

# 000 001 002 003 004 005 006 007 I-DRUID: LAYOUT TO IMAGE GENERATION VIA 008 INSTANCE-DISENTANGLED REPRESENTATION AND 009 UNPAIRED DATA 010 011

012 **Anonymous authors**  
013 Paper under double-blind review  
014  
015  
016  
017  
018  
019  
020  
021  
022  
023  
024  
025  
026  
027  
028  
029  
030

## 031 ABSTRACT 032

033 Layout-to-Image (L2I) generation, aiming at coherently generating multiple instances conditioned on the given layouts and instance captions, has raised substantial attention in the recent research. The primary challenges of L2I stem from 1) attribute leakage due to the entangled instance features within attention and 2) limited generalization to novel scenes caused by insufficient image-text paired data. To address these issues, we propose I-DRUID, a novel framework that leverages **instance-disentanglement representations** (IDR) and **unpaired data** (UID) to improve L2I generation. IDR are extracted with our instance disentanglement modules, which utilizes information among instances to obtain semantic-related features while suppressing spurious parts. To facilitate disentangling, we require semantic-related features to trigger more accurate attention maps than spurious ones, formulating the instance-disentangled constraint to avoid attribute leakage. Moreover, to improve L2I generalization, we adapt L2I with unpaired, prompt-only data (UID) to novel scenes via reinforcement learning. Specifically, we enforce L2I model to learn from unpaired, prompt-only data by encouraging / rejecting the rational / implausible generation trajectories based on AI feedback, avoiding the need for paired data collection. Finally, our empirical observations show that IDM and RL cooperate synergistically to further enhance L2I accuracies. Extensive experiments demonstrate the efficacy of our method.  
034  
035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045  
046  
047  
048  
049  
050  
051  
052  
053

## 1 INTRODUCTION

Recent advances in text-to-image (T2I) generation have achieved remarkable success, primarily driven by diffusion models (Rombach et al., 2022; Ho et al., 2020; Esser et al., 2024; Ramesh et al., 2021). By employing UNet (Ronneberger et al., 2015) or multi-modal diffusion transformers (MM-DiT) (Li et al., 2024; Esser et al., 2024) for noise / velocity prediction, these models learn to generate high-quality images with given prompts. To achieve finer control over the generation process, recent works have explored various spatial controls, such as semantic masks (Couairon et al., 2023; Kim et al., 2023; Zhang et al., 2025a), sketches (Voynov et al., 2023; Zhang et al., 2023), or bounding boxes (Zhang et al., 2025b; Zhou et al., 2024a; Wang et al., 2024). Among these methods, bounding box-based control has become a particularly prevalent controlling factor (Li et al., 2023; Wang et al., 2024; Xie et al., 2023; Zhang et al., 2025b; Zhou et al., 2024b) due to its compatibility with downstream vision tasks. This has spurred the development of layout-to-image (L2I) generation (Zhou et al., 2024a; Wang et al., 2024; Xie et al., 2023; Li et al., 2023; Zhang et al., 2025b), which aims to synthesize multiple objects coherently based on a given spatial layout and corresponding captions.

Concurrent studies show promising results in L2I (Zhou et al., 2024a; Zhang et al., 2025b; Wang et al., 2024), but two key challenges remain. (1) The information fusion within attention layer hinders instance rendering (Dahary et al., 2024; Wang et al., 2024), leading to the persistence of attribute leakage. Previous works attempt to address attribute leakage through attention map manipulation (Dahary et al., 2024; Zhou et al., 2024a; Wang et al., 2024), but the inherent difficulty of CLIP (Radford et al., 2021) in differentiating singular attributes from complex prompts remains a bottleneck (Feng et al., 2023). Moreover, few studies have considered L2I under MM-DiT scenario (Zhang et al., 2025b). Therefore, it is necessary to explore precise instance-level representation control for both

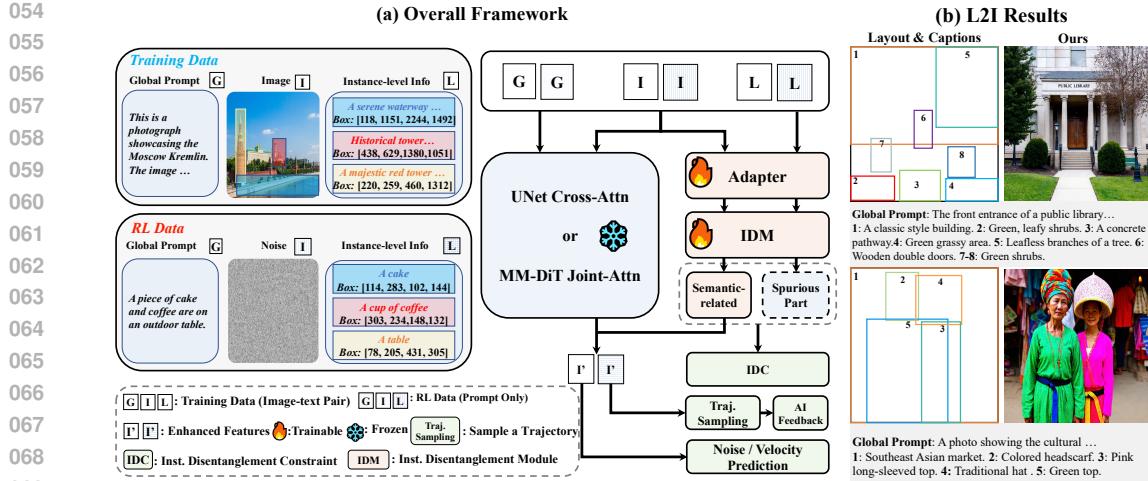


Figure 1: (a) Overall framework of I-DRUID. I-DRUID jointly considers attribute leakage and improves model generalization via AI feedback. (b) Visual results from our method.

UNet- and MM-DiT-based architectures. (2) L2I models may encounter performance degradation when being deployed to novel scenes. *e.g.*, models trained on training set with long captions (Wang et al., 2024; Zhang et al., 2025b) get poor performance when deployed under testing set with short coarse captions (Zhou et al., 2024b), demonstrating poor generalization when training with limited pair-wise training data. Collecting more pair-wise data is an empirical solution, but suffers from huge time cost. Inspired by the success of RL in scaling large language models to complex tasks (Guo et al., 2025; Jaech et al., 2024), we would like to advance L2I with RL by using prompt-only unpaired data, letting the novel-scene adaptation to be exempted from pair-wise data collection.

As shown in Fig. 1-(a), we present I-DRUID, which is comprised of instance disentanglement learning, coupled with reinforcement learning. (1) We first introduce adapters (Ye et al., 2023) to inject layout information into training process, then decompose attention features into “semantic-related features” and “spurious part” with our instance disentanglement modules (IDM). The core of disentanglement through IDM is based on our instance disentangling constraint (IDC). Our key insight of designing IDC is that semantic-related features must elicit more precise attention maps than spurious features, thereby facilitating the disentanglement process during L2I optimization. (2) To generalize our L2I model to OOD prompts, we further introduce a novel reinforcement learning (RL) framework (Agarwal et al., 2019; Liu et al., 2025b), encouraging L2I model to learn from unpaired novel prompts based on AI feedback. Specifically, we conduct trajectory sampling from Gaussian noise and the given novel prompt, producing images for the assessment of visual language model (VLM) in terms of spatial accuracy and instance consistency in an online manner. The feedback is used as a guidance in analyzing best generation policy, thus encouraging L2I capability. (3) We also demonstrate IDC and RL could mutually benefit each other as IDM provides a more accurate generation policy for RL, advancing L2I results as shown in Fig. 1-(b). Our method also has a high flexibility, which could be easily utilized for both UNet-based (*e.g.*, SD 1.5) and MM-DiT-based architectures (*e.g.*, SD3).

To sum up, our contributions are three-fold:

- We seek instance-disentangled representation, which is achieved with our IDC-supervised IDM. IDC leverages collective instance information to extract semantic-related features, which triggers more precise attention maps and thus avoids attribute leakage.
- We equip our L2I model with a novel Reinforcement Learning (RL) strategy to improve its generalization. RL enables L2I to learn from unpaired data, letting novel-scene adaptation to be exempted from pair-wise data collection.
- We demonstrate that these two components can be synergistically cooperated to further enhance L2I accuracy under both UNet and MM-DiT-based architectures. Our approach achieves state-of-the-art results on multiple L2I benchmarks, demonstrating its efficacy and flexibility.

