Cycle-Consistent Learning for Joint Layout-to-Image Generation and Object Detection

Xinhao Cai[1]*, Qiuxia Lai[2]*, Gensheng Pei[1], Xiangbo Shu[1], Yazhou Yao[1,3]†, Wenguan Wang[4,5]†

[1] Nanjing University of Science and Technology [2] Communication University of China

[3] State Key Laboratory of Intelligent Manufacturing of Advanced Construction Machinery [4] Zhejiang University

[5] National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, Xi'an Jiaotong University

https://github.com/NUST-Machine-Intelligence-Laboratory/GDCC

Abstract

In this paper, we propose a ***generation-detection cycle consistent*** *(*GDCC*) learning framework that jointly optimizes both layout-to-image (L2I) generation and object detection (OD) tasks in an end-to-end manner. The key of* GDCC *lies in the inherent duality between the two tasks, where L2I takes all object boxes and labels as input conditions to generate images, and OD maps images back to these layout conditions. Specifically, in* GDCC*, L2I generation is guided by a layout translation cycle loss, ensuring that the layouts used to generate images align with those predicted from the synthesized images. Similarly, OD benefits from an image translation cycle loss, which enforces consistency between the synthesized images fed into the detector and those generated from predicted layouts. While current L2I and OD tasks benefit from large-scale annotated layout-image pairs, our* GDCC *enables more efficient use of auto-synthesized data, thereby further enhancing data efficiency. It is worth noting that our* GDCC *framework is computationally efficient thanks to the perturbative single-step sampling strategy and a priority timestep re-sampling strategy during training. Besides,* GDCC *preserves the architectures of L2I, OD models, and the generation pipeline within the framework, thus maintaining the original inference speed. Extensive experiments demonstrate that* GDCC *significantly improves the controllability of diffusion models and the accuracy of object detectors.*

1. Introduction

Recent advancements in both layout-to-image (L2I) generation [36] and object detection (OD) [20] tasks have achieved remarkable success, largely driven by the availability of large-scale annotated datasets. Specifically, L2I generation methods incorporate image-based [33, 36, 79] or prompt-based [6, 77] conditional controls into text-to-image (T2I) diffusion models [52] to achieve more precise control over the instance placement during image synthesis. These methods train diffusion models to generate realistic images from structured layouts, which include bounding boxes and object class labels that define the spatial positioning and types of objects in the scene. On the other hand, OD takes an image as input and identifies the objects within it by predicting their bounding boxes and class labels. Current advancements have led to significant improvements in the precision of instance placement for L2I generation and the prediction accuracy of OD models.

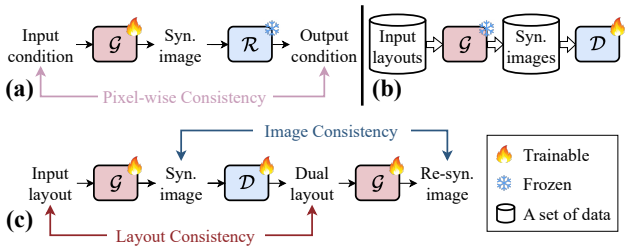

Figure 1. **Overall comparison. (a)** Some works such as [33] use a pre-trained discriminative reward model $\mathcal{R}$ to fine-tune the L2I generator $\mathcal{G}$. **(b)** Some [6, 70] show that the synthesized images provided by a pre-trained $\mathcal{G}$ can improve the performance of the object detector $\mathcal{D}$. **(c)** GDCC enables mutual enhancement between $\mathcal{G}$ and $\mathcal{D}$ through cycle-consistent learning. See §1.

Although both L2I generation and OD have been extensively studied, few have noticed the strong correlation between these two tasks, *i.e.*, they can be viewed as inverse tasks of each other, where L2I maps layouts to images and OD maps images to layouts. This natural duality between these two tasks has largely been overlooked in previous research. Our key finding is that such duality can be effectively leveraged to improve the performance of both tasks. Specifically, if we map an image to its corresponding layout using an OD model, and then map that layout back to an image using an L2I model, we should ideally recover the origi-

*Equal contribution.
†Corresponding author.

nal image. Similarly, mapping a layout to an image and then mapping that image back should yield the original layout. This *cycle consistency* not only enforces tighter alignment between two tasks but also provides a natural regularization that enhances the learning processes of both tasks. Moreover, the cycle consistency allows for the use of synthetic data, opening up possibilities for improving data efficiency.

