

See and Fix the Flaws: Enabling VLMs and Diffusion Models to Comprehend Visual Artifacts via Agentic Data Synthesis

Jaehyun Park^{1,*} Minyoung Ahn^{2,3,*} Minkyu Kim³ Jonghyun Lee³ Jae-Gil Lee¹ Dongmin Park^{3,†}

¹KAIST ²Seoul National University ³KRAFTON

{jhpark813, jaegil}@kaist.ac.kr, {minkyu.kim, jonghyunlee, dongmin.park}@krafton.com, michellahn02@snu.ac.kr

Abstract

*Despite recent advances in diffusion models, AI generated images still often contain visual artifacts that compromise realism. Although more thorough pre-training and bigger models might reduce artifacts, there is no assurance that they can be completely eliminated, which makes artifact mitigation a highly crucial area of study. Previous artifact-aware methodologies depend on human-labeled artifact datasets, which are costly and difficult to scale, underscoring the need for an automated approach to reliably acquire artifact-annotated datasets. In this paper, we propose **ArtiAgent**, which efficiently creates pairs of real and artifact-injected images. It comprises three agents: a perception agent that recognizes and grounds entities and subentities from real images, a synthesis agent that introduces artifacts via artifact injection tools through novel patch-wise embedding manipulation within a diffusion transformer, and a curation agent that filters the synthesized artifacts and generates both local and global explanations for each instance. Using ArtiAgent, we synthesize 100K images with rich artifact annotations and demonstrate both efficacy and versatility across diverse applications. Code is available at [link](#).*

1. Introduction

Recent advances in diffusion models have led to remarkable success in generating highly photorealistic images [45, 53]. While these models have demonstrated striking achievements in text-to-image alignment and aesthetic quality, they still suffer from producing *visual artifacts*, unintended distortions or anomalies in generated outputs. For example, in Figure 1(a), even state-of-the-art models such as NanoBanana [13] produce artifacts, e.g., *six-fingered hands* and *fused entities*, reducing user satisfaction, e.g., uncanny valley. Furthermore, mitigating such artifacts is especially critical in high-stakes diffusion applications where reliability is paramount, including image generation in medicine [23], robotics [17], autonomous driving [19], and so on.

Meanwhile, in image understanding, vision-language models (VLMs) have made substantial progress, showing strong capabilities in visual question answering and scene description [54]. These advances have allowed VLMs to serve as automatic systems for many vision tasks, including medical analysis [16] and robotics control [24]. However, we find that VLMs face challenges when confronted with visual artifacts. As shown in Figure 1(a), even state-of-the-art VLMs, such as GPT-5 [34] and Gemini-2.5-pro [8], exhibit limited ability to detect, localize, or explain artifacts in AI-generated images, nearly indistinguishable from random guessing (see § 6 for details). Consequently, VLMs cannot yet serve as reliable systems for artifact comprehension, which constrains their utility.

Recently, several approaches have been proposed to tackle visual artifacts in VLMs and diffusions [20, 48, 55]. However, existing efforts exhibit two major limitations: (1) they mainly target simple artifact types (e.g., Gaussian noise or blur), which were prevalent in early diffusion models such as SD1.0 [43], but are rarely observed in modern models, where more *plausible* physical distortions are predominant; and (2) they rely heavily on human-annotated artifacts (e.g., as many as 10K labels), which is costly and fundamentally limited in scalability to capture the full diversity of diffusion-generated artifacts. These limitations highlight the need for scalable, annotator-free methods to address plausible and extensive artifacts across diverse visual contexts produced by modern diffusion models.

To fill this gap, we introduce **ArtiAgent** (Figure 1(b)), a novel agentic framework that injects plausible artifacts without human intervention via inversion-restoration [9] by perturbing the attention of DiT (see § 4.2.2). ArtiAgent consists of three agents: (1) a *perception agent*, which identifies the most suitable objects or entities in a real image to perturb, (2) a *synthesis agent*, which selects and applies the tools to generate plausible artifacts, and (3) a *curation agent*, which filters low-quality results and refines their annotations. Through this agentic pipeline, ArtiAgent produces high-quality artifact-injected images with rich annotations, including binary labels, locations, and explanations suitable for detection, localization, and reasoning.

* Equal contribution; work done during an internship at KRAFTON.

† Corresponding author.

