REINFORCE YOUR LAYOUT: ONLINE REWARD-GUIDED DIFFUSION FOR LAYOUT-TO-IMAGE GENERATION

Ruijie Li **Xiaojun Shan** **Zheng Ding** **Zeyuan Chen** **Zhuowen Tu**
University of California, San Diego

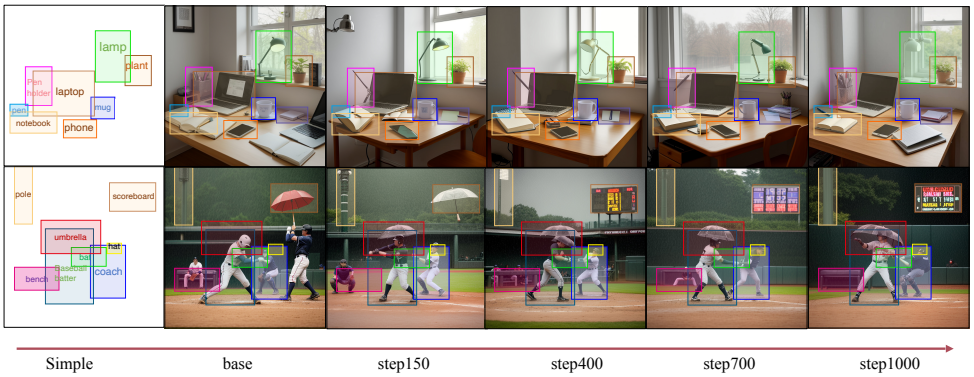

Figure 1: Qualitative visualization of the curriculum-guided reinforcement learning process of ARPO. Each row shows an example where the model starts from a simple layout (left) and progressively refines the generated image as training proceeds (base, step150, step400, step700, step1000). Bounding boxes of different colors denote the target objects. As training advances, object placement becomes increasingly accurate and visually coherent, demonstrating how our method incrementally improves spatial alignment and realism.

ABSTRACT

In this work, we tackle the layout-to-image generation task by proposing a novel online reinforcement learning (RL) framework that directly optimizes diffusion models to achieve consistency between images and layouts. We introduce RLLay, a method that overcomes a major limitation that lies in existing methods with their reliance on indirect side guidance—rather than direct supervision on layout alignment—which constrains these models' ability to accurately position and scale image content. Given a prompt, our approach generates multiple candidate images and ranks them using a reward model based on Intersection-over-Union (IoU) to quantify alignment between predicted and target layouts. To effectively utilize this ranking signal, we introduce a pairwise preference-based optimization strategy that fine-tunes the diffusion model by maximizing the likelihood of higher-ranked samples relative to lower-ranked ones (hard-negatives). Experimental results show that our RL-based fine-tuning significantly improves both spatial layout fidelity and text-image alignment, establishing a promising direction for more controlled and layout-aware image generation. Code is available at https://github.com/nerakoo/RLLay.

1 INTRODUCTION

With recent advances in text-to-image generative models[31, 27, 32, 4, 38], there has been growing interest in layout-to-image generation[19, 35, 10] allowing users can explicitly control the spatial locations [36, 35, 19], and counts [1, 41] of objects within generated images. While existing approaches achieve satisfactory results, they typically require extensive and accurately annotated bounding-box datasets for supervised training, resulting in significant data collection overhead.

Recent developments in reinforcement learning (RL) and preference-based learning, notably within large language models, have inspired new methods for image generation [47, 37]. Preference-based learning approaches such as DPO [30] offer stable training dynamics with lightweight frameworks compared to traditional RL-based methods [20], which often experience instability or suboptimal performance without careful hyperparameter tuning. Nevertheless, current offline preference methods like DPO rely heavily on pre-collected paired data, which is costly to acquire and limits adaptability. Furthermore, the optimization is indirect because preference comparisons often involve externally generated images rather than samples from the model itself, hindering training efficiency and adaptability in dynamic scenarios.

Online preference optimization [9, 2] emerges as an appealing alternative by directly optimizing model outputs in real time. However, transitioning preference-based methods to an online setting is nontrivial: generating paired data simultaneously with the training process yields only limited samples at each step, causing gradient updates to become unstable and negatively affecting image quality. This instability arises because gradients calculated from sparse or poorly differentiated image pairs produce noisy and inconsistent updates. We introduce Advanced Relative Policy Optimization (ARPO), which does not require training any additional reward model, and yields substantially more stable online training by directly optimizing pairwise preferences using explicit (trajectory) log-probabilities.

We summarize our contributions below:

1. We develop a **new pipeline called RLLay for layout-to-image generation** using reinforcement learning. To the best of our knowledge, this problem has received limited prior attention.

2. We introduce a novel preference-driven reinforcement learning algorithm **ARPO**, which achieves stable gains even when reward signals are complex and reliable pairwise data are scarce, advancing beyond existing approaches such as DPO, DSPO, and GRPO.

3. In detailed implementations, we propose an enhanced online preference optimization framework featuring to mitigate training instability under settings with limited online data. We present a hard negative mining strategy, which maximizes the utility of every generated data pair by leveraging the full diversity of online image generation, leading to improved image quality and more robust preference learning.

2 Related Work

2.1 Conditional Image Generation

Diffusion models have emerged as powerful generative frameworks, excelling in synthesizing high-quality images from textual descriptions. Early foundational work by Ho et al. [12] introduced denoising diffusion probabilistic models (DDPM), which inspired subsequent enhancements for text-to-image (T2I) generation such as Stable Diffusion [32], GLIDE [27], and Imagen [33]. These methods typically employ supervised training objectives with deterministic or stochastic denoising schedules, enabling precise yet flexible image generation conditioned on textual prompts.

Generating images consistent with spatial layouts has been explored extensively, employing various architectures and training strategies. Early CNN-based approaches [13, 18] relied on direct regression or adversarial losses to enforce spatial constraints. Transformer-based methods, such as LayoutTransformer [40], exploited attention mechanisms to enhance semantic and spatial coherence. Diffusion-based methods have recently emerged, with CreatiLayout [10] and HiCo [26] effectively leveraging diffusion architectures to improve spatial accuracy and visual realism via bounding box annotations, aiming to produce images that adhere to predefined layouts.. Despite their strengths, these methods largely depend on deterministic training paradigms, limiting their flexibility in explicitly optimizing layout adherence via external reward signals. Our method bridges this gap, combining diffusion models with RL-driven pairwise preference learning to explicitly and effectively enhance spatial fidelity and textual alignment in layout-to-image generation.

2.2 PREFERENCE LEARNING IN GENERATIVE TASKS

Reinforcement Learning (RL) has been effectively utilized in generative modeling to optimize models directly towards user-defined objectives. Early efforts such as GAN-based approaches (SeqGAN [42], RankGAN [21]) and policy-gradient methods [48] demonstrated RL's capability to guide generation via explicit reward signals. Recent advances integrate RL with diffusion models to enhance complex conditioning tasks. Black et al. [2] introduced DDPO, optimizing diffusion models through policy gradients for objectives that are difficult to capture explicitly by standard supervised learning, such as aesthetic quality and alignment with user preferences. Similarly, Fan et al. [9] demonstrated significant improvement in text-to-image alignment by incorporating RL-based fine-tuning with KL-divergence regularization.