Based on the above insight, in this paper, we are the first to propose a **generation-detection cycle consistent** (GDCC) learning framework that jointly optimizes L2I generation and OD in an end-to-end manner. In GDCC, consistency is maintained in two directions through two key components: (i) the **layout translation cycle loss**, which ensures consistency between the original layouts used to generate images and those predicted from the synthesized images, and (ii) the **image translation cycle loss**, which enforces consistency between the synthesized images and those reconstructed from the layouts predicted by the detector. These two losses guide the learning process in a cycle-consistent manner, ensuring tight alignment between the tasks during training and fostering mutual enhancement, which leads to more controllable diffusion models and more accurate object detectors.

Our GDCC framework offers several key advantages. First, GDCC enables mutual enhancement between L2I generation and OD, setting it apart from earlier approaches that focus on using one task to improve the other [6, 33, 70].

Such mutual enhancement results in more powerful L2I or OD models, as opposed to relying on pre-trained ones that are not fully optimized for improving the other task and may introduce errors during the training. Second, GDCC shows superior data efficiency by effectively utilizing auto-synthesized layout data, a capability not achieved by previous methods. Third, GDCC is computationally efficient in both training and inference. Our training process is accelerated by a perturbative single-step sampling strategy and a priority timestep re-sampling strategy. Fourth, GDCC serves as a training framework that retains the original architectures of both L2I and OD models, as well as the generation pipeline, ensuring inference speed is maintained. The key contributions of this paper are as follows:

- We are the first to identify the duality between L2I generation and OD, an insight that has previously been overlooked in the literature.
- Inspired by the task duality, we propose a **generation-detection cycle consistent** (GDCC) framework that jointly optimizes both tasks in an end-to-end manner and enables mutual enhancement between them.
- Our GDCC demonstrates both data and computational efficiency by allowing for the use of auto-synthesized data and incorporating a perturbative single-step sampling strategy along with a priority timestep re-sampling strategy to accelerate training.

Extensive experimental results confirm that GDCC establishes new benchmarks in both L2I generation and OD. For L2I generation, it achieves up to a 2.1% FID improvement over baseline L2I methods, and shows a 2.3% increase in YOLO score, indicating superior alignment between generated images and conditional layouts. For OD, GDCC achieves up to a 1.2% end-to-end improvement in AP, further validating the mutual enhancement between two tasks. With the incorporation of additional auto-synthesized training data, GDCC further achieves a 2.8% gain in detector mAP and a 3.0% enhancement in generator FID. These results confirm the effectiveness of our cycle-consistent framework in improving the controllability of diffusion models for image synthesis and the accuracy of detectors.

2. Related Work

Diffusion Models. Diffusion probabilistic models, first introduced in [57], have witnessed significant advancements both theoretically [13, 24, 31] and methodologically [25, 58, 59] in recent years. Latent Diffusion Model [52] further reduces computational costs by applying the diffusion process in the latent feature space rather than the pixel space. Due to their exceptional sample quality, diffusion models have set new standards across various benchmarks [11, 65, 75], including image editing [2, 22, 29, 40, 45], image-to-image transformation [32, 54, 64], and text-to-image (T2I) generation [16, 46, 47, 49, 50, 52, 55, 69]. Recent layout-to-image (L2I) studies seek to achieve more precise control over instance placement by extending pre-trained T2I models with layout conditions such as bounding boxes and object labels. Early approaches [27, 36, 60, 62, 76, 80] relied on a closed-set vocabulary from training labels (*e.g.*, COCO [3]) without using text prompts. With the emergence of image-text models such as CLIP [48], open-vocabulary methods became feasible [6, 7, 9, 10, 70, 72, 77, 82]. These methods incorporate layout information as text embeddings into pre-trained T2I diffusion models [52] to achieve more precise control over instance positioning.

In this paper, we boost L2I generation performance from a new perspective by proposing a cycle-consistent learning framework to achieve mutual benefits with OD, which naturally performs the inverse mapping of L2I from images to layouts. Our framework is computationally efficient thanks to the perturbative single-step sampling strategy and a priority timestep re-sampling strategy, while maintaining the same inference cost as the original L2I and OD models.