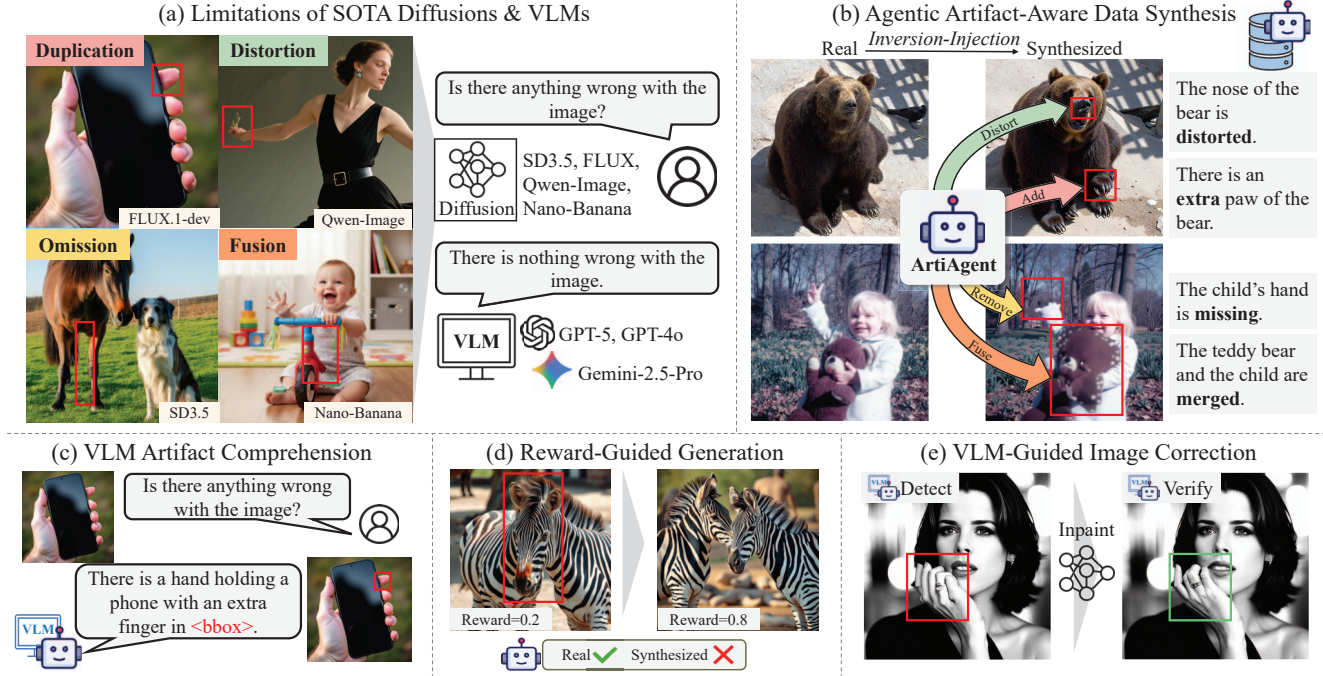


Figure 1. Overview of our challenges and approach. The red boxes indicate the regions with visual artifacts. (a) Examples of structural visual artifacts in state-of-the-art diffusion models and the inability of VLMs to recognize or explain them. (b) Overview of ArtiAgent, a novel agentic framework that synthesizes artifacts for arbitrary visual contexts without human intervention. (c) Example of VLM-based artifact comprehension via detection, explanation, and localization. (d) Application to reward-guided text-to-image generation. (e) Application to image correction, where artifact-aware VLM-guided inpainting removes the flawed regions.

For thorough validation, we present **ArtiBench**, a new benchmark of 1K AI-generated images produced by modern generative models such as FLUX-dev and Nano-Banana, where each image is carefully annotated by humans with binary artifact labels, their bounding box locations, and explanations. We evaluate ArtiAgent on ArtiBench as well as three existing benchmarks, showing that open-source VLMs (e.g., Qwen2.5-VL) fine-tuned on 100K training samples generated by ArtiAgent consistently outperform proprietary VLMs (e.g., GPT-5) and prior baselines across major artifact-perception tasks, including detection, localization, and reasoning. Furthermore, with these artifact-aware VLM, we benefit two downstream applications: (1) guiding diffusion sampling toward artifact-free generation with VLM-based artifact reward, and (2) automatically editing diffusion outputs that contain artifacts, substantially improving the image generation pipelines.

Our key contributions are summarized as follows:

- **Framework.** We introduce ArtiAgent, an agentic data synthesis framework that produces diverse plausible artifacts at scale, enriched with high-quality annotations.
- **Tools.** We develop agentic tools that inject artifacts via our novel inversion-injection method during image reconstruction, which can be used by any DiT model.
- **Datasets.** We synthesize a large-scale training set with 100K examples via ArtiAgent, along with ArtiBench¹, a

challenging benchmark of 1K images generated by modern generative models with human labels.

- **Experiments.** Extensive experiments demonstrate the superiority of ArtiAgent by scaling up VLM performance on core artifact-perception tasks. Moreover, we show its utility in downstream diffusion-based applications: artifact-free image generation and editing.

2. Related Work

Visual Artifact Datasets. Several training datasets have been introduced to supervise the understanding of visual artifacts in generative models. PAL [55] provides 10K images with pixel-level annotations of perceptual defects for segmentation-based training. SynthScars [20] provides 12K images with pixel-level masks and textual explanations. DiffDoctor [48] scales human labeling through semi-supervised expansion from a 25K seed set. Although these datasets offer essential supervision for artifact detection and correction, their reliance on human annotation makes them costly and difficult to scale.

Moreover, several benchmark datasets have been released to evaluate models' ability to understand visual artifacts. RichHF-18K [27] provides annotated artifact regions, while LOKI [51] adds natural-language explanations with bounding box labels. SynthScars also releases evaluation splits to test perception, localization, and explanation ca-

¹<https://huggingface.co/datasets/KRAFTON/ArtiBench>

pabilities. These benchmarks serve as reference points for measuring a model’s artifact awareness; however, most of them rely on artifacts from earlier diffusion models and tend to focus on degenerate artifacts (e.g., Gaussian noise), limiting their relevance to the richer and more diverse failure modes of modern generative systems.

Handling Visual Artifacts. Using these datasets, different modeling strategies have been explored. PAL [55] trains segmentation models to localize artifact regions and enables automated correction via inpainting. RichHF-18K [27] is used to train multimodal models that predict human-like feedback heatmaps, which can then refine diffusion models through preference learning. LEGION [20] trains GLaMM [39] with the SynthScars dataset that allows detection, localization, and explanation into a unified model. DiffDoctor [48] employs its dataset to train an artifact segmentation model, which is used for diffusion fine-tuning to alleviate artifact generation. While these studies emphasize the value of artifact detection and reasoning, they also reveal the necessity of reducing reliance on manual annotation to enable scalable and reliable dataset construction.

3. Understanding Visual Artifacts

3.1. Problem Scope

As generative models have matured, the types of visual artifacts that frequently appear have also changed. Early GANs [15] and U-Net-based diffusions [42] predominantly exhibited naive distortions such as Gaussian-like noise or low-level pixel corruption [55]. Modern diffusion models, equipped with DiT-based architectures and trained on higher-quality datasets, have largely overcome these naive distortions. However, they still generate plausible *structural* visual artifacts, which are the main focus of this work.

Definition 3.1. (Informal) *Structural visual artifacts* refer to defects in which the inherent physical structures of objects are distorted in the generated image. That is, while the contents specified in the prompt are represented, their form violates common-sense plausibility (e.g., a generated dog with two noses). This definition excludes text-to-image misalignments where the prompt content itself is misrepresented (e.g., generating a cat given the prompt of a dog).

3.2. Artifact Analysis in Modern Generative Models

We have categorized structural artifacts into four representative types: **duplication**, **omission**, **distortion**, and **fusion**. Figure 1(a) provides illustrative examples of each case, showing how diffusion models can compromise reliability while still producing visually high-quality pixels. To systematically analyze these artifacts, we randomly sampled 100 captions from the MS-COCO dataset [7] and used them as text-to-image prompts across five state-of-the-art diffusion models, including Stable Diffusion 3.5 [11], FLUX [5],

Table 1. Artifact frequency of modern diffusion models.