Pairwise preference learning has become prominent due to its effectiveness in aligning generative models with subjective human judgments. Christiano et al. [6] and Ouyang et al. [28] initially proposed preference-based methods in reinforcement learning from human feedback (RLHF), popularizing pairwise comparison paradigms. Rafailov et al. [30] introduced Direct Preference Optimization (DPO), enabling training without explicit reward modeling by directly optimizing preference pairs. Recent extensions such as RankDPO [15] generalized this approach to ranking-based preferences, further enhancing generative models' alignment with nuanced human judgments. In the image generation domain, ImageReward [37] utilized human-rated image pairs to train robust reward models, subsequently guiding diffusion models towards outputs aligned closely with human aesthetic preferences. DSPO (Direct Score Preference Optimization) [47] is a novel fine-tuning algorithm for diffusion-based text-to-image models that aligns the pretraining and preference alignment objectives via score matching, enabling human-preference-consistent.

3 METHOD

This section presents **RLLay** (Reinforce Your Layout): an online reinforcement learning fine-tuning method for diffusion-based layout-to-image models that aims to substantially improve *layout consistency* without sacrificing fidelity or semantic coherence. First (§ 3.1), we formalize the task with a text prompt and layout meta-information as inputs, and define a unified *layout reward* R_{layout} as the mean IoU between predicted and target boxes in the generated image. Next (§3.2.1), we introduce a curriculum by layout difficulty, training in a progressive easy-to-hard schedule to stabilize optimization and preserve separable preference signals. Then (§3.2.2), to obtain stronger supervision, we generate multiple candidates per (prompt, layout), rank them by R_{layout}, and construct *extreme* preference pairs to amplify reward gaps and log-probability differences, improving learning efficiency under the same inference budget. Additionally (§A.1), since policy gradients require *differentiable* action probabilities, we extend SD-3's deterministic Flow–ODE to a variance-time SDE and derive a one-step explicit log-probability $\log \pi_\theta$. Building on this (§3.2.3), we optimize a loss that combines *pairwise logistic advantage* with an *IoU-aware KL* regularizer, yielding a learning signal that is both directional and stable.

3.1 PROBLEM FORMULATION

We formulate the task as follows. Given a text prompt $\mathbf{p}$ and a set of layout meta-information

$$\mathcal{M} = \big\{(b_i,\, t_i)\big\}_{i=1}^{K}, \tag{1}$$

where

$$b_i = \big(x_{1,i},\, y_{1,i},\, x_{2,i},\, y_{2,i}\big)$$

is the bounding box of the i-th object specified by its top-left corner $(x_{1,i}, y_{1,i})$ and bottom-right corner $(x_{2,i}, y_{2,i})$. All coordinates are normalized to $[0, 1]$ with the image origin at the top-left $(0, 0)$, the x-axis pointing right, and the y-axis pointing down. The term t_i denotes the textual description of the region (e.g., "red car"). Our goal is to generate an image I that is semantically aligned with $\mathbf{p}$ and spatially consistent with $\mathcal{M}$.

To quantify spatial consistency, we compare predicted bounding boxes in the generated image with the target layout. Let o_i be the object instance in I corresponding to description t_i, and let $\text{Box}_I(o_i)$

denote its predicted bounding box. We define the layout reward as

$$R_{\text{layout}}(I, \mathcal{M}) = \frac{1}{K} \sum_{i=1}^{K} \text{IoU}\big(\text{Box}_I(o_i),\, b_i\big). \tag{2}$$

Here, $\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|}$ denotes the Intersection-over-Union between two boxes—a standard measure of spatial overlap in object detection and localization [8].

3.2 Reinforce Your Layout

3.2.1 Curriculum Learning by Layout Difficulty

Motivation. Online fine-tuning over the full data distribution is unreliable: at early stages, the base model typically fails to satisfy complex layout constraints, so parallel sampling yields candidate pairs of uniformly poor quality with barely separable preference/reward signals, making policy improvement difficult. In addition, the wide distributional span within a single dataset further amplifies training instability. We therefore adopt a progressive curriculum by layout difficulty: the model first consolidates basic spatial alignment on easy layouts, and is then gradually exposed to harder scenes characterized by higher object counts, stronger overlaps, and tighter inter-object relations. This staged schedule stabilizes optimization while preserving useful preference signal as complexity increases.

Difficulty factors. We characterize difficulty using three signals:

- *Box count* K: more boxes increase combinatorial and occlusion complexity.
- *Overlap*: measured by average pairwise IoU or total overlap ratio (higher overlap makes localization harder) [17].
- *Preference bias (co-occurrence overfitting)*: diffusion models tend to overfit frequent co-occurrences in training data (e.g., *baseball player* $\rightarrow$ *bat*), which often overrides rare or counterfactual combinations (e.g., *baseball player with an umbrella*) [34, 3]. This factor captures compositional failures from semantic priors.

Level definitions. Based on these factors, we partition samples into four levels:

- **Easy:** $K \in [1, 3]$, Overlap $< 10\%$, no preferences/relations.
- **Medium:** $K \in [3, 4]$, Overlap $< 20\%$, with preferences.
- **Hard:** $K \in [4, 5]$, Overlap $> 30\%$, with preferences and *containment*.
- **Very Hard:** $K > 5$, Overlap $> 30\%$, with preferences, *containment*, and *over four bounding boxes crossing*.

Scheduling. We train using stage-wise shuffled mixtures of the four difficulty levels; the exact level definitions and per-stage mixing ratios are reported in Sec. 4.1.2.

Outcome. This curriculum first yields stable, separable signals on easy cases, then increases exposure to hard/very-hard layouts so the policy continues to improve under heavy overlap and complex relations without collapsing. The results of the training stage are shown as in Fig 2.

3.2.2 Hard Negative Mining via Extreme Pairing

For each (prompt, meta) pair, we generate six candidate images in parallel. To assess their spatial alignment quality, we compute the layout reward R_{layout} for each image using GroundingDINO [24], based on the alignment between generated content and the provided bounding boxes. Let the resulting scores be sorted in descending order, and denote by $x_{(i)}$ the image with the i-th highest score.

Based on this ranking, we construct a set of four distinct preference pairs:

$$\mathcal{P} = \{(x_{(1)}, x_{(6)}), (x_{(2)}, x_{(6)}), (x_{(1)}, x_{(5)}), (x_{(2)}, x_{(5)})\}, |\mathcal{P}| = 4. \tag{3}$$

where each pair (x^w, x^ℓ) consists of a higher-scoring image x^w (the *winner*) and a lower-scoring image x^ℓ (the *loser*). These pairs serve as supervised preference signals for fine-tuning.

