Where, What, Why, and Importance: Structured Defect Grounding for Text-to-Image Feedback

Huaisong Zhang[1,2,*,§] **Hao Yu**[1,*] **Yuxuan Zhang**[2,3,4,§] **Jiahe Wang**[1] **Xinrui Chen**[1]
Haoxiang Cao[2,5,§] **Feng Lu**[1] **Wendong Zhang**[2,†] **Changqian Yu**[2,‡] **Chun Yuan**[1,†]
[1]Tsinghua University [2]Kolors Team, KlingAI Research
[3]University of British Columbia [4]Vector Institute [5]South China Normal University

Abstract

Despite generating increasingly photorealistic images, text-to-image (T2I) models still exhibit localized, subtle, and structurally complex failures. Diagnosing these failures requires instance-level feedback that answers *where* a defect occurs, *what* type it is, *why* it is defective, and its *importance* to overall image quality. While recent dense-feedback methods move beyond scalar supervision, their heatmap-centric representations still formulate diagnosis as pixel-field regression, making it difficult to localize variable-cardinality defects and bind semantic reasons to individual failures. To address this representation bottleneck, we propose **Structured Defect Grounding (SDG)**, which casts T2I diagnosis as structured set prediction by modeling each defect as a *(location, type, reason, importance)* tuple. To make this formulation trainable and measurable, we introduce **SDG-30K**, a 30K-image dataset with box-grounded annotations across four modern T2I generators, together with a dedicated evaluation protocol, **SDG-Eval**. Building on this structured representation, we further present a diagnosis-to-alignment framework in which a Vision-Language Model (VLM) serves as the SDG detector, and **BoxFlow-GRPO** converts predicted defect sets into box-derived, importance-weighted spatial rewards for diffusion model alignment. Extensive experiments show that our SDG detector outperforms leading proprietary VLMs on structured defect grounding, while SDG-guided rewards consistently improve T2I alignment and support localized image refinement. These results establish SDG as a unified, instance-level interface for diagnosing, evaluating, and enhancing modern generative models.

1 Introduction

Text-to-image (T2I) models [Esser et al., 2024, Betker et al., 2023, Black Forest Labs, 2025, Cai et al., 2025, Ma et al., 2025a, Xie et al., 2025] produce increasingly photorealistic images, yet their failures often remain localized, subtle, and structurally heterogeneous, such as malformed text, implausible geometry, and semantic mismatches. Evaluating such outputs with scalar preference scores [Wu et al., 2023b,a, Xu et al., 2023, Kirstain et al., 2023] collapses these defects into a single global value. Although useful for ranking images, scalar feedback cannot answer the four key questions required for defect diagnosis: *where* the error occurs, *what* type of defect it is, *why* the region is defective, and its *importance* to overall image quality. This limitation reveals a mismatch between global image-level supervision and the localized, instance-level nature of T2I failures.

Recent work attempts to address this need by moving beyond scalar supervision toward dense feedback. RichHF [Liang et al., 2024] introduces artifact and misalignment heatmaps, while ImageDoctor [Guo et al., 2025] augments heatmap-based diagnosis with VLM reasoning. Although

[*]Equal contribution. [‡]Project lead. [†]Corresponding authors.
[§]Work conducted during an internship at Kolors Team, KlingAI Research.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

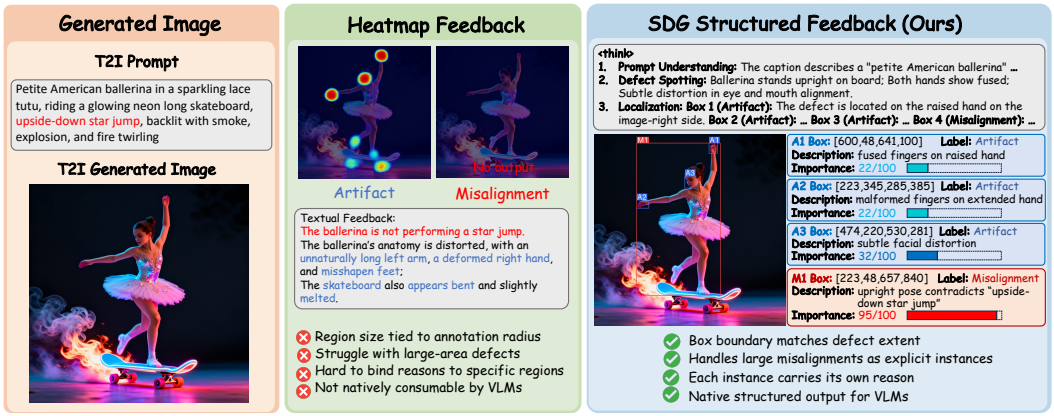

Figure 1: Comparison of dense feedback paradigms for T2I defect diagnosis. **Left**: a T2I generated image with its prompt. **Middle**: heatmap-style feedback produces separate artifact and misalignment maps with textual feedback. **Right**: SDG reasons via a chain-of-thought (CoT) trace, then outputs structured defects with bounding boxes, types, descriptions, and importance scores.

heatmaps provide dense signals, they still formulate defect diagnosis as pixel-field regression rather than instance-level understanding. As illustrated in Figure 1, this heatmap-centric paradigm exposes representation bottlenecks: spatially, point-derived maps depend on annotator-chosen radii rather than true defect extents [Liang et al., 2024]; semantically, continuous severity fields cannot bind defect types, reasons, or importance scores to individual failures; and architecturally, pixel-level maps are not native outputs of autoregressive VLMs and often require additional decoders or regression heads. These limitations suggest that the key challenge is not merely to make feedback dense, but to represent defects as spatially grounded, semantically explicit, and VLM-compatible instances.

Motivated by this bottleneck, we move from continuous pixel-field regression to structured instance prediction. We introduce **Structured Defect Grounding (SDG)**, which models each defect as a structured *(location, type, reason, importance)* tuple and formulates diagnosis as predicting a variable-length set of such tuples. SDG unifies two defect types, *artifacts* (i.e., image-intrinsic visual flaws) and *misalignments* (i.e., prompt-conditioned semantic errors), within a single instance space, naturally supporting images with multiple heterogeneous defects. To make this formulation trainable and measurable, we construct **SDG-30K**, a 30,096-image dataset with box-grounded artifact and misalignment annotations across four modern T2I generators, and define **SDG-Eval**, a dedicated protocol for structured defect-set evaluation.

Beyond diagnosis, SDG also provides a natural interface for model alignment: boxes define spatial support, defect types and reasons provide semantic diagnoses, and importance scores calibrate reward strength. We convert SDG predictions into box-derived, importance-weighted spatial reward maps and post-train diffusion models with our proposed **BoxFlow-GRPO**, enabling spatially targeted alignment beyond scalar preference optimization. Code and released assets are available online.[1] See Appendix D.6 for details. Overall, this paper makes following main **contributions**:

- We introduce **Structured Defect Grounding (SDG)**, an instance-level representation that formulates dense T2I diagnosis as variable-cardinality set prediction over structured *(location, type, reason, importance)* tuples.
- We construct **SDG-30K**, a 30,096-image box-grounded defect dataset across four modern T2I generators, and define **SDG-Eval** for image-level and defect-level evaluation.
- We develop a diagnosis-to-alignment framework where a VLM-based **SDG detector** predicts structured defect sets and **BoxFlow-GRPO** converts them into importance-weighted spatial rewards for diffusion model alignment.
- Extensive experiments demonstrate that our SDG detector outperforms leading proprietary VLMs on structured defect grounding, while SDG-guided rewards improve T2I alignment and support more faithful, actionable image refinement.

[1] https://github.com/nianbai006/SDG

2 Related Work

From scalar evaluation to dense T2I feedback. Most T2I evaluators produce scalar scores [Wu et al., 2023b,a, Xu et al., 2023, Kirstain et al., 2023]. RichHF [Liang et al., 2024] introduces heatmap-based dense feedback, ImageDoctor [Guo et al., 2025] predicts heatmaps via a VLM-plus-decoder architecture, and HEIE [Yang et al., 2025] and MagicMirror [Wang et al., 2025a] further enriches feedback with hierarchical explanations and fine-grained artifact taxonomy, respectively. Beyond T2I evaluation, HAD [Wang et al., 2024] and AbHuman [Fang et al., 2024] demonstrate box-level supervision for human-body artifact detection, and LEGION [Kang et al., 2025] combines mask-level localization with natural-language explanations. However, they do not provide a unified instance-level formulation that jointly localizes *artifact* and *misalignment* with localized descriptions.

Structured spatial reasoning in VLMs. Modern VLMs increasingly support explicit spatial structure within an autoregressive generation framework. Qwen2.5-VL [Bai et al., 2025b] supports box- and point-based grounding with structured outputs, while Qwen3-VL [Bai et al., 2025a] further strengthens image-grounded reasoning and spatial understanding. At a finer granularity, SimpleSeg [Song et al., 2026] reformulates segmentation as point-sequence generation entirely in language space. These advances motivate our use of VLMs to generate structured defect instances.

RL for diffusion alignment and image refinement. RL has become a key tool for aligning diffusion models with human preferences. DDPO [Black et al., 2023] frames denoising as a multi-step MDP and applies policy gradients, Diffusion-DPO [Wallace et al., 2024] adapts direct preference optimization to diffusion training, and Flow-GRPO [Liu et al., 2025] applies group relative policy optimization [Shao et al., 2024] to flow-matching models. Related work adapts diffusion RL across denoising steps through entropy regulation [Yan et al., 2025] and stage-specific reward objectives [Yan et al., 2026]. TempFlow-GRPO [He et al., 2026] uses trajectory branching and noise-aware timestep weighting to assign credit across denoising steps. ImageDoctor [Guo et al., 2025] proposes extending scalar rewards to spatially varying dense reward maps. On the correction side, ReflectionFlow [Zhuo et al., 2025] and HumanRefiner [Fang et al., 2024] demonstrate that defect-guided refinement can improve generation quality. To our knowledge, our work is the first to realize spatially dense advantages in diffusion RL, and further uses structured dense feedback to guide image refinement.

3 SDG-30K: Dataset and Evaluation

3.1 Task Formulation

Given a generated image I and its prompt T, the goal is to predict a variable-cardinality set of structured defect instances $\{(b_i, t_i, r_i, s_i)\}_{i=1}^{K}$. Here, $K \in \mathbb{N}$ denotes the number of defects in the image. The i-th instance consists of (1) a quantized bounding box $b_i \in \{0, \ldots, 1000\}^4$; (2) a defect type $t_i \in \{\textit{artifact}, \textit{misalignment}\}$; (3) a free-form natural-language description r_i; and (4) an integer importance score $s_i \in \{1, \ldots, 100\}$ reflecting the defect's perceptual impact on the overall image quality. The detailed definitions and distinctions of these defect types are provided in Appendix A.

3.2 Evaluation Metrics

We define **SDG-Eval**, an evaluation protocol that scores SDG at both image and defect levels for each defect type $t \in \{\textit{artifact}, \textit{misalignment}\}$; full definitions are provided in Appendix C. At the image level, **DetTypeF1** measures whether a defect type is present on each test image, while **ClnAcc** reports true-negative accuracy on images with no ground-truth instance of that type ($GT{=}0$). At the defect level, predictions are matched to ground-truth instances of the same type via class-aware Hungarian matching by IoU. We report **BoxF1**@0.1 and **BoxF1**@0.5 for localization, **DescCos**@0.1 using Qwen3-Embedding-0.6B for description alignment, and **ImpAcc**@0.1 as normalized importance accuracy over matched pairs.