Generative Model	Freq.
SD3.5-Large	36%
FLUX-schnell	28%
Qwen-Image	17%
FLUX-dev	12%
Nano-Banana	5%

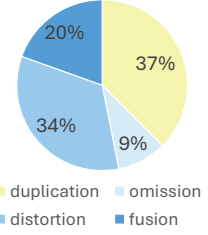


Figure 2. Artifact type distribution of diffusion models.

Qwen-Image [49], and Nano-Banana [13]. For each generated image, a human annotator manually inspected the outputs and identified whether they contained any artifacts, further categorizing them into the four structural types. Based on this evaluation, Table 1 summarizes the frequency of artifacts for each model, while Figure 2 presents the relative occurrence of different artifact types. These findings underscore the importance of addressing structural artifacts in modern generative modeling, highlighting the need for scalable methods to detect, analyze, and mitigate them.

4. Agentic Pipeline for Artifact Synthesis

We propose **ArtiAgent**, a fully automated agentic pipeline that synthesizes visual artifacts in clean images. As shown in Figure 3, three agents (perception, synthesis, curation) select injection candidates, inject artifacts, and curate outputs with local and global explanations. Leveraging recent LLMs and our artifact-injection tools, ArtiAgent determines the artifact types and locations for each image and produces high-quality annotations. Full agent details and visualizations are in Appendices A.

4.1. Perception Agent (Figure 3(a))

The perception agent aims to analyze a real image and decompose it into meaningful semantic units that can serve as reliable candidates for the synthesis agent.

4.1.1. Entity-Subentity Vocabulary Generation

The agent uses out-of-the-box VLMs to decompose the input image into a hierarchical vocabulary of entities (e.g., dog) and subentities (e.g., nose). These are grouped into two semantic levels: peripheral subentities, such as fingers or legs, and intermediate subentities, such as body or face. The prompt template for this module is in Appendix A.2.1.

4.1.2. Entity-Subentity Grounding

The agent then employs Grounded-SAM [41] to ground entities and subentities visible in the image. Once the segmentation masks for both entities (e.g., dog) and subentities (e.g., leg) are obtained, the agent performs a containment analysis to associate each subentity with its parent entity (e.g., a leg of a dog) by computing the overlap ratio.

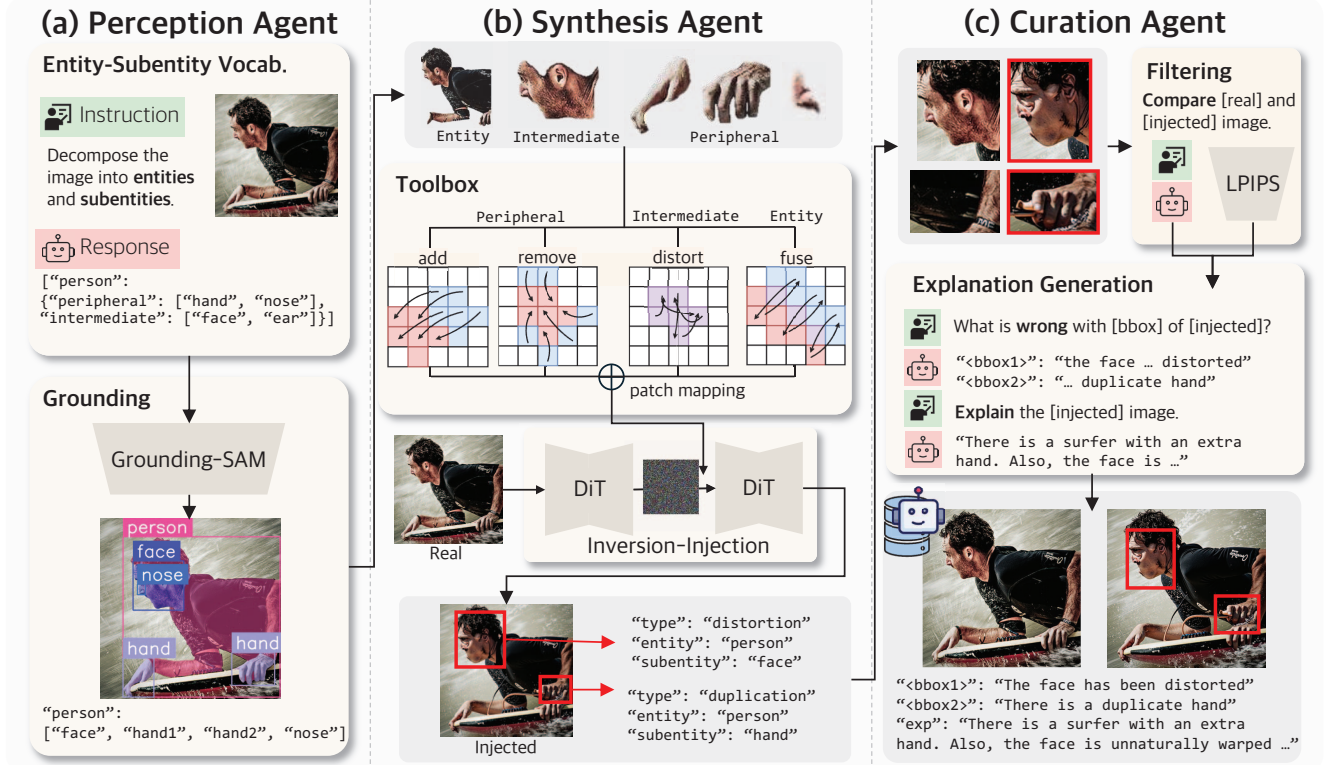


Figure 3. ArtiAgent consists of three coordinated agents: (1) the *perception agent* detects entities and subentities using Grounded-SAM; (2) the *synthesis agent* injects artifacts through patch mapping tool and the inversion-injection paradigm; and (3) the *curation agent* filters low-quality results and generates localized and global textual explanations.

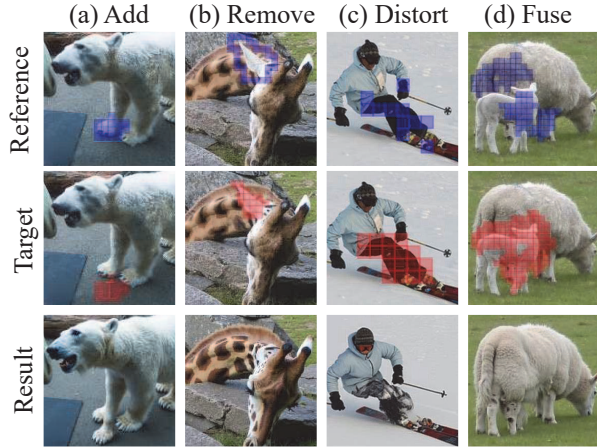


Figure 4. Visualization of each target-reference patch mapping and its resulting artifact-injected image.