L2I Generation and OD. Several works have involved both L2I and OD tasks, but primarily use one to enhance the other. For example, ControlNet++ [33] uses pre-trained discriminative reward models to fine-tune controllable diffu-

sion models. However, these reward models are constrained by their original training data and struggle to adapt to the styles of synthesized images, which hinders their ability to provide more accurate feedback signals for training L2I models. On the other hand, methods [6, 61, 81] explore using synthetic data from diffusion models to improve object detection and segmentation. GeoDiffusion [6] demonstrates that OD can benefit from high-quality synthesized data generated by L2I models. DetDiffusion [70] leverages perceptive models (e.g., semantic segmentation) to enhance generation controllability and improve downstream OD performance, but introduces an extra perceptual model and lacks end-to-end joint optimization between the L2I and OD models. Despite these advances, the potential of tuning L2I models to generate samples specifically designed to boost OD performance remains underexplored.

This paper, for the first time, fully recognizes the duality between L2I and OD tasks and proposes a unified framework GDCC that enables *mutual enhancement* between the two tasks. Furthermore, in addition to leveraging large-scale paired layout-image data, our framework can utilize synthetic layout data, resulting in superior data efficiency.

Cycle-Consistent Learning. Cycle-consistent learning is a technique that leverages cyclic transformations to regularize the training process, ensuring that the data or tasks remain aligned when converted back and forth between representations. It can be applied within a single task through sample cycling, such as object tracking [43, 67, 71], temporal representation learning [14], visual acoustic matching [44], and image generation [8, 30, 33, 37, 74, 83]. It has also been shown to improve model performance across related tasks such as question answering *v.s.* question generation [34, 56, 63], captioning *v.s.* grounding [18, 68], vision-language navigation *v.s.* instruction generation [66], *etc*.

In this paper, we explore the uncharted potential of cycle-consistent learning between L2I generation and OD tasks, wherein the correlation and inherent duality have long been overlooked. These two tasks are seamlessly integrated into an end-to-end cycle-consistent learning framework, where their symmetrical structures provide informative feedback signals that enhance each other. Moreover, our framework allows for the usage of synthetic layout data, leading to superior data efficiency.

3. Methodology

In §3.1, we first introduce the preliminaries of diffusion-based L2I generation and OD. We then explore the inherent duality between these two tasks and show how GDCC utilizes cycle consistency to achieve mutual improvement (§3.2.1). Finally, we show GDCC (§3.2.2) and GDCC with extra auto-synthesized data (§3.2.3).

3.1. Preliminary

Diffusion-based L2I Generation. Diffusion models (DMs) [11, 13, 25], functioning by progressively transforming an initial random noise distribution into a coherent image, have arisen as renowned T2I generation methods. DMs define a T-step Markovian diffusion forward process to add Gaussian noise $\boldsymbol{\epsilon}$ into input image $\boldsymbol{x_0}$:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I), \tag{1}$$

where $\boldsymbol{x}_t$ is the perturbed image, t is the timestep, $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$, and $\alpha_t = 1-\beta_t$ is a differentiable function of t determined by the denoising sampler.

Diffusion-based L2I generation introduces additional control over DMs by incorporating layout conditions. Given a text prompt $\boldsymbol{y}$ and a layout condition $\boldsymbol{l}$, the training loss can be formulated as:

$$\mathcal{L}_{\text{dm}} = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{y},\boldsymbol{l},\boldsymbol{\epsilon}\sim\mathcal{N}(0,1)} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\big(t, \boldsymbol{x}_t, \boldsymbol{y}, \boldsymbol{l}\big) \right\|_2^2, \tag{2}$$

where $\boldsymbol{\epsilon}_\theta$ is the noise predictor realized as a U-Net [53].

