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**

Reviewed on OpenReview: `https://openreview.net/forum?id=2JxXGhpCP4&invitationId=TMLR/Paper9259/`

Figure 1: Given prompts with explicit per-class counts, CountLoop produces images whose detected counts align with targets, even at extreme cardinalities (*e.g.*, 140 oranges and 31 birds), where competing methods suffer from count saturation, semantic leakage, and grid-like layouts. Unlike prior approaches, CountLoop requires no retraining: a VLM-guided planning graph structures the layout, instance-driven attention masking prevents attribute leakage, and a Critic VLM iteratively refines the scene until the count and quality criteria are met. Accurate count-faithful synthesis unlocks practical applications (right): (a) augmenting object-counting datasets (Ranjan et al. (2021)) with high-instance scenes, and (b) Compositional Image Generation in complex out-of-distribution scenarios for example it is rare to find peacock in the middle of traffic situation.

Abstract

Diffusion models excel at photorealistic synthesis but struggle with accurate object counts, especially in high-density settings. We introduce CountLoop, a training-free framework that achieves accurate instance control through iterative, structured feedback. Our method alternates between synthesis and evaluation: a VLM-based planner generates structured scene layouts, while a VLM-based critic provides explicit feedback on object counts, spatial arrangements, and visual quality to refine the layout iteratively. Instance-driven attention masking and cumulative attention composition further prevent semantic leakage, ensuring clear object separation even in densely occluded scenes. Evaluations on COCO-Count, T2I-CompBench, and two newly introduced high instance benchmarks show that Count-

LOOP reduces counting error by up to 57% and achieves the highest or comparable spatial quality scores across all benchmarks, while maintaining photorealism. Project page is at https://mondalanindya.github.io/CountLoop/.

1 Introduction

Digital creators, designers, and artists increasingly use text-to-image diffusion models like DALL-E 3 (Betker et al. (2023)), SDXL (Podell et al. (2024)), and FLUX (Black-Forest-Labs (2024)) to produce high-quality visuals. However, these models struggle with scenes containing many distinct yet related object instances (Paiss et al. (2023)), limiting their effectiveness in applications where cardinality is crucial, such as Compositional Image Generation and augmenting object-counting datasets. Current image diffusion models typically saturate at around 10 instances per category (Binyamin et al. (2024)), with accurate quantity being a known long-tail compositional failure (Ye et al. (2025)), 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 three categories: (1) text-to-image (T2I) models, sometimes augmented with gradient-based counting guidance (Kang et al. (2025); Chefer et al. (2023)); (2) layout-to-image (L2I) pipelines (Li et al. (2023); Feng et al. (2023); Binyamin et al. (2024); Zhou et al. (2024); Wang et al. (2024a); Zhou et al. (2025)); and (3) agentic diffusion frameworks (Wu et al. (2024b); Wang et al. (2024b); Yang et al. (2024); Wu et al. (2025)). 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 but often introduce artifacts or worsen semantic leakage as object density increases (Dahary et al. (2024; 2025)). Attribute or semantic leakage denotes the phenomenon where the cross-attention of the UNet lets the query tokens of one instance attend to features of other instances, causing attributes to bleed between objects. For example, in fig. 2(b), although the prompt asks for "79 Eggs", leakage distorts the cross-attention features and changes the egg textures to look like stones. L2I pipelines guide diffusion using bounding boxes or masks, but single-pass generation causes cross-attention leakage, and autoregressive layout biases (Xiong et al. (2024)) produce unnatural, grid-like arrangements (see fig. 2(a)). Agentic frameworks use LLM-based critique but lack explicit scene structure, leading to overcorrection or object omission, and their focus on aesthetics over spatial precision makes them unreliable for dense, count-sensitive generation. We present COUNTLOOP, a training-free framework that treats high-instance image generation as an iterative design process. Inspired by how human designers refine compositions, COUNTLOOP parses the input prompt into a planning graph encoding object attributes and spatial relationships, which guides layout-conditioned image synthesis. A VLM critic then evaluates (a) spatial coherence and appearance fidelity via a pretrained encoder (Wu et al. (2024a)), and (b) counting accuracy via an off-the-shelf detector — since VLMs alone struggle with precise counting in dense scenes (Gavrikov et al. (2025)). Critic feedback updates the planning graph and prompt, repeating until quality criteria are met.

Figure 2: Issues in High-instance image generation

Our COUNTLOOP also introduces a cumulative attention mechanism during the denoising process to mitigate semantic leakage (Dahary et al. (2024; 2025)), a common issue in high-instance scenes. Inspired by the multi-turn image generation (Cheng et al. (2024)), rather than generating all subjects simultaneously, it synthesises one instance at a time, providing per-instance grounding that prevents semantic entanglement and maintains the identity of individual objects. By imposing attention locality within instance-specific regions (Chefer et al. (2023); Dahary et al. (2024)), 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 training-free closed-loop pipeline. Unlike gradient-guided methods that introduce artifacts or L2I pipelines that produce

rigid layouts, CountLoop acts as a plug-and-play enhancement to standard diffusion backbones, scaling gracefully to dense, high-instance scenes while maintaining accurate counts and natural spatial arrangements. While CountLoop may not achieve perfect counts in every single instance, its outputs remain highly practical for downstream tasks like image classification where image quality is more important than exact quantity. Furthermore, for strict applications requiring perfect ground-truth, CountLoop serves as an exceptionally efficient generator. By drastically reducing the initial error rate (e.g., 43–48% on high-instance scenes), it increases the yield of perfectly counted, compositionally coherent images that can be easily isolated via a simple post-hoc verification filter, saving substantial computational resources compared to existing baselines.

Our contributions: **(1)** CountLoop, a training-free iterative pipeline for high-instance generation with accurate counts and strong aesthetics; **(2)** a cumulative attention mechanism that sequentially injects objects via instance-specific masks, mitigating semantic leakage and preserving identity in dense scenes; **(3)** a VLM critic that evaluates count consistency and appearance fidelity, providing structured feedback to iteratively refine layout and prompt; **(4)** evaluation on COCO-Count, T2I-CompBench, and two new high-instance benchmarks shows CountLoop reduces counting error by up to 57% on standard benchmarks and 43–48% on high-instance scenes, with the highest or comparable spatial quality across all four.

2 Related Work

Count Control in Text-to-Image Generation: Modern text-to-image diffusion models such as LDM (Rombach et al. (2022)), Imagen (Saharia et al. (2022)), SDXL (Podell et al. (2024)), and FLUX (Black-Forest-Labs (2024)) achieve high photorealistic fidelity through iterative denoising, but 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 (Chefer et al. (2023); Dahary et al. (2024); Binyamin et al. (2024)). 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 (Kang et al. (2025); Zeng et al. (2025)) offer partial remedies at inference time: Counting Guidance (Kang et al. (2025)) steers denoising via a regression-based counting network, while YOLO-Count (Zeng et al. (2025)) introduces a differentiable cardinality map for token-level optimisation, improving count. However, these methods treat counting as a global scalar constraint by optimising a signal that tells the model how many objects to produce but not where to place them or how to keep them visually distinct. As density grows, this leads to object merging and spatial collapse, fixing which requires explicit layout structure and per-instance attention control rather than stronger counting gradients.

Layout-to-Image Generation: Layout-to-image methods condition diffusion on boxes or masks (Li et al. (2023)), LLM-derived layouts (Lian et al. (2023); Feng et al. (2023); Zhao (2024); Anonymous (2024)), or per-instance conditioning signals such as instance-decomposed cross-attention with shading aggregation (Zhou et al. (2024); Alimohammadi et al. (2025)) and flexible bbox/point/scribble inputs (Wang et al. (2024a)). Scene-graph pipelines (Johnson et al. (2018)) encode pairwise relations but depend on expensive graph annotations. More recent approaches decouple layout planning from rendering via intermediate depth-map synthesis (Zhou et al. (2025)), improving spatial coherence across diverse backbones, while retrieval-based layout adaptation (Binyamin et al. (2024)) avoids manual annotation but depends on retrieval coverage and the downstream generator. However, shared limitations persist: single-pass generation causes cross-attention leakage and identity confusion as objects crowd together (Chefer et al. (2023); Dahary et al. (2024)), and the absence of closed-loop feedback means counting errors are irreversible. Robustness under high-instance prompts ($\gg$20) remains under-explored (Binyamin et al. (2024)).

Agentic Diffusion Correction: Recent frameworks employ LLM/VLM agents as planners or critics to iteratively refine diffusion generation. SLD (Wu et al. (2024b)) applies LLM-directed latent-space corrections (addition, deletion, repositioning) but lacks a persistent scene representation, limiting global spatial consistency. GenArtist (Wang et al. (2024b)) uses MLLM-driven tree planning over specialist tools, improving compositionality but not targeting high-instance count control. RPG-DiffusionMaster (Yang et al. (2024)) decomposes prompts via chain-of-thought and applies regional diffusion, but rectangular non-overlapping partitions preclude dense or occluded multi-instance layouts. Qwen-Image (Wu et al. (2025)) combines a VLM backbone with a diffusion transformer for strong semantic fidelity, yet lacks explicit layout

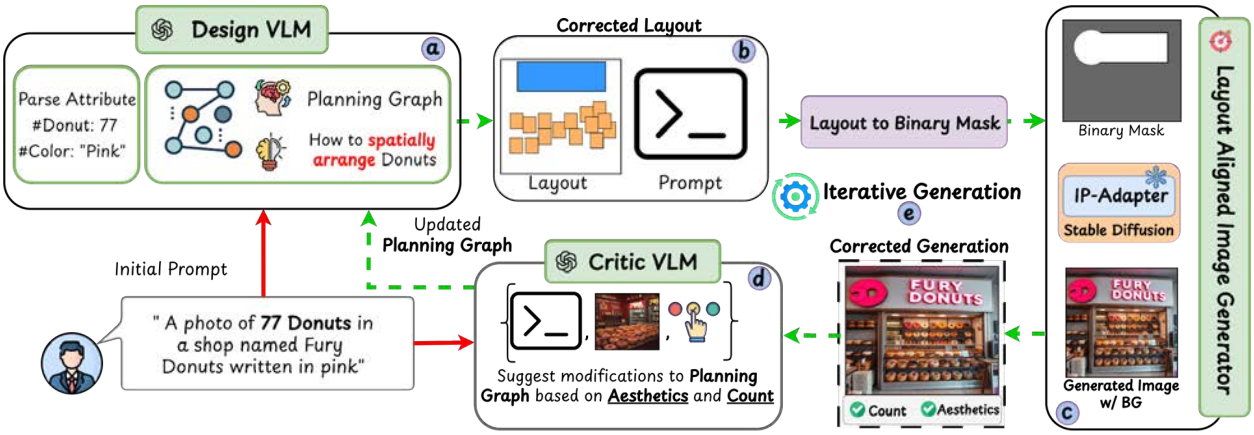


Figure 3: Given a text prompt, ⓐ The Design VLM parses the prompt to construct a planning graph, which is converted into a pixel-aligned layout ⓑ. ⓒ This layout guides an IP-Adapter-enhanced T2I backbone for image generation. ⓓ A Critic VLM evaluates the generated image's count and aesthetics, providing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

conditioning and iterative correction. Despite advances in compositional control, none maintain a structured, editable scene graph, reducing reliability when precise counting and spatial coherence are required at high instance densities.