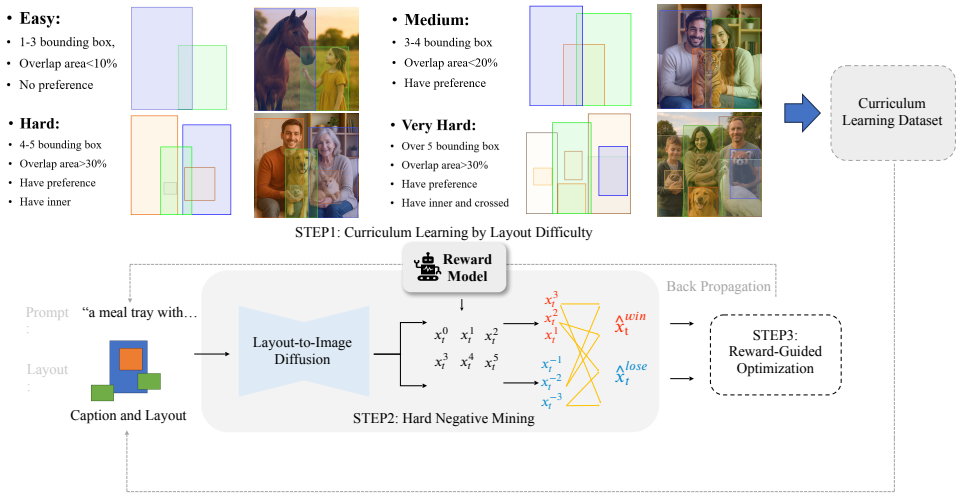

Figure 2: Overall pipeline of our curriculum learning and reward-guided optimization framework. Step 1 groups training samples into four difficulty levels (Easy, Medium, Hard, Very Hard) based on bounding-box number, overlap ratio, and preference conditions. Step 2 performs hard negative mining with a reward model to distinguish high- and low-quality generations. Step 3 applies reward-guided optimization to update the layout-to-image diffusion model, completing a closed training loop.

This extreme pairing strategy provides two key advantages.

First, by forming several preference pairs within each mini-batch, our strategy uses the same inference budget more efficiently; the enlarged candidate pool yields a denser set of preference signals, providing richer and more informative supervision.

Second, extreme pairing amplifies both the layout reward gap and the log-probability difference under the current sampling policy:

$$\Delta \log p = \log p_\theta(x^w) - \log p_\theta(x^\ell),$$

which concentrates gradient updates on image pairs exhibiting the most significant spatial misalignment. Compared to random or arbitrary pairings, this targeted supervision produces stronger preference signals and accelerates convergence during fine-tuning.

3.2.3 Loss Function

To explicitly improve layout alignment without sacrificing fidelity or semantic consistency, we minimize the following preference-guided objective over preference pairs (x_w, x_ℓ) (constructed by the extreme pairing in Sec. 3.2.2):

$$\mathcal{L}_\theta = \mathbb{E}_{(x_w,x_\ell)\sim\mathcal{P}}\Big[-\log\sigma\big(\beta\,\Delta\log p\big) + \lambda\big(1-\sigma(\kappa\,\Delta r)\big)\,\tfrac{1}{2}\sum_{x\in\{x_w,x_\ell\}}\|\delta_\varepsilon(x)\|_2^2\Big]. \tag{4}$$

The first term is a pairwise logistic advantage that pushes the log-probability gap $\Delta\log p$ to increase in favor of the better-aligned sample (thus shifting probability mass toward high-IoU regions). The second term is an IoU-aware KL-style regularizer: when the preference gap Δr is weak (i.e., $\sigma(\kappa\,\Delta r)$ is small), it pulls the current model toward a frozen reference to stabilize training. All symbols are defined *in situ* as

$$\begin{aligned}\Delta\log p &= \log p_\theta(x_w) - \log p_\theta(x_\ell),\\ \Delta r &= r_w - r_\ell,\\ \delta_\varepsilon(x) &= \log p_\theta(x) - \log p_{\text{ref}}(x).\end{aligned} \tag{5}$$

where $r_i = R_{\text{layout}}(x_i, \mathcal{M})$ is the layout reward, $\sigma(z) = 1/(1+e^{-z})$ is the sigmoid, p_θ denotes the sampler-induced distribution (see Sec. A.1), and p_{ref} is a frozen reference model. The scalars $\beta > 0$,

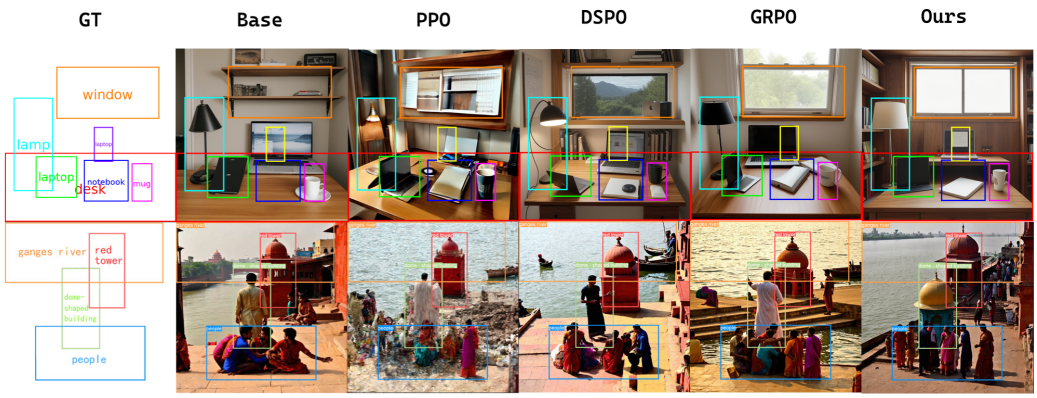

Figure 3: We present visual comparisons across various reinforcement learning-based preference optimization methods for layout-to-image generation, including PPO, DSPO, GRPO, and our proposed approach. More visual presentations will be included in the appendix.

$\kappa > 0$, and $\lambda \geq 0$ control the sharpness of the logistic term, the temperature on the reward gap, and the strength of the regularization, respectively.

This objective can be derived from the constrained problem

$$\max_{\theta} \; \mathbb{E}_{x \sim p_\theta}\big[R_{\text{layout}}(x, \mathcal{M})\big] \quad \text{s.t.} \quad \mathrm{KL}\big(p_\theta \| p_{\text{ref}}\big) \leq \varepsilon. \tag{6}$$

We approximate the expected reward by a weighted pairwise advantage $\mathbb{E}[\sigma(\kappa\,\Delta r)\,\Delta \log p]$, and locally approximate the KL by a second-order expansion $\mathrm{KL}(p_\theta \| p_{\text{ref}}) \approx \frac{1}{2}\,\mathbb{E}_x\big[\|\delta_\varepsilon(x)\|_2^2\big]$ (up to constants), yielding Eq. equation 4.