During the sampling stage of L2I generation, the denoising process progressively eliminates the noise estimated by the diffusion model from a randomly sampled noise to predict the final image. Given noise $\boldsymbol{\epsilon}$, conditional text $\boldsymbol{y}$, and layout $\boldsymbol{l}$, the sampling process can be simplified to:

$$\boldsymbol{x}^{\text{syn}} = \mathcal{G}^T\big(t, \boldsymbol{\epsilon}, \boldsymbol{y}, \boldsymbol{l}\big), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I), \tag{3}$$

where $\boldsymbol{x}^{\text{syn}} \in \mathbb{R}^{H\times W\times 3}$ represents the synthesized image, and $\mathcal{G}^T$ denotes an L2I generator that performs T denoising steps. The layout $\boldsymbol{l} = \{(\boldsymbol{b}_n, c_n)\}_{n=1}^{N} \in \mathbb{R}^{N\times 5}$ consists of N bounding boxes, where each bounding box $\boldsymbol{b}_n = [x_{n,1}, y_{n,1}, x_{n,2}, y_{n,2}]$ defines the spatial location of object n, and $c_n \in \mathcal{C}$ denotes its corresponding semantic class.

Object Detection. This task aims to train a detector $\mathcal{D}(\cdot)$ to identify and localize objects within an image by predicting bounding boxes and their corresponding class labels:

$$\boldsymbol{l} = \mathcal{D}\big(\boldsymbol{x}\big), \tag{4}$$

where $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$ denotes the input image, and $\boldsymbol{l} = \{(\boldsymbol{b}_n, c_n)\}_{n=1}^{N'} \in \mathbb{R}^{N'\times 5}$ is the N' predicted layouts for the N objects in the image.

3.2. Generation-Detection Cycle-Consistent Learning Framework

3.2.1. Task Duality and Cycle-Consistency

From §3.1, it becomes evident that L2I and OD can be viewed as inverse tasks of each other, where the input and output of L2I generation correspond to the output and input of OD, respectively.

Though largely overlooked in previous research, such task duality can be effectively leveraged to improve both tasks through cycle consistency learning.

Specifically, if a layout is mapped to an image using an L2I generator $\mathcal{G}$, and then mapped back to a layout using an object detector $\mathcal{D}$, the process should recover the original layout. This forces consistency in what we term a **layout translation cycle**, ensuring more precise and realistic image generation that faithfully reflects input layouts.

Similarly, mapping an image to a layout and then back again should ideally recover the original image. This ensures consistency in an **image translation cycle**, which enhances its ability to accurately predict layouts from images.

These two cycle-consistent learning processes improve both $\mathcal{G}$ and $\mathcal{D}$ in an end-to-end manner, with each receiving feedback from the other.

In the following, we will present GDCC (§3.2.2) and GDCC with extra auto-synthesized data (§3.2.3).

3.2.2. GDCC

In the paired data setting, each image $\boldsymbol{x}_0 \in \mathbb{R}^{H\times W\times 3}$ is annotated with a structured layout $\boldsymbol{l} \in \mathbb{R}^{N\times 5}$ that includes bounding boxes and class labels for the objects in the image. The framework is shown in Fig. 2. Below, we detail the learning process of GDCC in this context.

Layout Translation Cycle.

As discussed in §3.2.1, in this process, $\mathcal{G}$ is optimized to minimize the discrepancy between the predicted and the original input layouts to achieve more precise and realistic image generation that faithfully reflects the input layout.

Specifically, given an L2I generation model $\mathcal{G}$ and the layout input $\boldsymbol{l} \in \mathbb{R}^{N\times 5}$, a conditionally synthesized images $\boldsymbol{x}_1^{\text{syn}} \in \mathbb{R}^{H\times W\times 3}$ can be obtained as follows:

$$\boldsymbol{x}_1^{\text{syn}} = \mathcal{G}^T\big(t, \boldsymbol{\epsilon}, \boldsymbol{y}, \boldsymbol{l}\big). \tag{5}$$

Next, a pre-trained object detector $\mathcal{D}$ is employed to map $\boldsymbol{x}_1^{\text{syn}}$ back into the layout space:

$$\hat{\boldsymbol{l}} = \mathcal{D}\big(\boldsymbol{x}_1^{\text{syn}}\big), \tag{6}$$

where a score threshold s_{thre} is applied to filter the predicted bounding boxes, leading to a more stable training process. The **layout translation cycle loss** $\mathcal{L}_{\text{layoutTC}}$ is then computed by measuring the similarity between the input layout $\boldsymbol{l}$ and its dual layout $\hat{\boldsymbol{l}} \in \mathbb{R}^{N\times 5}$:

$$\begin{aligned}\mathcal{L}_{\text{layoutTC}} &= \mathcal{L}_{\text{bbox}}\big(\boldsymbol{l}, \hat{\boldsymbol{l}}\big) \\ &= \mathcal{L}_{\text{reg}}\big(\{\boldsymbol{b}_n\}_{n=1}^{N}, \{\hat{\boldsymbol{b}}_n\}_{n=1}^{N'}\big) \\ &\quad + \mathcal{L}_{\text{cls}}\big(\{\boldsymbol{c}_n\}_{n=1}^{N}, \{\hat{\boldsymbol{c}}_n\}_{n=1}^{N'}\big),\end{aligned} \tag{7}$$

where N' is the number of detected objects and the bounding box loss $\mathcal{L}_{\text{bbox}}$ consists of a smooth L1 loss $\mathcal{L}_{\text{reg}}$ for regression and a cross-entropy loss $\mathcal{L}_{\text{cls}}$ for classification.

Perturbative Single-step Sampling. The T-step samplings process to generate $\boldsymbol{x}_1^{\text{syn}}$ in Eq. (5) is time-consuming and requires gradient storage at each timestep to facilitate back-propagation, which reduces the efficiency of layout translation cycle. Inspired by [33], we implement a *perturbative single-step denoising strategy* to accelerate the L2I process. Instead of generating $\boldsymbol{x}_1^{\text{syn}}$ from Gaussian noise, we obtain a special noise $\boldsymbol{x}_t^{\text{pert}}$ by perturbing image $\boldsymbol{x}_0$ with a small noise $\boldsymbol{\epsilon}_0$ for $t \le t_{\text{thre}}$ diffusion steps, where t_{thre} is a hyper-parameter that constrains $\boldsymbol{\epsilon}_0$ to be relatively small:

$$\boldsymbol{x}_t^{\text{pert}} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0. \tag{8}$$

We then perform a single-step denoising process on $\boldsymbol{x}_t^{\text{pert}}$ to achieve L2I generation and obtain $\boldsymbol{x}_1^{\text{syn}}$:

$$\begin{aligned}\boldsymbol{x}_1^{\text{syn}} &= \frac{\boldsymbol{x}_t^{\text{pert}} - \sqrt{1-\alpha_t}\,\boldsymbol{\epsilon}_\theta\big(t-1, \boldsymbol{x}_t^{\text{pert}}, \boldsymbol{y}, \boldsymbol{l}\big)}{\sqrt{\alpha_t}} \\ &= \mathcal{G}\big(t, \boldsymbol{x}_t^{\text{pert}}, \boldsymbol{y}, \boldsymbol{l}\big),\end{aligned} \tag{9}$$

where $\mathcal{G}$ denotes the L2I generator that performs perturbative single-step denoising, which is guided by the diffusion model loss $\mathcal{L}_{\text{dm}}$ defined in Eq. (2).

In summary, the total loss for training $\mathcal{G}$ in the layout transition cycle for the paired data setting is defined as follows:

$$\mathcal{L}_{\text{gen}} = \begin{cases}\mathcal{L}_{\text{dm}} + \lambda_1 \cdot \mathcal{L}_{\text{layoutTC}} & \text{if } t \le t_{\text{thre}} \\ \mathcal{L}_{\text{dm}} & \text{otherwise}\end{cases}. \tag{10}$$

Here, λ_1 adjusts the weight of the layout translation cycle loss $\mathcal{L}_{\text{layoutTC}}$, and t_{thre} denotes a threshold beyond which $\mathcal{L}_{\text{layoutTC}}$ is no longer applied, as the noise introduced in the perturbative single-step sampling process becomes too large to yield desired $\boldsymbol{x}_t^{\text{pert}}$ and $\boldsymbol{x}_1^{\text{syn}}$ for consistency learning.

Image Translation Cycle.

As discussed in §3.2.1, in this process, $\mathcal{D}$ is optimized to minimize the difference between the predicted and original images, thereby improving its ability to accurately predict layouts.