3 CountLoop

Overview: CountLoop is a training-free, VLM-guided framework for high-instance image generation operating in three stages (see fig. 1). First, a Design VLM interprets the prompt to produce realistic, non-grid layouts (fig. 2(a)) with natural object placement. Second, a cumulative attention mechanism guides style-consistent generation, mitigating attribute leakage (fig. 2(b)) and preserving object clarity under overlap. Finally, a Critic VLM assesses counting accuracy and aesthetic quality, providing structured feedback to refine both layout and prompt.

3.1 VLM-Guided Layout Generation

Precise multi-instance layout generation remains challenging: LLM-based approaches (Lian et al. (2023)) suffer from limited spatial reasoning (Ramachandran et al. (2025)) and autoregressive bias, producing rigid grid-like structures (fig. 2(a)), while VLMs (Wu et al. (2023a)) offer richer multimodal reasoning but still lack sufficient spatial flexibility. We address this by augmenting VLM Chain-of-Thought with explicit relational and spatial priors via planning graphs (Chen et al. (2024)). Building on Qwen3-VL (Yang et al. (2025)), our Design VLM produces more consistent object placement, attributes, and relations, reducing grid artifacts and yielding 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 (*e.g.*, Qwen3-VL (Yang et al. (2025))) to analyze the prompt and return a JSON dictionary. Each node carries `id`, `category`, `pos [x,y]`, `size [w,h]`, `depth`, and `color`; edges encode `relation`, `dist`, and `angle`; a `context` field captures the background. These object-attribute relations serve as the foundation for the planning graph. The full prompt schema and a worked example are provided in the Supplementary.

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 $[x, y] \in [0, 1]^2$, size $[w_i, h_i]$, 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:

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

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, combined with in-context examples for grounding. Both P_G and the in-context examples (denoted by P_{icl}) are fed into the Design VLM as follows:

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

where $\mathbb{J}$ is the VLM's output in JSON format, from which we extract per-instance layouts $l_i = (x_i, y_i, w_i, h_i)$ forming the layout set $\mathbb{L} = \{l_1, \ldots, l_N\}$, the scene description prompt P_d, and background prompt P_{bg}.

3.2 Layout Aligned Image Generation

Given layouts $\mathbb{L}$, layout-grounded generation commonly exhibits attribute leakage (Dahary et al. (2024; 2025)), a phenomenon where UNet cross-attention allows query tokens of one instance to attend to features of other instances, causing attributes like color or texture to bleed between objects, yielding correct counts but degraded visual quality (fig. 2(b)); for instance, although the prompt requests "79 Eggs", they appear as stones due to this leakage. Inspired by multi-turn generation (Cheng et al. (2024)), we avoid synthesising all instances in a single pass.

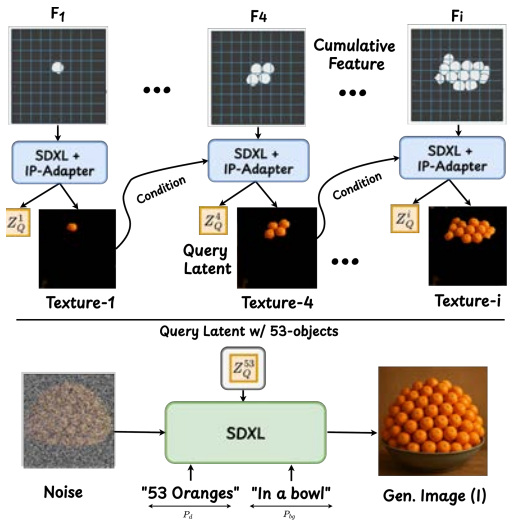


Figure 4: Cumulative latent composition, along with disentangled query feature extraction, mitigates attribute leakage.

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. Following (Lian et al. (2023)), we adapt GLIGEN(Li et al. (2023)) adapter (denoted by $\mathbb{E}$), which encodes each per-instance layout boxes $l_i \in \mathbb{L}$ into latent tokens $Q_i = \mathbb{E}(l_i)$ where $Q_i \in \mathbb{R}^D$. The full set of embeddings is represented as $Q = \{Q_1, \ldots, Q_N\}$ Since GLIGEN was built on top of SD(McLean (2023)), each layout token Q_i has same dimensions D as the intermediate latents of U-Net based diffusion models. These tokens are injected into the Diffusion U-Net via GLIGEN's gated self-attention mechanism in a training-free manner, conditioning the denoising process on the layouts. During denoising, the U-Net's cross-attention layers compute spatial attention between the latent feature map infused with the layout embeddings and the text embedding of P_d to obtain cross-attention feature A_{cross} thereby grounding the features with the textual description.

However, directly using A_{cross} for generation introduces semantic leakage (Dahary et al. (2024)) 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}.

To obtain shape-aware instance boundaries, we refine $\hat{M}_i$ via the self-segmentation of (Dahary et al. (2024)), partitioning the mask into foreground/background via k-means (k=2).

This produces a binary, shape-aware mask that tightly follows object contours rather than bounding-box boundaries. 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 spatial region, preventing feature mixing and semantic leakage across instances.

Cumulative Latent Composition: Once instance-level attention maps A^i_{mask} have been computed for each object layout $l_i \in L$, we build the global latent feature map F by sequentially placing each object's latent features in the diffusion latent space. We initialize $F_0 = 0$ as a zero tensor in $\mathbb{R}^{H_\ell \times W_\ell \times D}$ where H_ℓ, W_ℓ are the spatial feature dimensions and D is the fixed feature dimension. Hence for $i = 0, 1, \ldots$ we update

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

Here $\mathbb{1}_{(x,y)}$ is the binary indicator that pixel (x, y) lies within the spatial extent of l_i. In other words, for each pixel covered by layout l_i, we replace the previous feature with the new attention feature A^i_{mask}, and elsewhere we retain the existing feature. Because we never change the feature dimensionality in this process (each F_i and $A^i_{mask} \in \mathbb{R}^D$), the dimension D, remains fixed (e.g. $D = 1280$), ensuring compatibility with a frozen backbone. All object positioning, scale, and depth ordering are pre-specified by the layout l_i; we apply no further latent-space warping or scaling. In practice, instances are composed in order of increasing depth (Far $\rightarrow$ Near), so that each nearer object i overwrites any existing features in its mask.

The result is a composite latent feature map F that faithfully encodes each object's appearance, position, and scale according to the input layouts, with nearer objects occluding farther ones without any spurious blending.

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 (Podell et al. (2024))) on the foreground texture of the previously generated output using IP-Adapter (Ye et al. (2023)). Because leakage occurs when query tokens attend to different instances during self-attention (Dahary et al. (2024)), we further preserve the per-instance query representation (Z_q) before its interaction with keys and values, maintaining instance-level semantics. Formally:

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

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: $\mathbb{A}(Z^N_q, K, V)$, 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 (Wu et al. (2023b); Alimohammadi et al. (2025)), 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 image I, we run an iterative refinement loop that (i) evaluates I, (ii) identifies flaws, and (iii) updates the planning graph and prompt until count and aesthetic targets are met.

Critic VLM: We reconfigure a Qwen3-VL (Yang et al. (2025)) agent as a Critic VLM that analyses generated images and suggests layout revisions. Since LLM behaviour varies sharply with instruction design (Madaan et al. (2023); Sun et al. (2023)), the same model can serve as either creator or critic depending on the prompt. 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 counting (Guo et al. (2025)), the count signal inside the Critic VLM never comes from a VLM, but rather from an off-the-shelf detection model (e.g., GroundingDINO Liu et al. (2024) or CountGD++ Amini-Naieni and Zisserman (2026)), distinct from the evaluation detector (section 4.1). Aesthetic alignment s_a is scored by an external estimator (Wu et al. (2024a)). A composite score : $S = \alpha \cdot s_c + (1 - \alpha) \cdot s_a$, where $s_c, s_a \in [0, 1]$, is used to capture the overall quality of the generation (formulation details in the Supplementary). The Critic produces structured feedback in the form of text (denoted by P_{feed}) which is used to update the nodes and relations of the planning graph P_G, thereby altering the object layout size and spatial locations in the canvas.

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, we employ a parameter-free textual refinement operator inspired by (Yuksekgonul et al. (2025)). 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 optimisation prompt P_{opt}, the operator produces an updated graph: $G' = \Psi(G, P_{\text{feed}}, P_{\text{opt}})$. Mirroring how PyTorch's AutoGrad (Paszke et al. (2017)) performs backpropagation, instead of gradient tensors as feedback, the feedback here is in text form. Analogous to how backpropagation in PyTorch updates model weights, $\Psi(\cdot)$ edits (updates) the graph nodes and edges based on the textual feedback. $\Psi(\cdot)$ interprets the input feedback and estimates a textual analogue of a *gradient*, using a loss function defined as a pre-defined textual prompt template in P_{opt}. It then applies gradient-like edits to G via textual modifications rather than numerical parameter updates.

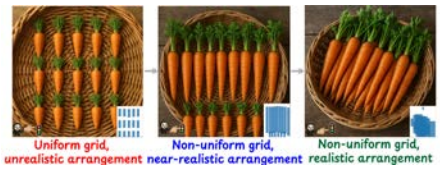


Figure 5: Successive layout refinement by Critic VLM. Layouts in the inset.

Operating entirely on textual representations, Ψ applies targeted structural edits to G. For example, if the feedback identifies overlapping objects, Ψ increases their spatial separation in G. If the feedback indicates a missing object count, Ψ inserts the required missing nodes. (A full multi-round generation case and explicit reasoning trace for this process is provided in the Supplementary.) This parameter-free refinement is compatible with any frozen diffusion model. After obtaining G', we derive $P_{G'}$ (eq. (1)) to generate a refined layout $\mathbb{L}$ (eq. (2)), followed by updated image synthesis I (see fig. 5). The process terminates when the composite score exceeds a quality threshold, *i.e.*, the detector confirms the correct count and the aesthetic score is acceptable, or after a fixed number of rounds to prevent diminishing returns from further refinement. In practice, the majority of prompts converge within three rounds (see convergence analysis in the Supplementary).

4 Experiments

4.1 Dataset and Evaluation

Datasets and Metric: We evaluate on four sets spanning instance count and compositional difficulty: COCO-Count (MS-COCO subset (Lin et al. (2014))); T2I-CompBench Count (subset of (Huang et al. (2023))); newly proposed CountLoop-S (single category, 200 prompts, 30–200 instances); and CountLoop-M (multi-category, 200 prompts, 30–200 instances). Benchmark construction details and prompt lists are in the supplementary. We report *counting accuracy* using MAE metric (Kang et al. (2025)) where OWLv2 (Minderer

Figure 6: CountLoop maintains precise object counts and natural arrangements in dense scenes, while methods like LMD (Lian et al. (2023)), SLD (Wu et al. (2024b)), Counting Guidance (Kang et al. (2025)), and CountGen (Binyamin et al. (2024)) exhibit abnormal counts, spatial collapse, and grid artifacts. More visuals in the supplementary.

et al. (2023)) is used as an evaluator for *all* methods, configured with a confidence threshold of 0.25 and class-agnostic NMS IoU of 0.5. Additionally, we report exact-count matches evaluated using the state-of-the-art counting model CountGD++ (Amini-Naieni and Zisserman (2026)). For counting convention, we count an object if it exceeds the minimum bounding box dimensions (3×3) and if its center is visible (determined via the objectness token in OWLv2). Any systematic detection bias at high densities therefore affects baselines and CountLoop equally, leaving relative MAE comparisons valid. Spatial alignment is measured via CLIP–FlanT5 encoder from VQAScore (Li et al. (2024)).