This loss has two useful properties. First, the derivative of the logistic term with respect to the log-probability gap is strictly negative:

$$\frac{\partial\,\mathcal{L}_\theta}{\partial\,\Delta \log p} = -\,\beta\,\sigma\big(-\,\beta\,\Delta \log p\big) \;<\; 0, \tag{7}$$

so gradient-based updates monotonically increase $\Delta \log p$, raising the relative likelihood of high-IoU samples. Second, when the preference gap is weak ($\Delta r \to 0$), we have $\sigma(\kappa\,\Delta r) \to \frac{1}{2}$ and the regularizer reduces to $\frac{\lambda}{2}\,\mathbb{E}_x[\|\delta_\varepsilon(x)\|_2^2]$, which suppresses semantic drift and preserves image quality under noisy or uninformative comparisons.

Overall, combining the pairwise logistic advantage with the IoU-aware regularizer provides a learning signal that is both directional and stable; in our experiments (Sec. 4.1.2), this objective improves expected IoU without degrading fidelity or semantic consistency.

4 Experiments

4.1 Experiment Setup

4.1.1 Baselines

To explore the effectiveness and generalizability of our approach across different diffusion backbones, we implement and fine-tune two model variants. Specifically, **HiCoNet-Layout** [5] is built upon Stable Diffusion 1.5, a widely adopted open-source architecture known for its balance between generation quality and computational efficiency. In contrast, **CreatiLayout** [44] leverages Stable Diffusion 3, a more recent and powerful backbone offering enhanced semantic understanding and generation fidelity. This dual-backbone setup allows us to evaluate how our layout generation method adapts to varying model capacities and architectural changes, and to verify whether the improvements introduced by our approach consistently hold across diffusion model versions.

Table 1: Curriculum schedule of difficulty ratios for CreatiLayout and HicoNet across training iterations. **Detailed training curves and additional analyses are provided in the appendix.**

Method	Iterations	Easy	Medium	Hard	Very Hard
CreatiLayout [44]	1–150	70%	30%	0%	0%
	150–400	30%	60%	10%	0%
	400–700	20%	45%	30%	5%
	700–1000	10%	20%	40%	30%
HicoNet [5]	1–150	50%	50%	0%	0%
	150–400	20%	50%	30%	0%
	400–700	10%	40%	40%	10%
	700–1000	0%	20%	40%	40%

4.1.2 DATASET SCHEDULING

We adopt a *stage-wise mixed curriculum* covering four difficulty levels (Easy / Medium / Hard / Very Hard). We first construct four pools from *real* layouts (7,000 samples each). At the beginning of each training stage, we resample with replacement according to the preset ratios, shuffle, and form a 10,000-sample subset for that stage; within the stage, this subset is reshuffled every epoch. At the next stage, a new 10,000-sample subset is re-sampled under the updated ratios. The concrete categorization and schedule are provided in Table 1.

4.1.3 TRAINING DETAILS

Unlike traditional supervised fine-tuning, **RLLay** does not rely on a pre-fixed corpus of real image–layout pairs. To keep the training distribution close to real scenes, we *do not synthesize* bounding boxes; instead, we *select real layouts* directly from the annotations of LAYOUTSAM and COCO 2017 [22]. According to our difficulty criteria (object count, overlap, and preferences/relations), we construct four difficulty pools—*Easy / Medium / Hard / Very Hard*—each containing **7,000** layout samples (preserving the original categories and text descriptions). At each stage of the curriculum, we resample and mix from the four pools according to the stage-specific ratios to form a **10,000**-sample training subset for that stage.

Evaluation protocol. We evaluate on two dataset families: the fine-grained, open-set LAYOUTSAM and the coarse-grained, closed-set COCO 2017. **LayoutSAM-Eval:** following [44], we evaluate on LAYOUTSAM-EVAL, which contains **5,000** prompts with metadata (category and layout annotations). **COCO-Eval:** following prior work [39, 45], we filter out images with too small effective area and too few bounding boxes from the COCO validation set, resulting in **4,652** validation images.

Metrics. To assess *layout compliance*, we use QWEN2.5-7B-INSTRUCT [14] to detect objects in generated images and match them with the ground-truth boxes, computing standard **IoU**, **AP**, **AP50**, and **AR**. To measure *overall generation quality*, we report **FID** [11] and **CLIP-Score** [29]. We also report a **VQA** score (based on a visual question answering model that answers prompt-related questions) to assess the *semantic accuracy* of the generations.

Training setup. We use the ADAMW optimizer [25] with distributed training on **8× NVIDIA A6000** GPUs; the local batch size is **32 preference pairs per GPU**, and training runs for **1000 Iterations**. Resolutions are model-specific: HICONET is trained at 512×512, whereas CREATILAYOUT is trained at 1024×1024.

4.2 RESULTS

4.2.1 EVALUATION ON LAYOUTSAM-EVAL

Table 2 reports quantitative results on the LayoutSAM-Eval [44] benchmark. Our reinforcement learning framework RLLay, consistently improves both spatial and visual quality metrics across two strong baselines. When applied to HiCO, RLLay yields clear gains on all spatial measures—IoU, AP, AP50, and AR—while further lowering FID and enhancing semantic accuracy (VQA). A similar

Table 2: Overall performance of RLLay (ours) compared to existing state-of-the-art approaches on the LayoutSAM-Eval [44] benchmark.Bold and underline denote the best and second best methods, respectively.

Method	Spatial				Quality		Accuracy
	IoU↑	**AP**↑	**AP50**↑	**AR**↑	**FID**↓	CLIP-Score(Global)↑	VQA↑
GLIGEN [19]	60.09	18.34	37.18	29.77	25.93	24.61	79.82
3DIS-FLUX [46]	69.42	36.90	<u>54.96</u>	39.15	19.47	27.62	85.35
Instance Diffusion [35]	<u>77.03</u>	<u>37.16</u>	46.55	<u>40.10</u>	23.16	27.70	<u>91.47</u>
HiCO-baseline	72.05	29.42	41.55	33.50	18.64	**29.52**	91.45
+ RLLay (ours)	**80.12**(+8.07↑)	**38.40**(+8.98↑)	47.75(+6.20↑)	**41.30**(+7.80↑)	17.12(-1.52↓)	<u>29.03</u>(-0.49↓)	**93.18**(+1.73↑)
CreatiLayout-baseline	68.01	29.55	50.17	33.20	<u>15.26</u>	27.89	82.26
+ RLLay (ours)	74.97(+6.96↑)	33.15(+3.60↑)	**55.37**(+5.20↑)	35.60(+2.40↑)	**13.27**(-1.99↓)	27.31(-0.58↓)	86.08(+3.82↑)

Table 3: Overall performance of RLLay compared to existing state-of-the-art approaches on the COCO [22] benchmark.Bold and underline denote the best and second best methods, respectively.