108  
109  
2 RELATED WORK110  
111 **Layout to Image Generation.** Diffusion models like SD (Ho et al., 2020; Rombach et al., 2022),  
112 SD3 (Esser et al., 2024), and FLUX (Labs, 2024) are powerful tools to achieve text to image  
113 generation. To achieve fine-level control, additional spatial control like bounding boxes (Wang et al.,  
114 2024; Li et al., 2023; Zhou et al., 2024b;a; Zhang et al., 2025b; Lee et al., 2024) are introduced  
115 into generation process to craft entities within the given location and instance-level prompts, termed  
116 as layout to image generation. Generally speaking, these methods can be categorized as training-  
117 based (Zhou et al., 2024b; Li et al., 2023) and training-free approach (Xie et al., 2023; Lee et al.,  
118 2024). The former introduce adapters (Ye et al., 2023; Mou et al., 2023) and additional modules  
119 to encode locations for subsequent attention. The latter focus on manipulating attention map at  
120 test-time to achieve location control for each entity. Although effective, most of them are conducted  
121 based on UNet architecture, while ignoring modern MM-DiT architectures like SD3 (Esser et al.,  
122 2024). Recently, Creati-Layout (Zhang et al., 2025b) first devise SD3-based L2I scheme, but ignores  
123 explicit constraint to alleviate attribute leakage (Wang et al., 2024). Different from previous methods,  
124 our approach devise disentanglement modules to advance L2I task under both UNet-based and  
125 MM-DiT-based scenarios.126 **Reinforcement Learning.** Reinforcement learning (RL) (Schulman et al., 2017) is a widely-used  
127 strategy to align model’s response to human preference, both for large language models (Rafailov  
128 et al., 2023; Wang et al., 2025; Yan et al., 2024) and diffusion models (Chen et al., 2024; Fan  
129 et al., 2023). The core idea of RL is training a model to interact with environment, typically a  
130 reward model, that provides feedback for the model’s responses. This feedback guides the model  
131 in exploring and learning an optimal policy. Early RL methods for alignment often relied on  
132 either a separate reward model (Ramamurthy et al., 2022; Peng et al., 2019) or extensive human  
133 annotation of samples (Rafailov et al., 2023; Yuan et al., 2024) to obtain preference data. However,  
134 the recent development of large-scale models has led to a paradigm shift. Learning from AI feedback  
135 has emerged as a promising and efficient alternative for scaling the alignment process (Bai et al.,  
136 2022; Fan et al., 2023). The RL under diffusion scenario can be categorized into online and  
137 offline scheme. DPOK (Fan et al., 2023) and DDPO (Black et al., 2023) are pioneering works that  
138 introduce RL into image generation by fine-tuning generation policy with feedback from AI like  
139 ImageReward (Xu et al., 2024) in an online manner. DiffDPO (Wallace et al., 2024) proposes the first  
140 DPO-based (Rafailov et al., 2023) RL method to fine-tune diffusion in an offline manner. Although  
141 these methods show promising results, most of them relies on the randomness of generation trajectory  
142 to achieve environmental exploration, which is not compatible with SD3 (Esser et al., 2024) or  
143 FLUX (Labs, 2024) with deterministic ODE forward process. Recently, the advent of SDE-based  
144 forwarding equivalence (Albergo et al., 2023; Liu et al., 2025a) enables efficient exploration during  
145 the forwarding process of flow-matching methods. However, the application of RL under L2I scenario  
146 is still under-explored, inspiring us to scale L2I model with the help of AI feedback and RL.147  
148  
149 3 METHODOLOGY150  
151 3.1 OVERVIEW152 **Problem Formulation.** Layout to image generation (L2I) assumes the users to give the following  
153 information during inference. (1) a global prompt  $p_G$  defining the overall semantic information for the  
154 generated image; (2)  $n$  instance captions  $\mathcal{P}_L = \{p_1, p_2, \dots, p_n\}$ , which describes detailed information  
155 for each instance; (3) layout  $\mathcal{B} = \{b_1, b_2, \dots, b_n\}$ , where  $b_i = \{x_{i0}, y_{i0}, x_{i1}, y_{i1}\}$ ,  $(1 \leq i \leq n)$   
156 containing top-left coordinate  $(x_{i0}, y_{i0})$  and bottom-right coordinate  $(x_{i1}, y_{i1})$  for each instance. The  
157 goal of L2I is generating images that follows the above instructions. Although our method is capable  
158 of L2I for both UNet-based (e.g., SD-1.5) and MM-DiT-based architectures (e.g., SD3), *we mainly*  
159 *introduce our method with SD3 and the SD-1.5 variant could be easily derived.*160 **Overall Pipeline.** As shown in Fig. 2-(a), Our L2I solution is comprised of (a) instance disentangle-  
161 ment learning and (b) reinforcement learning. The former is built upon adapter-based L2I (Li et al.,  
162 2023; Wang et al., 2024), further refining features based on our IDM and IDC. The latter enforces  
163 L2I model to seek for the optimal generation policy based on the given prompt and feedback from  
164 reward, thus generalizing the model to novel scenes without using image-text paired data. These two

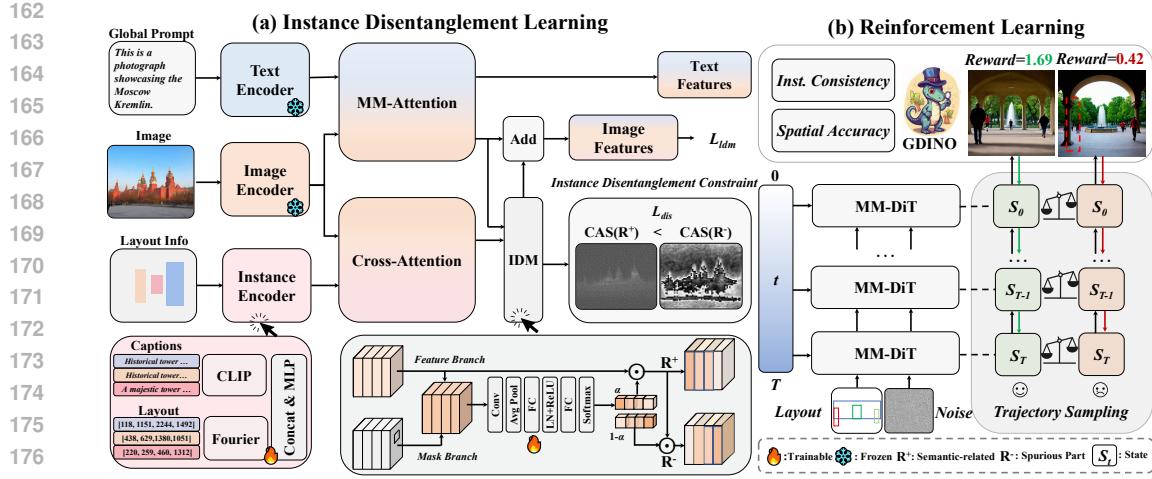


Figure 2: Overall Training Process. Our method is comprised of: (a) Instance Disentanglement Learning and (b) Reinforcement Learning Stage. At (a), our instance disentanglement module (IDM) is supervised by instance disentanglement constraint (IDC) for extracting semantic-related representations  $\mathbf{R}^+$ , while discarding spurious part  $\mathbf{R}^-$  based on the proposed metric CAS. Moreover, we sample various generation trajectories with unpaired data and adopt GDINO to evaluate generated images. Better results will be encouraged at (b), adapting L2I model to novel scenes.

contributions mutual benefit each other and further improves L2I capability. Next, we introduce our solution in detail.

### 3.2 INSTANCE DISENTANGLEMENT STAGE

**Instance Disentanglement Module.** Previous L2I methods (Li et al., 2023; Wang et al., 2024; Zhang et al., 2025b) achieve layout control by injecting layout-caption coupled features with cross-attention for optimization. However, as demonstrated in previous literature (Dahary et al., 2024; Zhou et al., 2024b;a), attention tends to fuse instance-level features, leading to the attribute leakage. We therefore devise IDM as feature refiner to alleviate this. The IDM specifications are shown in Fig. 2-(a), which receives  $n + 1$  enhanced features and layout masks as inputs. IDM learns to extract “semantic-related features”  $\mathbf{R}^+$  while discarding “spurious part”  $\mathbf{R}^-$  for subsequent generation. Specifically, we compute the given  $n$  instances’ enhanced features with corresponding layout and captions, coupled with global-prompt-enhanced features to formulate  $n + 1$  inputs  $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{n+1}\} \in \mathbb{R}^{(n+1) \times C \times W \times H}$ . These features formulate the input of *feature branch*. For *mask branch*, we obtain layout masks for instance  $i$  with its bounding box  $b_i = \{x_{i0}, y_{i0}, x_{i1}, y_{i1}\}$ . Pixels within bounding box  $b_i$  will be assigned with 1 while others are set to 0. The mask for global prompt is set to all-one matrix, formulating  $n + 1$  masks  $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{n+1}\}$ . IDM jointly considers  $\mathcal{E}$  and  $\mathcal{M}$  to get the optimal channel-wise weighting scheme:

$$\alpha = \text{IDM}(\mathcal{E}, \mathcal{M}) \in \mathbb{R}^{n+1}, \quad (1)$$

where  $\alpha$  is the weighting scores assigned to each enhanced feature in  $\mathcal{E}$ . We thus obtain “semantic-related features”  $\mathbf{R}^+$  while discarding “spurious part”  $\mathbf{R}^-$  through simple multiplication:

$$\mathbf{R}^+ = \alpha_i \mathcal{E}_i; \quad \mathbf{R}^- = (1 - \alpha_i) \mathcal{E}_i, \quad (2)$$

where  $\mathbf{R}^+$  and  $\mathbf{R}^-$  are “semantic-related features” and “spurious part” with the size of  $(n + 1) \times C \times W \times H$ .

**Instance Disentanglement Constraint.** To facilitate the disentangling process, we further devise instance disentanglement constraint (IDC) by measuring  $\mathbf{R}^+$ ’s ability in triggering accurate attention map. Specifically,  $\mathbf{R}^+$  will be used to obtain cross-attention map  $\mathbf{R}_{CA}^+$ . We then adopt layout mask  $\mathcal{M}$  to extract attention value at background and evaluate overall attention accuracy with the following criterion:

$$\text{CAS}(\mathbf{R}_{CA}^+, \mathcal{M}) = \sum_{i=1}^n |\mathbf{R}_{CA,i}^+ - \text{AVG}(\mathbf{R}_{CA,i}^+ \odot (1 - \mathcal{M}_i))| \odot (1 - \mathcal{M}_i), \quad (3)$$

216 where  $\mathcal{M}_i$  and  $\mathbf{R}_{CA,i}^+$  are the mask and triggered attention map for instance  $i$ , respectively.  $\text{AVG}(\cdot)$   
 217 is the averaging operation. In our design,  $\text{AVG}(\mathbf{R}_{CA,i}^+ \odot (1 - \mathcal{M}_i))$  extracts instance  $i$ 's averaged  
 218 background attention values. Eq. 3 subsequently quantifies the total absolute deviation beyond  
 219 instance  $i$ 's scope. Higher CAS indicates potential bounding box mis-alignment for the generated  
 220 instance as the instance will lead to high dispersion at background. Based on CAS, it is obvious that  
 221  $\mathbf{R}^+$  should trigger lower CAS value than  $\mathbf{R}^-$ , formulating an inequality  $\text{CAS}_{\mathbf{R}^+} < \text{CAS}_{\mathbf{R}^-}$ . The  
 222 inequality could be further transformed into our “Instance Disentanglement Constraint”:

$$224 \quad L_{dis}(\mathbf{R}_{CA}^+, \mathbf{R}_{CA}^-, \mathcal{M}) = \text{Softplus} \left[ \text{CAS}(\mathbf{R}_{CA}^+, \mathcal{M}) - \text{CAS}(\mathbf{R}_{CA}^-, \mathcal{M}) \right], \quad (4)$$

225 where  $\text{Softplus} = \ln(1 + \exp(\cdot))$  is a monotonically increasing function. Minimizing Eq. 4 is  
 226 equivalent to encourage lower CAS score for  $\mathbf{R}^+$  and higher CAS for  $\mathbf{R}^-$ , thus enabling our  
 227 disentanglement to heuristically learn semantic-related features for L2I generation.