Competitors: We compare CountLoop with representative T2I (SDXL (Podell et al. (2024)), FLUX (Black-Forest-Labs (2024)), SDXL-Turbo (Sauer et al. (2024)), SD3.5 (Stability-AI (2025)), Counting Guidance (Kang et al. (2025))), Agentic (Qwen-Image (Wu et al. (2025)), GenArtist (Wang et al. (2024b)), SLD (Wu et al. (2024b)), RPG-DiffusionMaster (Yang et al. (2024))), and L2I (LMD (Lian et al. (2023)), MIGC (Zhou et al. (2024)), CountGen (Binyamin et al. (2024)), 3DIS (Zhou et al. (2025)), InstanceDiffusion (Wang et al. (2024a))) methods. Note that LMD and CountGen generate layouts with their own native modules. Box-conditioned methods without a native text-to-layout stage (MIGC, InstanceDiffusion, 3DIS) receive layouts from LMD's LLM layout generator, not from our planning graph. Thus, no baseline benefits from or is handicapped by our layout prior. Implementation details are provided in the supplementary.

4.2 Main Results

Quantitative Results: table 1 reports counting error (MAE, lower is better), exact-count match (Ex.↑), and spatial quality across all benchmarks. In the low-instance regime of COCO-Count, CountLoop achieves the lowest overall MAE (0.45), outperforming strong agentic competitors such as Qwen-Image (1.04) and SLD (1.15). These gains are modest in absolute terms because existing methods already perform well at low counts, where the failure modes CountLoop targets, like semantic leakage, layout rigidity, and count saturation, have not yet manifested. The critical differentiator emerges at scale: on CountLoop-S, CountLoop records a MAE of 7.59, less than half the error of the strongest agentic competitor (Qwen-Image: 17.30) and the best L2I baseline (3DIS: 14.55). Methods that perform competitively at low counts suffer clear collapse at scale: CountGen's MAE rises from 1.88 to 34.44 and SLD's from 1.15 to 29.65. Even recent baselines evaluated only on CountLoop-S (InstanceDiffusion: 16.07, Qwen-Image: 17.30) remain 2× above CountLoop. This robustness extends to multi-category scenes (CountLoop-M, MAE: 2.13). This capability is further

Table 1: **Counting and spatial quality across benchmarks.** We report counting error (MAE↓), exact-count match (Ex.↑), and spatial quality (Spatial↑) for single-category and multi-category prompts. Agentic baselines are run with their native self-correction loops fully enabled. **Best** in bold, second-best underlined.

Family	Model	Single Category						Multi Category					
		COCO-Count			T2I-CompBench			CountLoop-S			CountLoop-M		
		MAE↓	Ex.↑	Sp.↑	MAE↓	Ex.↑	Sp.↑	MAE↓	Ex.↑	Sp.↑	MAE↓	Ex.↑	Sp.↑
T2I	SDXL (Podell et al. (2024))	2.37	25%	0.38	2.72	25%	0.75	29.96	0%	0.63	9.89	10%	0.55
	FLUX (Black-Forest-Labs (2024))	1.40	55%	0.53	1.48	55%	0.78	17.47	4%	0.65	9.62	12%	0.58
	SD 3.5 (Stability-AI (2025))	1.10	55%	0.46	1.58	40%	0.76	21.81	0%	0.64	8.40	16%	0.56
	SDXL-Turbo (Sauer et al. (2024))	2.50	25%	0.23	3.76	10%	0.53	51.14	0%	0.39	9.95	10%	0.37
	Counting Guidance (Kang et al. (2025))	1.68	40%	0.63	3.90	10%	0.56	42.49	0%	0.47	8.43	16%	0.41
L2I	LMD (Lian et al. (2023))	3.09	10%	0.24	5.56	2%	0.73	16.62	5%	0.66	6.34	25%	0.64
	MIGC (Zhou et al. (2024))	1.83	40%	0.36	2.96	25%	0.65	17.54	4%	0.65	6.28	25%	0.62
	CountGen (Binyamin et al. (2024))	1.88	40%	0.61	5.22	2%	0.75	34.44	0%	0.72	6.46	24%	0.69
	InstanceDiffusion (Wang et al. (2024a))	1.77	40%	0.40	2.83	25%	0.68	16.07	5%	0.74	6.11	26%	0.66
	3DIS (Zhou et al. (2025))	1.56	40%	0.42	2.56	25%	0.70	14.55	6%	0.76	5.75	27%	0.69
Agentic	GenArtist (Wang et al. (2024b))	1.50	40%	0.45	1.50	40%	0.70	32.47	0%	0.60	4.93	11%	0.57
	SLD (Wu et al. (2024b))	1.15	55%	0.70	1.44	55%	0.77	29.65	0%	0.75	3.74	20%	0.65
	RPG (Yang et al. (2024))	1.28	55%	0.60	1.47	55%	0.75	31.85	0%	0.70	4.34	15%	0.62
	Qwen-Image (Wu et al. (2025))	1.04	55%	0.58	1.26	55%	0.77	17.30	4%	0.74	6.92	22%	0.66
Ours	**CountLoop (w/ GDINO critic)**	0.45	85%	0.93	1.23	55%	0.79	7.59	20%	0.93	2.13	33%	0.73
	CountLoop (w/ CountGD++ critic)	**0.32**	**85%**	**0.98**	**0.87**	**70%**	**0.95**	**5.33**	**29%**	**0.97**	**1.02**	**42%**	**0.85**

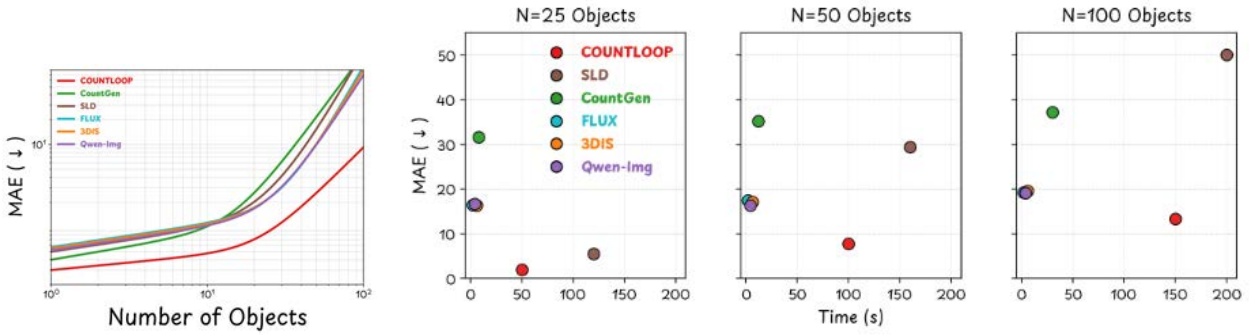
highlighted by the exact-count match (Ex.): CountLoop achieves 33% on CountLoop-S, outperforming the best baseline (3DIS) by nearly a factor of five (6%). Notably, CountLoop achieves this without sacrificing generation quality: its spatial score on CountLoop-S is 0.93 versus 0.75 (SLD) and 0.74 (Qwen-Image), demonstrating that the Critic VLM resolves the count-quality trade-off that constrains all prior paradigms. To provide a commercial reference, we evaluated a 100-prompt CountLoop-S subsample on Nano Banana Pro, which yielded an MAE of 19.06 and spatial score of 0.70 (compared to our 7.59/0.93), demonstrating that even commercial systems saturate in the 15–20 instance range.

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). CountLoop consistently avoids semantic drift, grid artifacts, and count inaccuracies that plague baselines, outperforming all competitors on high-instance scenes.

4.3 Ablations and Analysis

Key Components: table 2b progressively builds CountLoop from a plain baseline to quantify the contribution of each component on CountLoop-S. Both the Planning Graph (**PG**) and Cumulative Attention (**CA**) independently halve the counting error relative to the baseline (MAE: 29.91→ 14.98 and 14.39, respectively), confirming that structured prompt decomposition and leakage-free attention each address a distinct and roughly equal source of failure. As shown, layout grounding helps the model by 49% compared to mere text conditioning alone. Combining them (PG+CA: 11.27) yields further gains, outperforming standard single-pass L2I methods (e.g., 3DIS at 14.55) by structuring the prompt before generation. Iterative Refinement (**IR**) closes the remaining gap with a 33% additional MAE reduction (11.27 → 7.59), recovering instances that a single forward pass cannot place correctly. A random-retry baseline (best of 3 attempts with new seeds) only reaches 10.90, confirming that directed feedback—not merely repetition—drives these gains. Notably, pairing our IR critic with a standard generation backbone (simulated by dropping CA) causes errors to double (7.59 → 14.98), proving the critic alone is insufficient without instance-driven attention. We also show that beyond a critical instance count, the physical area per instance on a fixed-resolution canvas becomes too small for objects to remain visually distinguishable, which is a limit shared by all generation methods and that CountLoop reaches this floor at substantially higher N than competitors (fig. 7a).

Runtime Analysis: We evaluate end-to-end runtime on CountLoop-S by measuring MAE vs. wall-clock time at $N \in \{25, 50, 100\}$ (fig. 7b). One-shot baselines (FLUX (Black-Forest-Labs (2024)), Qwen-Image (Wu



(a) Accuracy vs object count. (b) Accuracy vs runtime.

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

Table 2: Analysis of CountLoop components, critic choices, and human evaluation. **(a)** Design-critic combinations on CountLoop-S; the default is in **bold** and the best per designer is underlined. **(b, c)** Ablations of the design components (**PG**: Planning Graph, **CA**: Cumulative Attn., **IR**: Iterative Refinement) and critic modules (**OVD**: Open-vocab Detector, **AS**: Aesthetic Scorer). **(d)** User scores on a 0–5 scale (higher is better).

(a) Design–critic configurations

Design	Critic	MAE↓	Spatial↑
Qwen3-VL (Yang et al. (2025))	Qwen3-VL	**7.59**	**0.93**
	Llava-1.5B	12.40	0.70
	Pixtral	11.83	0.72
Llava-1.5B (Lin et al. (2024))	Qwen3-VL	11.27	0.74
	Llava-1.5B	10.85	0.73
	Pixtral	10.51	0.75
Pixtral (Agrawal et al. (2024))	Qwen3-VL	10.18	0.76
	Llava-1.5B	11.05	0.72
	Pixtral	10.68	0.73

(b) Component ablation. *Random-retry.