Method	Spatial				Quality		Accuracy
	IoU↑	**AP**↑	**AP50**↑	**AR**↑	**FID**↓	CLIP-Score(Global)↑	VQA↑
GLIGEN [19]	51.60	19.48	32.10	30.68	27.94	21.63	68.47
3DIS-FLUX [46]	69.33	25.83	39.70	20.17	27.74	24.54	72.36
Eligen [43]	71.06	32.66	<u>47.83</u>	40.25	25.84	**28.31**	<u>81.75</u>
Instance Diffusion	**84.40**	**38.91**	47.60	<u>43.10</u>	25.66	25.65	79.49
HiCO-baseline	70.57	22.12	33.26	29.00	26.05	<u>27.17</u>	75.12
+ RLLay (ours)	<u>78.29</u> (+7.72↑)	<u>37.33</u> (+15.21↑)	**48.19** (+14.93↑)	**44.30** (+15.30↑)	<u>25.03</u>(-1.02↓)	26.98(-0.19↓)	**81.98** (+6.86↑)
CreatiLayout-baseline	56.59	12.09	26.08	17.60	26.18	26.19	69.74
+ RLLay (ours)	62.58 (+5.99↑)	15.86 (+3.77↑)	32.19 (+6.11↑)	18.30 (+0.70↑)	**21.09**(-5.09↓)	25.14(-1.05↓)	75.54 (+5.80↑)

trend is observed on CreatiLayout, where RLLay substantially boosts localization accuracy (IoU and AP50) and reduces FID, confirming that the generated layouts and final images are both more precise and more visually faithful. Compared with diffusion-based methods such as GLIGEN and Instance Diffusion, RLLay-enhanced models not only maintain competitive generative quality but also deliver markedly better alignment between textual descriptions and visual layouts, as reflected in higher CLIP-Score and VQA. These results demonstrate that the proposed reinforcement-guided optimization effectively strengthens spatial reasoning while preserving high-fidelity image synthesis.

4.2.2 Evaluation on COCO

Table 3 presents a comprehensive comparison of RLLay with representative state-of-the-art layout-to-image generation approaches on the COCO [22] benchmark. Our method consistently achieves superior spatial alignment, as reflected by higher IoU, AP, AP50, and AR scores, while maintaining strong perceptual quality (lower FID and higher CLIP-Score) and semantic accuracy (higher VQA). Compared with diffusion-based baselines such as GLIGEN and Instance Diffusion, RLLay delivers more precise object localization and competitive visual fidelity. Relative to reinforcement-learning-enhanced methods (e.g., HiCO + RLLay), our approach further improves detection metrics and semantic faithfulness. In particular, integrating our reward-guided curriculum with CreatiLayout (CreatiLayout + RLLay) yields notable gains across all three categories of metrics, confirming the effectiveness of our design in balancing spatial control and generative quality. Noted that CreatiLayout does not release the version after finetuning on the COCO dataset, leading to lower performance than the numbers reported in its original paper.

4.2.3 Evaluation on OverLayBench-Hard

To rigorously assess model robustness in challenging scenarios, we evaluate on the Hard split of OverLayBench, which features dense multi-object layouts with significant bounding box overlap and high semantic similarity. In contrast to prior benchmarks like COCO and LayoutSAM that are dominated by simpler, minimally overlapping layouts, OverLayBench-Hard provides a focused testbed for spatial and relational reasoning, with the highest OverLayScore[17] cases reflecting severe entanglement and ambiguous instance boundaries.

Table 4: Comprehensive comparison between training-based methods on OverLayBench-Hard.

Method	mIoU(%) ↑	CLIP ↑	FID ↓
GLIGEN	47.20	33.92	31.27
3DIS-FLUX	50.65	35.11	29.18
Instance Diffusion	53.68	32.33	36.17
Eligen	52.53	36.18	28.87
HiCO-baseline	46.56	33.15	29.21
+ RLLay (ours)	**60.12** (+13.56↑)	33.01	28.71
CreatiLayout-baseline	44.24	36.55	27.51
+ RLLay (ours)	55.01 (+10.77↑)	36.39	26.74

Table 5: Ablation study on the COCO Benchmark for **CreatiLayout** and **HicoNet**. Metrics include AP, AP50, AR, FID, CLIP-Score(Global), and IoU.

Method	AP ↑	AP50 ↑	AR ↑	FID ↓	CLIP-Score(Global) ↑	IoU ↑
CreatiLayout	12.09	26.08	17.60	26.18	26.19	56.59
+ DPO	12.87	27.35	18.20	26.01	25.79	58.35
+ DSPO	15.37	30.93	18.20	21.96	25.76	60.46
+ GRPO	15.72	31.72	18.30	26.25	22.63	61.33
+ ARPO	**15.86**	**32.19**	**18.30**	**21.09**	25.14	**62.58**
HicoNet	22.12	33.26	29.00	26.05	**27.17**	70.57
+ DSPO	29.59	41.24	35.80	25.23	26.85	75.85
+ GRPO	29.45	40.60	35.70	29.35	25.23	74.95
+ ARPO	**37.33**	**48.19**	**44.30**	**25.03**	26.80	**78.29**

Table 4 shows that although all models struggle on OverLayBench-Hard, adding RLLay to HiCO and CreatiLayout yields clear gains (+13.56 and +10.77 mIoU), while maintaining competitive FID and CLIP scores.

4.2.4 Ablation Study

Table 3 reports the quantitative comparison on the COCO benchmark for CreatiLayout and HicoNet. The proposed reinforcement learning method (ARPO) consistently improves both spatial and visual quality metrics over all baselines. For CreatiLayout, ARPO achieves the best spatial alignment, raising AP and AP50 to 15.86 and 32.19 while increasing IoU to 62.58, and simultaneously lowering FID to 21.09, indicating sharper and more faithful image synthesis. A similar trend is observed for HicoNet, where ARPO delivers substantial gains across detection-oriented metrics (AP from 22.12 to 37.33, AP50 from 33.26 to 48.19, AR from 29.00 to 44.30) while also reducing FID and maintaining competitive CLIP-Score. Compared with other reinforcement-preference optimization approaches such as DPO, DSPO, and GRPO, our method provides a more balanced improvement on both spatial precision and visual fidelity, confirming its effectiveness in aligning generated layouts and images with the intended semantics

5 Conclusion

In this work, we presented **RLLay**, a reinforcement-learning-driven pipeline for preference-based optimization of layout-to-image diffusion models. Unlike conventional methods that rely heavily on paired layout–image datasets, **RLLay** leverages pairwise human or model feedback to guide policy updates, significantly reducing data collection requirements while improving generative alignment. Built upon our novel reinforcement learning algorithm **ARPO**, the framework integrates curriculum scheduling, hard negative mining, and preference-guided reward modeling to achieve stable and sample-efficient optimization. Extensive experiments on COCO and LayoutSAM-Eval benchmarks demonstrate consistent improvements in both spatial accuracy and perceptual quality over strong diffusion and preference-optimization baselines, including DPO, DSPO, and GRPO. These results highlight the effectiveness and generality of our approach, opening avenues for broader applications of reinforcement preference optimization in complex generative modeling tasks.

Acknowledgment

This work is supported by NSF award IIS-2127544 and NSF award IIS-2433768.