### 229 3.3 REINFORCEMENT LEARNING STAGE

231 **RL Formulation.** We build our algorithm based on classical PPO (Schulman et al., 2017). The  
 232 elements of conventional RL could be summarized into four essentials: (1) state (2) action (3) policy  
 233 (4) reward. We then introduce these elements in the context of flow matching model SD3. For  
 234 the given time  $t$  and condition  $\mathbf{y} = \{\mathbf{b}, \mathbf{p}\}$  ( $\mathbf{b}$  and  $\mathbf{p}$  are bounding box and their corresponding  
 235 instance captions), “state” is  $\mathbf{s}_t \triangleq (\mathbf{x}_t, t, \mathbf{y})$  “action” denotes denoised latent at  $t - 1$ , i.e.,  $\mathbf{a}_t \triangleq \mathbf{x}_{t-1}$ .  
 236 “Policy” denotes the transition probability between two time steps  $\pi(\mathbf{a}_t | \mathbf{s}_t) \triangleq \pi(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{y})$ . By  
 237 inferring from  $T$  to 0, the model generate final output  $\mathbf{x}_0$ , which could be subsequently decoded by  
 238 VAE to obtain image  $\mathbf{X}_0$ , and evaluated with reward model  $r(\cdot)$  in terms of spatial accuracies and  
 239 instance consistency.  $r(\cdot)$  assigns score as a guidance to supervise the overall generation process, thus  
 240 promoting the L2I accuracy when novel prompts emerges. For UNet-based diffusion model, policy  
 241 is derived based on DDPM (Ho et al., 2020), which contains randomness to achieve exploration.  
 242 However, MM-DiT-based diffusion is built upon flow matching (Esser et al., 2024), which relies  
 243 on deterministic ODE and hinders environmental exploration. We thus divert deterministic ODE  
 244 sampling to SDE to formulate our policy. Specifically, we follow (Domingo-Enrich et al., 2024;  
 245 Albergo et al., 2023; Liu et al., 2025a) and change the policy as follow:

$$246 \quad \mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \left[ v_\theta(\mathbf{x}_t, t, \mathbf{y}) + \frac{\sigma_t^2}{2t} (\mathbf{x}_t + (1-t)v_\theta(\mathbf{x}_t, t, \mathbf{y})) \right] \Delta t + \sigma_t \sqrt{\Delta t} \epsilon, \quad (5)$$

247 where  $\mathbf{x}_t$  is the partially denoised image latent at time  $t$ ,  $v_\theta$  is the predicted velocity.  $\epsilon$  is the  
 248 random term sampled from standard Gaussian distribution  $\mathcal{N}(0, 1)$ .  $\sigma_t = a \sqrt{\frac{t}{1-t}}$  and  $a = 0.7$  is a  
 249 hyper-parameter defined in (Liu et al., 2025a).

250 **Reward Definition.** Considering the layout-control nature of L2I task, we choose Grounding-DINO  
 251 (GDINO) (Liu et al., 2024) as the reward model  $r(\cdot)$  to assess the overall spatial accuracy and  
 252 instance consistency for generated images. GDINO receives instance captions and  $\mathbf{X}_0$  to detect  
 253 the location of objects within the input, then returns detected bounding box and confidence score  
 254  $\mathbf{o}_{pred} = \{\mathbf{b}_{pred}, \mathbf{c}_{pred}\}$ . We thus define the reward based on  $\mathbf{o}_{pred}$  and ground-truth  $\mathbf{o}$ :

$$257 \quad r(\mathbf{o}, \mathbf{o}_{pred}) = \sum_i \left[ \text{IoU}(b_{pred,i}, b_i) + c_{pred,i} \right], \quad (6)$$

258 where  $b_{pred,i}$  is the ground-truth bounding box for instance  $i$ ,  $c_{pred,i}$  is the confidence score in  
 259 detecting instance  $i$  based on instance caption  $p_{pred,i}$ . IoU is the function to compute IoU score  
 260 between ground-truth bounding box and GDINO-predicted counterpart. Based on these formulations,  
 261 we require L2I model to perform actions and interact with environment, seeking the optimal action  
 262 by optimizing underlying policy. Following standard RL formulation, we term our diffusion model as  
 263 “actor net”.

264 **Critic-Net.** During RL fine-tuning, we also introduce critic-net  $\phi$  as a collaborator to further improve  
 265 actor-net's generation capability. The critic-net is a light-weighted MLP, which receives model state  
 266  $s_t$  and embedded user inputs  $\mathbf{y}$  to predict final scalar reward value. Its training goal is minimizing the  
 267 discrepancy between predicted and actual reward given by GDINO to learn the average reward:

$$269 \quad L_{critic}(s_t, \mathbf{o}, \mathbf{o}_{pred}) = \left[ \phi(s_t) - r(\mathbf{o}, \mathbf{o}_{pred}) \right]^2. \quad (7)$$

270 With critic-net, we could further define advantage score by subtracting critic-net's prediction from  
 271 reward:

$$272 \quad A(s_t, \mathbf{o}, \mathbf{o}_{pred}) = r(\mathbf{o}, \mathbf{o}_{pred}) - \phi(s_t), \quad (8)$$

273 the advantage function measures the expected reward over the average performance, which facilitates  
 274 actor-critic collaboration.

275 **Actor-Critic Collaboration via PPO.** The goal of RL is fine-tuning actor-net to obtain high advantage  
 276 scores for generated images. By following classical PPO (Schulman et al., 2017) algorithm, we use  
 277 importance sampling to optimize actor net. Specifically, we compute probability ratio  $\rho_t$  at  $t$ -th step  
 278 by comparing the policy for current and old model:

$$280 \quad \rho_t = \frac{\pi(a_t | s_t, \mathbf{y})}{\pi(a_t^{old} | s_t^{old}, \mathbf{y})}, \quad (9)$$

283 where  $a_t^{old}$  and  $s_t^{old}$  are action and state for old actor net, which is commonly the EMA model (Klinker,  
 284 2011). The final loss for actor-net could be formulated as:

$$285 \quad L_{rl} = \min \left[ \rho_t A_t, \text{clip}(\rho_t, 1 - \zeta, 1 + \zeta) A_t \right] + \text{KL}(\pi(a_t | s_t, \mathbf{y}) || \pi(a_t^{old} | s_t^{old}, \mathbf{y})), \quad (10)$$

287 where  $\text{clip}(\cdot)$  is the clipping function, limiting the probability ratio in the range of  $[1-\zeta, 1+\zeta]$  to ensure  
 288 stable training (Schulman et al., 2017),  $\text{KL}(\cdot || \cdot)$  is the KL divergence to avoid training collapse. It  
 289 should be noted that  $L_{rl}$  instructs and optimizes actor-net through  $\rho_t$  to achieve the enhancement of  
 290 advantage scores  $A_t$ .  $A_t$  does not directly supervise actor-net's training.

291 **Joint Optimization.** We incorporate flow matching loss (Esser et al., 2024) to optimize our  
 292 actor-net:

$$293 \quad L_{ldm} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, 1), t, \mathbf{y}, \mathbf{x}_t} \|\mathbf{v} - \mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{y})\|_2^2, \quad (11)$$

295 where  $\mathbf{v}$  is the ground-truth velocity, obtained through encoded image and Gaussian noise. For  
 296 UNet-based diffusion model like SD 1.5 (Rombach et al., 2022), the objective of training should be  
 297 noise prediction (Ho et al., 2020). We will introduce our UNet-based solution in the supplementary.  
 298 The final loss for actor-net's optimization are formulated as follow:

$$299 \quad L_{act} = L_{ldm} + \lambda_{dis} L_{dis} + \lambda_{rl} L_{rl}, \quad (12)$$

300 where  $\lambda_{dis}$  and  $\lambda_{rl}$  are balancing factors. In practice, we utilize model pool  $\mathcal{MP}$  to collect old  
 301 model's policies.  $\mathcal{MP}$  is a memory pool, storing old model's policies to facilitate RL. When  
 302 computing Eq. 12, we randomly sample an old policy for image generation and optimization. The  
 303 overall training pipeline is shown in our appendix, at Alg. 1.

305 **RL Speed-up.** Our original RL scheme requires trajectory sampling for reward computation and  
 306 action-based fine-tuning, consuming large amount of time due to the inference of model with full  
 307 time-steps. Recently, the finding that optimization at early time steps (Zhou et al., 2024c; Kang  
 308 et al., 2025; Zheng et al., 2025) yields similar or even better results inspiring us to conduct RL at  
 309 early time steps. We therefore conduct RL only at the first 20% time steps and save corresponding  
 310 actions to avoid reference model's action computation. Moreover, our PPO-based RL naturally has  
 311 computational advantage over the popular GRPO-based methods (Liu et al., 2025a) as GRPO requires  
 312 sampling several trajectories at the same time for group-level advantage computation.

## 313 4 EXPERIMENT

### 315 4.1 EXPERIMENTAL SETTING

317 **Training Sets.** We train our model with LayoutSAM (Zhang et al., 2025b) and COCO-2014-  
 318 MIG (Zhou et al., 2024b). LayoutSAM contains 2.7 million image-text pairs and 10.7 million  
 319 fine-grained instance-level captions derived from SAM (Kirillov et al., 2023) dataset. Different from  
 320 LayoutSAM, COCO-2014-MIG (Zhou et al., 2024b) adopts stanza (Qi et al., 2020) to recognize  
 321 entities, while obtaining instance-level bounding boxes with Grounding-DINO. Therefore COCO-  
 322 2014-MIG has coarse instance captions while LayoutSAM has detailed instance captions. We only  
 323 use the prompt (layout and instance captions) from COCO-2014-MIG for RL while adopt image-text  
 pairs from LayoutSAM for diffusion training.

324 Table 1: Comparison with state-of-the-art methods on COCO-MIG. We compare our method with  
 325 state-of-the-art generation methods, including BoxDiff (Xie et al., 2023), Reco (Yang et al., 2023),  
 326 GLIGEN (Li et al., 2023), RichContext Cheng et al. (2024b), InstanceDiff (Wang et al., 2024),  
 327 MIGC (Zhou et al., 2024b), and Creati-Layout (Zhang et al., 2025b). \*: Evaluated based on MIGC’s  
 328 code. **Red** and **Green** are best and second best results.