3.3 Dataset Construction

Figure 2 (left) illustrates the dataset construction pipeline. SDG-30K contains **30,096** images at $1024{\times}1024$ resolution, generated from Pick-a-Pic prompts [Kirstain et al., 2023] using four T2I generators ($\sim$7.5K each): FLUX.2-dev [Black Forest Labs, 2025], Z-Image-Turbo [Cai et al., 2025], LongCat-Image [Ma et al., 2025a], and SANA-1.5-1.6B [Xie et al., 2025].

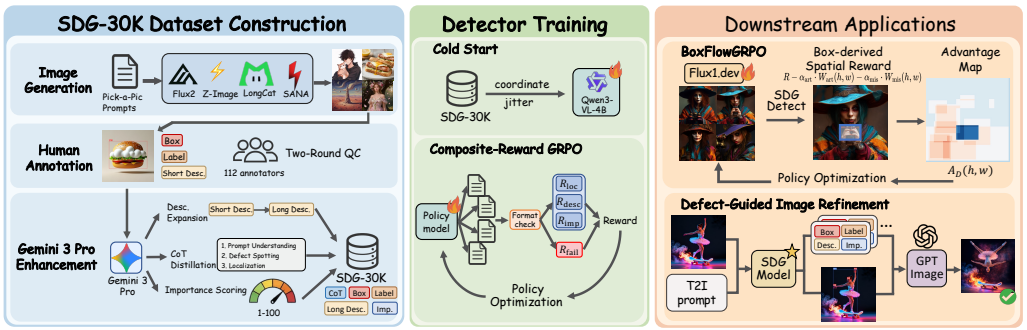

Figure 2: Overview of the SDG framework. **Left**: SDG-30K construction combines human box-level defect annotation across four T2I generators with Gemini 3 Pro enhancement. **Middle**: Two-stage SDG detector training via SFT with coordinate jitter followed by GRPO with a format-gated composite reward. **Right**: Downstream applications. BoxFlow-GRPO converts SDG detections into box-derived spatial rewards for diffusion alignment; defect-guided refinement feeds box overlays and text feedback to GPT-Image-1.5 for image refinement.

Table 1: Comparison with related datasets. SDG-30K is the first to jointly cover *artifact* and *misalignment* within a unified instance-level annotation space with natural-language reasons.

Dataset	Images	Spatial Ann.	Defect Types	NL Reason	Instance-Level	Generators
RichHF-18K [Liang et al., 2024]	18K	Heatmap (point-derived)	Art. + Mis.	×	×	SD-family
HAD [Wang et al., 2024]	37K	Bounding box	Artifact (human body)	×	✓	SDXL, DALL-E, MJ
AbHuman [Fang et al., 2024]	56K	Bounding box	Artifact (human body)	×	✓	SDXL
MagicMirror [Wang et al., 2025a]	343K	Taxonomy label	Artifact	×	×	FLUX, SD3, Kolors, MJ, etc.
LEGION [Kang et al., 2025]	12K	Pixel mask	Artifact	✓	✓	SD, FLUX, GLIDE, etc.
SDG-30K (Ours)	30K	Bounding box	Art. + Mis.	✓	✓	FLUX, Z-Image, LongCat, SANA

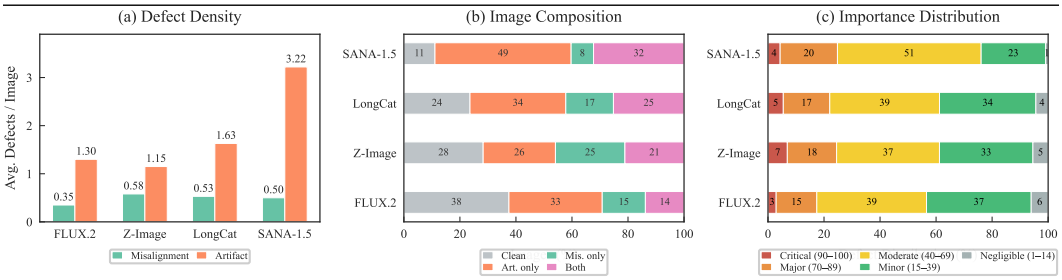

Figure 3: SDG-30K dataset statistics across four generators. (a) Average number of defects per image by type. (b) Image composition showing the proportion of clean, artifact-only, misalignment-only, and both-type images. (c) Distribution of defect importance scores across five severity tiers.

112 annotators (∼**1,085** person-hours) examined each prompt–image pair, drew bounding boxes, assigned top-level labels, and wrote concise Chinese descriptions (≤30 characters). Two rounds of review resolved disagreements. To quantify inter-annotator agreement, 16 held-out annotators who did not participate in the original labeling independently re-annotated the test set. The resulting BoxF1@0.5 of 0.278 (*artifact*) and 0.409 (*misalignment*) against the primary annotations serves as a human upper bound for localization (Table 2). Detailed annotation rules are in Appendix A.

Human annotations provide boxes, labels, and short Chinese descriptions but lack reasoning traces and importance scores. We use Gemini 3 Pro [Team et al., 2023] to augment each sample with three components: (1) *description expansion*: from Chinese to detailed English; (2) *reasoning trace distillation*: as a three-step CoT trace (prompt understanding, defect spotting, localization); and (3) *importance scoring*: based on a multi-criteria rubric. Details are in Appendix B.2. Human calibration on 500 distinct defect boxes yields 64.8% exact-tier agreement, 97.2% agreement within one adjacent tier, and Spearman correlation 0.786 with Gemini scores (Appendix E.6). Table 1 compares SDG-30K with related datasets.

3.4 Statistics and Splits

Dataset split. The split follows the Pick-a-Pic prompt partition (prompt-disjoint): **28,945** training and **1,151** test images, ensuring that no prompt appears in both splits.

Defect frequency and image composition. Figure 3 summarizes the dataset statistics. *Artifact* instances are more frequent than *misalignment* across all four generators, with SANA-1.5 exhibiting the highest artifact frequency (3.22 per image). Weighting the per-generator proportions in Figure 3b by image count, approximately 25.1% of images are defect-free, 35.5% contain only artifacts, 16.3% contain only misalignments, and 23.1% contain both. This confirms that localized failures remain common in recent T2I generators, motivating dense, instance-level feedback.

Importance distribution. Importance scores estimate each defect's impact on image quality and prompt faithfulness, enabling failure prioritization and reward weighting. They are centered around the Moderate tier (40–69), mostly between 30 and 80 (Figure 3c), with rare extremes and balanced coverage for training.

4 Structured Defect Grounding Framework

In this section, we instantiate SDG as a diagnosis-to-alignment framework built around the structured-defect representation. We first present the SDG detector, which produces structured defect sets that inherently align with VLM output formats. We then introduce BoxFlow-GRPO to convert SDG's outputs into box-derived spatial rewards for diffusion alignment.

4.1 SDG Detector

We formulate defect diagnosis as a structured vision-language generation task. Given a generated image and its prompt, SDG first produces a reasoning trace $\mathcal{R}$ and then emits a structured defect set $\mathcal{D}$, where each instance specifies its location, type, reason, and importance.

The SDG detector is trained in two stages (Figure 2, middle). Supervised fine-tuning (SFT) teaches the model to follow the defect-grounding instruction and emit the required structured format, while group relative policy optimization (GRPO) [Shao et al., 2024] further improves localization, description consistency, and importance estimation under a format-validity gate.

Cold Start. We fine-tune Qwen3-VL-4B-Instruct on the SDG-30K training split (Section 3.3). For each image-prompt pair (I, T), the target sequence concatenates the Gemini-distilled reasoning trace and the structured defect set, denoted as $y = [\mathcal{R}; \mathcal{D}]$. We optimize the model with the standard teacher-forced negative log-likelihood over the target tokens:

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid I, T, y_{<t}). \tag{1}$$

To reduce sensitivity to exact coordinate values, we apply *coordinate jitter* to the `<answer>` segment when preparing the SFT data. For each box $[x_0, y_0, x_1, y_1]$ in normalized $[0, 1000]$ space, each coordinate is independently perturbed by $\delta \sim \mathcal{U}(-10, 10)$, then clamped to $[0, 1000]$ with valid box ordering enforced. The `<think>` segment is kept unchanged because it captures high-level visual reasoning rather than exact numerical coordinates. With independently sampled jitter offsets for the repeated training examples, the model observes multiple plausible coordinate variants of the same defect, improving tolerance to minor spatial variation and providing a more robust SFT initialization for subsequent GRPO training.

Composite-Reward GRPO. After SFT, we apply GRPO to directly optimize the structured output using rewards for spatial accuracy, description consistency, and importance estimation. For each prompt, we sample S=8 responses, compute the composite reward R_s for each response, and form group-normalized advantages $A_s = (R_s - \bar{R})/\sigma_R$. The policy is then optimized with the clipped GRPO objective:

$$\mathcal{L}_{\mathrm{GRPO}} = -\mathbb{E}[\min(\rho_s A_s,\ \mathrm{clip}(\rho_s, 1{-}\epsilon, 1{+}\epsilon)\, A_s)] + \beta\, \mathrm{KL}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}), \tag{2}$$

where $\rho_s = \pi_\theta(y_s \mid I, T)/\pi_{\mathrm{old}}(y_s \mid I, T)$ is the importance ratio, π_{old} is the rollout policy, π_{ref} is the fixed reference policy, and β=0.01.

The composite reward is gated by a format-validity check:

$$R = \begin{cases} \lambda_{\mathrm{loc}}\, R_{\mathrm{loc}} + \lambda_{\mathrm{desc}}\, R_{\mathrm{desc}} + \lambda_{\mathrm{imp}}\, R_{\mathrm{imp}}, & \text{if } \mathrm{Format}(y) = \mathrm{true}, \\ R_{\mathrm{fail}}, & \text{otherwise}, \end{cases} \tag{3}$$

where $\lambda_{\text{loc}}, \lambda_{\text{desc}}, \lambda_{\text{imp}} \geq 0$ with $\lambda_{\text{loc}} + \lambda_{\text{desc}} + \lambda_{\text{imp}} = 1$ control the relative importance of each reward component, and $R_{\text{fail}} < 0$ is a fixed penalty for malformed outputs. The format predicate $\text{Format}(y)$ verifies that the response contains well-formed reasoning and answer delimiters, a parseable JSON defect list, and geometrically valid bounding boxes (i.e., $x_0 < x_1$ and $y_0 < y_1$).

The three reward components are defined as follows. R_{loc} measures spatial grounding accuracy: predicted and ground-truth boxes are matched via the Hungarian algorithm with Distance-IoU (DIoU) cost, and the reward reflects match quality with explicit penalties for false negatives and false positives. R_{desc} measures description consistency via embedding cosine similarity (Qwen3-Embedding-0.6B), linearly mapped to $[0, 1]$ and averaged over matched pairs. R_{imp} measures importance estimation accuracy via a clipped absolute-error metric. In practice, we set λ_{loc}=0.6, λ_{desc}=0.25, λ_{imp}=0.15, and R_{fail}=-1. Detailed formulas and the treatment of boundary cases are provided in Appendix D.

4.2 SDG-Guided BoxFlow-GRPO

Once the SDG detector is trained, we translate its dense feedback into spatially varying rewards to guide the reinforcement learning of diffusion models. The closest prior work is ImageDoctor's DenseFlow-GRPO [Guo et al., 2025], which extends Flow-GRPO [Liu et al., 2025] with a heatmap signal. In its public training code[2], each image is assigned an image-level scalar reward R, normalized across the prompt group into a scalar advantage A; the predicted heatmap $H \in [0, 1]^{H \times W}$ enters only as a multiplicative mask on the policy-gradient loss, $\mathcal{L} \propto -A \cdot \rho \cdot (1-H)$, where ρ is the importance ratio. The gradient signal is therefore still driven by an image-level scalar, with the heatmap merely down-weighting locations the detector flags as defective rather than contributing a true per-location advantage at the latent grid.