Formally, the layout $\hat{\boldsymbol{l}}$ obtained from $\boldsymbol{x}_1^{\text{syn}}$ (*cf.*, Eq. (6)) can be remap to image space by $\mathcal{G}$, resulting in $\boldsymbol{x}_2^{\text{syn}} \in \mathbb{R}^{H\times W\times 3}$. The **image translation cycle loss** $\mathcal{L}_{\text{imageTC}}$ is then computed by evaluating the similarity between $\boldsymbol{x}_1^{\text{syn}}$ (*cf.* Eq. (9)) and $\boldsymbol{x}_2^{\text{syn}}$:

$$\begin{aligned}\mathcal{L}_{\text{imageTC}} &= \|\mathcal{G}\big(t, \boldsymbol{x}_t^{\text{pert}}, \boldsymbol{y}, \boldsymbol{l}\big) - \mathcal{G}\big(t, \boldsymbol{x}_t^{\text{pert}}, \boldsymbol{y}, \hat{\boldsymbol{l}}\big)\|_2^2 \\ &= \Big\| \big[\boldsymbol{x}_t^{\text{pert}} - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta\big(t, \boldsymbol{x}_t^{\text{pert}}, \boldsymbol{y}, \boldsymbol{l}\big)\big] / \sqrt{\bar{\alpha}_t} \\ &\quad - [\boldsymbol{x}_t^{\text{pert}} - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta\big(t, \boldsymbol{x}_t^{\text{pert}}, \boldsymbol{y}, \hat{\boldsymbol{l}}\big)]/\sqrt{\bar{\alpha}_t}\Big\|_2^2 \\ &= \big(\sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t}\big)\|\boldsymbol{\epsilon}_\theta\big(t, \boldsymbol{x}_t^{\text{pert}}, \boldsymbol{y}, \boldsymbol{l}\big) \\ &\quad - \boldsymbol{\epsilon}_\theta(t, \boldsymbol{x}_t^{\text{pert}}, \boldsymbol{y}, \hat{\boldsymbol{l}})\|_2^2.\end{aligned} \tag{11}$$

We obtain $\mathcal{L}_{\text{imageTC}} = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{y},\boldsymbol{l},\boldsymbol{\epsilon}\sim\mathcal{N}(0,1)}\|\boldsymbol{\epsilon}_\theta(t, \boldsymbol{x}_t^{\text{pert}}, \boldsymbol{y}, \boldsymbol{l}) - \boldsymbol{\epsilon}_\theta(t, \boldsymbol{x}_t^{\text{pert}}, \boldsymbol{y}, \hat{\boldsymbol{l}})\|_2^2$ by omitting the scaling factor. As seen, with the above perturbative single-step denoising strategy, the image translation cycle only requires to compute the noise predicted by the U-Net denoiser $\boldsymbol{\epsilon}_\theta$ at timestep t during two generation forward translations, which significantly improves the efficiency of GDCC.