Methods	Instance Success Rate ↑						mIoU ↑					
	L2	L3	L4	L5	L6	Avg	L2	L3	L4	L5	L6	Avg
BoxDiff	24.61	19.22	14.20	11.92	9.31	15.85	32.54	29.88	25.39	23.81	21.19	26.56
GLIGEN	42.30	35.55	32.66	28.18	30.84	33.89	37.58	32.34	29.95	26.60	27.70	30.83
RichContext	<b>40.31</b>	<b>30.83</b>	<b>30.78</b>	<b>26.50</b>	<b>25.42</b>	<b>30.76</b>	<b>37.88</b>	<b>31.43</b>	<b>30.35</b>	<b>28.42</b>	<b>26.59</b>	<b>30.93</b>
InstanceDiff	58.00	52.16	55.03	47.59	47.12	51.98	52.14	48.64	50.36	42.64	42.86	47.33
MIGC	67.70	59.61	58.09	<b>56.16</b>	56.88	59.68	59.39	52.73	51.45	49.52	49.89	52.60
Reco	65.50	56.10	52.30	52.40	<b>58.30</b>	56.90	55.70	46.70	47.20	43.30	48.80	47.60
Creati-Layout*	65.93	65.41	56.40	50.62	50.00	57.67	56.61	56.29	50.30	45.85	45.66	50.94
Ours (SD-1.5)	<b>79.10</b>	<b>70.24</b>	<b>65.48</b>	<b>63.87</b>	<b>66.97</b>	<b>69.13</b>	<b>70.61</b>	<b>62.10</b>	<b>58.63</b>	<b>56.04</b>	<b>58.18</b>	<b>68.18</b>
Ours (SD-3)	<b>76.87</b>	69.16	<b>62.96</b>	52.37	52.39	<b>62.75</b>	63.35	59.01	53.80	51.32	50.54	55.60

340  
 341 **Evaluation Protocol.** We evaluate our method on two large-scale benchmarks: COCO-MIG (Zhou  
 342 et al., 2024b) and LayoutSAM-eval (Zhang et al., 2025b). COCO-MIG is composed of 800 randomly  
 343 sampled multi-instance prompts from COCO-2014, each prompt contains 2 to 6 instances and  
 344 therefore the whole benchmark could be split into 5 levels (L2-L6) in terms of instance numbers.  
 345 As COCO-MIG focuses on evaluating spatial correctness, all generated images will be checked by  
 346 Grounded-SAM (Ren et al., 2024) to compute instance success rate (ISR) and mIoU. ISR evaluates  
 347 percentage that all instances are successfully recognized by Grounded-SAM, while mIoU calculates  
 348 the mean of the maximum IoU for all instances. Please refer to (Zhou et al., 2024b) for more  
 349 details. LayoutSAM-eval (Zhang et al., 2025b) is another large-scale benchmark for L2I with 5,000  
 350 prompts. The evaluation on LayoutSAM-eval is based on MINI-CPM’s inquiry (Yao et al., 2024) in  
 351 terms of each instance’s spatial location, color, texture, and shape. As LayoutSAM-eval provides  
 352 real images for the given prompts, we also report general image quality metrics, including FID,  
 353 PickScore (Kirstain et al., 2023), and IS on this benchmark.

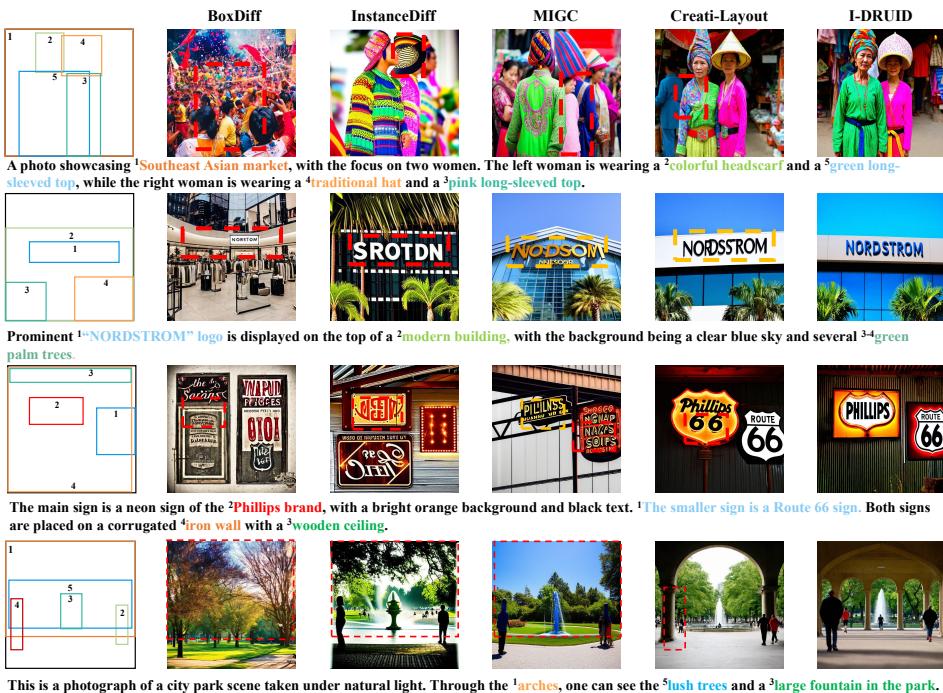
354 **Implementation Details.** We validate our method on both SD-1.5 (Rombach et al., 2022) and  
 355 SD3-mid (Esser et al., 2024). Conducting paired image-text training with LayoutSAM while un-  
 356 paired RL with COCO-2014-MIG. *The optimized model is evaluated on COCO-MIG to check OOD*  
 357 *generalization while assessed on LayoutSAM-eval to check basic L2I capability.* For SD-1.5 variant,  
 358 please check our supplementary for details. For SD3-based variant, we deploy our IDM and IDC  
 359 on every joint-attention transformer layers. During training, we set training batch size as 128, RL  
 360 batch size as 16, and optimize the model for 20 epochs with a learning rate of  $1 \times 10^{-4}$ , which will  
 361 take 4 days on 8 NVIDIA H20 GPUs. Note that RL will not be included into training for the first  
 362 10 epochs. During RL, the critic-net and actor-net are optimized in a GAN-like manner. Critic-net  
 363 will be optimized for 5 times before the collaboration with actor-net. During the inference, we set  
 364 classifier free guidance scale as 7.5. We follow the inference-time trick used in (Zhang et al., 2025b),  
 365 activating IDM for the first 30% time steps during inference, while conduct vanilla joint attention for  
 366 the rest 70%. We set  $\lambda_{dis} = \lambda_{rl} = 1$  and  $\zeta = 1 \times 10^{-4}$ . We set training resolution to  $1024 \times 1024$   
 367 for SD3. We set maximum number of instances as 10 for each training sample, *i.e.*, only the first 10  
 368 instances will be kept during training. Samples with less than 10 instances will be padded with zero  
 369 for both text and layout embeddings.

## 370 4.2 COMPARISON WITH STATE-OF-THE-ARTS

371 **Results on COCO-MIG.** We first report the results on COCO-MIG to check L2I generalization.  
 372 As shown in Tab. 1, we have two conclusions. (1) Previous methods are struggling in adapting to  
 373 novel L2I scenes. Specifically, InstanceDiff (Wang et al., 2024) and Creati-Layout (Zhang et al.,  
 374 2025b) are trained with detailed instance captions, which is different from MIGC (Zhou et al., 2024b)  
 375 that is trained with short captions. As COCO-MIG assigns short captions for each instance, both  
 376 InstanceDiff and Creati-Layout fail to outperform MIGC, demonstrating the necessity of improving  
 377 L2I model’s generalization. (2) Our method outperforms all previous works in terms of averaged ISR  
 378 and mIoU under both SD-1.5 and SD3 scenario, showing its strong flexibility and L2I generalization.

378 Table 2: Comparison on LayoutSAM-eval. We compare our method with InstanceDiff (Wang et al.,  
379 2024), Ranni (Feng et al., 2023), BeYourself (Dahary et al., 2024), MIGC (Zhou et al., 2024b),  
380 HiCo (Cheng et al., 2024a), and CreatiLayout (Zhang et al., 2025b).

Methods	Spatial $\uparrow$	Color $\uparrow$	Texture $\uparrow$	Shape $\uparrow$	FID $\downarrow$	PickScore $\uparrow$	IS $\uparrow$
Ranni	41.38	24.10	25.57	23.35	27.24	20.49	19.81
BeYourself	53.99	31.73	35.26	32.75	28.10	20.20	17.98
MIGC	85.66	66.97	71.24	69.06	21.19	20.71	19.65
InstanceDiff	87.99	69.16	72.78	71.08	19.67	21.01	20.02
HiCo	87.04	69.19	72.36	71.10	22.61	21.70	20.15
Creati-Layout	92.67	74.45	77.21	75.93	19.10	22.02	22.04
Ours (SD-1.5)	86.95	70.49	73.56	72.30	22.92	21.98	17.95
Ours (SD-3)	<b>93.14</b>	<b>75.37</b>	<b>78.35</b>	<b>77.20</b>	<b>17.21</b>	<b>23.16</b>	<b>22.45</b>



413 Figure 3: Visualization of generated samples with I-DRUID and state-of-the-arts. **Dashed red box:**  
414 wrongly generated. **Dashed yellow box:** Distorted text.

416 By introducing unpaired data from COCO-2014-MIG, I-DRUID learns to generate with short captions  
417 under the guidance from GDINO and thus enhances its L2I generalization in novel COCO-MIG.

418 **Results on LayoutSAM-eval.** We also evaluate our method on LayoutSAM-eval in terms of spatial,  
419 color, texture, and shape to show its basic L2I capability. As LayoutSAM-eval provides image-text  
420 pair, we also report the general image quality metrics like FID at Tab. 2. It is obvious our SD3  
421 counterpart outperforms other methods. SD3 provides better image quality, enabling our method to  
422 achieve lower FID scores on LayoutSAM-eval. Moreover, our method also outperforms other state-  
423 of-the-arts in all aspects (spatial, color, texture, and shape), demonstrating its efficacy on large-scale  
424 L2I benchmark. It should be noted that MIGC (Zhou et al., 2024b) fails to achieve better results  
425 than InstanceDiff (Wang et al., 2024) and Creati-Layout (Zhang et al., 2025b), demonstrating the  
426 significance of improving L2I’s generalization to different scenes, especially when there is large  
427 domain gap between training and testing sets.