Motivated by this gap, we implement BoxFlow-GRPO around two design choices: box-derived spatial rewards and spatially normalized per-location advantages.

Given a base scalar reward R from any predefined reward model[3] and the defect bounding boxes detected by our SDG model, we construct a spatially varying reward map in the latent space. For each latent spatial location (h, w), weighted masks $W_{\text{art}}, W_{\text{mis}}$ aggregate the predicted importance of all defect boxes covering that location, and the per-location reward subtracts type-specific penalties from the scalar reward:

$$\begin{aligned} W_{\text{type}}(h, w) &= \max_{k \in \mathcal{B}_{\text{type}}(h,w)} \hat{s}_k / 100, \\ \alpha_{\text{art}} = c_{\text{art}} \cdot \sigma_R^{(\text{group})}, &\quad \alpha_{\text{mis}} = c_{\text{mis}} \cdot \sigma_R^{(\text{group})}, \\ R_D(h, w) &= R - \alpha_{\text{art}} W_{\text{art}}(h, w) - \alpha_{\text{mis}} W_{\text{mis}}(h, w). \end{aligned} \tag{4}$$

where $\mathcal{B}_{\text{type}}(h, w)$ denotes the set of boxes of a given type covering location (h, w) and $\hat{s}_k \in \{1, \ldots, 100\}$ is the importance predicted by SDG for box k (Section 3.2). We set $W_{\text{type}}(h, w) = 0$ when no box of that type covers (h, w). We set c_{art}=0.5 and c_{mis}=0.05 to approximately normalize spatial penalty mass: an average misalignment box covers 9.90× the area of an artifact box. Among three coefficient settings evaluated after 500 training steps, only this setting improves all five metrics relative to Base (Appendix E.7). Scaling by $\sigma_R^{(\text{group})}$ adapts penalties to reward variability, while importance weights prioritize severe defects.

After constructing R_D, per-location advantages are computed by normalizing across the K samples within each prompt group at each spatial location:

$$A_D^{(k)}(h, w) = \frac{R_D^{(k)}(h, w) - \mu_D(h, w)}{\sigma_D(h, w) + \epsilon}, \tag{5}$$

where $\mu_D(h, w)$ and $\sigma_D(h, w)$ are the mean and standard deviation of $\{R_D^{(k)}(h, w)\}_{k=1}^K$ computed over the K=8 samples in the prompt group, and ϵ is a small constant for numerical stability. Defining the per-location likelihood ratio $\rho_t^{(k)}(h, w) = \pi_\phi(x_{t-1,h,w}^{(k)} \mid x_t^{(k)}, c) / \pi_{\phi_{\text{old}}}(x_{t-1,h,w}^{(k)} \mid x_t^{(k)}, c)$, the BoxFlow-GRPO objective is (KL regularization omitted for brevity):

$$\mathcal{J}_{\text{BoxFlow}}(\phi) = \frac{1}{KTHW} \sum_{k,t,h,w} \min\Big(\rho_t^{(k)}(h, w)\, A_D^{(k)}(h, w),\ \text{clip}\Big(\rho_t^{(k)}(h, w), 1-\varepsilon, 1+\varepsilon\Big)\, A_D^{(k)}(h, w)\Big) \tag{6}$$

[2] https://github.com/EthanG97/ImageDoctor

[3] We use UnifiedReward-2.0 (UR2) [Wang et al., 2025b] in our experiments

where T is the number of denoising steps and H, W are the latent spatial dimensions. Compared with a scalar-advantage implementation, this objective preserves spatial variation in both the advantage and the likelihood ratio.

5 Experiments

We evaluate SDG from three complementary perspectives: (1) defect grounding quality on SDG-30K, (2) diffusion alignment with structured-feedback rewards via BoxFlow-GRPO, and (3) defect-guided image refinement driven by structured feedback.

5.1 Defect Grounding Results

5.1.1 Setup

Implementation. We fine-tune Qwen3-VL-4B-Instruct on $16\times$A100-80G GPUs with DeepSpeed ZeRO-2 for one epoch over a $3\times$ pre-augmented SFT corpus (effective batch size 16, learning rate 3×10^{-5}, cosine schedule). The vision encoder is frozen throughout SFT; Table 4 shows that unfreezing it degrades grounding quality at this dataset scale. For GRPO, we train on 2×8 A100 GPUs (16 total) with $S{=}8$ sampled responses per prompt (temperature 1.0, top-p 0.85) for 2 epochs at learning rate 5×10^{-6}.

Baselines. We compare **SDG** (SFT and GRPO variants) against zero-shot GPT-5.4 and Gemini 3 Pro, both prompted with the same structured output format but without task-specific training. A human reference from 16 independent re-annotators provides a localization upper bound.

5.1.2 Main Results

Table 2: Defect grounding results on SDG-30K test set. Human row: localization upper bound from 16 independent re-annotators. **Bold**: best; underline: second best.

	Artifact						*Misalignment*					
	Image-Level		Defect-Level				Image-Level		Defect-Level			
	All	$GT{=}0$	$GT{>}0$				All	$GT{=}0$	$GT{>}0$			
Model	DetTypeF1	ClnAcc	BoxF1@0.1	BoxF1@0.5	DescCos@0.1	ImpAcc@0.1	DetTypeF1	ClnAcc	BoxF1@0.1	BoxF1@0.5	DescCos@0.1	ImpAcc@0.1
GPT-5.4	0.736	0.118	0.237	0.035	0.868	0.812	0.654	0.513	0.457	0.292	0.837	0.820
Gemini 3 Pro	0.733	0.076	0.337	0.200	0.885	0.824	**0.696**	0.672	0.464	0.307	0.891	0.851
SDG (SFT)	**0.776**	**0.697**	0.402	0.255	**0.904**	0.883	0.636	**0.799**	0.499	0.376	**0.893**	0.892
SDG (GRPO)	0.772	0.560	**0.404**	**0.263**	**0.904**	**0.887**	0.675	0.732	**0.511**	**0.387**	0.888	**0.893**
Human	0.792	0.774	0.462	0.278	–	–	0.712	0.818	0.563	0.409	–	–

Quantitative analysis. Table 2 presents defect grounding results for artifact and misalignment separately. GRPO achieves the strongest BoxF1@0.5 (0.263/0.387) and importance accuracy (0.887/0.893), while SFT remains competitive and attains the highest clean-image accuracy (0.697/0.799). Compared with zero-shot VLM baselines, both SDG variants substantially reduce localization error: GPT-5.4 obtains reasonable image-level artifact detection (F1 0.736) but weak precise artifact localization (BoxF1@0.5 0.035), and Gemini 3 Pro improves localization (0.200/0.307) but remains below SDG. GRPO narrows the gap to the human reference on both artifact (0.263 vs. 0.278) and misalignment (0.387 vs. 0.409), while maintaining high description cosine similarity.

Qualitative comparison. Figure 4 shows qualitative comparisons on SDG-30K across three representative cases: an *artifact*-only image (top), a *misalignment*-only image (middle), and a clean image (bottom). SDG produces instance-level bounding boxes with per-defect descriptions, accurately localizing defects in the first two rows and correctly recognizing the clean case as defect-free. In contrast, ImageDoctor tends to predict heatmap responses on faces and hands even when they are anatomically correct, and struggles to detect prompt-conditioned misalignments.

Cross-dataset generalization. To evaluate generalization, we test SDG on the RichHF-18K test set without any fine-tuning on this dataset, comparing against ImageDoctor [Guo et al., 2025] which is trained on RichHF-18K. Since RichHF-18K provides heatmap annotations, we threshold ImageDoctor heatmaps at two operating points (0.10 and 0.33) for image-level evaluation.

As shown in Table 3, SDG achieves substantially higher misalignment F1 (0.655 vs. 0.250/0.007), demonstrating that structured defect grounding generalizes to unseen data and captures prompt-conditioned misalignments more effectively than heatmap-based methods. ImageDoctor achieves



Figure 4: Qualitative comparison on SDG-30K. Rows from top to bottom: artifact-only, misalignment-only, and clean images.

Table 3: Image-level defect detection on RichHF-18K. ImageDoctor is trained in-domain; SDG is zero-shot.

	Artifact			*Misalignment*		
Method	Prec.	Rec.	F1	Prec.	Rec.	F1
ImageDoctor @0.10	0.965	0.939	0.952	0.983	0.143	0.250
ImageDoctor @0.33	0.977	0.762	0.856	1.000	0.004	0.007
SDG (ours)	0.970	0.698	0.812	0.951	0.500	0.655

Table 4: Key ablation results on SDG-30K. Full metrics are in Table 7.

Variant	Art. BoxF1@0.5	Mis. BoxF1@0.5
SFT	0.255	0.376
GRPO	0.263	0.387
SFT w/o step 1	0.218	0.329
SFT w/o step 3	0.253	0.281
GRPO w/o CoT	0.288	0.352
SFT unfreeze ViT	0.203	0.305
SFT w/o jitter	0.253	0.360

higher artifact F1 at the loose threshold (0.952), as expected from in-domain training, but its misalignment recall is notably poor (0.143/0.004), suggesting that heatmap-based representations struggle with prompt-conditioned misalignment in this setting.

5.1.3 Ablation Study

Table 4 summarizes the main ablation trends, with full metrics in Appendix E. GRPO improves localization over SFT (artifact/misalignment BoxF1@0.5: 0.263/0.387 vs. 0.255/0.376), confirming that policy optimization refines spatial precision. Removing CoT steps hurts prompt-conditioned misalignment more than artifact detection, and removing the reasoning trace from GRPO lowers misalignment BoxF1@0.5 from 0.387 to 0.352. Unfreezing the vision encoder substantially degrades localization, while coordinate jitter mainly improves image-level robustness (full results in Table 7).

5.2 Downstream Applications

5.2.1 BoxFlow-GRPO

Setup. We apply the dense reward construction of Section 4.2 to FLUX.1-dev, following Flow-GRPO [Liu et al., 2025]. The scalar base reward is UnifiedReward-2.0 (UR2) [Wang et al., 2025b]. We use the GRPO checkpoint of the SDG detector (Qwen3-VL-4B-Instruct) to identify defects and construct a dense reward by subtracting importance-weighted artifact and misalignment penalties at covered latent locations. Training uses Pick-a-Pic [Kirstain et al., 2023] prompts held out from SDG-30K (prompt-disjoint), for 500 optimization steps at 512×512 resolution on $8\times$A100 GPUs with LoRA (rank 64, α=128) and learning rate 3×10^{-4}. We evaluate on DrawBench [Saharia et al., 2022] using PickScore [Kirstain et al., 2023], CLIPScore [Hessel et al., 2021], HPSv3 [Ma et al.,



Figure 5: Qualitative comparison of BoxFlow-GRPO.

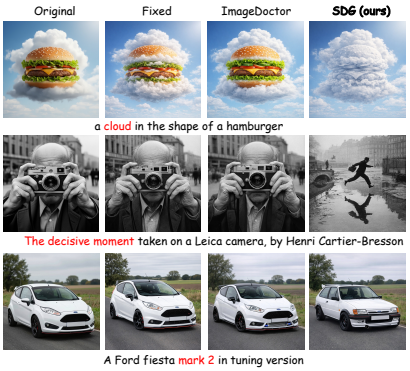

Figure 6: Qualitative comparison of defect-guided image refinement via GPT-Image-1.5.