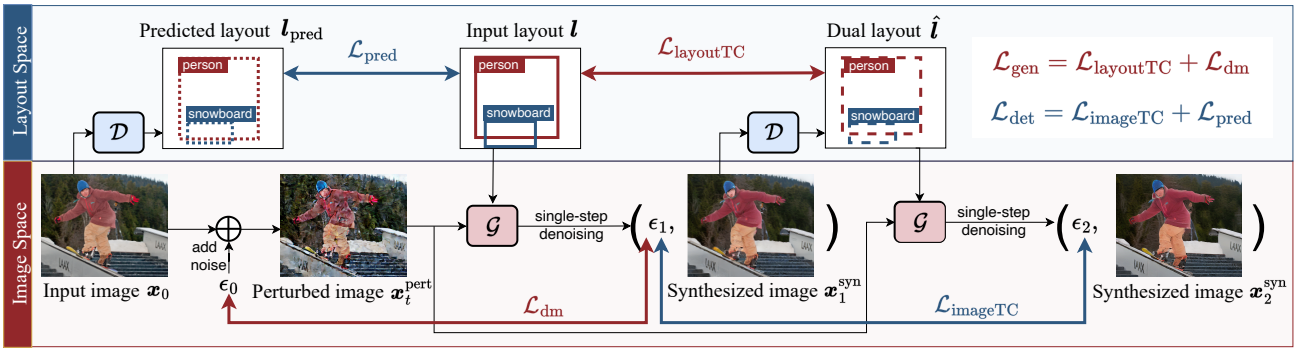

Figure 2. **GDCC framework in paired data setting.** The L2I generator $\mathcal{G}$ maps from the layout space to the image space, while the object detector $\mathcal{D}$ performs the inverse mapping. Given a paired data with an input image $\boldsymbol{x}_0$ and its corresponding layout $\boldsymbol{l}$, $\mathcal{G}$ is trained with the layout translation cycle loss $\mathcal{L}_{\text{layoutTC}}$ and the diffusion model loss $\mathcal{L}_{\text{dm}}$, and $\mathcal{D}$ is trained with the image translation cycle loss $\mathcal{L}_{\text{imageTC}}$ and the prediction loss $\mathcal{L}_{\text{pred}}$. See §3.2.2 for details.

To maintain the performance of $\mathcal{D}$ on real-world data, we make full use of the paired data by predicting the layout $\boldsymbol{l}_{\text{pred}} \in \mathbb{R}^{N\times 5}$ from image $\boldsymbol{x}_0$, and minimizing the prediction loss between $\boldsymbol{l}_{\text{pred}}$ and the annotated layout $\boldsymbol{l}$, defined as $\mathcal{L}_{\text{pred}} = \mathcal{L}_{\text{bbox}}(\boldsymbol{l}, \boldsymbol{l}_{\text{pred}})$, during the training of $\mathcal{D}$. In summary, the total loss for training $\mathcal{D}$ in the image translation cycle in paired data setting is as follows:

$$\mathcal{L}_{\text{det}} = \begin{cases} \mathcal{L}_{\text{pred}} + \lambda_2 \cdot \mathcal{L}_{\text{imageTC}} & \text{if } t \le t_{\text{thre}} \\ \mathcal{L}_{\text{pred}} & \text{otherwise} \end{cases}. \tag{12}$$

Similar to Eq. (10), λ_2 is the weight of $\mathcal{L}_{\text{imageTC}}$. Image translation cycle is performed within t_{thre} timesteps to fulfill the constraint of the perturbative single-step denoising.

Priority Timestep Re-Sampling. In the process of perturbative single-step sampling, the t_{thre} value is supposed to be small to ensure that ϵ_0 remains relatively constrained. However, the traditional uniform sampling strategy leads to a low cycle reward probability (*i.e.*, $t_{\text{thre}}/t_{\text{max}}$), resulting in slow convergence. Thus, we introduce the re-weighting factor w to increase the reward probability from $t_{\text{thre}}/t_{\text{max}}$ to $w * t_{\text{thre}}/t_{\text{max}}$. The re-weighted timestep sampling probability $p_{\text{reweight}}(t)$ for each interval is given by:

$$p_{\text{reweight}}(t) = \begin{cases} w * t_{\text{thre}}/t_{\text{max}} & \text{if } t \le t_{\text{thre}} \\ 1 - w * t_{\text{thre}}/t_{\text{max}} & \text{otherwise} \end{cases}. \tag{13}$$

When $t \le t_{\text{thre}}$, the layout and image translation cycle losses, as defined in Eq. (10) and (12), are triggered. The effectiveness of this re-sampling strategy is demonstrated by the results shown in Table 5b. The re-sampling strategy increases the reward frequency, simultaneously regulating the balance between the reward and the original loss terms (*i.e.*, $\mathcal{L}_{\text{dm}}$ and $\mathcal{L}_{\text{pred}}$). An appropriately chosen w enhances reward training efficiency, achieving superior results within a fraction of the original training time.

3.2.3. GDCC with Extra Auto-Synthesized Data

In this section, we explore GDCC learning with extra auto-synthesized data. In addition to leveraging large-scale annotated layout-image pairs to achieve mutual improvement of the L2I generator and object detector, GDCC also facilitates more efficient use of *annotation-free* auto-synthesized data, thereby further enhancing data efficiency.