### 4.3 VISUAL COMPARISON

431 We visualize some L2I results and compare them with state-of-the-arts for an intuitive evaluation. As  
432 shown in Fig. 3, we conclude that (1) Our method achieves better image quality for L2I. Specifically,

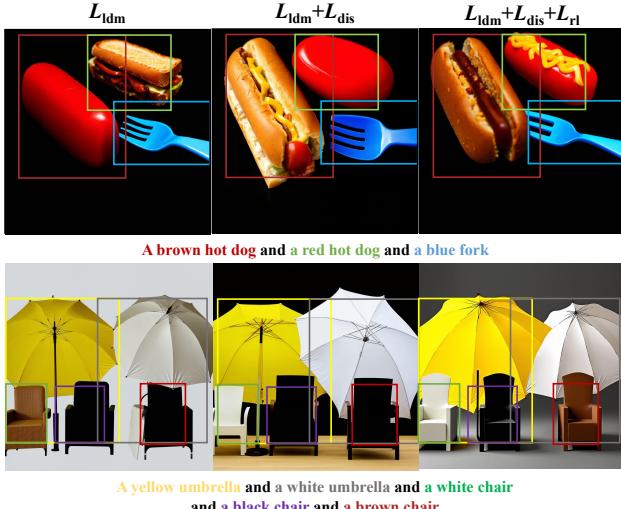
432 Table 3: Ablation study. We report the averaged ISR on COCO-MIG and four main criteria on  
 433 LayoutSAM-eval to check each component’s efficacy. **Green** and **red** are best and second best results.  
 434

435 436 No.	Attributes				COCO-MIG Avg ISR	LayoutSAM-eval			
	437 IDM	438 RL-PPO	439 RL-GRPO	440 SFT		441 Spatial	442 Color	443 Texture	444 Shape
1	×	×	×	×	56.82	86.96	71.08	73.20	72.44
2	✓	×	×	×	57.64	88.53	73.12	74.93	75.62
3	✓	×	×	✓	66.92	89.75	72.21	74.33	73.19
4	✓	×	✓	×	61.64	92.86	74.67	78.65	75.80
5	✗	✓	✗	✗	60.23	91.47	72.45	74.79	72.93
6	✓	✓	✗	✗	62.75	93.14	75.37	78.35	77.20

443 our method and SD3-based Creati-Layout generate human face and characters with high quality, while  
 444 other methods like MIGC fail to achieve this. (2) Our method achieves better detail interpretation. *e.g.*,  
 445 at the third row of Fig. 3, I-DRUID correctly rendered the given instance prompt “Phillips brand”,  
 446 while Creati-layout additionally generates “66”, which is potentially caused by the attribute leakage  
 447 problem. By explicitly considering the instance-disentangled representation for L2I, the problem is  
 448 alleviated. Therefore, we demonstrate that our method show competitive capability in L2I.  
 449

#### 450 4.4 ABLATION STUDY

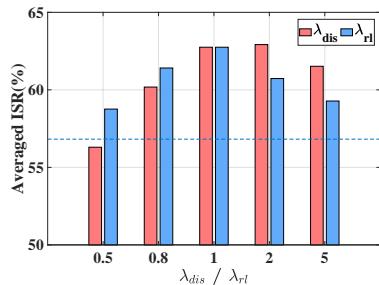
451 **Ablated Visual Comparison.** We  
 452 gradually add each component of our  
 453 proposed method and check the direct  
 454 influence of each component on the  
 455 final generated images through visu-  
 456 alization. The results are shown in  
 457 Fig. 4. From the visualization, at-  
 458 tribute leakage could be easily found  
 459 when solely using  $L_{ldm}$  for opti-  
 460 mization (at the column of “ $L_{ldm}$ ”).  
 461 Specifically, at the first row of Fig. 4,  
 462 “a brown hot dog” is mistakenly gen-  
 463 erated as a red hot dog due to nearby  
 464 instance prompt “a red hot dog”. Sim-  
 465 ilar results could also be noted at the  
 466 second row, where the “white chair”  
 467 is mistakenly generated as “a brown  
 468 chair”. After introducing  $L_{dis}$ , the  
 469 attribute leakage problem is alleviated  
 470 as all instances are generated in their  
 471 corresponding locations. Moreover, after adding  $L_{rl}$ , the L2I model achieves fine-level attribute  
 472 control for instances. *e.g.*, at the second row, the instance “a brown chair” adjust its color to align  
 473 with the attribute “brown”.



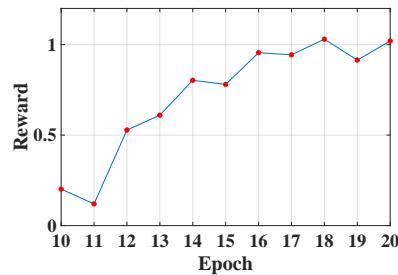
474 Figure 4: Ablated Visual Comparison.  
 475

476 **Effectiveness of using IDM.** To check the efficacy of each component, we also conduct ablation  
 477 study on COCO-MIG and LayoutSAM-eval and report the results in Tab. 3. We mainly check four  
 478 attributes of our method. “IDM”: using our IDM+IDC to disentangle features during L2I; “RL-PPO”:  
 479 PPO fine-tuning; “RL-GRPO”: replacing PPO with GRPO. GRPO does not require critic net for  
 480 advantage estimation, but needs to sample multiple trajectories at the same time; “SFT”: We  
 481 annotate COCO-2014-MIG with stanza (Qi et al., 2020) and GDINO in an offline manner while integrating  
 482 the annotated data into training. *Note that SFT introduces images from COCO-2014-MIG for joint*  
 483 *optimization and thus cannot be conducted with unpaired data.* By comparing “No. 1” and “No.  
 484 2”, we found that introducing IDC is beneficial to improving L2I accuracies on both benchmarks,  
 485 yielding improvement on L2I accuracies on all metrics. The results indicate IDC’s effectiveness in  
 486 generating multiple samples with accurate attribute and layout.

487 **Effectiveness of RL.** By further comparing the results between “No. 1”, “No. 3”, “No. 4”, “No.  
 488 5” and “No. 6”, we conclude RL is beneficial to improving L2I accuracies. Specifically, we only  
 489 adopt unpaired prompt-only data from COCO-2014-MIG in our RL-based experiments, but the



496 Figure 5: Sensitivity to  $\lambda_{dis}$  and  $\lambda_{rl}$ . “Blue  
497 dashed line”: Result when  $\lambda_{rl} = \lambda_{dis} = 0$ .



505 Figure 6: Reward stability.

506 results on COCO-MIG demonstrate RL’s efficacy in improving L2I model’s generalization with  
507 unpaired data. It should also be noted that “SFT” variant in “No. 3” is optimized with paired data  
508 from COCO-2014-MIG and violates our experimental setting that requires unpaired data for training.  
509 Although “SFT” achieves better results on COCO-MIG, it fails to enhance L2I accuracies under  
510 LayoutSAM-eval. Therefore, *SFT may lead to performance degradation due to the domain gap*  
511 *between different training sets*. Moreover, introducing RL does not lead to performance degradation,  
512 which further demonstrates RL’s efficacy.

513 **Effectiveness of using Different RL scheme.** We also compare PPO-based RL with GRPO-  
514 based (Shao et al., 2024; Liu et al., 2025a) RL. By comparing “No. 4” and “No. 6”, we note that using  
515 different RL strategies yields similar improvement over baseline. The results demonstrate the efficacy  
516 of RL even when different RL strategies. Moreover, GRPO-based counterpart requires sampling  
517 several trajectories at the same time to obtain group-level advantage, significantly increasing the  
518 training time cost. As advantage in our solution is obtained through light-weighted critic-net, training  
519 under our scheme is much faster than GRPO-based method. Our method achieves training speed of  
520 nearly 15s / iter on NVIDIA H20, while GRPO-based method needs more than 60s / iter when setting  
521 group size to 4 and conducting RL on full time steps.

#### 522 4.5 SENSITIVITY ANALYSIS

523 **Sensitivity to  $\lambda_{dis}$  and  $\lambda_{rl}$ .** We first check model’s sensitivity to hyper-parameters  $\lambda_{dis}$  and  $\lambda_{rl}$  with  
524 averaged ISR over all instance-levels. At this experiment, we change  $\lambda_{dis} / \lambda_{rl}$  from 0.5 to 5 while  
525 keeping  $\lambda_{rl} / \lambda_{dis}$  at 1. The results are shown in Fig. 5, we visualize baseline method’s averaged ISR  
526 (results of “No.1” in Tab. 3) with blue dashed line. From the results, we draw two conclusions. (1)  
527 Our method is not sensitive to both hyper-parameters. Specifically, we note in Fig. 5 that changing  
528 these two parameters does not bring significant degradation on performance and the best results  
529 are achieved when setting  $\lambda_{dis} = \lambda_{rl} = 1$ . (2)  $\lambda_{dis}$  plays an important role in our L2I algorithm.  
530 Specifically, we note performance degradation than baseline when setting  $\lambda_{dis} = 0.5$  and  $\lambda_{rl} = 1$ .  
531 However, setting  $\lambda_{dis} = 1$  and  $\lambda_{rl} = 0.5$  will not cause such problem. We speculate disentangling  
532 process brings better representation for L2I, generalizing model in a more efficient manner as correct  
533 samples’ trajectory could be easily collected than models without disentangling.

534 **Reward stability.** We also check the reward stability during the RL process. As RL is introduced  
535 after the 10-th training epoch, we compute the averaged reward for sampled trajectories and visualize  
536 them in Fig. 6. From the figure, we conclude that although RL may encounter fluctuation during  
537 training, the training process is generally steady.

## 538 5 CONCLUSION

539 This paper advances layout to image (L2I) generation by addressing attribute leakages and generaliza-  
540 tion. For the first challenge, we devise instance disentanglement module (IDM) and utilize instance  
541 disentanglement constraint (IDC) for disentangling semantic-related features. IDC requires semantic-  
542 related features to yield more accurate attention maps than spurious ones, and thus avoids attribute  
543 leakage. For the second challenge, we formulate a reinforcement learning (RL) framework, enabling  
544 our method to learn with unpaired prompt-only data to improve model’s generalization. These two  
545 contributions formulates our I-DRUID and achieves high accuracies on several benchmarks.

540 REFERENCES  
541

542 Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and  
543 algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32:96, 2019.

544 Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying  
545 framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

546 Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna  
547 Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness  
548 from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

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

551 Chaofeng Chen, Annan Wang, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin.  
552 Enhancing diffusion models with text-encoder reinforcement learning. In *Proceedings of the  
553 European Conference on Computer Vision*, 2024.

554 Bo Cheng, Yuhang Ma, Liebucha Wu, Shanyuan Liu, Ao Ma, Xiaoyu Wu, Dawei Leng, and Yuhui  
555 Yin. Hico: Hierarchical controllable diffusion model for layout-to-image generation. In *Advances  
556 in Neural Information Processing Systems*, 2024a.

557 Jiaxin Cheng, Zixu Zhao, Tong He, Tianjun Xiao, Zheng Zhang, and Yicong Zhou. Rethinking the  
558 training and evaluation of rich-context layout-to-image generation. *Advances in Neural Information  
559 Processing Systems*, 2024b.

560 Guillaume Couairon, Marlène Careil, Matthieu Cord, Stéphane Lathuilière, and Jakob Verbeek.  
561 Zero-shot spatial layout conditioning for text-to-image diffusion models. In *Proceedings of the  
562 International Conference on Computer Vision*, pp. 2174–2183, 2023.