4.2. Synthesis Agent (Figure 3(b))

The synthesis agent uses the perception agent’s grounding to inject artifacts into an image via two components: a *toolbox* that generates target-reference patch mappings and an *inversion-injection* module that applies them to the image.

4.2.1. Target-Reference Patch Mapping Toolbox

The toolbox has four artifact-injection tools: **add**, **remove**, **distort**, and **fuse**, each producing a target-reference patch

mapping. It applies add and remove to peripheral subentities, distort to intermediate subentities, and fuse to two overlapping entities. Figure 4 shows one example per tool. Detailed algorithms are in Appendix A.3.

- **Add.** Reference patches are the original subentity region, while the tool chooses the best target-patch candidate from surrounding patches. Nearby patches without overlaps or subentity class conflicts are preferred.
- **Remove.** Target patches are given as the original subentity region, while the tool creates reference patch mappings to replace them with the surrounding context.
- **Distort.** Target patches are the original subentity region, while the tool applies kernels to target patches to generate distorted mappings to reference patches. We use three kernels: jitter for random displacement, strip for circular shifts over bands, and a random permutation kernel.
- **Fuse.** The fuse tool generates mappings that blend content across multiple entities. The target patches contain the overlapping region between two entity instances, and the chunks of reference patches from one entity are assigned to the target patches of the other entity.

4.2.2. Inversion-Injection Method for Artifact Synthesis

We propose an *inversion-injection* module, which extends the inversion-restoration paradigm from image editing [9,

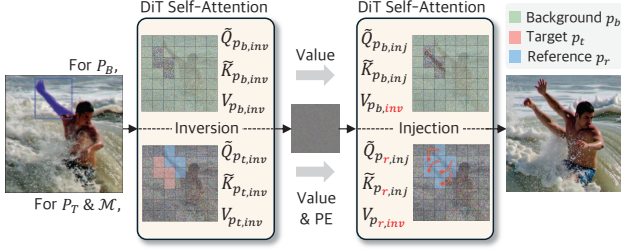


Figure 5. Inversion-injection module. In this example, the right arm from the reference patches is added to target patches below it.

29, 31, 47]. Specifically, the inversion-injection module employs the target-reference patch mapping to manipulate the positional information in the self-attention layers of DiT, allowing realistic structural artifact synthesis.

Notations. Let $X^{(\ell)} \in \mathbb{R}^{N \times d}$ denote the input to a transformer layer ℓ , where N is the number of image patches and d is the embedding dimensionality. We denote the set of all patch indices as $\mathcal{P} = \{1, \dots, N\}$. During artifact injection, we accept the target-reference patch mapping $\mathcal{M} = \{(p_t, p_r) \in \mathcal{P} \times \mathcal{P}\}$ provided by the toolbox, where p_t is a target patch to be modified and p_r its reference patch. The set of all target patches is $\mathcal{P}_T = \{p_t \mid (p_t, p_r) \in \mathcal{M}\}$, the set of all reference patches is $\mathcal{P}_R = \{p_r \mid (p_t, p_r) \in \mathcal{M}\}$, and the set of background patches becomes $\mathcal{P}_B = \mathcal{P} \setminus \mathcal{P}_T$.

Inversion Stage. The inversion stage maps the input image into its corresponding noisy latent representation. Queries, keys, and values for the self-attention layer are

$$Q^{(\ell)} = X^{(\ell)} W_Q^{(\ell)}, K^{(\ell)} = X^{(\ell)} W_K^{(\ell)}, V^{(\ell)} = X^{(\ell)} W_V^{(\ell)}, \quad (1)$$

where $W_Q^{(\ell)}, W_K^{(\ell)}, W_V^{(\ell)} \in \mathbb{R}^{d \times d}$ are learnable. We encode positions with rotary embeddings (RoPE) [46] such that

$$\tilde{Q}_p^{(\ell)} = \text{RoPE}(Q^{(\ell)}, p), \tilde{K}_p^{(\ell)} = \text{RoPE}(K^{(\ell)}, p), p \in \mathcal{P}. \quad (2)$$

Then, the layer’s attention output is

$$\text{Attn}^{(\ell)}(X^{(\ell)}) = \text{Softmax} \left(\frac{\tilde{Q}^{(\ell)} \tilde{K}^{(\ell)\top}}{\sqrt{d}} \right) V^{(\ell)}. \quad (3)$$

While executing Equation (3), we cache the per-layer value embeddings $V_{\text{inv}}^{(\ell)} \leftarrow V^{(\ell)}$ for use in the injection stage.

Injection Stage. As in Figure 5, during the denoising process, the injection stage injects structural artifacts into $\mathcal{P}_T$ by borrowing spatial semantics from $\mathcal{P}_R$, while keeping the original semantics of $\mathcal{P}_B$.

- **Target Region.** For $p_t \in \mathcal{P}_T$ mapped to p_r , we replace the positional embedding (PE) and the value embedding of p_t with those of p_r : $\tilde{Q}_{p_t}^{(\ell)} = \text{RoPE}(Q_{p_t}^{(\ell)}, p_r)$, $\tilde{K}_{p_t}^{(\ell)} = \text{RoPE}(K_{p_t}^{(\ell)}, p_r)$, and $V_{p_t}^{(\ell)} \leftarrow V_{p_r, \text{inv}}^{(\ell)}$.

- **Background Region.** For a non-target patch $p_b \in \mathcal{P}_B$, we keep their original positional information and reuse the $V_{\text{inv}}^{(\ell)}$ values to maintain the context of the original image: $\tilde{Q}_{p_b}^{(\ell)} = \text{RoPE}(Q_{p_b}^{(\ell)}, p_b)$, $\tilde{K}_{p_b}^{(\ell)} = \text{RoPE}(K_{p_b}^{(\ell)}, p_b)$, and $V_{p_b}^{(\ell)} \leftarrow V_{p_b, \text{inv}}^{(\ell)}$.

Then, self-attention is performed as in Equation (3).