PG	CA	IR	MAE↓	Spatial↑
✗	✗	✗	29.91	0.61
✓	✗	✗	14.98	0.68
✗	✓	✗	14.39	0.71
✓	✓	✗	11.27	0.81
✓	✗	✓	14.98	0.68
✓	✓	✗*	10.90	0.81
✓	✓	✓	**7.59**	**0.93**

(c) Critic VLM configs

OVD	AS	MAE↓	Spatial↑
✗	✗	29.59	0.67
✗	✓	15.49	0.70
✓	✗	9.27	0.83
✓	✓	**7.59**	**0.93**

(d) User evaluation

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

et al. (2025))) are fast (< 10 s) but incur high, irreducible error. L2I methods invest moderate compute yet plateau at comparable error. Among iterative methods, CountLoop dominates SLD (Wu et al. (2024b)) at every scale: at N=100, 3× lower error (MAE ≈ 13 vs. 50) in 25% less time (∼150 s vs. 200 s); at N=25, MAE ≈ 5 in ∼50 s while SLD needs ∼125 s yet plateaus at MAE ≈ 18. This mirrors fig. 7a, where competing methods plateau beyond ∼10–20 objects. We detail the exact per-stage runtime breakdown in the supplementary material.

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 under matched evaluation conditions. Results are in table 2a. The default Qwen3-VL (as design and critic role) achieves the best performance. However, even the weakest configuration (Qwen3-VL+Llava-1.5B) still outperforms the best competitor 3DIS(Zhou et al. (2025)) which is training based, demonstrating that

CountLoop is robust to VLM choice while successful in mutually guiding each other to generate plausible images.

Critic Composition: table 2c isolates each Critic component. A VLM-only critic yields the weakest performance (MAE 29.59), confirming VLMs struggle with dense counting (Guo et al. (2025)). AS alone substantially reduces MAE to 15.49 (−14.1), while OVD alone achieves a larger gain (MAE 9.27, −20.3), identifying it as the primary driver of count accuracy. Together they reach MAE 7.59, showcasing the importance of OVD and AS in guiding our critic-VLM.

Human Evaluation: We ran a 30-participant study (20 designers, 10 AI artists) across all four benchmarks. Each participant rated 15 blinded sets of 5 images (CountLoop, FLUX (Black-Forest-Labs (2024)), LMD (Lian et al. (2023)), SLD (Wu et al. (2024b)), and CountGen (Binyamin et al. (2024))) on a 5-point scale for Prompt Alignment, Aesthetic Quality, Count Accuracy, and Overall Preference. CountLoop was preferred across all axes (table 2d), with clear margins over its competitors. For our human count rating, we map each rating scale (0–5) to a 17% accuracy jump (e.g., 0 means 0% accuracy, 1 means 17% accuracy, etc.). Human count accuracy scores are consistent with OWLv2-reported trends across all methods, providing the strongest detector-agnostic validation available: if OWLv2 introduced systematic bias at high counts, human rankings would diverge from detector rankings; they do not. Procedure, demographics, and the survey interface are detailed in the supplementary.

Performance across different T2I backbones: To assess 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. table 3 reports counting 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. Counting performance remains stable (MAE $\leq$ 8.1) across all three backbones, indicating that CountLoop's instance-control mechanism is largely model-agnostic.

Table 3: Backbone swap.

Backbone	MAE↓	Spatial↑
SD v1.5	8.05	0.88
SD 3.5	7.44	0.90
SDXL	**7.59**	**0.93**

Robustness to Critic Components: The Critic VLM relies on two external modules: an open-vocabulary detector (OVD) for count verification and an aesthetic scorer (AS) for visual quality assessment. By default, we use the base GroundingDINO (Liu et al. (2024)) checkpoint as the OVD and Q-Align (Wu et al. (2024a)) as the AS across all experiments. GroundingDINO is selected for its strong performance on dense open-vocabulary detection, while Q-Align is chosen for its discrete text-defined level design, which produces stable scalar scores with lower variance than continuous VLM-based scoring, a critical property for a reliable termination signal in the iterative critic loop. This stability advantage is consistent with recent findings (Cao et al. (2025)). The critic detector is intentionally distinct from the evaluation detector (OWLv2 (Minderer et al. (2023))) to prevent the critic from optimising directly against the evaluation metric; we fix the GroundingDINO confidence threshold to 0.3 across all experiments for reproducibility.

To verify that CountLoop's gains are driven by the iterative loop architecture rather than a specific component pairing, we swap each module independently (table 4). Replacing GroundingDINO with OWLv2 in the critic role introduces critic–evaluator overlap that gives OWLv2 an inherent advantage, yet MAE remains comparable to the default. Replacing Q-Align with ImageReward (Xu et al. (2023)) increases MAE modestly but still substantially outperforms the best external baseline (3DIS: 14.55). Even the weakest configuration in the table beats all published baselines by a wide margin, confirming that Count-

Table 4: Component swap on CountLoop-S. Evaluation detector is OWLv2 throughout. †OWLv2 as both critic and evaluator gives an inherent advantage, yet MAE remains comparable.

Critic OVD	Aesthetic	MAE↓	vs Best Baseline
GroundingDINO	Q-Align (default)	**7.59**	+48%
OWLv2†	Q-Align	8.57	+41%
GroundingDINO	ImageReward	9.21	+36%

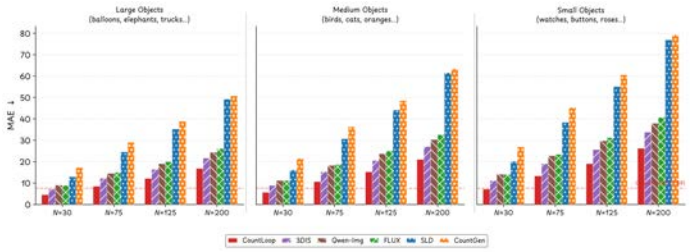


Figure 8: **Per-regime MAE at representative instance counts on CountLoop-S.** Categories are grouped by object size (Large / Medium / Small); bars show MAE at $N \in \{30, 75, 125, 200\}$ for all six methods. The Large < Medium < Small ordering is method-agnostic, reflecting canvas-density pressure and OWLv2 precision loss on small, densely-packed instances. CountLoop (red) achieves the lowest MAE in every regime at every N; the gap over next-best 3DIS widens with N, reaching $\Delta\text{MAE} \approx 7.6$ at $N = 200$ in the small-object regime (87% vs. 83% count accuracy). The dashed reference line marks CountLoop's overall MAE on CountLoop-S.

Loop is robust to component choice. Notably, the †OWLv2-as-critic row is an implicit cross-detector check: despite critic–evaluator overlap giving OWLv2 an inherent advantage, MAE remains comparable to the GroundingDINO default, confirming gains are not a detector artefact.

Instance Count Scalability by Object Regime: fig. 8 disaggregates the MAE-N relationship by object-size regime, grouping categories into *large* (balloons, elephants, trucks), *medium* (birds, cats, oranges), and *small* (watches, buttons, roses) classes, directly addressing the question of whether CountLoop has a practical upper bound on reliable instance generation.

Across all six methods, MAE increases monotonically with N, consistent with the aggregate trend. The rank ordering Large < Medium < Small is *method-agnostic*: it reflects two compounding factors independent of generation strategy. First, small objects occupy a larger fraction of the canvas at high instance counts, intensifying cross-attention identity confusion during diffusion sampling. Second, OWLv2 localisation precision degrades on densely packed, sub-pixel instances, introducing systematic upward bias in the detector-based MAE estimate for all methods equally. Since both factors are architectural constants of the evaluation setup rather than properties of any single method, the tier ordering cannot be attributed to CountLoop's design, and the shared bias does not inflate CountLoop's relative advantage, as it affects every method in the comparison identically.

Critically, CountLoop maintains the lowest MAE in every regime at every N, with the advantage *widening* rather than narrowing at high density: the gap over next-best 3DIS reaches $\Delta\text{MAE} \approx 7.6$ in the small-object regime at $N = 200$. CountLoop achieves 87% count accuracy in the hardest setting (small, $N = 200$) and 92% in the large-object regime, well beyond all prior methods. The performance ceiling is *category-dependent* and scales with object size; CountLoop raises it substantially across all regimes, with its advantage widening at the benchmark ceiling of $N = 200$.

5 Conclusion

We presented CountLoop, a training-free, iterative framework for high-instance image generation with accurate object counts and strong visual quality. VLM-based planning graphs, instance-driven attention, and cumulative attention composition overcome count saturation, semantic leakage, and rigid layouts, with a critic-in-the-loop refining generation. On COCO-Count, T2I-CompBench, and new high-instance benchmarks, CountLoop reduces counting error by up to 57% on standard benchmarks and 43–48% on high-instance scenes, achieving the highest or comparable spatial quality throughout.

Figure 9: Failure cases

Limitations: As a training-free system, CountLoop inherits
the limitations of its frozen VLM and detector. Dense occlusions, especially in human scenes, can degrade attention quality and spatial consistency. Lacking explicit 3D priors, CountLoop struggles with generating objects in different poses and complex perspectives. Moreover, strong layout guidance can reduce intra-class diversity by biasing toward canonical poses for count accuracy. Some of these limitations are shown in fig. 9.

References

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.

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

Niki Amini-Naieni and Andrew Zisserman. Countgd++: Generalized prompting for open-world counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 37725–37734, 2026.

Anonymous. Theatron: Layout generation, 2024.

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`.

Lital Binyamin, Yoad Tewel, et al. Make it count: Text-to-image generation with an accurate number of objects, 2024. arXiv:2406.10210.

Black-Forest-Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

Deanna McLean. Stable diffusion: A guide to dreamstudio. *Elegant Themes Blog*, 2023.

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

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.

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.

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.

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.

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

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.

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.

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

Stability-AI. sd3.5. `https://github.com/Stability-AI/sd3.5`, 2025.

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.

Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. InstanceDiffusion: Instance-Level Control for Image Generation . In *CVPR*, pages 6232–6242, Los Alamitos, CA, USA, 2024a. IEEE Computer Society.

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, 2024b.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

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, 2024a. JMLR.org.

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, 2023a.

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, 2023b.

Tsung-Han Wu, Long Lian, Joseph E Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting llm-controlled diffusion models, 2024b.

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.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *NeurIPS*, 36: 15903–15935, 2023.

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.

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.

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.

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.

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.

Guanning Zeng, Xiang Zhang, Zirui Wang, Haiyang Xu, Zeyuan Chen, Bingnan Li, and Zhuowen Tu. Yolo-count: Differentiable object counting for text-to-image generation. In *ICCV*, pages 16765–16775, 2025.

Lvmin Zhao. Omost: Your image is almost there. `https://github.com/lllyasviel/Omost`, 2024.

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.

Dewei Zhou, Ji Xie, Zongxin Yang, and Yi Yang. 3dis: Depth-driven decoupled image synthesis for universal multi-instance generation. In *ICLR*, 2025.