Table 5: Downstream performance of BoxFlow-GRPO and baselines. Parentheses show changes relative to Base (green: improvement; red: regression); Avg averages these changes. Only BoxFlow-GRPO improves all five dimensions.

Method	Reward	PickScore	CLIPScore	HPSv3	DeQA	$P(\text{real})$	Avg
Base	–	22.84	0.912	11.75	0.876	0.211	–
Flow-GRPO	UR2	22.78 (-0.3%)	0.908 (-0.4%)	11.27 (-4.1%)	**0.882** (+0.7%)	0.149 (-29.4%)	-6.7%
	ImageDoctor	**23.17** (+1.4%)	0.937 (+2.7%)	**12.23** (+4.1%)	0.857 (-2.2%)	0.201 (-4.7%)	+0.3%
DenseFlow-GRPO	ImageDoctor	22.88 (+0.2%)	**0.945** (+3.6%)	11.57 (-1.5%)	0.864 (-1.4%)	0.199 (-5.7%)	-1.0%
BoxFlow-GRPO (ours)	UR2+SDG	22.91 (+0.3%)	0.915 (+0.3%)	12.14 (+3.4%)	0.877 (+0.1%)	**0.228** (+8.1%)	**+2.4%**

2025b], DeQA [You et al., 2025], and the real-image probability $P(\text{real})$ from Forensic-Chat [Lin et al., 2025], obtained by a 2-class softmax over its "real" vs. "fake" token logits.

Results. Table 5 shows that baseline RL variants can increase reward-model scores while drifting toward illustration- or anime-like outputs, with lower $P(\text{real})$ than Base. BoxFlow-GRPO achieves the best average relative change ($+2.4\%$) and highest $P(\text{real})$ (0.228), while maintaining competitive preference and quality scores. Figure 5 illustrates reduced artifacts and misalignment with preserved realism. On Qwen-Image, after 1,200 training steps, BoxFlow-GRPO again achieves the best average relative improvement ($+1.1\%$; Appendix E.9).

5.2.2 Defect-Guided Image Refinement

Setup. We use the GRPO checkpoint of the SDG detector to diagnose defects, then provide GPT-Image-1.5 with a box overlay and structured text feedback. We compare it with **Fixed** caption-only editing and **ImageDoctor** heatmap-based feedback, using the same editor for all methods. Two annotators independently and blindly compare paired outputs on 873 valid samples retained after GPT-Image-1.5 filtering, assigning Good/Same/Bad labels following HunyuanImage 3.0 [Cao et al., 2025], where Good means SDG is preferred.

Table 6: GSB rates between SDG and baseline methods over 873 valid samples.

Comparison	Good (%)	Same (%)	Bad (%)
SDG vs ImageDoctor	11.00	85.11	3.90
SDG vs Fixed	10.31	86.94	2.75

Results. Table 6 shows higher Good than Bad rates against ImageDoctor (11.00% vs. 3.90%) and Fixed (10.31% vs. 2.75%). The high Same rate (85.1–86.9%) reflects the strong editor and high-quality inputs. Re-diagnosing paired images with the same frozen SDG detector shows that SDG-guided refinement reduces detected boxes by 37.55% and total importance by 42.37%, the largest reductions among the three methods (Appendix E.8). This complements the human preference results with detector-level evidence of effective refinement. Qualitative examples appear in Figure 6 and Appendix E.5.

6 Conclusion

We presented **Structured Defect Grounding (SDG)**, an instance-level formulation for dense text-to-image (T2I) feedback that models each defect as a *(location, type, reason, importance)* tuple and casts diagnosis as variable-cardinality structured set prediction. This formulation supports **SDG-30K**, a 30K-image box-grounded dataset across four modern T2I generators, and **SDG-Eval** for structured defect-set evaluation. It also enables a diagnosis-to-alignment pipeline in which a VLM-based detector predicts defect sets and **BoxFlow-GRPO** converts them into dense spatial rewards for diffusion alignment. Experiments demonstrate that SDG outperforms leading proprietary VLMs in zero-shot defect grounding, improves T2I alignment, and supports actionable defect-guided image refinement, making it a practical interface for evaluating and improving modern generative models.

Acknowledgments and Disclosure of Funding

This work was supported by the SSTIC Grants (KJZD20230923115106012, KJZD20230923114916032, GJHZ20240218113604008).

References

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

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. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Black Forest Labs. FLUX.2. https://github.com/black-forest-labs/flux2, 2025.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

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 *Forty-first international conference on machine learning*, 2024.

Guian Fang, Wenbiao Yan, Yuanfan Guo, Jianhua Han, Zutao Jiang, Hang Xu, Shengcai Liao, and Xiaodan Liang. Humanrefiner: Benchmarking abnormal human generation and refining with coarse-to-fine pose-reversible guidance. In *European Conference on Computer Vision*, pages 201–217. Springer, 2024.

Yuxiang Guo, Jiang Liu, Ze Wang, Hao Chen, Ximeng Sun, Yang Zhao, Jialian Wu, Xiaodong Yu, Zicheng Liu, and Emad Barsoum. Imagedoctor: Diagnosing text-to-image generation via grounded image reasoning. *arXiv preprint arXiv:2510.01010*, 2025.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. TempFlow-GRPO: When timing matters for GRPO in flow models. In *International Conference on Learning Representations*, 2026.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 7514–7528, 2021.

Hengrui Kang, Siwei Wen, Zichen Wen, Junyan Ye, Weijia Li, Peilin Feng, Baichuan Zhou, Bin Wang, Dahua Lin, Linfeng Zhang, et al. Legion: Learning to ground and explain for synthetic image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18937–18947, 2025.

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. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, et al. Rich human feedback for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19401–19411, 2024.

Kaiqing Lin, Zhiyuan Yan, Ruoxin Chen, Junyan Ye, Ke-Yue Zhang, Yue Zhou, Peng Jin, Bin Li, Taiping Yao, and Shouhong Ding. Seeing before reasoning: A unified framework for generalizable and explainable fake image detection. *arXiv preprint arXiv:2509.25502*, 2025.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, Xunliang Cai, et al. Longcat-image technical report, 2025a. URL https://arxiv.org/abs/2512.07584.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. pages 15086–15095, 2025b.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Tianhui Song, Haoyu Lu, Hao Yang, Lin Sui, Haoning Wu, Zaida Zhou, Zhiqi Huang, Yiping Bao, Y Charles, Xinyu Zhou, et al. Towards pixel-level vlm perception via simple points prediction. *arXiv preprint arXiv:2601.19228*, 2026.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

Jia Wang, Jie Hu, Xiaoqi Ma, Hanghang Ma, Yanbing Zeng, and Xiaoming Wei. Magicmirror: A large-scale dataset and benchmark for fine-grained artifacts assessment in text-to-image generation. *arXiv preprint arXiv:2509.10260*, 2025a.

Kaihong Wang, Lingzhi Zhang, and Jianming Zhang. Detecting human artifacts from text-to-image models. *arXiv preprint arXiv:2411.13842*, 2024.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025b.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023a.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2096–2105, 2023b.

Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, Bingchen Liu, Daquan Zhou, and Song Han. SANA 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 68578–68598, 2025.

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. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

RenYe Yan, Jikang Cheng, Yaozhong Gan, Shikun Sun, You Wu, Yunfan Yang, Liang Ling, Jinlong Lin, Yeshuang Zhu, Jie Zhou, Jinchao Zhang, Junliang Xing, Yimao Cai, and Ru Huang. Entropy-adaptive diffusion policy optimization with dynamic step alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1924–1934, 2025.

Renye Yan, Jikang Cheng, You Wu, Wei Peng, Zongwei Wang, Ling Liang, and Yimao Cai. Explore or converge? stage-guided per-step optimization for diffusion models. *arXiv preprint arXiv:2608.06768*, 2026.

Fan Yang, Ru Zhen, Jianing Wang, Yanhao Zhang, Haoxiang Chen, Haonan Lu, Sicheng Zhao, and Guiguang Ding. Heie: Mllm-based hierarchical explainable aigc image implausibility evaluator. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3856–3866, 2025.

Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching large language models to regress accurate image quality scores using score distribution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14483–14494, 2025.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15329–15339, 2025.

A Annotation Guidelines

This appendix describes the annotation protocol used to construct the SDG-30K dataset. The annotation task was conducted as data curation for generated images rather than as a user study. Annotators were compensated above the local minimum wage in the data-collection location.

A.1 Interface and Workflow

The annotation interface consists of two synchronized panels: a raw image canvas on which annotators draw the final bounding boxes and a metadata panel showing the generation prompt together with annotation controls.

Each image is processed in two passes. In the *initial annotation pass*, annotators examine the prompt-image pair, draw defect boxes from scratch, assign top-level labels, and write concise descriptions for each confirmed defect. In the *global scan pass*, annotators re-examine the full image to identify missed defects, adjust box boundaries, and improve annotation completeness. No machine-generated candidate boxes or defect descriptions are shown during manual annotation.

A.2 Defect-Type Taxonomy

We define two top-level defect categories, each with fine-grained subtypes.

Misalignment captures inconsistencies between the generation prompt and image content. Subtypes include missing objects, extra objects, attribute mismatches (color, count, material), spatial-relation errors, action mismatches, and style mismatches. For example, a prompt requesting "two cats on a red sofa" paired with an image showing three cats constitutes a count-based misalignment.

Artifact captures visual plausibility defects independent of the prompt. Subtypes include anatomical distortions (e.g., malformed hands), geometric deformations, texture abnormalities, edge and contour defects, text garbling, and lighting inconsistencies. For example, fused fingers or impossible joint angles constitute anatomical artifacts regardless of the prompt content.

The taxonomy explicitly separates semantic mismatch from visual corruption: green eyes are a misalignment when the prompt requests blue eyes, while geometrically distorted eyes are artifacts. Intentional artistic stylization (e.g., cubist deformation) is not labeled as a defect; only non-intentional structural or perceptual inconsistencies qualify.

A.3 Boxing and Description Principles

Although bounding boxes are coarser than pixel masks, they provide a practical trade-off between spatial specificity, annotation efficiency, and compatibility with VLM outputs. Bounding boxes should tightly cover the defective location while excluding unnecessary background. Each box corresponds to one concrete defect: when a large defect location contains multiple independent issues, annotators split them into separate boxes; when multiple small detections reflect a single coherent failure mode, they may be merged.

For each box, annotators provide a concise reason statement (target length $\leq$30 Chinese characters) that specifically describes the defect. Vague comments such as "looks wrong" are prohibited; descriptions must reference concrete visual evidence (e.g., "left hand has six fingers" or "hat mentioned in prompt is absent").

A.4 Clean Images

The annotation platform required at least one bounding box per submission. For images without defects, annotators drew a placeholder box and assigned an explicit *no-problem* tag. During post-processing, these samples are converted to empty defect sets, preserving them as valid negative examples for training and evaluation.

A.5 Prompt Interpretation Protocol

To ensure the correctness of prompt interpretation, annotators were instructed to consult search engines whenever a prompt involved unfamiliar named entities, cultural references, artistic styles, products, landmarks, or other domain-specific concepts. External search was used only to clarify the semantic background of the prompt, rather than to provide defect labels directly. This protocol reduces annotation noise caused by incomplete prior knowledge and improves the consistency of prompt-conditioned *misalignment* annotation.