In general, PE injection controls *where* the model believes denoising is occurring, while value injection provides *what* semantic content fills that position. Their combination allows for local injections of realistic artifacts, while the background remains consistent with the original image. Although *value injection* [9, 47] is a well-established approach in image editing research, our method introduces the novel idea of manipulating their positions during injection. Furthermore, *PE injection* itself has not been used in previous image editing works. Combining PE and value injections offers a particularly effective methodology for generating structural artifacts, as it enables direct manipulation of spatial information during the reconstruction process. To prevent shortcut-feature learning (e.g., edge discontinuities), we restrict PE and value injections to early-to-middle layers and disable them during the final denoising steps. We provide attention visualizations and ablation studies (including injection-step ablations) to support these design choices in Appendices A.4 and A.5. Furthermore, our training-free inversion-injection method is inherently model-agnostic, making it readily generalizable to other emerging DiT-based image generation models.

4.3. Curation Agent (Figure 3(c))

The curation agent is the pipeline’s quality assurance and enrichment stage, which refines the synthesis agent’s output into training-ready datasets for downstream tasks. Given paired real and artifact-injected images, the curation agent performs data filtering and explanation generation. This paired input enables reliable filtering and explanation by *contrasting* artifact-injected regions with real counterparts.

4.3.1. Data Filtering

This stage applies one of two filtering methods depending on which tool was used for artifact injection.

- **LPIPS-Based Filtering.** Distortion artifacts are validated using the LPIPS [56] metric, which measures the perceptual difference between two images by comparing their feature representations, producing a metric that aligns much more closely with human judgments. For each cropped original-injected pair, we compute the LPIPS score and retain the pair if it satisfies $\tau_1 \leq 1 - d_{\text{LPIPS}}(x_{\text{original}}, x_{\text{artifact}}) \leq \tau_2$, with τ_2 filtering out pairs too similar with unidentifiable changes, while τ_1 filters out severely corrupted samples with implausible damage in the region.

- **VLM-Based Filtering.** For duplication, omission, and fusion artifacts, we employ a VLM to validate whether the injected change is perceptible and localized within the designated region. The VLM receives the original image with the target region masked out (to provide the global scene context), the original image cropped to the target region (to show the unaltered content), and the artifact-injected image cropped to the same region (to focus on the modification). From this triplet, the VLM makes a binary judgment, confirming whether a new instance has appeared (duplication), an expected object has become missing (omission), or two objects have been unnaturally merged (fusion). For a detailed prompt template used for VLM-based filtering, see Appendix A.2.2.

4.3.2. Explanation Generation

- **Local Explanation.** For each candidate region, the VLM receives the same triplet used in filtering. We prompt the VLM to synthesize a short local description by guiding the VLM to describe what is different in the artifact region compared to the real counterpart. For the detailed prompt template used, see Appendix A.2.3.
- **Global Explanation.** After generating local explanations for all artifacts, the curation agent generates a global explanation for the whole image. The VLM accepts the artifact image and a list of artifact-injected bounding boxes and their corresponding local explanations. The VLM is prompted to explain why this artifact injected image is indeed an artifact. For the detailed prompt template used, see Appendix A.2.3.

4.4. Data Collection

With our novel three-stage agentic pipeline, we collect 50K pairs of artifact-injected images and the corresponding original images, along with the metadata consisting of artifact explanation shown in the final output of Figure 3. Here, we reconstruct the source images using the inversion-restoration method to alleviate pairwise differences originating from diffusion-generated image traits. The source images are composed of four different datasets, COCO [7], Caltech-101 [12], 11K Hands [1], and Celeba HQ [21], broadening the distribution from diverse real-world scenes to specific single entity images. We employ GPT-4o [33] as the VLM that contributes to our agentic flow. In addition, the inversion-injection method in Section 4.2.2 uses FLUX.1-dev [5] for the DiT and FireFlow [9] for the inversion-injection module.

5. ArtiBench: Artifact Detection Benchmark

Existing artifact detection benchmarks, such as RichHF [25], LOKI [51], and SynthScars [20] were built primarily using earlier diffusion models, including Stable Diffusion 1 or 2 and Midjourney. These datasets,

Table 2. **Comparison of artifact benchmark datasets.** If a benchmark reused another dataset as a source, we describe the generative models used in that source. The table with the full citations is provided in Appendix B.1.

Benchmark	Source Models		Sample	Bin.	Loc.	Exp.
	Oldest	Newest				
RichHF	SD2.1	Dreamlike Photoreal	955		✓	
LOKI	pix2pix	FLUX	229		✓	✓
SynthScars	Midjourney	DALL-E3	1K		✓	✓
ArtiBench	SD3.5	Nano-Banana	1K	✓	✓	✓

though valuable at the time, no longer capture the characteristics of modern artifacts produced by current diffusion and multimodal transformers. As a result, the evaluation on these benchmarks may not accurately reflect the artifact-handling capabilities of today’s models.

To address this issue, we introduce **ArtiBench**, a new benchmark that reflects the current state of artifact phenomena in recent generative models. As summarized in Table 2, previous benchmarks were limited either in sample diversity, recency of generative sources, or task coverage. Our benchmark is designed to overcome these limitations by including data from recent models and providing comprehensive annotations for multiple artifact-related tasks.

We construct **ArtiBench** with 1K images generated by five state-of-the-art diffusion models, Stable Diffusion3.5 [11], FLUX-schnell/dev [5], Qwen-Image [49], and Nano-Banana [13], with the prompts sampled from three datasets, MS-COCO [7], PartiPrompts [52], and FuseCap [44]. For annotation, we involve 12 human annotators and label each image with: (1) a binary indicator denoting the presence or absence of artifacts, (2) bounding boxes for all artifact regions, and (3) concise descriptions of the observed abnormalities. The dataset is balanced with an equal ratio of artifact-free and artifact-containing samples. Further details on the construction and annotation pipeline are provided in Appendix B.2.

6. Experiments

6.1. Understanding Artifacts with VLMs

We assess the efficacy of ArtiAgent by training VLMs with visual question-answering (VQA) samples generated from ArtiAgent and measuring their performance on artifact-aware benchmarks, including ArtiBench.

6.1.1. Setup

We consider *three* tasks, artifact detection, localization, and explanation. For evaluation metrics, we use accuracy and F1 score for detection, mIoU and F1 score for localization, and ROUGE and CSS for explanation. More details of the evaluation protocol are provided in Appendix B.3. For baselines, we use three artifact segmentation algorithms,

Table 3. **Artifact understanding performance across (a) binary detection, (b) localization, and (c) explanation.** The benchmarks are listed in the order in which they were published.