Supplementary Material for 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**

Reviewed on OpenReview: `https://openreview.net/forum?id=2JxXGhpCP4&invitationId=TMLR/Paper9259/`

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 (Lian et al. (2023)), SLD (Wu et al. (2024b)), CountGen (Binyamin et al. (2024)), MIGC (Zhou et al. (2024)), GenArtist (Wang et al. (2024b)), RPG-DiffusionMaster (Yang et al. (2024)), 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 (Li et al. (2023)) layout encoder (box+text mode), and cross-instance texture consistency was enforced using the IP-Adapter (public checkpoint from (Ye et al. (2023))). 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 (MAE/Spatial) were computed at the original resolution to avoid any bias.

T2I baselines: For FLUX (Black-Forest-Labs (2024)), 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. SDXL (Podell et al. (2024)), SD 3.5 (Stability-AI (2025)), and SDXL-Turbo (Surkov et al. (2025)) are all run using their official pipelines with default guidance scales, VAE settings, and sampling schedules. For Counting Guidance (Kang et al. (2025)), we use the authors' released code with the default SDXL backbone and counting loss weights. Across all T2I baselines, only the text prompt is changed; all model-specific system prompts and hyperparameters remain untouched.

L2I baselines: For LMD (Lian et al. (2023)), 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 (Zhou et al. (2024)) and CountGen (Binyamin et al. (2024)) are run with their released code, pre-trained diffusion

Figure 1: Spatial reasoning in image generation. Vanilla LLM (LMD (Lian et al. (2023))) fails to identify directions.

backbones, and unmodified layout encoders. For
InstanceDiffusion (Wang et al. (2024a)), we use the official per-instance conditioning pipeline (bbox mode) with the released SDXL checkpoint and default guidance scale. For 3DIS (Zhou et al. (2025)), we run the authors' depth-conditioned generation pipeline with the released monocular depth estimator and default diffusion backbone. Across all L2I baselines, we preserve the authors' layout formats, conditioning methods, and refinement logic without any tuning.

Agentic baselines: For SLD (Wu et al. (2024b)), 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 (Wang et al. (2024b)), 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 (Yang et al. (2024)), 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. For Qwen-Image (Wu et al. (2025)), we use the publicly released model with its default VLM backbone and diffusion decoder, generating images at 1024×1024 resolution with the authors' recommended inference settings; only the text prompt is substituted. Across all agentic baselines, we avoid tuning hyperparameters or increasing the number of refinement rounds, ensuring a fair comparison with CountLoop .

CountLoop configuration: Both the Design and Critic VLMs in CountLoop were instantiated from the Qwen3-VL (Yang et al. (2025)) 8B variant. We used the base variant of GroundingDINO (Liu et al. (2024)) as the detector guide for the Critic and the pretrained image encoder Q-Align (Wu et al. (2024a)) as the aesthetic guide. The composite score weight was set to $\alpha = 0.6$, giving a count accuracy weight of 0.6 and an aesthetic weight of $(1 - \alpha) = 0.4$, with a GroundingDINO confidence threshold of 0.3. The loop terminates as soon as the composite score $S \geq \tau$ (*early stopping*), or after K refinement rounds, whichever comes first (τ=0.85, K=3 for all main-paper experiments). 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. Code, prompts, planning-graph templates, and both benchmarks (CountLoop-S, CountLoop-M) will be released upon acceptance.

Cumulative Attention Composition implementation details:

Layout encoding and injection. The layout encoder $\mathbb{E}$ is the grounding tokeniser of GLIGEN (Li et al. (2023)). Each instance layout $l_i = (t_i, b_i) \in \mathbb{L}$, comprising noun phrase t_i and bounding box b_i, is encoded into a *1-D* grounding token $g_i = \mathbb{E}(t_i, b_i) \in \mathbb{R}^d$; the full set is $G = \{g_1, \ldots, g_N\}$. These tokens are injected into the U-Net via GLIGEN's *gated self-attention* mechanism as a layout-conditioning pathway that is structurally separate from text-conditioned cross-attention. This distinction is important for resolving apparent dimensionality ambiguity: the cross-attention query is the spatial latent feature map (not the 1-D grounding tokens), and its output is always a spatial tensor of the same resolution as the latent.

Cross-attention and mask dimensionality. At each U-Net layer, text-conditioned cross-attention is computed between the spatial latent $z \in \mathbb{R}^{h \times w \times d}$ and the text embedding sequence $T \in \mathbb{R}^{L \times d}$ produced by the text encoder from P_d:

$$A_{\text{cross}} = \text{softmax}\left(\frac{(z\,W_Q)\,(T\,W_K)^\top}{\sqrt{d_k}}\right) T\,W_V \;\in\; \mathbb{R}^{h \times w \times d}, \tag{1}$$

where W_Q, W_K, W_V are learned projection matrices. The result is a *spatially-resolved* feature map matching the latent resolution $h \times w$ at each block, never a set of 1-D vectors.

For each instance i, a binary spatial mask $M_i \in \{0,1\}^{W \times H}$ (1 inside bounding box b_i, 0 elsewhere) is resized via bilinear interpolation to $\hat{M}_i \in \{0,1\}^{h \times w}$ to match the latent resolution of A_{cross}. When applied, $\hat{M}_i$ is broadcast along the channel dimension as $\hat{M}_i \in \{0,1\}^{h \times w \times 1}$, giving:

$$A^i_{\text{mask}} = A^i_{\text{cross}} \odot \hat{M}_i \;\in\; \mathbb{R}^{h \times w \times d}. \tag{2}$$

All spatial dimensions $h \times w$ are consistent across the operation; the mask broadcast is scalar-only in the channel axis.

Self-segmentation block scope. Shape-aware mask refinement follows (Dahary et al. (2024)): self-attention maps are averaged across heads and layers at the U-Net's *middle block and first up-block only*, which yield more noise-robust representations than earlier or later blocks (Dahary et al. (2024)), and partitioned via k-means (k=2) into foreground/background clusters. Each cluster is then labelled by computing the Intersection-over-Minimum (IoM) between the cluster mask and the per-instance cross-attention map of the corresponding noun token, assigning each spatial region to the subject with the highest IoM score. This produces a binary, shape-aware mask that tightly follows object contours rather than bounding-box boundaries.

Query extraction and cumulative composition. The per-instance query representation Z_q^i is extracted from the self-attention layers of the denoising U-Net immediately before the query tokens interact with keys and values, *i.e.*, after the linear projection W_Q but before the $\text{softmax}(QK^\top/\sqrt{d})V$ computation, thereby preserving instance-level semantics without altering the denoising trajectory.

The cumulative feature map F is built by the binary indicator accumulation of Eq. 4 (main paper): starting from $F_0 = \mathbf{0} \in \mathbb{R}^{h\times w\times d}$, for each instance in Far→Near depth order:

$$F_{i+1}(x,y) = \mathbb{1}_{(x,y)\in l_i} \cdot A^i_{\text{mask}} + \big(1 - \mathbb{1}_{(x,y)\in l_i}\big) \cdot F_i. \tag{3}$$

For every pixel covered by layout l_i the existing feature is replaced by A^i_{mask}; all other pixels retain their current value. The feature dimensionality D remains fixed throughout (D=1280 for SDXL), ensuring full compatibility with the frozen backbone without any architectural modification or latent-space warping. Because the cumulative feature map F is consumed by a full denoising pass that attends over shared keys and values (Eq. (5)), the diffusion process naturally harmonises boundary transitions between adjacent instances; no explicit post-hoc blending or seam regularisation is required.

IP-Adapter and final composition pass. Each instance i+1 is generated from F_{i+1} via:

$$I_{i+1},\, Z_q^{i+1} = \Phi\big(F_{i+1},\, P_d,\, \theta(I_i)\big), \quad i = 1, \ldots, N{-}1, \tag{4}$$

where θ is IP-Adapter conditioning on the previously generated image I_i. The first image I_1 is generated without an IP-Adapter due to the absence of prior texture. After all N passes, the *last* query latent Z_q^N, which encodes all N objects with consistent accumulated appearance, drives the final composition pass:

$$\mathbb{A}(Z_q^N,\, K,\, V), \tag{5}$$

where K and V are the shared key-value matrices of the frozen backbone, and the textual condition is the concatenation of the scene description prompt P_d and background prompt P_{bg} (main paper, Sec. 3.2). Each object-specific feature in Z_q^N attends to this shared key-value set, enforcing semantic coherence across all foreground instances while keeping the background disentangled. Full layer indices and denoising schedules will be released with the code.

Critic VLM design rationale: LLM behaviour varies sharply with instruction design (Madaan et al. (2023); Sun et al. (2023)): the same model can function as either creator or critic based on the prompt, integrating creator/critique signals into its chain-of-thought reasoning. We exploit this by supplying a critique-style prompt P_{crit} (Fig. Fig. 9) that evaluates count fidelity via an open-vocabulary detector (Liu et al. (2024)) and visual aesthetics via Q-Align (Wu et al. (2024a)). The Critic's output is restricted to a closed edit vocabulary (`move` | `add` | `remove` | `resize` | `degrid`); since no aesthetic edit type exists, s_a influences termination via $S \geq \tau$ but cannot propagate into planning-graph updates, eliminating cognitive conflict by design (see also Section 2 for the full edit-space definition). 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. 8 (Design VLM) and Fig. 9 (Critic VLM).

2 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.

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

Input : Prompt p (class c, target count 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); count-accuracy weight α; quality threshold τ; max rounds K.

Output : Image $I^\star$ with score $S \geq \tau$.

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 depth-ordered (Far→Near) instance composition order.

Set $G \leftarrow G_0$; $k \leftarrow 0$.

Iterative Synthesize-Critique-Refine.

`/* Early stopping: terminate as soon as` $S \geq \tau$. K `is a hard cap and computational safeguard only (`τ`=0.85, `K`=3 by default). */`

repeat

Synthesize (cumulative, instance-aware generation).

Encode per-instance layouts $l_i \in \mathbb{L}$ with $\mathbb{E}$ to obtain Q_i; apply layout-aligned attention masks $A^i_{\text{mask}} = A^i_{\text{cross}} \odot \hat{M}_i$ and accumulate cumulative features F : replace features within l_i with A^i_{mask}, retain elsewhere.

Generate instances sequentially with IP-Adapter: $I_{i+1} = \Phi(F_{i+1}, P_d, \theta(I_i))$; compose final image I with background inpainting using P_{bg}.

Critique (count and aesthetics).

Run OVD on I to obtain normalized count score $s_c = \max(0, 1 - \frac{|\hat{c} - c_{gt}|}{c_{gt}})$, where $\hat{c}$ is the raw detector count and c_{gt} is the target count from the prompt ; compute $s_a = \text{AS}(I)$.

Compute composite score:

$$S = \alpha \cdot s_c + (1 - \alpha) \cdot s_a$$

Obtain structured feedback $P_{\text{feed}} = V_{\text{crit}}(I, P_d, P_{\text{crit}}, S, s_c, s_a)$ restricted to edit vocabulary `move`|`add`|`remove`|`resize`|`degrid`.

Refine (parameter-free textual operator Ψ).