563 Omer Dahary, Or Patashnik, Kfir Aberman, and Daniel Cohen-Or. Be yourself: Bounded attention  
564 for multi-subject text-to-image generation. In *Proceedings of European Conference on Computer  
565 Vision*, pp. 432–448, 2024.

566 Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching:  
567 Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control.  
568 *arXiv preprint arXiv:2409.08861*, 2024.

569 Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam  
570 Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for  
571 high-resolution image synthesis. In *Proceedings of International Conference on Machine Learning*,  
572 2024.

573 Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel,  
574 Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for  
575 fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*,  
576 2023.

577 Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato  
578 Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for  
579 compositional text-to-image synthesis. In *Proceedings of International Conference in Learning  
580 Representation*, 2023.

581 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,  
582 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms  
583 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

584 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in  
585 Neural Information Processing Systems*, 2020.

586 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec  
587 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint  
588 arXiv:2412.16720*, 2024.

594 Junyong Kang, Seohyun Lim, Kyungjune Baek, and Hyunjung Shim. Rethinking direct preference  
 595 optimization in diffusion models. *arXiv preprint arXiv:2505.18736*, 2025.

596

597 Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image  
 598 generation with attention modulation. In *Proceedings of the International Conference on Computer  
 599 Vision*, pp. 7701–7711, 2023.

600 Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete  
 601 Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings  
 602 of the International Conference on Computer Vision*, pp. 4015–4026, 2023.

603

604 Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-  
 605 a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural  
 606 Information Processing Systems*, 2023.

607 Frank Klinker. Exponential moving average versus moving exponential average. *Mathematische  
 608 Semesterberichte*, pp. 97–107, 2011.

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

610

611 Phillip Y. Lee, Taehoon Yoon, and Minhyuk Sung. Groundit: Grounding diffusion transformers via  
 612 noisy patch transplantation. *Advances in Neural Information Processing Systems*, 2024.

613

614 Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li,  
 615 and Yong Jae Lee. GLIGEN: open-set grounded text-to-image generation. In *Proceedings of the  
 616 Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023.

617

618 Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang,  
 619 Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution  
 620 diffusion transformer with fine-grained chinese understanding. *arXiv preprint arXiv:2405.08748*,  
 2024.

621

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

624

625 Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin  
 626 Wang, Wenyu Qin, Menghan Xia, et al. Improving video generation with human feedback. *arXiv  
 627 preprint arXiv:2501.13918*, 2025b.

628

629 Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan  
 630 Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for  
 631 open-set object detection. In *Proceedings of European Conference on Computer Vision*, pp. 38–55,  
 2024.

632

633 Zhengyao Lv, Tianlin Pan, Chenyang Si, Zhaoxi Chen, Wangmeng Zuo, Ziwei Liu, and Kwan-Yee K  
 634 Wong. Rethinking cross-modal interaction in multimodal diffusion transformers. In *Proceedings  
 635 of the International Conference on Computer Vision*, 2025.

636

637 Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and  
 638 Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image  
 639 diffusion models. In *AAAI Conference on Artificial Intelligence*, 2023.

640

641 Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression:  
 642 Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

643

644 Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. Stanza: A python  
 645 natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*,  
 2020.

646

647 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,  
 648 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual  
 649 models from natural language supervision. In *Proceedings of International Conference on Machine  
 650 Learning*, pp. 8748–8763. PMLR, 2021.

648 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea  
 649 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances*  
 650 *in Neural Information Processing Systems*, 2023.

651

652 Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian  
 653 Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. Is reinforcement learning (not) for natural  
 654 language processing: Benchmarks, baselines, and building blocks for natural language policy  
 655 optimization. *arXiv preprint arXiv:2210.01241*, 2022.

656 Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen,  
 657 and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of International Conference*  
 658 *on Machine Learning*, pp. 8821–8831. Pmlr, 2021.

659

660 Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang,  
 661 Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing  
 662 Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks.  
 663 *arXiv preprint arXiv:2306.03514*, 2024.

664 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-  
 665 resolution image synthesis with latent diffusion models. In *Proceedings of the Conference on*  
 666 *Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

667

668 Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical  
 669 image segmentation. In *Medical Image Computing and Computer-assisted Intervention*, pp.  
 670 234–241, 2015.

671 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy  
 672 optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

673

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

677

678 Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford,  
 679 Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Advances*  
 680 *in Neural Information Processing Systems*, 2020.

681 Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models.  
 682 In *ACM SIGGRAPH Conference Proceedings*, pp. 1–11, 2023.

683

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

688

689 Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion:  
 690 Instance-level control for image generation. In *Proceedings of the Conference on Computer*  
 691 *Vision and Pattern Recognition*, pp. 6232–6242, 2024.

692

693 Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai  
 694 He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language  
 695 models with one training example. *arXiv preprint arXiv:2504.20571*, 2025.

696

697 Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and  
 698 Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained dif-  
 699 fusion. In *Proceedings of the International Conference on Computer Vision*, pp. 7452–7461,  
 700 2023.

701 Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao  
 702 Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In  
 703 *Advances in Neural Information Processing Systems*, 2024.

702 Xue Yan, Yan Song, Xidong Feng, Mengyue Yang, Haifeng Zhang, Haitham Bou Ammar, and  
 703 Jun Wang. Efficient reinforcement learning with large language model priors. In *International*  
 704 *Conference in Learning Representation*, 2024.

705  
 706 Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng  
 707 Liu, Ce Liu, Michael Zeng, et al. Reco: Region-controlled text-to-image generation. In *Proceedings*  
 708 *of the Conference on Computer Vision and Pattern Recognition*, pp. 14246–14255, 2023.

709 Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li,  
 710 Weilin Zhao, Zihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint*  
 711 *arXiv:2408.01800*, 2024.

712 Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt  
 713 adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

714  
 715 Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion  
 716 models for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2024.

717 Hong Zhang, Zhongjie Duan, Xingjun Wang, Yingda Chen, and Yu Zhang. Eligen: Entity-level  
 718 controlled image generation with regional attention. In *AAAI conference on artificial intelligence*,  
 719 2025a.

720 Hui Zhang, Dexiang Hong, Tingwei Gao, Yitong Wang, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-  
 721 Gang Jiang. Creatilayout: Siamese multimodal diffusion transformer for creative layout-to-image  
 722 generation. In *Proceedings of International Conference in Computer Vision*, 2025b.

723 Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image  
 724 diffusion models. In *Proceedings of International Conference on Computer Vision*, 2023.

725 Tianyi Zheng, Jiayang Zou, Peng-Tao Jiang, Hao Zhang, Jinwei Chen, Jia Wang, and Bo Li. Bidirec-  
 726 tional beta-tuned diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,  
 727 pp. 1–15, 2025. doi: 10.1109/TPAMI.2025.3604039.

728 Dewei Zhou, You Li, Fan Ma, Zongxin Yang, and Yi Yang. Migc++: Advanced multi-instance  
 729 generation controller for image synthesis. *IEEE Transactions on Pattern Analysis and Machine  
 730 Intelligence*, 2024a.

731 Dewei Zhou, You Li, Fan Ma, Xiaoting Zhang, and Yi Yang. Migc: Multi-instance generation  
 732 controller for text-to-image synthesis. In *Proceedings of the Conference on Computer Vision and  
 733 Pattern Recognition*, pp. 6818–6828, 2024b.

734 Zikai Zhou, Shitong Shao, Lichen Bai, Zhiqiang Xu, Bo Han, and Zeke Xie. Golden noise for  
 735 diffusion models: A learning framework. *arXiv preprint arXiv:2411.09502*, 2024c.

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756 APPENDIX  
757758 **Algorithm 1** The Process of I-DRUID.  
759

---

760 **Inputs:** Sample with  $b$  bounding boxes, a global prompt  $p_g$  and  $n$  instance captions  $p$ . Reward  
761 model  $r(\cdot)$ , critic-net, actor-net, hyper-parameters  $\lambda_{dis}$ ,  $\lambda_{rl}$ , training epochs  $E$ , action pool  $\mathcal{MP}$ ,  
762 unpaired prompt  $\{\mathbf{p}_u, \mathbf{b}_u, p_{g,u}\}$ .

763 1: // Disentangling Stage.  
764 2: Initialize L2I model with SD3-mid;  
765 3: **for**  $e$  in  $E$  **do**  
766 4:     Sample a batch of training data  $\{\mathbf{b}_e, \mathbf{p}_e, p_g\}$ ;  
767 5:     Sample time  $t$ ;  
768 6:     Obtain  $\mathbf{R}^+$  with IDM and compute Eq. 4;  
769 7:     Compute Eq. 11 with  $\mathbf{R}^+$  and time  $t$ ;  
770 8:     // RL Stage.  
771 9:     **if**  $e \geq 10$  **then**  
772 10:         Sampling trajectory with unpaired data  $\{\mathbf{p}_u, \mathbf{b}_u, p_{g,u}\}$  and enqueue actions into  $\mathcal{MP}$   
773 11:         (obtain enqueue actions for the first 20% time steps);  
774 12:         Obtaining generated images  $X_0$ ;  
775 13:         Obtaining reward with Eq. 6;  
776 14:         Computing Eq. 7 and optimize critic-net;  
777 15:         Computing advantage scores with Eq. 8;  
778 16:         Sampling old actions of the same input from  $\mathcal{MP}$ ;  
779 17:         Obtaining policy for old and current actor;  
780 18:         Computing Eq. 10 to obtain  $L_{rl}$ ;  
781 19:     **else**  
782 20:         Setting  $L_{rl}$  to 0;  
783 21:     **end if**  
784 22:     Optimizing L2I model through backpropagation with Eq. 12;  
785 **end for**

---

785 **LLM Statement.** We use LLM to polish the writing, such as correcting grammar and other errors.  
786

787 A OVERALL TRAINING PIPELINE  
788

789 We demonstrate our overall training pipeline in Alg. 1. Our training contains two parts, *i.e.*, disen-  
790 tangling stage and RL stage. The former will be conducted for the whole training process to ensure  
791 basic L2I capability while the latter will be utilized only for the last 10 epochs to improve L2I model's  
792 generalization with unpaired data. It should be noted that action pool  $\mathcal{MP}$  will be activated only for  
793 the last 10 epochs and will only save the actions for the first 20% time steps for faster RL. Moreover,  
794 during RL, actor- and critic-net will be optimized in a GAN-like manner. critic-net will be optimized  
795 5 times before joining actor-net's optimization.