A.6 Importance Scoring Rubric

Each defect instance receives an importance score (integer from 1 to 100) reflecting how much it affects overall image quality and caption faithfulness, assessed via a rubric-guided Gemini protocol. The four criteria are considered in order of priority:

1. *Visual prominence* — how easily a typical viewer can spot the defect at normal viewing distance.
2. *Semantic impact* — whether the defect changes the meaning, identity, or key content relative to the prompt.
3. *Area coverage* — larger defects affecting more of the image score higher.
4. *Location* — defects on the main subject or focal area score higher than those in the background.

Scores are grouped into five tiers: Critical (90–100), Major (70–89), Moderate (40–69), Minor (15–39), and Negligible (1–14). Importance annotations are obtained by prompting Gemini with the image, the original prompt, and the human-annotated defect set, and are used both as SFT supervision and as a GRPO reward signal.

B Prompt Templates

This appendix provides the full prompt templates used in SDG. The same user prompt (Appendix B.1) is used for SFT training, GRPO training, and inference. The system prompt uses the default Qwen chat template ("You are a helpful assistant.").

B.1 SFT / GRPO / Inference User Prompt

SDG User Prompt

```
You are an AI image quality evaluator. You will be given one image to analyze.
### Definitions
Misalignment: Areas where the image content does NOT match the text caption,
including:
- Missing objects: Objects mentioned in caption but not present in image
- Extra objects: Objects present in image but not mentioned in caption
- Wrong attributes: Incorrect color, size, material, count, or other properties
- Wrong spatial relationships: Incorrect positions, orientations, or
arrangements
Artifact: Visual defects in the generated image, including:
- Distorted anatomy: Malformed hands, extra/missing limbs, wrong number of
fingers
- Duplicated/missing parts: Repeated or absent body parts, objects
- Warped geometry: Perspective errors, impossible shapes
- Texture issues: Melted, smeared, or overly smooth textures
- Unnatural edges: Jagged, broken, or blurry boundaries
- Garbled text: Unreadable or malformed text/letters
- Lighting inconsistencies: Wrong shadows, reflections, or light sources
Text Caption: {caption}
Goal: Produce a detailed analysis of the image quality and output bounding boxes
with severity scores for all detected issues.
### Strict Output Rules
Output TWO blocks in this exact order:
```

```
1) <think> - Your detailed analysis
2) <answer> - JSON list of bounding boxes
### Think Format (STRICT)
<think>
### Step 1: Caption Understanding
- Briefly summarize what the caption requires (subject, key attributes, actions,
setting, style/composition if mentioned).
### Step 2: Visual Analysis & Defect Spotting (Issue Summary)
- Describe the quality issues you observe in the image.
- You MAY merge similar issues into a single bullet.
- Each bullet MUST include:
(a) the issue category (artifact or misalignment, or both if needed)
(b) what is affected (object/part)
(c) concrete visual evidence (what specifically looks wrong/missing/mismatched)
- Do NOT mention numeric coordinates.
### Step 3: Localization (Box-by-Box Grounding)
- Provide a detailed, precise localization statement for EACH defect instance
(one line per box).
- Do NOT mention numeric coordinates.
- Each localization line MUST include all of the following information in natural
language:
1) Anchor: the exact object/part involved
2) Position: image-based cues (image-left/right, upper/lower, center, near the
border)
3) Interaction cue (when applicable): holding/touching/overlapping/merging/etc.
4) Scale description: tiny localized detail, part-sized area, large area, or
whole image
5) Shape/orientation: compact, elongated, runs along a boundary, wraps around an
object
- Do NOT invent new defects; each line must correspond to exactly one defect
instance.
</think>
### Answer Format (for <answer>)
[{"box_2d": [x0, y0, x1, y1], "label": "misalignment"|"artifact",
"description": "brief description of the issue", "importance": N}]
Bounding box coordinates are in normalized 0-1000 space: [x0, y0, x1, y1].
If there are no issues, output an empty list.
### Importance Scoring
For EACH box, assign an integer importance score from 1 to 100:
- 90-100: Critical -- immediately obvious, ruins the image.
- 70-89: Major -- clearly visible at normal viewing distance.
- 40-69: Moderate -- noticeable on closer inspection.
- 15-39: Minor -- only visible on careful examination.
- 1-14: Negligible -- barely perceptible.
### Description Style Guide
- 18 to 45 words per description.
- Must mention the affected object/part.
- Must include at least one concrete evidence phrase.
- Do NOT use vague words like "weird/strange/bad".
- Do NOT include numeric coordinates in description.
Now analyze the image and produce your output:
```

B.2 Gemini Distillation Prompt

During data preparation, we use Gemini 3 Pro to translate Chinese defect descriptions into English, expand them with richer detail, and assign importance scores. The model receives the image, caption, and human-annotated defect boxes as hidden context. The full prompt is shown below.

Gemini Distillation Prompt

```
You are an AI image quality evaluator. You will be shown an image and asked to
identify quality issues.
Hidden context (DO NOT reveal):
The human annotations below are Ground Truth. You MUST keep the same number of
boxes, the same coordinates, and the same labels in the final JSON. You must
behave as if you discovered the issues independently from the image and caption.
Ground Truth annotations (DO NOT modify coordinates or labels; you may
rewrite/expand description in English):
- Artifact boxes: {artifact_bbox_text}
- Misalignment boxes: {misalignment_bbox_text}
Text Caption: {caption}
====== GLOBAL CONSTRAINTS (MUST FOLLOW) ======
1) NEVER mention annotations, boxes, ground truth, translation, or any external
hints.
2) Do NOT invent extra defects or remove any defect. The final JSON must contain
exactly the same set of boxes/labels as provided.
3) IMPORTANT: box_2d uses RELATIVE coordinates on a 0-1000 scale.
- Format: [x_min, y_min, x_max, y_max] (xyxy)
- You MUST output these coordinates exactly as given in the final JSON.
4) Do NOT write numeric coordinates in the <think> block.
5) Output MUST contain exactly TWO blocks in this order:
(1) <think> ... </think> (2) <answer> ... </answer>
6) <answer> MUST be a JSON list in xyxy format:
[{"box_2d":[x0,y0,x1,y1],"label":"artifact"|"misalignment",
"description":"...","importance":N}, ...]
====== IMPORTANCE SCORING (FIELD: "importance") ======
For EACH box, assign an integer importance score from 1 to 100:
- 90-100: Critical -- immediately obvious, ruins the image.
- 70-89: Major -- clearly visible at normal viewing distance.
- 40-69: Moderate -- noticeable on closer inspection.
- 15-39: Minor -- only visible on careful examination.
- 1-14: Negligible -- barely perceptible.
Scoring criteria (in order of priority):
a) Visual prominence -- How easily can a typical viewer spot this defect at
normal viewing distance?
b) Semantic impact -- Does this defect change the meaning, identity, or key
content of the image relative to the caption?
c) Area coverage -- Larger defects affecting more of the image score higher.
d) Location -- Defects on the main subject or focal area score higher than those
in background/periphery.
====== THINKING FORMAT (STRICT) ======
<think>
### Step 1: Caption Understanding
### Step 2: Visual Analysis & Defect Spotting (Issue Summary)
### Step 3: Localization (Box-by-Box Grounding)
</think>
====== DESCRIPTION STYLE GUIDE (FOR <answer>) ======
For EACH box in the final JSON, rewrite description into English with richer
detail:
- 18 to 45 words per description.
- Must mention the affected object/part.
- Must include at least one concrete evidence phrase.
- Do NOT use vague words like "weird/strange/bad".
- Do NOT mention translation or ground truth.
====== FEW-SHOT EXAMPLES ======
[EXAMPLE A: NON-EMPTY GT]
Caption: a rock musician sticking tongue out holding a sign that says hail Satan
<think>
### Step 1: Caption Understanding
```

```
- The caption asks for a rock musician sticking their tongue out while holding a
sign reading ‘‘hail Satan’’.
### Step 2: Visual Analysis & Defect Spotting
- Misalignment: The person does not convincingly match the specified celebrity
identity.
- Misalignment: The sign text is not rendered as the requested phrase.
- Artifact: Ring edges blend into skin and local finger texture appears
unnaturally bumpy.
### Step 3: Localization
- Box 1 (artifact): Ring-skin contact on image-right hand, tiny localized
detail.
- Box 2 (artifact): Mid-finger ring edge, compact boundary location.
- Box 3 (artifact): Below rings on image-right fingers, tiny skin patch.
- Box 4 (artifact): Image-left hand ring-to-finger junction, tiny localized.
- Box 5 (misalignment): Upper-central face location, part-sized.
- Box 6 (misalignment): Central sign surface with lettering, part-sized.
</think>
<answer>
[{‘‘box_2d’’:[777,604,796,627], ‘‘label’’:‘‘artifact’’,
‘‘description’’:‘‘Ring appears fused into finger with smeared metal-to-skin
boundary...’’,
‘‘importance’’:45}, ...]
</answer>
[EXAMPLE B: EMPTY GT]
Caption: a girl with pink curly hair wearing a witch hat
<think> ... No defects identified. </think>
<answer> [] </answer>
Now analyze the image and respond with exactly TWO blocks (<think> then
<answer>).
```

C Evaluation Metric Details

Let $\mathcal{T} = \{\textit{artifact}, \textit{misalignment}\}$ and let $\mathcal{D}$ be the evaluation set. For image d, denote the ground-truth and predicted defect sets as $G(d) = \{(b_i, t_i, r_i, s_i)\}_{i=1}^{N_d}$ and $P(d) = \{(\hat{b}_j, \hat{t}_j, \hat{r}_j, \hat{s}_j)\}_{j=1}^{M_d}$. For type t, let $G_t(d)$ and $P_t(d)$ be the subsets whose type is t.

Image-level metrics. We define binary indicators $y_{t,d} = \mathbf{1}[|G_t(d)| > 0]$ and $\hat{y}_{t,d} = \mathbf{1}[|P_t(d)| > 0]$. DetTypeF1 is the type-specific image-level F1:

$$Pr_t = \frac{\sum_{d\in\mathcal{D}} y_{t,d}\hat{y}_{t,d}}{\sum_{d\in\mathcal{D}} \hat{y}_{t,d}}, \quad Re_t = \frac{\sum_{d\in\mathcal{D}} y_{t,d}\hat{y}_{t,d}}{\sum_{d\in\mathcal{D}} y_{t,d}}, \quad \text{DetTypeF1}_t = \frac{2Pr_tRe_t}{Pr_t + Re_t}. \tag{7}$$

Clean-image accuracy is the true-negative rate on $\mathcal{D}_t^- = \{d \in \mathcal{D} \mid y_{t,d} = 0\}$:

$$\text{ClnAcc}_t = \frac{1}{|\mathcal{D}_t^-|} \sum_{d\in\mathcal{D}_t^-} \mathbf{1}[\hat{y}_{t,d} = 0]. \tag{8}$$

Defect-level metrics. For each d, we compute a class-aware Hungarian matching [Kuhn, 1955] between $G(d)$ and $P(d)$ under the constraint $t_g = t_p$. Let $\mathcal{M}_{\tau,t}$ be the union of matched pairs of type t whose IoU is at least τ. For $\tau \in \{0.1, 0.5\}$, localization precision, recall, and F1 are

$$\text{BoxPr}_t = \frac{|\mathcal{M}_{\tau,t}|}{\sum_{d\in\mathcal{D}} |P_t(d)|}, \quad \text{BoxRe}_t = \frac{|\mathcal{M}_{\tau,t}|}{\sum_{d\in\mathcal{D}} |G_t(d)|}, \quad \text{BoxF1}_t = \frac{2|\mathcal{M}_{\tau,t}|}{\sum_{d\in\mathcal{D}}(|G_t(d)| + |P_t(d)|)}. \tag{9}$$

For each valid matched pair $(g, p) \in \mathcal{M}_{\tau,t}$, DescCos is the mean cosine similarity between Qwen3-Embedding-0.6B embeddings of r_g and $\hat{r}_p$, and ImpAcc is normalized absolute-error accuracy:

$$\text{DescCos}_t = \frac{1}{|\mathcal{M}_{\tau,t}|} \sum_{(g,p)\in\mathcal{M}_{\tau,t}} \langle r_g, \hat{r}_p \rangle, \quad \text{ImpAcc}_t = \frac{1}{|\mathcal{M}_{\tau,t}|} \sum_{(g,p)\in\mathcal{M}_{\tau,t}} \left(1 - \frac{|s_g - \hat{s}_p|}{100}\right). \tag{10}$$

D Experimental Details

This appendix provides the complete reward computation and training details for the GRPO stage described in Section 4.1.