REFERENCES

[1] Lital Binyamin, Yoad Tewel, Hilit Segev, Eran Hirsch, Royi Rassin, and Gal Chechik. Make it count: Text-to-image generation with an accurate number of objects. *arXiv preprint arXiv:2406.10210*, 2024.

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

[3] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, and Eric Wallace. Extracting training data from diffusion models. In *Proceedings of the 32nd USENIX Security Symposium (USENIX Security 2023)*, pp. 5253–5270. USENIX Association, 2023.

[4] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

[5] Bo Cheng, Yuhang Ma, Liebucha Wu, Shanyuan Liu, Ao Ma, Xiaoyu Wu, Dawei Leng, and Yuhui Yin. Hico: Hierarchical controllable diffusion model for layout-to-image generation. *arXiv preprint arXiv:2410.14324*, 2024.

[6] Paul F Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, 2017.

[7] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024. URL `https://arxiv.org/abs/2403.03206`.

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010.

[9] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *arXiv preprint arXiv:2305.16381*, 2023.

[10] Ziqi Guo, Baoxiong Wu, Yixin Zhu, et al. Creatilayout: Enhanced layout-to-image generation with compositional diffusion. In *CVPR*, 2024.

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[13] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *CVPR*, 2018.

[14] B. Hui, J. Yang, Z. Cui, J. Yang, D. Liu, L. Zhang, others, and J. Lin. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

[15] S. Karthik, H. Coskun, Z. Akata, S. Tulyakov, J. Ren, and A. Kag. Scalable ranked preference optimization for text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18399–18410, 2025.

[16] H. Lamba, J. C. Mattingly, and A. M. Stuart. An adaptive euler–maruyama scheme for sdes: convergence and stability. *IMA Journal of Numerical Analysis*, 27(3):479–506, 07 2007. doi: 10.1093/imanum/drl032. URL `https://academic.oup.com/imajna/article-abstract/27/3/479/744803`.

[17] Bingnan Li, Chen-Yu Wang, Haiyang Xu, Xiang Zhang, Ethan Armand, Divyansh Srivastava, Xiaojun Shan, Zeyuan Chen, Jianwen Xie, and Zhuowen Tu. Overlaybench: A benchmark for layout-to-image generation with dense overlaps. *arXiv preprint arXiv:2509.19282*, September 2025. doi: 10.48550/arXiv.2509.19282. URL https://arxiv.org/abs/2509.19282. Accepted to NeurIPS 2025 Datasets & Benchmarks Track.

[18] Jiang Li, Dingdong Yang, Jianchao Wang, Zhenguo Liu, and Yizhou Wu. Layoutgan: Generating graphic layouts with wireframe discriminators. In *ICLR*, 2019.

[19] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22511–22521, June 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Li_GLIGEN_Open-Set_Grounded_Text-to-Image_Generation_CVPR_2023_paper.html.

[20] Zihao Li, Zhuoran Yang, and Mengdi Wang. Reinforcement learning with human feedback: Learning dynamic choices via pessimism, 2023. URL https://arxiv.org/abs/2305.18438.

[21] Kevin Lin, Dian Li, Xiaodong He, and Zhengyou Zhang. Adversarial ranking for language generation. In *NIPS*, 2017.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[23] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. URL https://arxiv.org/abs/2210.02747.

[24] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Computer Vision – ECCV 2024: Proceedings, Part XLVII*, volume 15105 of *Lecture Notes in Computer Science*, pp. 38–55, Cham, 2024. Springer Nature Switzerland. doi: 10.1007/978-3-031-72970-6_3. URL https://doi.org/10.1007/978-3-031-72970-6_3.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. URL https://arxiv.org/abs/1711.05101.

[26] Yuhang Ma, Shanyuan Liu, Ao Ma, Xiaoyu Wu, Dawei Leng, and Yuhui Yin. Hico: Hierarchical controllable diffusion model for layout-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 128886–128910, 2024.

[27] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

[28] Long Ouyang, Jeff Wu, Xu Jiang, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[30] Rafael Rafailov, Tianyi Zhang, Yian Fu, et al. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

[31] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *ICML*, 2022.

[34] Kaiyue Sun, Ke Huang, Xinyu Liu, Y. Wu, Z. Xu, Z. Li, and Xiaoming Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8406–8416, 2025.

[35] Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6232–6242, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wang_InstanceDiffusion_Instance-level_Control_for_Image_Generation_CVPR_2024_paper.html.

[36] Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7452–7461, 2023.

[37] 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. In *NeurIPS*, 2023.

[38] Zeyue Xue, Guanglu Song, Qiushan Guo, Boxiao Liu, Zhuofan Zong, Yu Liu, and Ping Luo. Raphael: Text-to-image generation via large mixture of diffusion paths. *Advances in Neural Information Processing Systems*, 36, 2024.

[39] Binbin Yang, Yi Luo, Ziliang Chen, Guangrun Wang, Xiaodan Liang, and Liang Lin. Layoutdiffusion: Complex scene generation by diffusion with layouts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22669–22679, October 2023.

[40] Shitao Yang, Xuan Liu, Qinxian Huang, et al. Layouttransformer: Scene layout generation with multimodal transformers. In *CVPR*, 2021.

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

[42] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017.

[43] Hao Zhang, Zhi Duan, Xiaoyu Wang, Yifan Chen, and Yulin Zhang. Eligen: Entity-level controlled image generation with regional attention. *arXiv preprint arXiv:2501.01097*, 2025.

[44] Hui Zhang, Dexiang Hong, Tingwei Gao, Yitong Wang, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-Gang Jiang. Creatilayout: Siamese multimodal diffusion transformer for creative layout-to-image generation. *arXiv preprint arXiv:2412.03859*, 2024. doi: 10.48550/arXiv.2412.03859. URL https://arxiv.org/abs/2412.03859. v2 (2025-03-14).

[45] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22490–22499, June 2023.

[46] Dewei Zhou, Ji Xie, Zongxin Yang, and Yi Yang. 3dis-flux: simple and efficient multi-instance generation with dit rendering. *arXiv preprint arXiv:2501.05131*, 2025.

[47] Huaisheng Zhu, Teng Xiao, and Vasant G. Honavar. Dspo: Direct score preference optimization for diffusion model alignment. In *International Conference on Learning Representations (ICLR)*, 2025. URL `https://github.com/huaishengzhu/DSPO`.

[48] Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.

A APPENDIX

A.1 STOCHASTIC SD-3 SAMPLER WITH EXPLICIT LOG-PROB

SD-3 [7] uses a deterministic Flow–ODE scheduler [23] that converges to a unique solution for any fixed noise vector. To introduce stochasticity *while approximately preserving the sampling marginals* and to obtain *differentiable* action probabilities for RL, we extend the ODE

$$\mathrm{d}x_t = v_\theta(x_t, t)\,\mathrm{d}t$$

to an SDE under a *variance-time* parameterization:

$$\mathrm{d}x_t = v_\theta(x_t, t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}W_t,$$

where $x_t \in \mathbb{R}^D$, v_θ is the network-predicted velocity field, $\sigma(t) = \sqrt{1/\,\mathrm{SNR}(t)}$ follows the SD-3 log-SNR schedule, and W_t is a standard Wiener process.