796 B IMPLEMENTATION DETAILS FOR UNET-BASED METHOD  
797

799 Our method is also compatible with UNet-based diffusion models like SD-1.5. When transferring to  
800 SD-1.5, (1) IDM's location for deployment, (2) the objective for noise prediction, and (3) RL actions  
801 should be modified. For (1), we deploy IDM at the  $8 \times 8$  mid-layer and  $16 \times 16$  decoder layers of  
802 UNet to conduct feature disentanglement and subsequent cross attention. For other cross-attention  
803 layers without IDM, we optimize them with global prompt. For (2), as SD-1.5's training objective is  
804 noise prediction,  $L_{ldm}$  should be changed (Rombach et al., 2022) accordingly. Finally, for (3), the  
805 actions for SD-1.5 follows original transitional probability formula at (Ho et al., 2020).

806 C FAILURE CASE STUDY  
807

809 We visualize some failed cases in Fig. 7 to show its limitation. Specifically, our method is liable to  
810 generate flawed cases when small characters and human body emerges. However, the generation with



Figure 7: Visualization of generated failure cases.

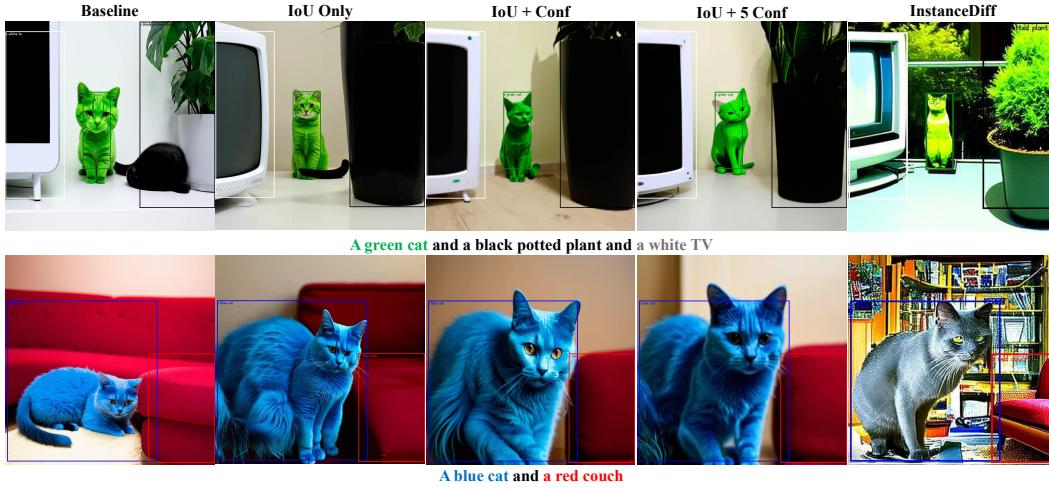


Figure 8: Justification of Reward Design.

small-scale text is a known issue in original stable diffusion (Esser et al., 2024), which will bring similar negative effects on all SD-based AIGC algorithms (Zhou et al., 2024b). To mitigate these negative effects, users could collect more data with text and human bodies to finetune the model.

We also note that previous SD1.5-based MIGC Zhou et al. (2024b) achieves better results than SD3-based Creati-Layout Zhang et al. (2025b) on COCO-MIG in terms of ‘‘Avg ISR’’. COCO-MIG is a benchmark highly focused on evaluating attribute leakage and rendering of each instance, suggesting weaker instance-level rendering capability for SD3-based L2I. We speculate the phenomenon is caused by the inherent problem in MM-attention, where text tokens are diluted in MM-attention Lv et al. (2025). Therefore, as MM-attention is heavily used in SD3, improving SD3-based L2I models generation capability in correctly rendering instance attributes becomes challenging. The problem further leads to the relatively lower L2I accuracies on COCO-MIG.

## D JUSTIFICATION OF REWARD DESIGN

As demonstrated in our experiments, we set the reward as ‘‘IoU + Confidence’’ to jointly consider spatial accuracies and instances’ generation quality. To explore the optimal reward design, we conduct experiments by setting different weights to ‘‘Confidence’’ term, and visualize generated images for different variants. The results are shown in Fig. 8, we also visualize the results from InstanceDiff Wang et al. (2024) for comparison. From the result, we draw two conclusions. (1) IoU term in RL enables better spatial control. Specifically, the ‘‘blue cat’’ at second row does not follow the given

Table 4: Quantitative Results for Reward Justification.

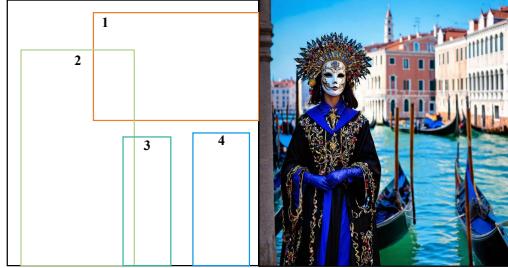
	ISR	L2	L3	L4	L5	L6	Avg
Baseline	63.30	61.04	57.46	52.20	50.10	56.82	
+ IoU	70.37	65.48	59.21	51.26	51.14	59.49	
+ Conf	76.87	69.16	62.96	52.37	52.39	62.75	
+ 5Conf	72.50	67.29	61.09	51.25	52.79	60.98	

Table 5: RL for initial states.

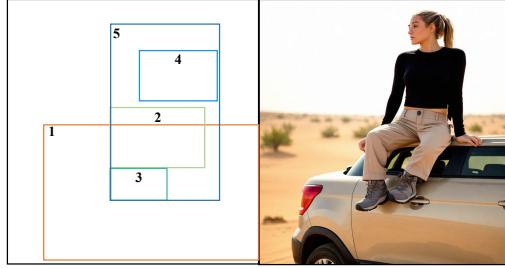
Method	Avg ISR
Full-Time Sampling	63.06
Ours	62.75

Table 6: Robustness to False Reward on COCO-MIG.

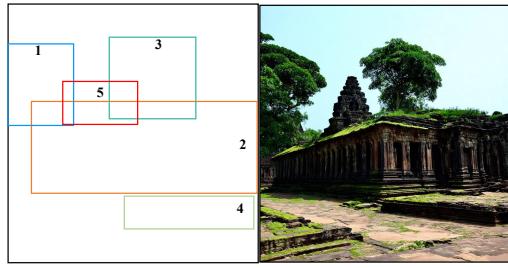
ISR	L2	L3	L4	L5	L6
w/o False Reward	76.87	69.16	62.96	52.37	52.39
w/ False Reward	67.18	66.25	57.34	51.20	51.92



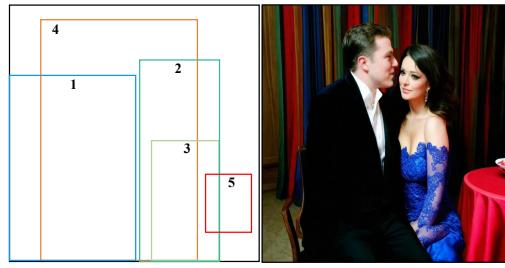
In the foreground, a figure dressed in exquisite traditional attire stands by the water. In the background, the iconic Venetian buildings and gondolas are clearly visible, with the water shimmering under the sunlight.



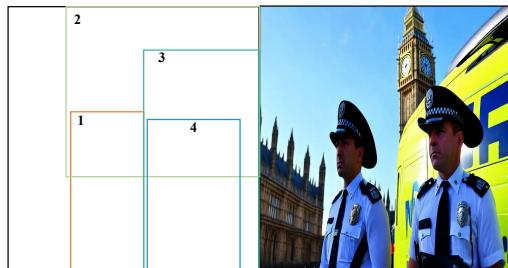
This is a photo of a woman sitting on the roof of a vehicle in a desert environment. The woman is dressed in a black top and beige trousers, wearing a pair of grey hiking shoes. She is sitting on the roof of a beige vehicle.



The temple is located in a tropical environment, with trees growing on its roof and walls. The walls of the temple are covered with moss. The ground is paved with stone slabs.



The man is dressed in a black suit, while the woman is wearing a blue evening gown with a lace design. They are sitting next to a table, with a curtain with red tablecloth in the background.



The officer on the left is wearing a black hat. The officer on the right is also wearing a black hat. They are standing in front of a yellow police van, with Big Ben and the surrounding buildings in the background.



The child in the middle is holding a small drum in his hand. The left is a young child in festive white dress and oversized hat, while the right child is focused on her dancing.

Figure 9: Visualization of more cases (Part-I).

layout when RL is not applied. After using “IoU-based” RL, the spatial control is further improved. (2) Introducing confidence score into RL further improves instance-level generation, but may lead to low image quality. Specifically, ignoring “confidence term” will lead to the unnatural generation of some instances (e.g., green cat with a black tail, or blue cat with multiple legs). These artifacts are alleviated after taking “confidence term” into RL. However, when changing the reward function to “IoU + 5 \* confidence”, we note over-saturation for some generated instances. We thus adopt “IoU + confidence” as our final reward. The quantitative results on COCO-MIG at Tab. 4 demonstrate the same conclusion.