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

Rebuild P_G and layouts $\mathbb{L}$ from G'; set $G \leftarrow G'$; $k \leftarrow k + 1$.

until $S \geq \tau$ ***or*** $k \geq K$

return $I^\star \leftarrow I$

Table 1: **Exact per-stage runtime breakdown** (mean over CountLoop-S prompts at each N, single A100 GPU).

Stage	$N = 25$	$N = 50$	$N = 100$
Design VLM planning (parse → graph → layout JSON)	6.8 s	7.6 s	8.8 s
GLIGEN layout encoding	4.5 s	4.5 s	4.5 s
Per-instance cumulative composition (IP-Adapter)	22.5 s (0.90 s/inst)	45.5 s (0.91 s/inst)	92.0 s (0.92 s/inst)
Final composition pass (SDXL)	6.5 s	6.5 s	6.5 s
Critic scoring: GroundingDINO + Aesthetic + Critic VLM	6.9 s	7.1 s	7.4 s
Round-1 subtotal	**47.2 s**	**71.2 s**	**119.2 s**
Graph refinement (Text Edit VLM) per round (3 rounds total)	2.7 s	4.5 s	7.8 s
Fig. 7b plotted value	**≈50 s**	**≈85 s**	**≈150 s**

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

$$S = \alpha \cdot s_c + (1 - \alpha) \cdot s_a, \tag{6}$$

where

$$s_c = \max(0, 1 - \frac{|\hat{c} - c_{gt}|}{c_{gt}})$$

is the normalized count score ($\hat{c}$ being the predicted count and c_{gt} being the target count), and $s_a \in [0, 1]$ is the aesthetic score. The weight α and quality threshold ($\tau = 0.85$) are fixed across all experiments, and sensitivity to τ is low because the count-match condition $\hat{c} = c_{gt}$ (*i.e.*, $s_c = 1$) is the binding constraint in practice. Once the detector confirms the target count, $S \geq \tau$ is typically satisfied simultaneously.

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

- *Local position updates:* nudging a node to reduce overlaps or break grid patterns (small $\Delta x, \Delta y$ in normalized coordinates).
- *Count corrections:* adding new nodes when $\hat{c} < c_{gt}$ in free regions, or slightly shrinking/moving nodes when heavy overlap causes under-detection.

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. The enumerated edit types constitute the complete output vocabulary available to the Critic VLM; no aesthetic edit type exists in this schema. Consequently, even when S is low due to poor s_a, the Critic cannot generate actionable aesthetic corrections and s_a enters only the termination condition $S \geq \tau$ and never propagates into graph edits, eliminating cognitive conflict by architectural constraint.

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. In practice, the combination of bounded per-iteration displacements (≤ 0.08 in normalized coordinates), a monotonically informative count signal from the OVD, and the K=3 hard cap prevents oscillatory behaviour; across all prompts in CountLoop-S, we observed no cases of composite-score regression between consecutive rounds.

Termination: At each iteration, we recompute (s_c, s_a, S) from the new image. The loop terminates as soon as $S \geq \tau$ (*early stopping*), and if S does not reach τ, a hard cap of K iterations prevents diminishing returns from further refinement, while also bounding compute. Fig. 2 shows the convergence trajectory with the iteration cap raised to K=5 to reveal the full saturation curve: MAE falls from 11.27 at iteration 1 to 9.72 at iteration 2 and 7.59 at iteration 3, after which further rounds yield negligible improvement. Under the default cap of K=3, all prompts terminate within three rounds across CountLoop-S: 60% converge by round 1, a further 30% by round 2, and the remaining 10% are resolved by the hard cap, confirming that the iteration budget is rarely the binding constraint (τ=0.85, K=3, see Section 1).

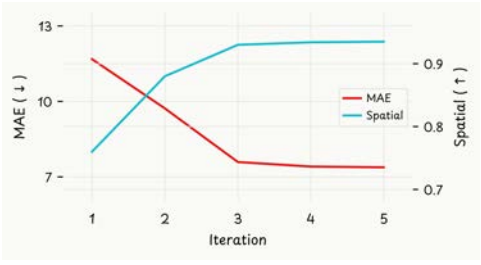


Figure 2: Convergence trajectory with the iteration cap raised to K=5 to reveal the full saturation curve. Under the default K=3, all prompts in CountLoop-S terminate within three rounds.

3 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 (Huang et al. (2023)) and COCO-Count (Binyamin et al. (2024)), 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 (Mondal et al. (2025)). 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 48 birds and 30 dogs"*), enabling assessment of compositional fidelity at scale. Representative generations are shown in Fig. 6; further qualitative examples are provided below.

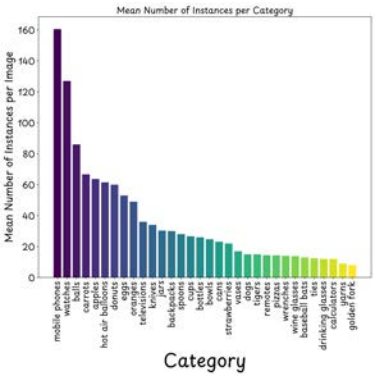


Figure 3: Statistics (instance per image vs category) for the CountLoop-S benchmark.

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:** CountLoop-S covers total instance counts from 30 to 200 per image; CountLoop-M covers 30-200 total instances split across multiple categories (individual per-class counts may be as low as 1).
- **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. 3.

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

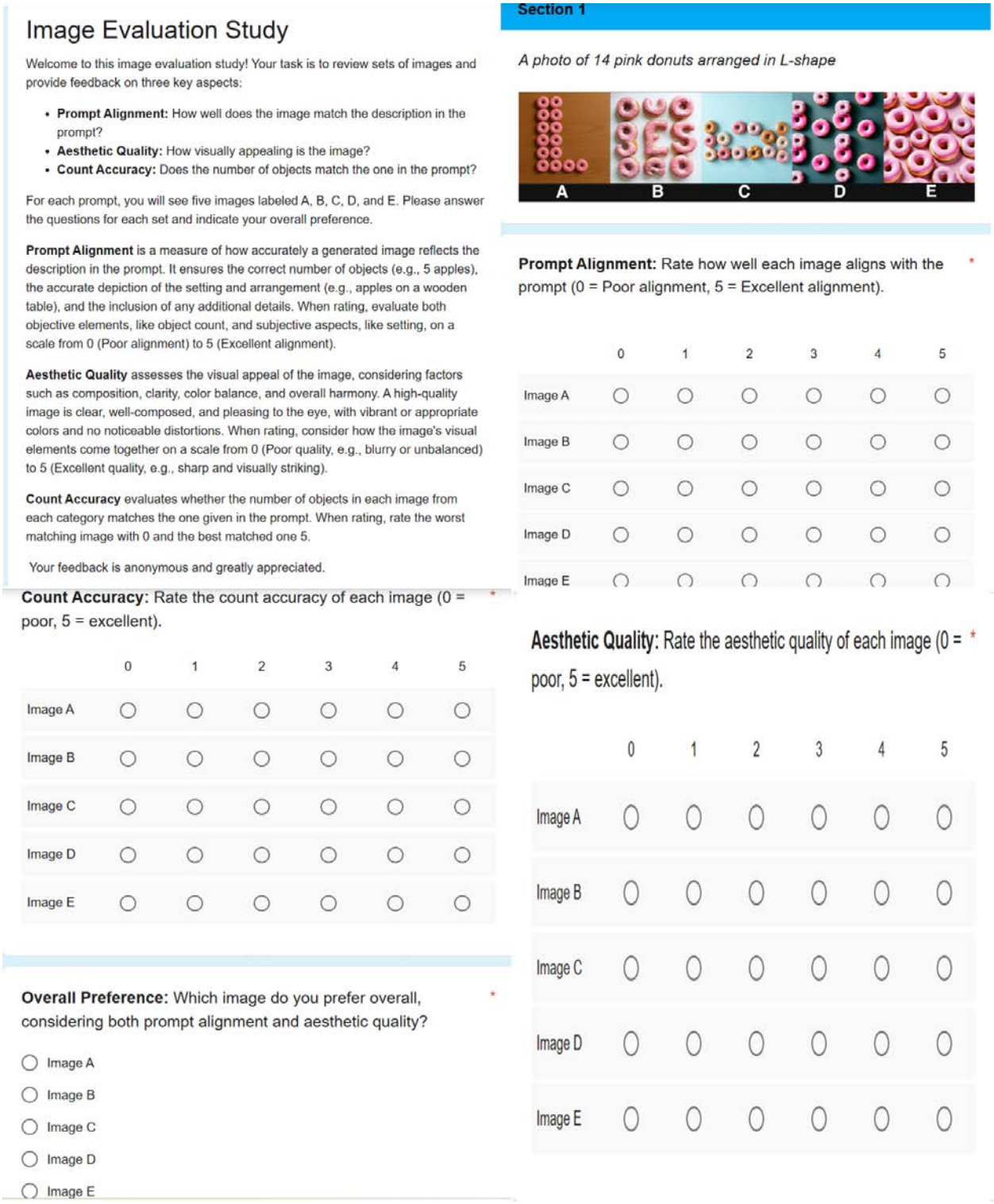


Figure 4: Human evaluation platform interface

4 Extended Ablations and Detailed Metrics

In this section, we provide further analysis of CountLoop's hyperparameters and exact-match performance. First, Table 2 demonstrates CountLoop's robustness to varying the GroundingDINO confidence threshold. Across a wide range of thresholds (0.15 to 0.35), CountLoop consistently maintains lower MAE than the strongest baselines. Second, in Table 3, we provide a detailed breakdown of the exact-count match and tolerance accuracy (±5%, ±10%) across different count tiers for CountLoop and the best baseline, 3DIS. CountLoop significantly outperforms 3DIS in exact matches and strict tolerance across all difficulty tiers.

Table 2: **Threshold sensitivity sweep on CountLoop-S (MAE↓).**

Model / Threshold	**0.15**	**0.20**	**0.25**	**0.30**	**0.35**
CountLoop	9.2	8.1	7.59	7.9	8.8
3DIS	16.4	15.1	14.55	15.0	16.1
SLD	31.5	30.2	29.65	30.1	31.2

Table 3: **Per-tier exact-match and tolerance results on CountLoop-S split.**

Tier / Method	**MAE↓**	**Exact↑**	±5%↑	±10%↑
CountLoop (30–60)	3.6	38%	79%	93%
CountLoop (61–120)	7.1	16%	61%	84%
CountLoop (121–200)	12.8	5%	47%	71%
3DIS (best baseline, 30–60)	8.2	12%	41%	58%
3DIS (61–120)	14.9	3%	26%	39%
3DIS (121–200)	24.1	0%	13%	22%

5 Potential Usecases

CountLoop allows for high-instance, count-faithful scene generation while adhering to explicit numeric constraints. This capability is particularly valuable for synthesizing large-scale, accurately labeled datasets where manual annotation is time-consuming and expensive, and for rendering complex multi-object scenes. 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. We highlight two representative use cases below.