Methods	(a) Binary detection		(b) Localization								(c) Explanation					
	ArtiBench		RichHF		LOKI		SynthScars		ArtiBench		LOKI		SynthScars		ArtiBench	
	Acc	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	ROUGE	CSS	ROUGE	CSS	ROUGE	CSS
PAL	-	-	0.079	0.028	0.021	0.037	0.035	0.053	0.040	0.066	-	-	-	-	-	-
DiffDoctor	-	-	0.077	0.139	0.175	0.274	0.083	0.136	0.081	0.137	-	-	-	-	-	-
LEGION	-	-	0.067	0.112	0.100	0.158	0.106 [†]	0.152 [†]	0.062	0.099	0.133	0.314	0.247[†]	0.589[†]	0.143	0.332
GPT-4o	0.619	0.601	0.086	0.040	0.037	0.056	0.032	0.052	0.049	0.084	0.107	0.266	0.125	0.404	0.143	0.433
Gemini-2.5-Pro	0.582	0.575	0.061	0.091	0.109	0.169	0.064	0.101	0.095	0.147	0.097	0.358	0.103	0.474	0.159	0.420
GPT-5	0.599	0.577	0.126	0.146	0.089	0.141	0.117	0.185	0.061	0.099	0.121	0.382	0.120	0.461	0.145	0.434
Qwen2.5-VL-7B	0.501	0.336	0.075	0.028	0.052	0.068	0.013	0.018	0.010	0.014	0.106	0.267	0.115	0.362	0.117	0.263
+ ArtiAgent	0.627	0.627	0.119	0.198	0.129	0.198	0.137	0.214	0.111	0.168	0.169	0.454	0.196	0.578	0.233	0.643
InternVL3.5-8B	0.498	0.357	0.013	0.022	0.015	0.025	0.019	0.033	0.010	0.015	0.081	0.189	0.050	0.180	0.126	0.256
+ ArtiAgent	0.630	0.620	0.100	0.170	0.126	0.196	0.140	0.217	0.119	0.176	0.137	0.401	0.179	0.513	0.226	0.625

[†]LEGION was trained on the SynthScars training dataset split.

PAL [55], DiffDoctor [48], and LEGION [20]; and three proprietary VLMs, GPT-4o [33], Gemini-2.5-Pro [14], and GPT-5 [34]. PAL and DiffDoctor are evaluated only on the localization task as they output only segmentations, LEGION is evaluated on the localization and explanation tasks, and the three VLMs are evaluated on all three tasks. We fine-tune Qwen2.5-VL-7B [4] and InternVL3.5-8B [57] on a 100K training set generated by ArtiAgent. The detailed procedure for VQA generation is provided in Appendix B.4.

6.1.2. Main Results

Overall, fine-tuning open-source VLMs with synthetic data generated by ArtiAgent consistently enhances their ability to detect, explain, and localize visual artifacts. Across all three tasks, ArtiAgent-trained models not only outperform their vanilla counterparts but also match or exceed the performance of proprietary systems such as GPT-5 and Gemini-2.5-Pro. These gains highlight the quality and scalability of ArtiAgent-generated supervision, demonstrating that artifact synthesis through agentic data generation provides rich, transferable signals for both spatial grounding and semantic reasoning.

Artifact Binary Detection. Table 3(a) shows that VLMs trained with ArtiAgent achieve clearly superior artifact detection performance. Specifically, ArtiAgent improves the accuracy of InternVL3.5-8B by 26.5%. At the same time, the overall low accuracy on ArtiBench highlights the difficulty and importance of understanding visual artifacts in AI-generated images, especially because modern diffusion models increasingly exhibit subtle, structured failures that current VLMs easily miss.

Artifact Localization. Table 3(b) shows that ArtiAgent consistently enhances the spatial grounding capability of open-source VLMs. Although DiffDoctor exhibits high accuracy in LOKI, it fails to generalize to more recent benchmarks, including ArtiBench. This outcome means that ArtiBench further extends the artifact detection field one step deeper, by capturing artifacts that prior artifact-detection models struggle to identify.

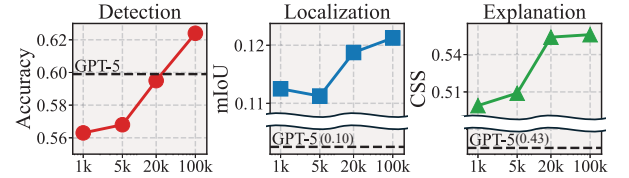


Figure 6. **Scaling effect** of data generated by ArtiAgent with Qwen2.5-VL-7B. We average the results of all the benchmarks.

Table 4. ArtiBench result of the VLM trained on a 1K dataset.

Methods	(a) Binary detection		(b) Localization		(c) Explanation	
	Acc	F1	mIoU	F1	ROUGE	CSS
SynthScars	0.555	0.548	0.094	0.147	0.156	0.521
ArtiAgent	0.555	0.550	0.074	0.121	0.222	0.606

Artifact Explanation. Table 3(c) shows that training with the ArtiAgent’s generated dataset greatly strengthens the reasoning and description capabilities of VLMs. The fine-tuned models exhibit clear improvements in the ROUGE and CSS metrics across all benchmarks.

6.1.3. Data Scaling Effect of ArtiAgent

Figure 6 shows how model performance grows as we increase the size of the training data with ArtiAgent. In all three tasks, we observe a clear upward trend, which means that more synthesized data consistently lead to better artifact understanding. Notably, for localization and explanation, the performance with subsets as small as 1K samples already surpasses that of GPT-5, showing that ArtiAgent provides highly sample-efficient supervision. In contrast, the binary detection task continues to improve up to the 100K scale, suggesting that detection benefits from larger and more diverse artifacts. These results highlight the rich supervision and the strong scaling potential of ArtiAgent.

6.1.4. Comparison with Human-Annotated Supervision

To compare the quality of human annotation and ArtiAgent-generated annotations, we train Qwen2.5-VL-7B with 500 SynthScars samples and 500 clean samples from ArtiAgent and compare it with a counterpart trained on 1K ArtiAgent samples. Table 4 shows that ArtiAgent’s synthetic supervision remains competitive with human annotation: SynthScars is slightly better in localization, while ArtiA-



Figure 7. **Reward-guided generation.** ArtiAgent can train a reward model that guides diffusion to generate artifact-free images.

gent matches in detection and performs better in explanation. We attribute the localization gap mainly to the patch-level granularity of ArtiAgent labels. Overall, these results indicate that ArtiAgent is a cost-effective and scalable alternative to human annotation while maintaining comparable supervision quality.