D.1 Reward Formulation

Composite reward. The composite reward is gated by a format check:

$$R = \begin{cases} 0.6\, R_{\text{diou}} + 0.25\, R_{\text{desc}} + 0.15\, R_{\text{imp}}, & \text{if format valid,} \\ -1, & \text{otherwise.} \end{cases} \tag{11}$$

The format gate verifies that the response contains valid `<think>` tags, a parseable JSON defect list in `<answer>`, and properly ordered `xyxy` coordinates.

Grounding accuracy (R_{diou}). Predicted and ground-truth boxes are matched using the Hungarian algorithm with DIoU as the cost metric, yielding optimal one-to-one assignments. For each matched pair $(i, j) \in \mathcal{M}$, the spatial reward is the DIoU score of that pair.

Edge cases are handled as follows:

- **Correct rejection** (both sets empty): $R_{\text{diou}} = 0.3$.
- **Miss** (ground-truth defects exist but no predictions): $R_{\text{diou}} = -0.8$.
- **False alarm** (predictions on a clean image): $R_{\text{diou}} = -0.3$.
- **Unmatched boxes**: each receives a penalty of -0.5.

The final score is normalized by $\max(|G|, |P|, 1)$ and clipped to $[-1, 1]$.

Description consistency (R_{desc}). For each matched pair $(i, j) \in \mathcal{M}$, we compute the cosine similarity between the predicted and ground-truth descriptions using Qwen3-Embedding-0.6B. The raw similarity is linearly transformed to $[0, 1]$ via:

$$\hat{s}_{ij} = \text{clip}\left(\frac{\text{sim}(r_i, \hat{r}_j) - 0.5}{0.4},\ 0,\ 1\right). \tag{12}$$

The description reward is the sum of transformed similarities over matched pairs, divided by $\max(|G|, |P|, 1)$. Unmatched boxes contribute zero, implicitly penalizing over- or under-prediction.

Importance estimation (R_{imp}). For each matched pair $(i, j) \in \mathcal{M}$, the importance reward is:

$$r_{\text{imp}}^{(i,j)} = \text{clip}\left(1 - \frac{|\hat{s}_j - s_i|}{50},\ 0,\ 1\right), \tag{13}$$

where s_i and $\hat{s}_j$ are the ground-truth and predicted importance scores. This provides a continuous reward that decreases linearly with absolute error, reaching zero when the error exceeds 50 points.

D.2 GRPO Objective

The full GRPO objective optimizes the policy using clipped importance ratios and KL regularization:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}\left[\min\left(\rho_s A_s,\ \text{clip}(\rho_s, 1-\epsilon, 1+\epsilon) A_s\right)\right] + \beta\, \text{KL}(\pi_\theta \,\|\, \pi_{\text{ref}}), \tag{14}$$

where

$$\rho_s = \frac{\pi_\theta(y_s \mid I, T)}{\pi_{\text{old}}(y_s \mid I, T)} \tag{15}$$

is the importance ratio for sampled response y_s, ϵ is the clipping range, and β=0.01 is the KL regularization coefficient.

D.3 SFT Hyperparameters

All generated images are resized so that the longer side is at most 1024 pixels. We use DeepSpeed ZeRO-2 with bfloat16 mixed precision on 2 nodes $\times$ 8 GPUs (16 A100-80G GPUs total). The learning rate is 3×10^{-5} with a cosine schedule and 5% warmup. We train for 1 epoch with per-device batch size 1 and gradient accumulation steps 1, yielding an effective batch size of 16; the $3\times$ pre-baked jitter augmentation effectively exposes the model to three passes over each example within this single epoch. The maximum sequence length is 5,100 tokens, and the vision encoder is frozen throughout.

For coordinate jitter augmentation, each coordinate receives an independent random offset sampled uniformly from $[-10, +10]$ in the $[0, 1000]$ normalized space, with clamping to ensure valid box constraints. We independently sample the offsets when preparing the three augmented copies, providing coordinate variation across repeated presentations of each example.

D.4 GRPO Hyperparameters

We use 2 nodes $\times$ 8 GPUs (16 GPUs total) with DeepSpeed ZeRO-2 and a learning rate of 5×10^{-6}. The KL coefficient is β=0.01. For each prompt, S=8 candidate responses are sampled via colocated vLLM rollout with temperature 1.0 and top-p 0.85 (max completion length 4,096 tokens). We train for 2 epochs with per-device batch size 4.

D.5 Compute Resources

All detector training experiments are run on A100-80G GPUs. A single SDG detector SFT run takes approximately 2 hours on 16$\times$A100-80G GPUs, while a single detector GRPO run takes approximately 36 hours on 16$\times$A100-80G GPUs. For diffusion alignment, one BoxFlow-GRPO run takes approximately 24 hours on 16$\times$A100 GPUs. The defect-guided image refinement experiments use GPT-Image-1.5 through an external API; their wall-clock time depends primarily on API throughput and the allowed request concurrency rather than local GPU compute. The estimated wall-clock time for the reported experiments is about 7 days in total. The full research project required additional compute beyond this estimate because it included preliminary studies and failed experimental runs that are not reported in the paper.

D.6 Data, Code, and Model Availability

The public code repository provides source code and reproduction instructions. The SDG collection provides SDG-30K, the SFT detector, the merged GRPO detector, and the BoxFlow-GRPO FLUX adapter. The GRPO detector is a full merged checkpoint and should be loaded directly rather than as a LoRA adapter.

D.7 Existing Assets and Licenses

We use existing assets only for research dataset construction, training, evaluation, or API-based editing, and cite their original sources throughout the paper. Pick-a-Pic prompts are used under the MIT License. Qwen3-VL-4B-Instruct and Qwen3-Embedding-0.6B are released under Apache-2.0. The T2I generators used for SDG-30K follow their respective public licenses or terms: FLUX.2-dev is governed by the FLUX Non-Commercial License, Z-Image-Turbo and LongCat-Image are released under Apache-2.0, and SANA-1.5 is released under NSCL v2-custom / NVIDIA License. Gemini 3 Pro, GPT-5.4, and GPT-Image-1.5 are accessed only through their official API terms. Public baselines and evaluation resources, including ImageDoctor, Flow-GRPO, UnifiedReward-2.0, PickScore, CLIPScore, HPSv3, DeQA, Forensic-Chat, DrawBench, and RichHF-18K, are cited and used according to their public releases or provider terms. We do not redistribute third-party weights, prompts, images, or code except where allowed by their licenses; released SDG assets will include attribution, license notices, intended-use documentation, and pointers to the original sources.

D.8 LLM Usage Declaration

LLMs and VLMs are used as core components of this work. Gemini 3 Pro is used during data preparation for description expansion, reasoning-trace distillation, and importance scoring; Qwen3-

VL-4B-Instruct is fine-tuned as the SDG detector; Qwen3-Embedding-0.6B is used for description-similarity evaluation and reward computation; and GPT-Image-1.5 is used for defect-guided image refinement. We also used general-purpose LLMs for manuscript writing assistance, including language polishing and wording refinement. All technical claims, experimental results, and final text were reviewed and edited by the authors.

E Extended Experimental Results

E.1 SDG Detector Ablation

Table 7 reports the full ablation results summarized in Section 5.1.

Table 7: Ablation study on SDG-30K test set. Table header follows Table 2. "–" indicates the component is ablated and the corresponding metric is undefined. Ablation groups are separated by dashed lines: training stage, CoT steps, output component, and architecture/augmentation.

	Artifact						*Misalignment*					
	Image-Level		Defect-Level				Image-Level		Defect-Level			
	All	GT=0	GT>0				All	GT=0	GT>0			
Variant	DetTypeF1	ClnAcc	BoxF1@0.1	BoxF1@0.5	DescCos@0.1	ImpAcc@0.1	DetTypeF1	ClnAcc	BoxF1@0.1	BoxF1@0.5	DescCos@0.1	ImpAcc@0.1
SFT	0.776	0.697	0.402	0.255	0.904	0.883	0.636	0.799	0.499	0.376	0.893	0.892
GRPO	0.772	0.560	0.404	0.263	0.904	0.887	0.675	0.732	0.511	0.387	0.888	0.893
SFT w/o step 1	0.719	0.793	0.357	0.218	0.894	0.880	0.597	0.798	0.436	0.329	0.875	0.892
SFT w/o step 3	0.731	0.537	0.387	0.253	0.897	0.884	0.496	0.840	0.364	0.281	0.846	0.878
GRPO (no think)	0.776	0.640	0.417	0.288	0.899	0.881	0.640	0.677	0.487	0.352	0.879	0.883
SFT w/o desc	0.757	0.718	0.386	0.250	–	0.884	0.609	0.782	0.468	0.358	–	0.894
SFT w/o importance	0.761	0.745	0.402	0.254	0.898	–	0.594	0.793	0.448	0.357	0.870	–
SFT unfreeze ViT	0.659	0.638	0.329	0.203	0.894	0.880	0.584	0.799	0.427	0.305	0.856	0.880
SFT w/o jitter	0.751	0.684	0.396	0.253	0.898	0.884	0.611	0.786	0.462	0.360	0.878	0.894

E.2 Extended Comparison with ImageDoctor on SDG-30K

Figure 7 extends the comparison in Figure 4 with six additional cases covering both *artifact* and *misalignment* defects, as well as images where the targeted defect type is absent. For ImageDoctor we show the artifact and misalignment heatmap heads separately; for SDG we overlay the predicted bounding boxes with per-instance labels. SDG consistently grounds prompt-conditioned misalignments (e.g., "Nucleosome" depicted as a double helix, "Dumbo" drawn without clown makeup) that ImageDoctor's misalignment head misses, while avoiding the spurious face/hand activations that ImageDoctor's artifact head produces on clean regions (rows 5–6).



Figure 7: Extended qualitative comparison on SDG-30K. Columns: original image, ground-truth SDG annotations, ImageDoctor artifact heatmap, ImageDoctor misalignment heatmap, and SDG (ours) predictions. Red text in the captions highlights the prompt span corresponding to the misalignment.

E.3 Full SDG Output Example

We provide the complete SDG model output for the example shown in Figure 1. The model receives the generated image and the prompt, and produces a structured reasoning trace (`<think>`) followed by a JSON defect set (`<answer>`).

SDG Full Output for Figure 1

Prompt: Psychedelic smoke, explosion, fire twirling, backlit, twisting, curled, petite American ballerina, wearing ballerina sparkling lace tutu, riding long glowing neon skateboard, star jump upside down, 8K, HD, highly detailed, rendered in octane

<think>

Step 1: Caption Understanding
- The caption describes a ‘‘petite American ballerina’’ in a ‘‘sparkling lace tutu’’ riding a ‘‘long glowing neon skateboard’’.
- Key action: ‘‘star jump upside down’’.
- Setting/Style: ‘‘psychedelic smoke’’, ‘‘explosion’’, ‘‘fire twirling’’, ‘‘backlit’’, ‘‘twisting’’, ‘‘curled’’, ‘‘8K’’, ‘‘highly detailed’’, ‘‘rendered in octane’’.