Define the cumulative variance

$$\Sigma(t) = \int_0^t \sigma^2(s)\,\mathrm{d}s,$$

and let $0 = t_0 < \cdots < t_N = 1$ be a partition of the time horizon with $\Sigma_i = \Sigma(t_i)$, $\Delta\Sigma_i = \Sigma_{i+1} - \Sigma_i$, and $\sigma_i = \sigma(t_i)$. A single Euler–Maruyama step [16] over $[\Sigma_i, \Sigma_{i+1}]$ yields

$$x_{i+1} \;=\; x_i + \frac{\Delta\Sigma_i}{\sigma_i^2}\, v_\theta(x_i, t_i) + \sqrt{\Delta\Sigma_i}\,\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I). \tag{8}$$

Because $\Sigma(t)$ is strictly increasing, reparameterizing time in Σ changes step sizes but preserves the *Gaussian form* of the one-step transition, enabling controlled noise injection without redesigning the sampler.

To strengthen stochasticity, we scale the noise magnitude to $\hat{\sigma}_i = (1+\gamma)\sigma_i$ ($\gamma \geq 0$), which implies

$$\hat{\Delta\Sigma}_i \;=\; (1+\gamma)^2\,\Delta\Sigma_i. \tag{9}$$

In Σ-time this can be viewed as adjusting the step size while keeping the transition form unchanged. For small γ, the induced marginal shift is minor in practice (see Appendix and ablations); as $\gamma \to 0$, the sampler reduces to the original deterministic Flow–ODE.

The one-step transition admits a closed-form log-probability:

$$\log \pi_\theta(a_i = x_{i+1} \mid s_i = x_i) = -\frac{\|x_{i+1} - \mu_i\|^2}{2\Delta\Sigma_i} - \frac{1}{2}\log\bigl(2\pi\Delta\Sigma_i\bigr), \mu_i \quad = x_i + \frac{\Delta\Sigma_i}{\sigma_i^2}\, v_\theta(x_i, t_i). \tag{10}$$

Differentiability flows through μ_i (and thus v_θ), providing well-behaved action-probability estimates for preference-based and policy-gradient updates. When noise is disabled ($\Delta\Sigma_i = 0$), the variance term vanishes and the sampler collapses to the deterministic Flow–ODE. For numerical stability we use $\Delta\Sigma_i + \varepsilon$ (e.g., 10^{-6}) in denominators and logarithms.

A.2 ADDITIONAL NOTES ON THE RANDOM SD-3 SAMPLER

What "approximately preserving marginals" means. Extending the Flow–ODE to a variance-time SDE does not *strictly* guarantee identical marginals to the original ODE. "Approximately preserving" indicates that: (i) as the noise scaling $\gamma \to 0$ and the step size $\Delta\Sigma_i \to 0$, the local Markov kernel of the SDE matches the probability flow of the ODE to first order, so the shift in the generated distribution is controlled; (ii) with finite steps and a small γ, empirical changes in image quality (e.g., FID/CLIP) and alignment (e.g., IoU/AP) remain acceptable. Increasing γ raises stochasticity and exploration but also enlarges marginal drift and quality variance.

Why variance-time reparameterization. In Σ-time the one-step transition is explicitly Gaussian:

$$x_{i+1} \sim \mathcal{N}\bigl(\mu_i,\ \Delta\Sigma_i\, I\bigr), \qquad \mu_i \;=\; x_i + \frac{\Delta\Sigma_i}{\sigma_i^2}\, v_\theta(x_i, t_i),$$

which yields two practical benefits: (i) a closed-form one-step log-probability $\log \pi_\theta(x_{i+1} \mid x_i)$ that backpropagates through v_θ; (ii) fine control of the exploration strength by tuning $\Delta\Sigma_i$ and γ without changing the Gaussian form.

Accumulating trajectory log-probabilities. While the main text gives the one-step log-probability, preference/policy learning uses the *trajectory* log-probability via stepwise accumulation:

$$\log p_\theta(x_{0:N}) \;=\; \sum_{i=0}^{N-1} \log \pi_\theta(x_{i+1} \mid x_i) \;+\; \log p(x_0),$$

where $\log p(x_0)$ is an initialization prior (constant or negligible). The preference gap $\Delta \log p$ and the reference difference $\log p_\theta - \log p_{\mathrm{ref}}$ are accumulated in the same way, which is numerically more robust than a single-step proxy.

Choosing and scheduling γ. *Fixed small values:* $\gamma \in [0.05, 0.20]$ often strikes a good balance between having differentiable action probabilities and limiting distributional drift. *Piecewise/annealed:* use larger γ early for exploration/separability and reduce it later to recover quality. *Step-adaptive:* employ smaller γ at larger $\Delta\Sigma_i$ to avoid excessive variance accumulation.

Numerical stability and implementation tips. Use $(\Delta\Sigma_i + \varepsilon)$ (e.g., $\varepsilon = 10^{-6}$) in denominators and logarithms to suppress blow-ups; compute $\|x_{i+1} - \mu_i\|^2$ in at least `float32` and optionally normalize by feature dimension. Ensure that sampling and $\log \pi_\theta$ share the same random seed and schedule to avoid "mismatch" bias. For long trajectories, gradient clipping and an EMA teacher help stabilize training.

Relationship to probability-flow ODE. The probability-flow ODE (PF-ODE) gives deterministic trajectories in the zero-diffusion limit. With small γ and small steps, the SDE's drift (mean update) aligns with PF-ODE locally, while the added diffusion supplies a stochastic policy required by RL. "Preserving marginals" here should be understood as a *local approximation* and *empirically controlled* behavior, rather than an exact equivalence.

Compatibility with guidance and conditioning. *Classifier-free guidance (CFG)* remains applicable; v_θ in μ_i simply includes the guided drift. *Text/layout conditioning* only enters through $v_\theta(x_i, t_i; \mathrm{cond})$, to which the SDE design is agnostic. *Reference model* log-probabilities should reuse the same discretization and noise draws to reduce variance and enable effective differencing.

Discretization and number of steps. A larger number of steps N approaches the continuous limit but increases compute linearly; in practice N is set equal to (or slightly larger than) the original SD-3 inference steps. Non-uniform Σ-grids are recommended: densify segments where the log-SNR schedule is most sensitive, which improves the stability of $\log \pi_\theta$ estimation.

A.3 Curriculum Learning Ablation Results

We evaluate curriculum learning via a controlled ablation on two backbones (HicoNet and CreatiLayout). A held-out validation set of 120 layouts is constructed with stratified difficulty (30 Easy / 30 Medium / 30 Hard / 30 Very Hard), none of which are used for training. During RL fine-tuning, we evaluate every 10 epochs on this set and report ΔIoU relative to the initial model. All settings are identical across runs—data pools, backbone, ARPO, SDE steps, candidate sampling K, hard-negative mining (pair$=4$), learning rate, and all hyperparameters—except for the difficulty schedule: the baseline uses a fixed $25\%/25\%/25\%/25\%$ mixture, while our method uses the progressive stage-wise schedule in Table 1.