6.2. Mitigating Artifacts in Diffusion Models

6.2.1. Reward-Guided Artifact-Free Generation

A key strength of ArtiAgent is its pairwise data design: for every instance, it provides two tightly matched images with the same content, one clean and one with artifact. This structure gives extremely rich supervision for learning artifact preferences. Using these pairs, we adopt the reward-guided test-time scaling framework [28] that steers diffusion models toward producing artifact-free images.

Setup. We use CLIP as the backbone of the Bradley-Terry [6] reward model. The model learns to assign a higher score to the real image over the artifact image. With this artifact-aware reward model, we apply test-time scaling to FLUX-schnell. We run six search rounds, each for 100 prompts sampled from MS-COCO and measure how much the reward increases as the search progresses. The detailed training schema and the test-time scaling procedure are provided in the Appendix C.1.

Results. As can be seen in Figure 7, the reward steadily improves throughout the search rounds, indicating that the diffusion model continues to generate images with fewer artifacts. Qualitatively, the examples in Figure 7 show clearer structures and reduced artifact patterns in later rounds, demonstrating that the reward model has successfully learned the real-artifact preference and enables guidance towards artifact-free images.

6.2.2. VLM-Guided Artifact Correction

Since our artifact-trained VLM can reliably detect and localize artifacts, we employ it to guide an image inpainting model to correct artifact regions in AI-generated images.

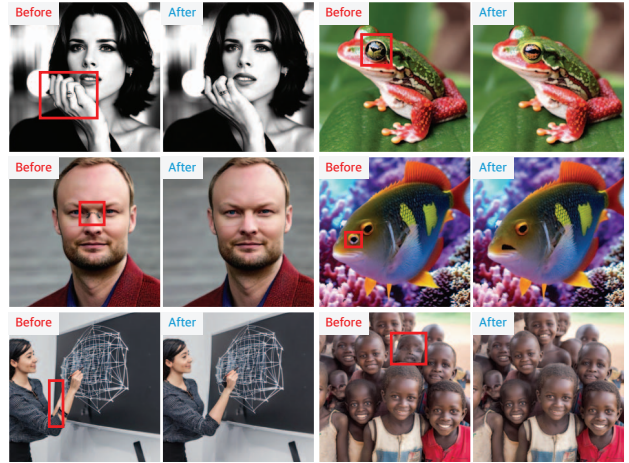


Figure 8. **Image correction.** The ArtiAgent-trained VLM can effectively guide image inpainting models to correct artifact regions.

Setup. We use Qwen2.5-VL-7B trained with ArtiAgent to determine whether an artifact is present in a given image and to localize the artifact region. This region is then processed by the FLUX inpainting pipeline [3], which synthesizes a corrected version of the localized area. Next, the corrected image is re-evaluated by the VLM to verify whether the artifact in the region has been fully resolved in the specified region. If the VLM continues to detect an artifact, the inpainting procedure is repeated. This iterative loop continues until the VLM confirms the absence of artifacts. The detailed procedure is provided in the Appendix C.2.

Results. Figure 8 presents qualitative results of our image correction pipeline. The VLM accurately identifies the artifact region and the inpainting model corrects the region with natural and structurally consistent content. These results demonstrate that the proposed pipeline can reliably locate and correct artifacts, depicting the usefulness of our artifact understanding VLM.

7. Conclusion

In this work, we introduce ArtiAgent, a scalable agentic framework that automatically synthesizes visual artifacts through positional embedding manipulation in diffusion transformers. By integrating perception, synthesis, and curation agents, our pipeline generates large-scale, richly annotated artifact datasets without human supervision. Experiments showed that VLMs fine-tuned on the ArtiAgent datasets achieved substantial gains in artifact detection, localization, and explanation, and that the resulting models can guide diffusion sampling toward artifact-free generations and perform automated artifact correction. Together, these results demonstrate that agentic data synthesis provides an effective and general pathway to perceiving and mitigating visual artifacts in modern generative models.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II200862, DB4DL: High-Usability and Performance In-Memory Distributed DBMS for Deep Learning, 50% and No. RS-2025-25442149, LG AI STAR Talent Development Program for Leading Large-Scale Generative AI Models in the Physical AI Domain, 50%).