Compositional Image Generation: Beyond data augmentation, the ability to control exact object counts while maintaining spatial and semantic coherence directly enables complex Compositional Image Generation (as illustrated in Figure 1 of the main paper). This requires models to synthesize scenes with challenging, out-of-distribution subject combinations and precise cardinalities—for example, generating a peacock amidst traffic or 140 oranges alongside 31 birds. Conventional diffusion models often suffer from attribute leakage or drop objects entirely in such dense, multi-category edge cases. We empirically validated CountLoop's capability on this task in the main paper (Table 1), where it achieves state-of-the-art exact-match and spatial fidelity scores on the CountLoop-M benchmark, demonstrating robust compositional alignment without retraining.

Data Augmentation for object counting models: Object counting (You et al. (2023); Ranjan et al. (2021); D'Alessandro et al. (2024); Mondal et al. (2025)) 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 (Mondal et al. (2025); Shi et al. (2024)). 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 (Ranjan et al. (2021)) 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 (Amini-Naieni et al. (2024)), 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.

Table 4: **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 (Liu et al. (2024))	Text	None	54.45	137.12	54.16	157.87	–
LOCA (Đukić et al. (2023))	Exemplar	None	10.24	32.56	10.79	56.97	–
CountGD (Amini-Naieni et al. (2024))	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.

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. 5) and a generation gallery (Fig. 6). The visual results provide compelling evidence of CountLoop's effectiveness in high-instance generation against SoTA models, under both single and multiple category scenarios.

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. 7 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.

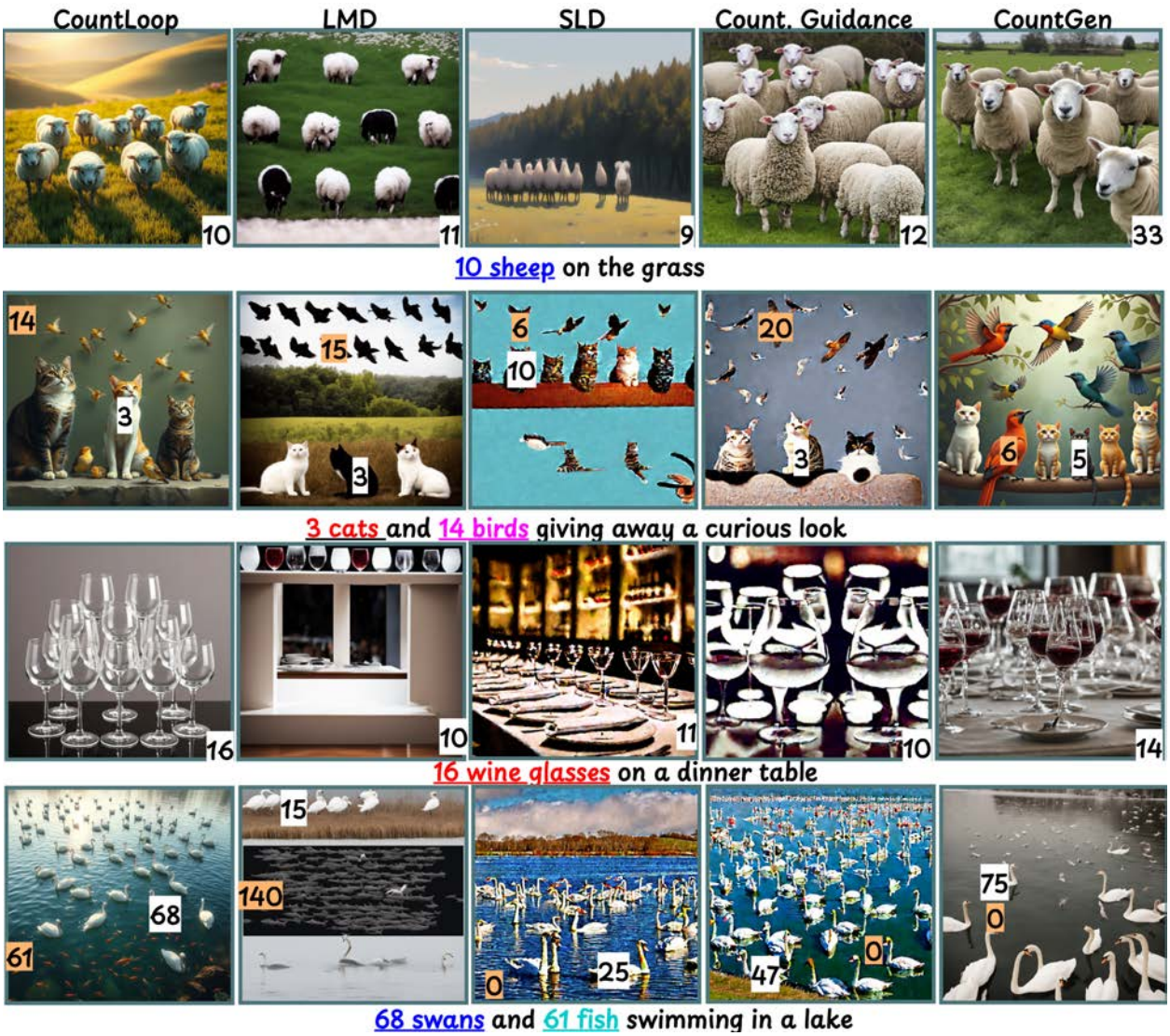


Figure 5: Comparison with SoTA

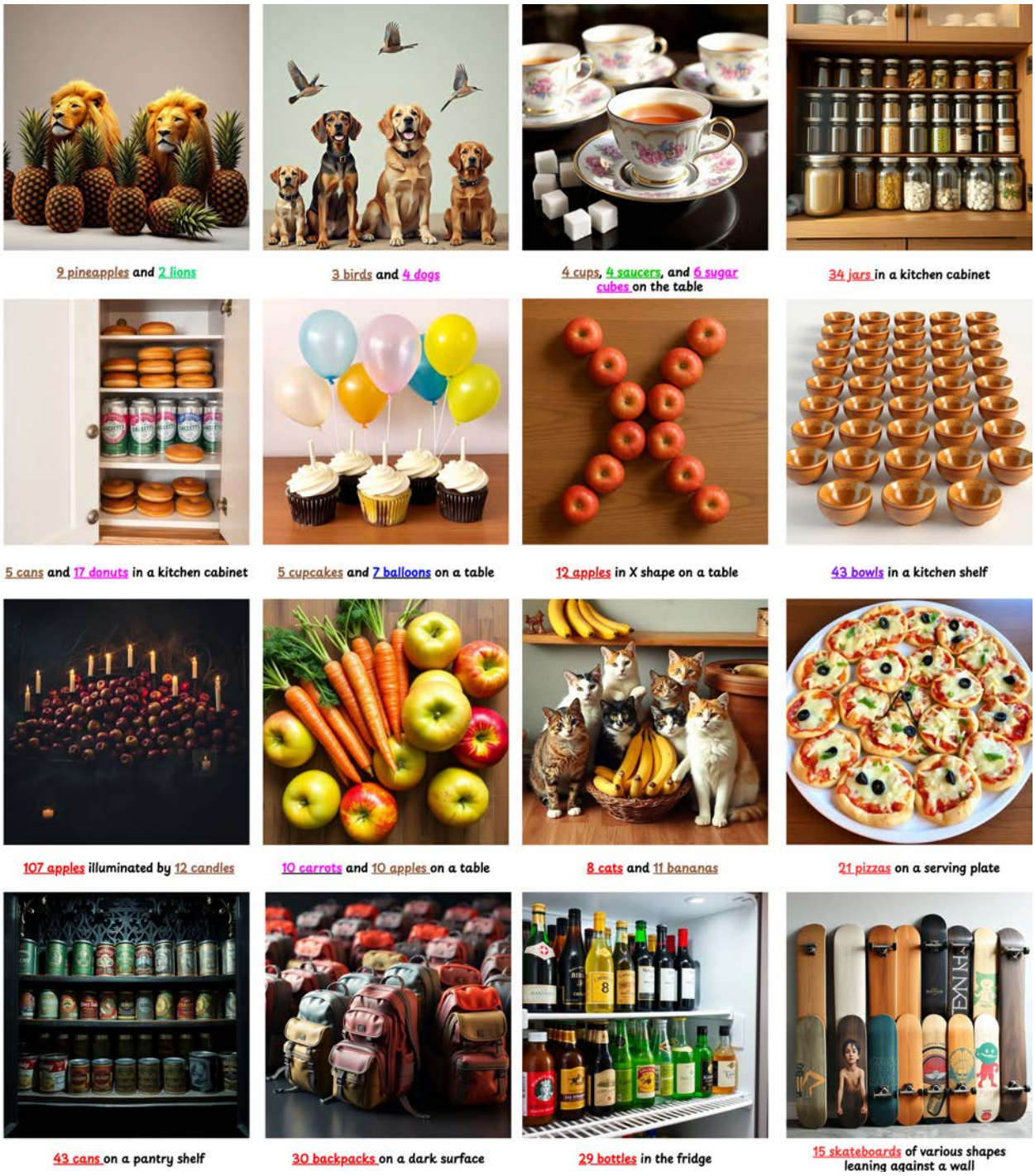


Figure 6: Visuals from our CountLoop-M & CountLoop-S benchmarks using CountLoop .

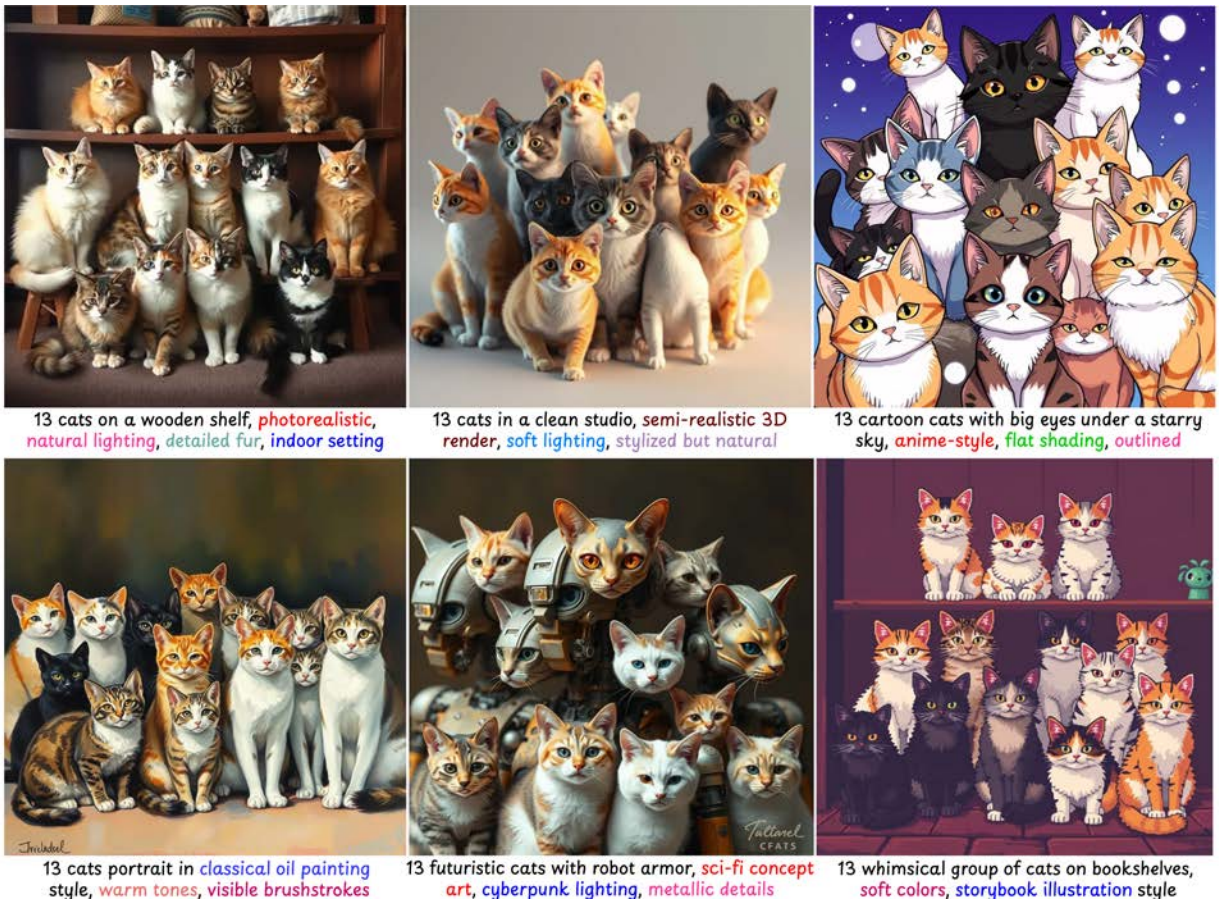


Figure 7: CountLoop's style control capability

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.