Step 2: Visual Analysis & Defect Spotting (Issue Summary)
- Misalignment (Action/Pose): The most prominent issue is the ballerina’s pose. The caption explicitly requests a ‘‘star jump upside down’’. However, the generated image shows the ballerina standing upright on the skateboard, balancing on one leg with the other extended. She is neither upside down nor performing a star jump.
- Artifact (Anatomy - Hands): Both hands show generation artifacts. The hand raised high in the air (image-right) appears to have fused or indistinct fingers. The hand extended to the side (image-left) also lacks clear finger definition and looks somewhat malformed or ‘‘melted’’.
- Artifact (Anatomy - Face): The facial features, while detailed, show some subtle distortion typical of AI generation, particularly in the alignment of the eyes and mouth, which appear slightly uncanny or warped.

Step 3: Localization (Box-by-Box Grounding)
- Box 1 (Artifact): The defect is located on the raised hand on the image-right side; it is a small localized region where the fingers are visible but appear fused and lack proper anatomical separation.
- Box 2 (Artifact): The defect is on the extended hand on the image-left side; it is a small localized region at the end of the arm where the fingers are blurry and indistinct, failing to resolve into a clear hand shape.
- Box 3 (Artifact): The defect is centered on the ballerina’s face; it is a part-sized region where the facial geometry (eyes, nose, mouth) appears slightly distorted and lacks the crisp symmetry expected of a high-quality render.
- Box 4 (Misalignment): The defect encompasses the entire figure of the ballerina; it is a large area covering the subject who is standing upright on the board, contradicting the prompt’s instruction for an ‘‘upside down’’ and ‘‘star jump’’ pose.

</think>

<answer>


```
[
  {
    "box_2d": [600, 48, 641, 100],
    "label": "artifact",
    "description": "On the raised image-right hand, the fingers
      appear fused and indistinct, lacking clear anatomical
      separation and joint definition, resulting in a distorted,
      unnatural shape.",
    "importance": 22
  },
  {
    "box_2d": [223, 345, 285, 385],
    "label": "artifact",
    "description": "On the extended image-left hand, the fingers
      are malformed and blurry, showing a lack of distinct digits
      and a melted appearance rather than a properly articulated
      hand.",
    "importance": 22
  },
  {
```

```
    "box_2d": [474, 220, 530, 281],
    "label": "artifact",
    "description": "The facial features in the upper center show
      subtle distortion, with the eyes and mouth appearing slightly
      warped or asymmetrical, failing to maintain a consistent
      realistic geometry.",
    "importance": 32
  },
  {
    "box_2d": [223, 48, 657, 840],
    "label": "misalignment",
    "description": "The ballerina is depicted standing upright on
      the skateboard, which directly contradicts the caption's
      explicit requirement for an 'upside down' star jump action.",
    "importance": 95
  }
]
</answer>
```


E.4 Extended BoxFlow-GRPO Qualitative Comparison

Figure 8 extends the qualitative comparison in Figure 5 with additional DrawBench prompts. Compared to FlowGRPO (UR2) and DenseFlow-GRPO (ImageDoctor), BoxFlow-GRPO (UR2+SDG) better respects fine-grained prompt attributes (e.g., correct color binding in "a red book and a yellow vase", coherent subject composition in "Darth Vader playing with raccoon") while preserving photographic realism, avoiding the illustration/anime drift that baseline RL variants exhibit.

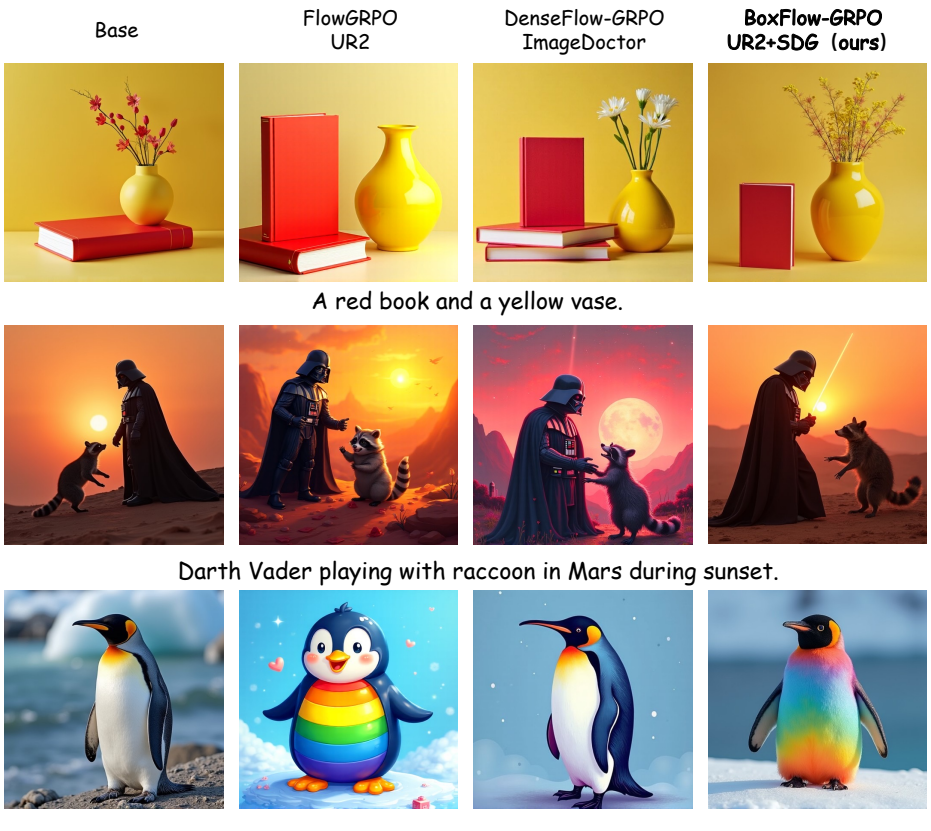

Figure 8: Extended qualitative comparison of BoxFlow-GRPO on DrawBench prompts. Columns: Base (FLUX.1-dev), FlowGRPO with UR2 reward, DenseFlow-GRPO with ImageDoctor heatmap reward, and BoxFlow-GRPO with UR2+SDG structured reward (ours).

E.5 Extended Defect-Guided Refinement Results

Figure 9 extends the refinement comparison in Figure 6 with additional GPT-Image-1.5 editing cases. Across these examples, SDG feedback provides localized, instance-level guidance that helps the editor make targeted corrections, such as removing an extra lion cub when the prompt asks for a single cub, identifying an incorrect webpage title and replacing it with the correct title, and eliminating a large poster-like obstruction from a city-skyline image.

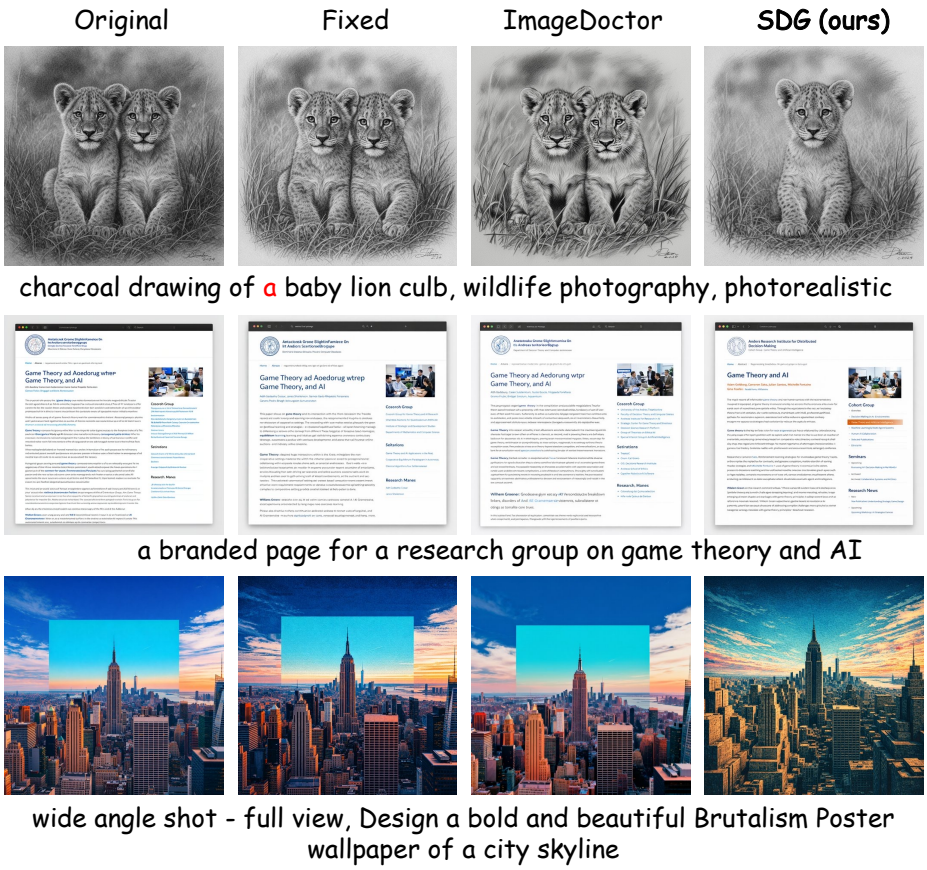

Figure 9: Extended qualitative comparison of defect-guided image refinement via GPT-Image-1.5. Columns: original image, Fixed caption-only editing, ImageDoctor heatmap/text-feedback editing, and SDG (ours) box-structured-feedback editing.

E.6 Validation of Importance Scores Against Human Judgments

Importance scores provide supervision for the SDG detector and contribute to both the GRPO reward and the importance-weighted spatial penalties in BoxFlow-GRPO. To assess whether Gemini-derived importance scores reflect human judgments of perceptual severity, we randomly sampled 500 distinct defect boxes and assigned 250 non-overlapping boxes to each of two human annotators. Each annotator reassessed severity using the same five-tier rubric without seeing the Gemini scores. Thus, each box received one human rating, which we compared with the corresponding Gemini assessment.

Table 8 summarizes the results. Exact-tier agreement requires the same severity tier, while adjacent-tier agreement allows a difference of at most one tier. Spearman correlation of 0.786 indicates strong agreement in severity ordering; 97.2% of assessments differ by at most one tier. These results support importance as a proxy for perceptual severity on the sampled defects. The disjoint assignments measure human–Gemini agreement, not inter-annotator reliability.

Table 8: Agreement between human severity judgments and Gemini-derived importance scores on 500 distinct defect boxes. Two annotators each assessed 250 non-overlapping boxes while blinded to Gemini scores.

Human calibration metric	Result
Exact-tier agreement	64.8%
Agreement within one adjacent tier	97.2%
Spearman rank correlation	0.786

E.7 Area-Normalized Penalty Coefficients and Sensitivity Analysis

Area-normalization rationale. The ratio between c_{art} and c_{mis} is motivated by the different spatial extents of the two defect types. An average misalignment box covers approximately $9.90\times$ the area of an artifact box. Since BoxFlow-GRPO applies penalties at covered latent locations, equal coefficients would assign a larger total spatial penalty to a larger box at comparable importance. Writing $\bar{A}_{\text{art}}$ and $\bar{A}_{\text{mis}}$ for the average box areas, approximate area normalization gives

$$\frac{c_{\text{mis}}}{c_{\text{art}}} \approx \frac{\bar{A}_{\text{art}}}{\bar{A}_{\text{mis}}} \approx \frac{1}{9.90}. \tag{16}$$

We therefore use c_{art}=0.5 and c_{mis}=0.05. Their area-scaled contributions have ratio $(0.05 \times 9.90)/0.5 = 0.99$, approximately balancing spatial penalty mass. This is an area-based normalization rationale rather than an exact equality of realized penalties, which also depend on importance weights and box overlap.

