami. Vlm-slideeval: Evaluating vlms on structured comprehension and perturbation sensitivity in ppt. In *NeurIPS Workshop*, 2025. 2

[25] Wonjun Kang, Kevin Galim, Hyung Il Koo, and Nam Ik Cho. Counting guidance for high fidelity text-to-image synthesis. In *WACV*, pages 899–908, Tucson, AZ, USA, 2025. IEEE, Computer Society. 2, 6, 7, 8

[26] Hari Chandana Kuchibhotla, Sai Srinivas Kancheti, Abbavaram Gowtham Reddy, and Vineeth N Balasubramanian. Semantic alignment for prompt-tuning in vision language models. *TMLR*, 2025. 2

[27] Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. Evaluating and improving compositional text-to-visual generation. In *CVPR*, pages 5290–5301, Seattle, WA, USA, 2024. IEEE. 6

[28] Yuheng Li, Haotian Liu, Jianwei Yang, et al. Gligen: Open-set grounded text-to-image generation. In *CVPR*, pages 22511–22521, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 2, 3, 5, 9

[29] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *Trans. Mach. Learn. Res.*, 2024, 2023. 3, 6, 7, 8, 9, 10

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, Zurich, Switzerland, 2014. Springer, Cham. 6

[31] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *ECCV*, pages 366–384. Springer, 2024. 9

[32] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, pages 38–55, Milan, Italy, 2024. Springer. 2, 6, 9, 10, 14

[33] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *NeurIPS*, 36:46534–46594, 2023. 6

[34] Microsoft Research Blog. Introducing muse: Our first generative ai model designed for gameplay ideation. https://www.microsoft.com/en-us/research/blog/introducing-muse-our-first-generative-ai-model-designed-for-gameplay-ideation/, 2025. Accessed: 2025-11-05. 1

[35] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *NeurIPS*, 36:72983–73007, 2023. 6

[36] Anindya Mondal, Sauradip Nag, Xiatian Zhu, and Anjan Dutta. Omnicount: Multi-label object counting with semantic-geometric priors. In *AAAI*, pages 19537–19545, -, 2025. AAAI. 11, 12

[37] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *ICCV*, pages 3170–3180, 2023. 1

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[39] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. In *ICLR*, pages –, -, 2024. OpenReview.net. 1, 2, 5, 6, 8, 9

[40] Rahul Ramachandran, Ali Garjani, Roman Bachmann, Andrei Atanov, Oğuzhan Fatih Kar, and Amir Zamir. How well does gpt-4o understand vision? evaluating multimodal foundation models on standard computer vision tasks. *arXiv preprint arXiv:2507.01955*, 2025. 3

[41] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *CVPR*, pages 3394–3403, 2021. 1, 12

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *CVPR*, pages 10674–10685, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 2

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Lit, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, Red Hook, NY, USA, 2022. Curran Associates Inc. 2

[44] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *ECCV*, pages 87–103. Springer, 2024. 6, 8

[45] Zenglin Shi, Ying Sun, and Mengmi Zhang. Training-free object counting with prompts. In *WACV*, pages 323–331, 2024. 12

[46] Stability-AI. sd3.5. https://github.com/Stability-AI/sd3.5, 2025. 6, 8, 9

[47] Jiashuo Sun, Yi Luo, Yeyun Gong, Chen Lin, Yelong Shen, Jian Guo, and Nan Duan. Enhancing chain-of-thoughts prompting with iterative bootstrapping in large language models. *arXiv preprint arXiv:2304.11657*, 2023. 6

[48] Viacheslav Surkov, Chris Wendler, Mikhail Terekhov, Justin Deschenaux, Robert West, and Caglar Gulcehre. Unpacking sdxl turbo: Interpreting text-to-image models with sparse autoencoders. In *Mechanistic Interpretability for Vision at CVPR 2025 (Non-proceedings Track)*, 2025. 9

[49] Nikola Đukić, Alan Lukežič, Vitjan Zavrtanik, and Matej Kristan. A low-shot object counting network with iterative prototype adaptation. In *ICCV*, pages 18872–18881, 2023. 14

[50] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1, 13

[51] Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *NeurIPS*, 37:128374–128395, 2024. 2, 3, 6, 8, 9

[52] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, Qiong Yan, Xiongkuo Min, Guangtao Zhai, and Weisi Lin. Q-align: teaching lmms for visual scoring via discrete text-defined levels. In *ICML*, Vienna, Austria, 2024. JMLR.org. 2, 6, 9, 10

[53] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S Yu. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256. IEEE, 2023. 3

[54] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie,

and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, pages 7623–7633, 2023. 5

[55] Tsung-Han Wu, Long Lian, Joseph E Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting llm-controlled diffusion models, 2024. 2, 3, 6, 7, 8, 9

[56] Jing Xiong, Gongye Liu, Lun Huang, Chengyue Wu, Taiqiang Wu, Yao Mu, Yuan Yao, Hui Shen, Zhongwei Wan, Jinfa Huang, et al. Autoregressive models in vision: A survey. *arXiv preprint arXiv:2411.05902*, 2024. 2

[57] Zhiyuan Yan, Junyan Ye, Weijia Li, Zilong Huang, Shenghai Yuan, Xiangyang He, Kaiqing Lin, Jun He, Conghui He, and Li Yuan. Gpt-imgeval: A comprehensive benchmark for diagnosing gpt-4o in image generation. *arXiv preprint arXiv:2504.02782*, -(-):–, 2025. 6, 8

[58] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, -(-):–, 2025. 2, 3, 6, 9

[59] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin Cui. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *ICML*, 2024. 2, 3, 6, 8, 9

[60] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, -(-):–, 2023. 5, 9

[61] Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, et al. Echo-4o: Harnessing the power of gpt-4o synthetic images for improved image generation. *arXiv preprint arXiv:2508.09987*, 2025. 1

[62] Zhiyuan You, Kai Yang, Wenhan Luo, Xin Lu, Lei Cui, and Xinyi Le. Few-shot object counting with similarity-aware feature enhancement. In *WACV*, pages 6315–6324, 2023. 12

[63] Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616, 2025. 6

[64] Dewei Zhou, You Li, Fan Ma, Xiaoting Zhang, and Yi Yang. Migc: Multi-instance generation controller for text-to-image synthesis. In *CVPR*, pages 6818–6828, -, 2024. IEEE. 2, 6, 8, 9