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].
- Per-instance bbox area: 1/100 <= w*h <= 1/25
  (at 1024x1024: min ~102x102 px, max ~205x205 px).
- Partially occluded instances: visible area >= 1/6 of full bbox area.
- Bounding boxes are not necessarily square.
- 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,
    "size":[w,h], "color":"string", "attrs":["optional"]} ],
  "relations":[ {"from":"id", "to":"id", "relation":"above|below|left-of|right-of|near",
    "dist":float, "angle":float} ],
  "context": "background description",
  "prompts": {"Pd":"foreground description", "Pbg":"background description"} }

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 (size 0.08*0.08=0.0064, within [0.01,0.04])
  ],
  "relations":[ {"from":"orange_01","to":"orange_02","relation":"left-of",
    "dist":0.12,"angle":0.0} ],
  "context":"wooden fruit crate on a rustic table, soft daylight",
  "prompts":{"Pd":"a crate filled with ripe oranges, 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 8: Full executable Design VLM prompt used in all CountLoop experiments. The bbox area bounds serve as soft spatial guidance targets for the Design VLM to encourage detectable instance sizes; in dense scenes, depth-ordered cumulative composition allows partial occlusion such that instances remain detectable via their visible foreground area (see Sec. 1.2).

Critic VLM Prompt (Textual Refinement Signal Ψ)

```
SYSTEM
You are the Critic VLM. You receive: foreground prompt Pd, generated image I,
target count N (N >= 1), predicted count s_c from a detector, aesthetic score
s_a in [0,1] from an external scorer, and current planning graph G.
Your job: (1) report scores and composite S, (2) provide structured local
suggestions for the textual refinement operator Psi to update G.

SCORING
- Composite: S = alpha * C_acc + (1 - alpha) * s_a, with alpha = 0.6.

TERMINATION (enforced by system, not by this prompt)
- Loop stops when S >= 0.85 OR after K=3 iterations, whichever comes first.
- The "continue" field below is advisory only; system applies the rule above.

OUTPUT FORMAT (JSON only)
{ "scores": {"s_c":int, "N":int, "s_a":float, "C_acc":float, "S":float,
    "alpha":0.6},
  "decision": {"continue":boolean, "reason":"short explanation"},
  "feedback": {
    "summary":"1-2 sentences on layout/style",
    "count":"1-2 sentences on count/visibility",
    "edits":[ {"type":"move|add|remove|resize|degrid",
      "targets":["obj_id_1","obj_id_2"], "hint":"concise instruction"} ] } }

GUIDELINES
- Prefer small, local edits: move a few instances, add/remove a few, break grids.
- Do not propose major scene changes.
- Hints must be specific enough for Psi but short and unambiguous.
- Edit types restricted to: move, add, remove, resize, degrid.
  No aesthetic-only edits; s_a governs termination but never propagates into G.
- At most 10 edit items.

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

Figure 9: Full executable Critic VLM prompt. The `edits` field is restricted to `move` | `add` | `remove` | `resize` | `degrid`; s_a cannot propagate into graph updates. The `continue` field is advisory: the system enforces $S \geq \tau$ or $k \geq K$.

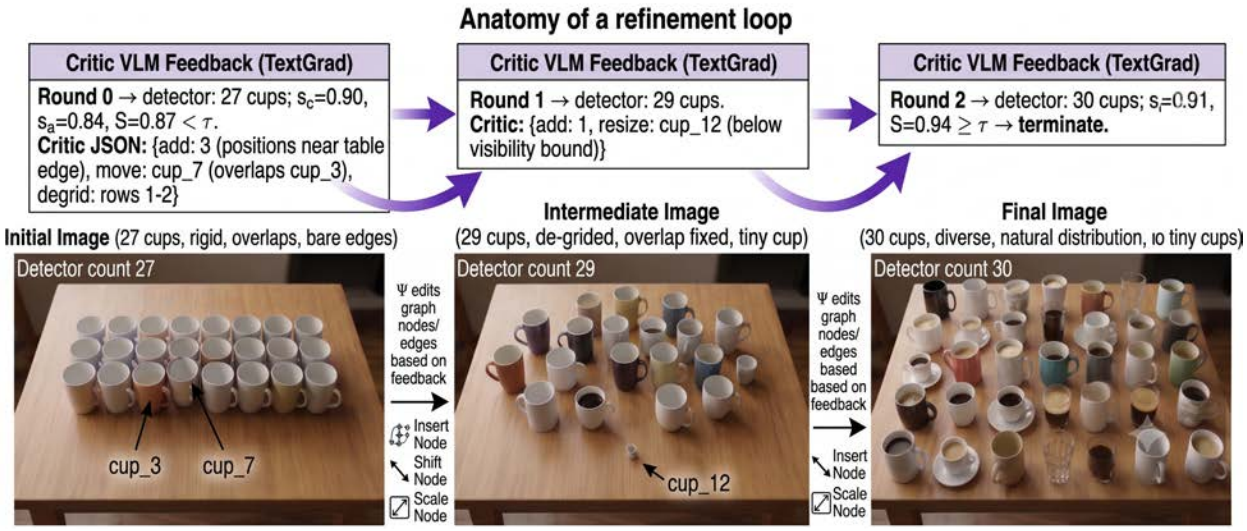


Figure 10: **Anatomy of a refinement loop.** Every feedback message and each round's layout and image for "30 cups on a wooden table". **Round 0** → detector: 27 cups; $s_c = 0.90, s_a = 0.84, S = 0.87 < \tau$. Critic JSON: `{add: 3 (positions near table edge), move: cup_7 (overlaps cup_3), degrid: rows 1-2}` → Ψ inserts 3 nodes, shifts `cup_7` by $(+0.06, -0.03)$, and perturbs row spacing. **Round 1** → detector: 29 cups. Critic: `{add: 1, resize: cup_12 (below visibility bound)}` → Ψ adds one node, scales `cup_12` to the minimum-area bound. **Round 2** → detector: 30 cups; $s_a = 0.91, S = 0.94 \geq \tau$ → terminate.

8 Parameter-Free Refinement and Reasoning Trace

As discussed in Section 3.3 of the main paper, our text editing agent Ψ operates similarly to backpropagation in PyTorch, but instead of gradient tensors, the feedback is in text form (akin to TextGrad). For example, during Round 0 in Fig. 10, the Critic VLM outputs the feedback JSON: `{add: 3 (positions near table edge), move: cup_7 (overlaps cup_3), degrid: rows 1-2}`. The text editing agent Ψ takes this information and generates an explicit reasoning trace before modifying the graph:

Reasoning Trace Example

"Thought: The Critic detected 27 cups instead of 30. I need to add 3 new nodes in the empty space near the table edge. The Critic also identified an overlap between cup_7 and cup_3; I will shift cup_7's bounding box by $(+0.06, -0.03)$ to resolve this. Lastly, rows 1 and 2 look too grid-like, so I will perturb their spacing slightly to create a more natural layout."
"Graph Edit: - Insert: 3 new nodes [$cup_{28}, cup_{29}, cup_{30}$] near the bottom edge. - Update: shift cup_7 box coordinates by $(+0.06, -0.03)$. - Update: add random jitter to y-coordinates of cups in rows 1 and 2."

This forces the agent to plan the update step-by-step. Much like how backpropagation in PyTorch updates model weights based on loss gradients, Ψ explicitly edits the graph nodes and edges based on this textual feedback gradient.

9 OWLv2 Sensitivity Sweep

In our main evaluation, we configure OWLv2 with a confidence threshold of 0.25 and class-agnostic NMS IoU of 0.5. To ensure robustness, we performed a sensitivity sweep over these parameters (varying threshold from 0.15 to 0.35). As shown in Table 5, while the absolute MAE values shift as expected, the relative performance ranking between methods remains entirely invariant.

Table 5: Threshold sweep on CountLoop-S (MAE)

Threshold	**0.15**	**0.20**	**0.25**	**0.30**	**0.35**
CountLoop	9.2	8.1	7.59	7.9	8.8
3DIS	16.4	15.1	14.55	15.0	16.1
SLD	31.5	30.2	29.65	30.1	31.2

Broader Impact

CountLoop targets precise, high-density instance generation for applications in data augmentation, simulation, and synthetic pre-training. The same capability could be exploited to produce misleading imagery — *e.g.*, fabricating crowd sizes or staging scenes for disinformation. The system inherits biases from its pre-trained VLM and diffusion backbone, which may reinforce stereotyped or unrepresentative visual patterns in generated data. Practitioners using CountLoop for downstream dataset construction should audit generated outputs for such biases before deployment.

Future Work: It would be interesting to extend CountLoop to layout-free generation with weak spatial priors, and improve human modeling in dense scenes.

References

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

Lital Binyamin, Yoad Tewel, et al. Make it count: Text-to-image generation with an accurate number of objects, 2024. arXiv:2406.10210.

Black-Forest-Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

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.

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

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.

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.

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.

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.

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.

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.

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.

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.

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

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

Stability-AI. sd3.5. `https://github.com/Stability-AI/sd3.5`, 2025.

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.

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.

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.

Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. InstanceDiffusion: Instance-Level Control for Image Generation . In *CVPR*, pages 6232–6242, Los Alamitos, CA, USA, 2024a. IEEE Computer Society.

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, 2024b.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

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, 2024a. JMLR.org.

Tsung-Han Wu, Long Lian, Joseph E Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting llm-controlled diffusion models, 2024b.

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.

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.

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.

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.

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.

Dewei Zhou, Ji Xie, Zongxin Yang, and Yi Yang. 3dis: Depth-driven decoupled image synthesis for universal multi-instance generation. In *ICLR*, 2025.