In Fig. 4 (HicoNet), the *with curriculum* curve departs from the baseline around 250–300 epochs and keeps increasing steadily; between 500–700 epochs it maintains an advantage of roughly $+0.015$–0.02 ΔIoU, and reaches a final value of ≈ 0.082 versus ≈ 0.051 for the baseline (an absolute gain of $\approx +0.031$). The curriculum variant crosses intermediate thresholds earlier—for example, $\Delta\mathrm{IoU} = 0.04$ is reached at approximately 680 epochs versus 780 for the baseline. The trajectory is monotonic without noticeable regressions.

In Fig. 4 (CreatiLayout), the two curves remain close through the early–mid phase (approximately 0–700 epochs). The *with curriculum* curve then exhibits two clear ramps near ≈ 820 and ≈ 880–900 epochs and establishes a stable lead, converging at ≈ 0.072 versus ≈ 0.047 for the baseline (an

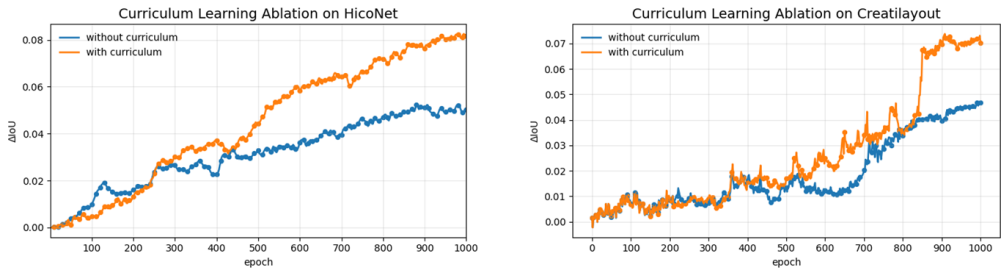

Figure 4: Curriculum learning ablation on the held-out validation set. We compare our progressive schedule with a fixed uniform difficulty mixture on both HicoNet and CreatiLayout backbones.

absolute gain of $\approx +0.025$). It also crosses $\Delta\text{IoU} = 0.02$ earlier (around 600 vs. 740 epochs). The overall training remains monotonic and stable.

Taken together, across both backbones curriculum learning consistently achieves earlier threshold crossing and a higher terminal plateau, under perfectly matched evaluation settings (same validation set, detector, and inference parameters). These curves provide direct evidence for the effectiveness of the progressive scheduling strategy on complex layout learning.

A.4 Curriculum Difficulty and Scheduling.

To instantiate curriculum learning, we build on the difficulty taxonomy already introduced in the main paper, where each layout is assigned to one of four levels (Easy, Medium, Hard, Very Hard) according to its object count, overlaps, and structural relations. In the appendix, we focus on how these difficulty labels are used to determine the stage-wise sampling ratios.

As a first step, we perform a difficulty-stratified evaluation of the two base models, HicoNet and CreatiLayout. On a disjoint validation subset, we group layouts by difficulty level and run each base model under a fixed inference configuration (same sampling steps and classifier-free guidance). For each difficulty, we compute IoU-based metrics with Qwen2.5-7B-Instruct against the target layouts. Both models exhibit the expected monotonic degradation as difficulty increases, but the drop for CreatiLayout is markedly more pronounced on Hard and Very Hard layouts: in these challenging cases, the generated images often contain severe misalignment and occlusion errors, which in turn yield highly noisy reward signals. This suggests that directly training RL on the full mixture of difficulties from the beginning would make it difficult for the CreatiLayout backbone, in particular, to reliably improve on the hardest layouts.

Guided by these observations, we adopt a simple "easy-to-hard" curriculum: early training stages are biased toward easier layouts, while harder layouts are gradually introduced with increasing weight. Concretely, we partition training into several stages and specify the per-stage difficulty ratios in Table 1. In the initial stage, the mini-batches are dominated by Easy layouts, with Medium layouts providing additional diversity. In intermediate stages, the proportion of Hard layouts is progressively increased, and in the final stage both Hard and Very Hard layouts receive substantially higher sampling weights so that difficult cases dominate the later phases of training. For fairness, the same curriculum schedule is used for both backbones; we do not tune separate schedules per model.

This design allows the base models, and especially the weaker CreatiLayout backbone on hard cases, to first strengthen their performance on simpler layouts before being exposed to the most challenging ones. As shown by the ablation results in Appendix A.2, this difficulty-aware curriculum scheduling consistently improves both convergence speed and final layout fidelity under our RL training framework.

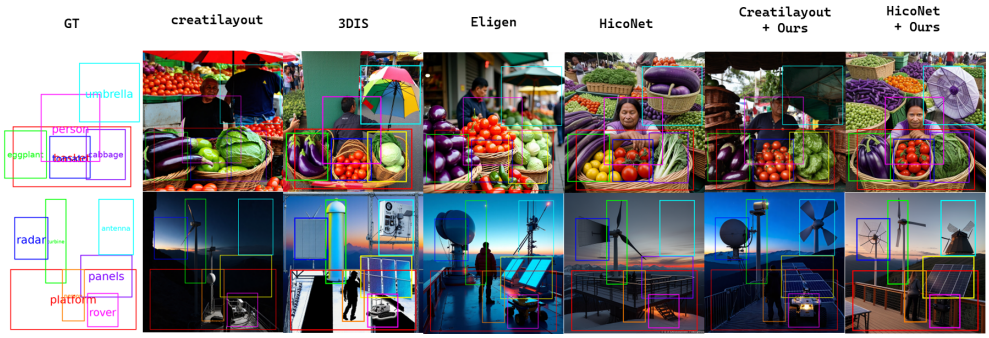

Figure 5: Additional qualitative comparison between baselines and our finetuned models on a shared set of layouts.

A.5 Additional qualitative comparisons between finetuned models and baselines

In this section, we present additional qualitative comparisons on a shared set of layouts and text prompts across a diverse set of layout-to-image models. For each selected layout, we generate images with Eligen, HicoNet, 3DIS, and CreatiLayout, and additionally include the outputs of our RLLay-finetuned HicoNet and RLLay-finetuned CreatiLayout. All methods are conditioned on exactly the same layout and prompt, and the resulting images are visualized side-by-side to enable direct comparison. Across multiple examples, the RLLay-finetuned models tend to produce images that more consistently respect the specified object positions, occlusion patterns, and relative sizes, particularly in crowded or strongly constrained layouts, with fewer severe overlaps or missing instances than the base models and other baselines. These qualitative results complement the quantitative metrics reported in the main paper and provide a more intuitive view of the practical gains achieved by our approach in controllable, layout-consistent image synthesis.