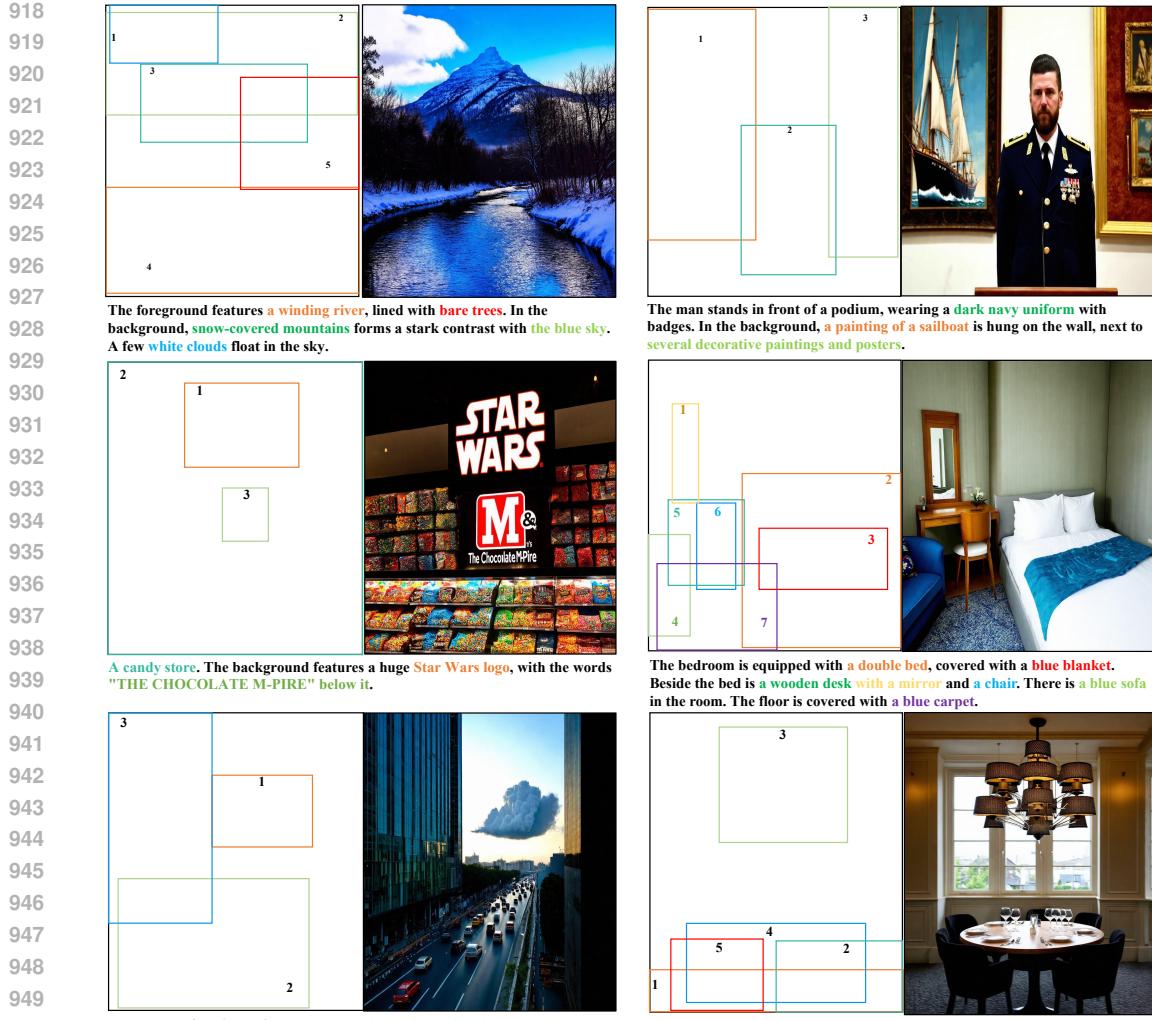


Figure 10: Visualization of more cases (Part-II).

## E RL AT EARLY TIME STEPS

We only conduct RL on the first 20% time steps to speed up the training process, raising the concern whether RL on initial states yields similar improvement on L2I trajectory sampling with full time steps (we term this variant as full-time sampling). We conduct experiment with full time trajectory sampling and evaluate L2I model on COCO-MIG to demonstrate this. As shown in Tab. 5, using full-time sampling yields slight improvement but requires higher computational cost as optimization will be conducted on the whole trajectory. Specifically, with RL training at early time steps, our method achieves training speed of nearly 15.4s / iter on NVIDIA H20, while the training speed with full trajectory sampling is nearly 82.6 s / iter. We thus propose to optimize initial states for faster RL.

## F MORE VISUALIZATION

We visualize more results in Fig. 9 and Fig. 10 to show our method’s efficacy in L2I. Specifically, our method achieves both spatial correctness and high instance image quality, demonstrating its potential in L2I field.

972 G ROBUSTNESS TO UNRELIABLE REWARD  
973974 We also conduct experiment to check our method’s robustness when encountering false reward by  
975 deliberately assigning false reward (minus of the reward). The results are shown in Tab. 6.  
976977 From the result, we conclude that our RL part suffers from performance degradation when wrong  
978 rewards are given, but the performance degradation does not cause catastrophic false generation. This  
979 is due to two reasons:980 (1) Joint optimization with paired data and KL terms. This is a commonly used strategy to overcome  
981 training instability during RL Stiennon et al. (2020); Fan et al. (2023); Black et al. (2023). By using  
982 KL term and keeping original pair-wise training loss, our method mitigates the negative effects  
983 brought by RL. Moreover, we only conduct RL at the last 10 epochs of training, where our model  
984 already has strong capability for L2I generation. During the trajectory sampling, the generated images  
985 will not introduce samples with completely false layout, alleviating the negative effects brought by  
986 false reward.987 (2) Feature Disentanglement. As demonstrated in Tab. 3, after using IDM for RL, the final L2I  
988 accuracies are improved. Therefore, our IDM provides better representation for subsequent RL.989 To further alleviate the negative effects brought by false reward, one can combine multiple reward  
990 functions (combining multiple detectors, *etc.*) to improve reward robustness, or carefully adjusting  
991 KL weights to strike a balance between aggressive exploration or stable training.  
992993 H PSEUDO CODE OF OUR I-DRUID  
994995 We provide the pytorch-like pseudo code for our disentangling module and RL training. Specifically,  
996 “CAS score computation” is the function to compute CAS score. “Compute  $L_{dis}$ ” is the function  
997 to compute  $L_{dis}$ , “Compute  $L_{rl}$ ” is the function to compute  $L_{rl}$ , while “SDE-ODE Conversion” is  
998 conducted on original “FlowMatchingScheduler” of SD3 to implement Eq. 5.  
999

## 1000 CAS score computation

```

1001
1002 def _get_bg_attn_loss(self, worse_attn, bg_mask):
1003     BPN, heads, HW, _ = ori_attnmap.shape
1004     B = bg_mask.shape[0]
1005     bg_mask = bg_mask.reshape(-1, 1, HW)
1006
1007     # get ori bg attn map
1008     ori_attnmap = torch.sum(ori_attnmap[:, :, :, 1:], dim=-1)
1009     ori_attnmap = ori_attnmap.reshape(B, -1, HW)
1010     bg_mask = bg_mask.reshape(B, 1, HW)
1011
1012     # get general bg info
1013     bg_attn_mean = (ori_attnmap * bg_mask).sum(dim=-1) / \
1014         ((bg_mask).sum(dim=-1) + 1e-6) # (B, PN*heads)
1015     loss_attn = (abs(ori_attnmap - bg_attn_mean[..., None].detach()) \
1016         * bg_mask).sum(dim=-1) / (bg_mask.sum(dim=-1) + 1e-6)
1017
1018     return loss_attn.mean()

```

1016 Compute  $L_{dis}$ 

```

1019
1020 def _get_disen_loss(self, worse_attn, better_attn, masks):
1021     ranking_loss = torch.nn.SoftMarginLoss()
1022     H_ori = self._get_bg_attn_loss(worse_attn, masks)
1023     H_better = self._get_bg_attn_loss(better_attn, masks)
1024     y = torch.ones_like(H_ori)
1025
1026     return ranking_loss(H_ori-H_better, y)

```

1024 Compute  $L_{rl}$

```

1026
1027
1028 def rl_train(self, state_dict, chosen_pid, height, width, \
1029     clip_range=1e-4):
1030     cur_glob = [state_dict["caption"][pid] for pid in chosen_pid]
1031     sub_prt = [state_dict["sub_prompts"][pid] for pid in chosen_pid]
1032     sub_bbox = [state_dict["bboxes"][pid] for pid in chosen_pid]
1033     ref_logp = state_dict["log_prob"][chosen_pid, :]
1034     cur_logp, cur_reward, infer_img = self.pipe.train_get_logp(
1035         cur_glob, height, width, num_inference_steps=50, \
1036         max_objs = 10, timesteps = None,
1037         bbox_phrases=sub_prt, bbox_raw=sub_bbox, \
1038         reward_func=self.reward_func,
1039         load_image=self.load_image, topk=10
1040     )
1041     topk = ref_logp.shape[-1]
1042     cur_adv = cur_reward.to(cur_logp.device).view(-1,1) - \
1043         self.value_func(infer_img)
1044
1045     ratio = torch.exp(cur_logp[:, :topk] - \
1046         ref_logp.to(cur_logp.device))
1047     unclipped_loss = -cur_adv * ratio
1048     clipped_loss = -cur_adv * torch.clamp(
1049         ratio, 1.0 - clip_range, 1.0 + clip_range,
1050     )
1051     return torch.mean(torch.maximum(unclipped_loss, clipped_loss)) + \
1052         self.kl_loss(cur_logp, ref_logp)
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079

```

## ODE-SDE Conversion

```

1050
1051
1052 def sde_step_with_logprob(
1053     self, model_output, timestep, sample = 0.7,
1054     prev_sample = None, generator = None,
1055     ):
1056     sample=sample.float()
1057     if prev_sample is not None:
1058         prev_sample=prev_sample.float()
1059
1060     step_index = [self.index_for_timestep(t) for t in timestep]
1061     prev_step_index = [step+1 for step in step_index]
1062     sigma = self.sigmas[step_index].view(
1063         -1, *(([1] * (len(sample.shape) - 1)))
1064     )
1065     sigma_prev = self.sigmas[prev_step_index].view(
1066         -1, *(([1] * (len(sample.shape) - 1)))
1067     )
1068     sigma_max = self.sigmas[1].item()
1069     dt = sigma_prev - sigma
1070     std_dev_t = torch.sqrt(sigma / \
1071         (1 - torch.where(sigma == 1, sigma_max, sigma))) * noise_level
1072
1073     # sde
1074     prev_sample_mean = sample* (1+std_dev_t**2/(2*sigma)*dt)+\
1075         model_output*(1+std_dev_t**2*(1-sigma)/(2*sigma))*dt
1076
1077     if prev_sample is None:
1078         variance_noise = randn_tensor(
1079             model_output.shape,
1080             generator=generator,
1081             device=model_output.device,
1082             dtype=model_output.dtype,
1083         )
1084     prev_sample = prev_sample_mean + \
1085         std_dev_t * torch.sqrt(-1*dt) * variance_noise
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
31
```

```

1080
1081     log_prob = (
1082         -((prev_sample.detach() - prev_sample_mean) ** 2) /
1083         (2 * ((std_dev_t * torch.sqrt(-1*dt))**2))
1084         - torch.log(std_dev_t * torch.sqrt(-1*dt))
1085         - torch.log(torch.sqrt(2 * torch.as_tensor(math.pi)))
1086     )
1087
1088     log_prob = log_prob.mean(dim=tuple(range(1,
1089                                         log_prob.ndim)))
1090     return prev_sample, log_prob, prev_sample_mean, std_dev_t
1091
1092 def compute_log_prob(self, noise_pred, sample, time):
1093     prev_sample, log_prob, prev_sample_mean, std_dev_t =
1094         self.sde_step_with_logprob(
1095             noise_pred.float(), time, sample.float(),
1096             prev_sample=None, noise_level=0.7,
1097         )
1098     return prev_sample, log_prob, prev_sample_mean, std_dev_t
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

```