References

- [1] Mahmoud Afifi. 11k hands: gender recognition and biometric identification using a large dataset of hand images. *Multimedia Tools and Applications*, 78:20835–20854, 2019. 6
- [2] Vatsal Agarwal, Matthew Gwilliam, Gefen Kohavi, Eshan Verma, Daniel Ulbricht, and Abhinav Shrivastava. Towards multimodal understanding via stable diffusion as a task-aware feature extractor. *arXiv preprint arXiv:2507.07106*, 2025. 1
- [3] AlimamaCreative. Flux.1-dev controlnet inpainting (beta). <https://huggingface.co/alimama-creative/FLUX.1-dev-Controlnet-Inpainting-Beta>, 2024. Model weights released under the FLUX.1 [dev] Non-Commercial License. 8, 18
- [4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibong Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and Others. Qwen2. 5-vl technical report. *Arxiv Preprint Arxiv:2502.13923*, 2025. 7, 17
- [5] Blackforestlabs. Flux: a powerful tool for text generation. <https://blackforestlabs.ai/>, 2024. 3, 6, 16
- [6] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. 8, 17
- [7] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: data collection and evaluation server. *Arxiv Preprint Arxiv:1504.00325*, 2015. 3, 6
- [8] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Bliestein, Ori Ram, Dan Zhang, Evan Rosen, and Others. Gemini 2.5: pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Arxiv Preprint Arxiv:2507.06261*, 2025. 1
- [9] Yingying Deng, Xiangyu He, Changwang Mei, Peisong Wang, and Fan Tang. Fireflow: fast inversion of rectified flow for image semantic editing. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025. 1, 4, 5, 6
- [10] Dreamlike Art. Dreamlike photoreal 2.0. <https://huggingface.co/dreamlike-art/dreamlike-photoreal-2.0>, 2023. 16
- [11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024. 3, 6, 16
- [12] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007. 6
- [13] Alisa Fortin, Guillaume Vernade, Kat Kampf, and Ammaar Reshi. Introducing gemini 2.5 flash image. <https://developers.googleblog.com/en/introducing-gemini-2-5-flash-image/>, 2025. Google AI / DeepMind blog post, August 26 2025. 1, 3, 6, 16
- [14] Gemini Team, Google DeepMind. Gemini 2.5: pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. Technical Report, 2025. 7
- [15] Ian Goodfellow, Jean Pouget-abadie, Mehdi Mirza, Bing Xu, David Warde-farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3
- [16] Iryna Hartsock and Ghulam Rasool. Vision-language models for medical report generation and visual question answering: a review. *Frontiers in Artificial Intelligence*, 7:1430984, 2024. 1
- [17] Xinyu Huang. A survey of domain adaptation in robotics using diffusion models. *Applied and Computational Engineering*, 179:1–8, 2025. 1
- [18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017. 16
- [19] Max Jiang, Yijing Bai, Andre Cornman, Christopher Davis, Xiukun Huang, Hong Jeon, Sakshum Kulshrestha, John Lambert, Shuangyu Li, Xuanyu Zhou, and Others. Scenediffuser: efficient and controllable driving simulation initialization and rollout. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 55729–55760, 2024. 1
- [20] Hengrui Kang, Siwei Wen, Zichen Wen, Junyan Ye, Weijia Li, Peilin Feng, Baichuan Zhou, Bin Wang, Dahua Lin, Linfeng Zhang, and Others. Legion: learning to ground and explain for synthetic image detection. *Arxiv Preprint Arxiv:2503.15264*, 2025. 1, 2, 3, 6, 7, 16
- [21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 6
- [22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8110–8119, 2020. 16
- [23] Firas Khader, Gustav Müller-Franzes, Soroosh Tayebi Arasteh, Tianyu Han, Christoph Haaburger, Maximilian Schulze-Hagen, Philipp Schad, Sandy Engelhardt, Bettina Baeßler, Sebastian Foersch, et al. Denoising

- diffusion probabilistic models for 3d medical image generation. *Scientific Reports*, 13(1):7303, 2023. 1
- [24] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, and Others. Openvla: an open-source vision-language-action model. *Arxiv Preprint Arxiv:2406.09246*, 2024. 1
- [25] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Arxiv Preprint Arxiv:2305.01569*, 2023. 6
- [26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 19730–19742. PMLR, 2023. 1
- [27] Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-tuset, Sarah Young, Feng Yang, and Others. Rich human feedback for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19401–19411, 2024. 2, 3, 16
- [28] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Others. Inference-time scaling for diffusion models beyond scaling denoising steps. *Arxiv Preprint Arxiv:2501.09732*, 2025. 8, 17
- [29] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 5
- [30] Midjourney, Inc. Midjourney. <https://www.midjourney.com/>, 2022. 16
- [31] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6038–6047, 2023. 5
- [32] Openai. Dall-e 3. <https://openai.com/research/dall-e-3>, 2023. 16
- [33] OpenAI. Gpt-4o system card. Technical Report arXiv:2410.21276, OpenAI, 2024. 6, 7, 1
- [34] Openai. Gpt-5: large language model. <https://openai.com/research/gpt-5>, 2025. 1, 7
- [35] Ji-Hoon Park, Yeong-Joon Ju, and Seong-Whan Lee. Explaining generative diffusion models via visual analysis for interpretable decision-making process. *Expert Systems with Applications*, 248:123231, 2024. 1
- [36] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 319–345. Springer, 2020. 16
- [37] Yong-Hyun Park, Mingi Kwon, Jaewoong Choi, Junghyo Jo, and Youngjung Uh. Understanding the latent space of diffusion models through the lens of riemannian geometry. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 24129–24142, 2023. 1
- [38] 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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 16
- [39] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13009–13018, 2024. 3
- [40] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982–3992. Association for Computational Linguistics, 2019. 16
- [41] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, and Others. Grounded sam: assembling open-world models for diverse visual tasks. *Arxiv Preprint Arxiv:2401.14159*, 2024. 3
- [42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 3, 16
- [43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 1
- [44] Noam Rotstein, David Bensaid, Shaked Brody, Roy Ganz, and Ron Kimmel. Fusecap: leveraging large language models for enriched fused image captions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5689–5700, 2024. 6
- [45] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo lopes, Burcu Karagol ayan, Tim Salimans, and Others. Photorealistic text-to-image diffusion models with deep language understanding. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 36479–36494, 2022. 1
- [46] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 5
- [47] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025. 5
- [48] Yiyang Wang, Xi Chen, Xiaogang Xu, Sihui Ji, Yu Liu, Yujun Shen, and Hengshuang Zhao. Diffdoctor: diagnos-

- ing image diffusion models before treating. *Arxiv Preprint Arxiv:2501.12382*, 2025. [1](#), [2](#), [3](#), [7](#)
- [49] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-Ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, and Others. Qwen-image technical report. *Arxiv Preprint Arxiv:2508.02324*, 2025. [3](#), [6](#), [16](#)
- [50] Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for AI-generated image detection. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. [16](#)
- [51] Junyan Ye, Baichuan Zhou, Zilong Huang, Junan Zhang, Tianyi Bai, Hengrui Kang, Jun He, Honglin Lin, Zihao Wang, Tong Wu, and Others. Loki: a comprehensive synthetic data detection benchmark using large multimodal models. *Arxiv Preprint Arxiv:2410.09732*, 2024. [2](#), [6](#), [16](#)
- [52] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. Featured Certification. [6](#)
- [53] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. Text-to-image diffusion models in generative ai: a survey. *Arxiv Preprint Arxiv:2303.07909*, 2023. [1](#)
- [54] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5625–5644, 2024. [1](#)
- [55] Lingzhi Zhang, Zhengjie Xu, Connelly Barnes, Yuqian Zhou, Qing Liu, He Zhang, Sohrab Amirghodsi, Zhe Lin, Eli Shechtman, and Jianbo Shi. Perceptual artifacts localization for image synthesis tasks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 7579–7590, 2023. [1](#), [2](#), [3](#), [7](#)
- [56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. [5](#)
- [57] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingdong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhui Wang. Internvl3: exploring advanced training and test-time recipes for open-source multimodal models. *Arxiv Preprint Arxiv:2504.10479*, 2025. [7](#), [17](#)