Sensitivity analysis. We fixed c_{art}=0.5 and varied $c_{\text{mis}} \in \{0.25, 0.05, 0.025\}$. All three variants were trained for 500 steps and evaluated on DrawBench, using the same evaluation set as Table 5. Table 9 reports five evaluation metrics and their mean relative change from Base (Avg.), following Table 5.

Table 9: Sensitivity to the artifact/misalignment penalty coefficients. All trained variants use 500 steps and are evaluated on DrawBench. Avg. is the mean relative improvement over Base across the five metrics. The default setting is bolded.

$c_{\text{art}}/c_{\text{mis}}$	PickScore	CLIPScore	HPSv3	DeQA	$P(\text{real})$	Avg.
Base	22.84	0.912	11.75	0.876	0.211	—
0.5/0.25	22.83	0.910	11.87	0.874	0.217	+0.7%
0.5/0.05	**22.91**	**0.915**	**12.14**	**0.877**	**0.228**	**+2.4%**
0.5/0.025	22.86	0.910	11.92	0.861	0.226	+1.3%

All three settings improve the average relative to Base. The area-normalized setting $(0.5, 0.05)$ achieves the largest average improvement $(+2.4\%)$ and is the only tested setting that improves all five metrics relative to Base. Increasing c_{mis} to 0.25 slightly reduces PickScore, CLIPScore, and DeQA relative to Base; decreasing it to 0.025 reduces CLIPScore and DeQA. These results support $(0.5, 0.05)$ as the most balanced operating point among the tested settings.

E.8 Closing the Diagnosis-to-Refinement Loop with Paired Re-diagnosis

Evaluation protocol. To test whether refinement reduces defects as assessed by the detector itself, we applied the same frozen SDG detector to paired images before and after editing. We compared SDG-guided refinement, ImageDoctor-guided refinement, and direct caption-only editing (the **Fixed** baseline in Section 5.2.2). We report the percentage reductions in detected defect boxes and in their total importance relative to the corresponding pre-edit images. Total importance sums the importance scores of detected defects, capturing their combined predicted severity rather than only their count.

Results and interpretation. All three editing methods reduce both detected defect count and total importance (Table 10). SDG-guided refinement achieves the largest reduction in both measures: 37.55% fewer boxes and 42.37% lower total importance. Thus, its improvements extend beyond

Table 10: Paired re-diagnosis before and after refinement with the same frozen SDG detector. Larger reductions indicate fewer detected defects and lower total predicted importance after editing.

Editing method	Box reduction	Importance reduction
SDG (ours)	**37.55%**	**42.37%**
ImageDoctor	25.81%	30.36%
Fixed (direct edit)	30.71%	34.17%

reducing the number of detected regions to reducing their combined predicted severity. With the detector held fixed, these changes support sensitivity to the corrected image content, although aggregate reductions alone do not rule out location-specific biases such as repeated activation on hands or faces.

This paired re-diagnosis complements the human preference evaluation in Table 6. Together, the human preference results and reductions in re-diagnosed defects provide evidence that structured diagnosis translates into effective image refinement, closing the diagnosis-to-refinement loop at the detector level.

E.9 Generalization to the Qwen-Image Base Model

To evaluate the method on another base model, we trained BoxFlow-GRPO, Flow-GRPO with UR2, and DenseFlow-GRPO with ImageDoctor on Qwen-Image for 1,200 steps each. We use the same evaluation dataset as the FLUX.1-dev experiments reported in Table 5. Table 11 reports the five metrics used in Table 5, with relative changes computed against the Qwen-Image Base model. Avg is the arithmetic mean of the five per-metric relative changes, rather than an average of their raw scores.

Table 11: Downstream performance on Qwen-Image after 1,200 training steps for each RL variant. Parentheses show relative changes from Qwen-Image Base (green: improvement; red: regression); Avg averages these changes across the five metrics. Best values are bold and second-best values are underlined, including Base. BoxFlow-GRPO achieves the best Avg.

Method	Reward	PickScore	CLIPScore	HPSv3	DeQA	$P(\text{real})$	Avg
Base	–	23.03	**0.969**	10.90	0.843	<u>0.230</u>	–
Flow-GRPO	UR2	**23.62** (+2.6%)	0.923 (-4.7%)	**12.65** (+16.1%)	**0.868** (+3.0%)	0.177 (-23.0%)	<u>-1.2%</u>
DenseFlow-GRPO	ImageDoctor	23.14 (+0.5%)	0.941 (-2.9%)	<u>11.07</u> (+1.6%)	0.764 (-9.4%)	0.195 (-15.2%)	-5.1%
BoxFlow-GRPO (ours)	UR2+SDG	<u>23.16</u> (+0.6%)	<u>0.965</u> (-0.4%)	11.05 (+1.4%)	<u>0.852</u> (+1.1%)	**0.237** (+3.0%)	**+1.1%**

BoxFlow-GRPO again achieves the best average relative improvement ($+1.1\%$), compared with -1.2% for Flow-GRPO (UR2) and -5.1% for DenseFlow-GRPO (ImageDoctor). It improves PickScore, HPSv3, DeQA, and $P(\text{real})$, with a small CLIPScore decrease (-0.4%). Flow-GRPO (UR2) obtains the highest PickScore, HPSv3, and DeQA, but its $P(\text{real})$ drops from 0.230 to 0.177 (-23.0%). BoxFlow-GRPO instead increases $P(\text{real})$ to 0.237 ($+3.0\%$), yielding a more balanced outcome across the five metrics. These results extend the favorable average performance observed on FLUX.1-dev to Qwen-Image.

F Limitations and Broader Impact

F.1 Limitations

SDG currently focuses on two defect types, *artifact* and *misalignment*. Other quality dimensions, such as aesthetics, style, composition, safety, and cultural appropriateness, fall outside the current label space and would require additional annotation guidelines and evaluation metrics. The dataset is constructed from four contemporary T2I generators and Pick-a-Pic prompts; although this provides broad coverage, performance may change for other generators, domains, resolutions, or prompt distributions.

Our importance scores are distilled from Gemini 3 Pro under a fixed rubric. Human calibration on 500 defect boxes supports their agreement with human severity judgments (Appendix E.6), but these scores may still differ from human preferences and inherit biases from the teacher model. The SDG detector can also miss subtle defects, hallucinate defects in clean locations, or produce boxes that are too coarse for very small or highly diffuse failures. Finally, BoxFlow-GRPO assumes that box-derived penalties can be meaningfully projected onto latent spatial locations; this approximation may be less reliable when the latent grid does not align cleanly with visible defect locations.

F.2 Broader Impact

Structured defect feedback can make T2I systems more interpretable by exposing localized failure modes, supporting dataset auditing, and enabling targeted image refinement. These properties can help users diagnose generation errors rather than relying only on scalar preference scores.

The same capabilities also carry risks. Better diagnosis and refinement may improve the realism of synthetic images, which could be misused for deceptive or harmful content. Our public release includes code, model weights, and SDG-30K. The released assets should preserve annotation provenance, document intended use, and include safeguards for controlled release where appropriate. Because SDG can make localized judgments about generated people or scenes, downstream deployments should also consider fairness and bias in the underlying prompts, generators, annotators, and teacher models.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the four main contributions: the SDG instance-level representation, the SDG-30K dataset, the SDG-Eval protocol, and the unified diagnosis-to-alignment framework with BoxFlow-GRPO. These claims are supported by the experimental results in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of SDG are discussed in Appendix F, covering label-space scope (only artifact and misalignment), generator and prompt-distribution coverage, the use of Gemini 3 Pro as teacher for importance scoring, detector failure modes, and the latent-grid projection assumption in BoxFlow-GRPO.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include theoretical results; all contributions are empirical (dataset, evaluation protocol, and training framework).

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 reports all training and inference details (backbone, optimizer, batch size, learning rate, schedule, GPU configuration, sampling temperature). Reward formulas, format-validity rules, prompts for human and Gemini 3 Pro annotation, and full evaluation-metric definitions are provided in Appendix D, Appendix B.2, Appendix A, and Appendix C. Public code, model weights, data, and reproduction instructions are described in Appendix D.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Appendix D.6 links the public code repository and the SDG collection containing model weights and SDG-30K, together with reproduction instructions.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 reports the data split (28,945 train / 1,151 test, prompt-disjoint), backbone (Qwen3-VL-4B-Instruct), optimizer schedule (cosine), batch size, learning rate, training epochs, and GPU configuration for SFT, GRPO, and BoxFlow-GRPO. The full reward formulation and training hyperparameters are provided in Appendix D, and full ablation metrics are provided in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars across multiple random seeds because the cost of running multiple full SFT and GRPO training runs (each on 8–16 A100-80G GPUs) is prohibitive. Statistical reliability is instead supported by reporting results on a 1,151-image test set with a

16-annotator human reference upper bound, and by ablations along multiple axes (Section 5.1, Appendix D).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix D.5 reports the GPU configuration and approximate wall-clock time for SDG detector SFT, detector GRPO, and BoxFlow-GRPO runs, estimates the reported experiments at about 7 days in total, and notes that the full project required additional compute for preliminary and failed runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics. All annotated images are generated by T2I models (no real-person or private data); annotators were compensated above the local minimum wage; defect annotations document quality failures rather than producing harmful content. Broader societal impacts are discussed in Appendix F.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix F discusses positive impacts, including interpretable defect diagnosis, dataset auditing, and targeted image refinement, as well as risks such as misuse for more realistic synthetic images and fairness or bias concerns in localized judgments.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Appendix F discusses responsible-release safeguards, including preserving annotation provenance, documenting intended use, and considering misuse and bias risks for released assets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Appendix D.7 lists the existing datasets, models, APIs, baselines, and evaluation resources used in the paper, states their licenses or governing provider terms, and notes that third-party assets are cited and not redistributed except where permitted by their licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces SDG-30K and SDG-Eval, which are documented in Section 3 and the appendices, including their construction process, annotation protocol, evaluation setup, asset licenses, intended use, limitations, and release status. Appendix D.6 links the public code, model weights, and dataset.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: Appendix A describes the annotation interface, workflow, defect taxonomy, boxing and description instructions, prompt-interpretation protocol, quality-control rules, and annotator compensation.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes].

Justification: The work involves paid annotation of synthetic T2I images for dataset curation, not a behavioral user study. The annotation task does not collect or analyze personal, demographic, medical, or sensitive information from annotators. Potential risks were limited to viewing synthetic images during annotation, and annotators were instructed about the task scope and compensated above the local minimum wage. According to the applicable institutional policy, this data-curation task is exempt from IRB review / does not constitute human-subject research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Appendix D.8 declares the use of LLMs and VLMs in the core methodology, including Gemini 3 Pro for annotation augmentation, Qwen3-VL-4B-Instruct as the SDG detector, Qwen3-Embedding-0.6B for description-similarity evaluation and reward computation, and GPT-Image-1.5 for defect-guided refinement. It also states that general-purpose LLMs were used for manuscript language polishing and wording refinement, with all final content reviewed by the authors.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.