To generate additional synthetic layouts, we employ VisorGPT [73], a recently developed generative model pre-trained on COCO [38] that autonomously samples layouts based on its learned visual priors. Specifically, we first input the class names and the corresponding number of instances from each image in the training set into VisorGPT to sample synthetic layouts $\boldsymbol{l}^{\text{syn}} \in \mathbb{R}^{N\times 5}$. Second, the synthetic layouts are fed into the generator $\mathcal{G}$ to obtain corresponding generated synthetic images $\boldsymbol{x}_0^{\text{syn}} \in \mathbb{R}^{H\times W\times 3}$. Third, to construct an augmented training set $\mathcal{S}^{\text{new}}$, we incorporate the synthetic data $\mathcal{S}^{\text{syn}}$ into the original training set $\mathcal{S}$: $\mathcal{S}^{\text{new}} = \mathcal{S} \cup \mathcal{S}^{\text{syn}}$. Experimental results are presented in Table 3. As shown, incorporating additional synthetic data further enhances the performance, demonstrating the data efficiency of GDCC and the great potential of leveraging synthetic data to improve both the generator and detector.

4. Experiments

4.1. Experimental Setup

Following [6], we train and evaluate the models on the COCO [3, 38] and NuImages [4] datasets. For L2I generation models, *fidelity* is evaluated using Frechet Inception Distance (FID) [23] and YOLO score [36], while *trainability* is measured by re-training object detection (OD) models on the synthetic and real data using Average Precision (AP). For OD models, detection fine-tuning performance is assessed using AP. Related details are shown in Appendix §B.

Training. We fine-tune the pre-trained generators *i.e.*, GeoDiffusion [6] and ControlNet [79], and a object detector,

Method	Res.	Epoch	FID ↓	mAP ↑	AP_{50} ↑	AP_{75} ↑
LostGAN [60] [ICCV 19]	256^2	200	42.55	9.1	15.3	9.8
LAMA [36] [ICCV 21]		200	31.12	13.4	19.7	14.9
CAL2IM [21] [CVPR 21]		200	25.95	10.0	14.9	11.1
Taming [27] [ArXiv 21]		128	33.68	-	-	-
TwFA [76] [CVPR 22]		300	22.15	-	28.2	20.1
Frido [15] [AAAI 23]		200	37.14	17.2	-	-
L.Diffusion† [82] [CVPR 23]		180	22.65	14.9	27.5	14.9
DetDiffusion‡ [70] [CVPR 24]		60	19.28	29.8	38.6	34.1
GeoDiffusion [6] [ICLR 24]		60	20.16	29.1	38.9	33.6
+ plain fine-tuning		2	20.13	29.3	39.0	33.9
+ **GDCC**		2	**18.02**	**31.4**	**41.2**	**36.4**
ReCo† [77] [CVPR 23]	512^2	100	29.69	18.8	33.5	19.7
L.Diffuse† [9] [ArXiv 23]		60	22.20	11.4	23.1	10.1
GLIGEN [35] [CVPR 23]		86	21.04	22.4	36.5	24.1
ControlNet [79] [ICCV 23]		60	28.14	25.2	46.7	22.7
+ plain fine-tuning		2	28.06	25.4	46.7	23.0
+ **GDCC**		2	**26.38**	**27.0**	**47.9**	**24.2**
GeoDiffusion [6] [ICLR 24]		60	18.89	30.6	41.7	35.6
+ plain fine-tuning		2	18.78	30.9	41.9	35.7
+ **GDCC**		2	**17.15**	**32.6**	**43.6**	**38.0**

Table 1. **Quantitative results of generation fidelity on COCO 2017** [38]. GDCC is fine-tuned for 2 epochs on pre-trained L2I methods. "plain fine-tuning" refers to continuing training L2I model for same extra epochs as GDCC. †: re-implementation from GeoDiffusion [6]. ‡: with additional mask annotations. YOLO score is reported as AP metrics. See §4.2 for details.

i.e., Faster R-CNN [51] for a few more epochs. For GeoDiffusion, experiments on both COCO [3, 38] and NuImages [4] are performed. In this process, only the U-Net denoiser parameters are updated, while all other parameters remain fixed. GeoDiffusion is fine-tuned for 2 epochs on COCO-Stuff and 3 epochs on NuImages, which is remarkably efficient. For ControlNet, we finetune the pretrained ControlNet using GDCC for 2 epochs by updating only the ControlNet-specific parameters and keep all others frozen. Related details are shown in Appendix §B.

Faster R-CNN [51], pre-trained separately on the COCO 2017 and the NuImages training sets, is employed for the respective datasets. A score threshold $s_{\text{thre}} = 0.5$ is used to filter the predicted bounding boxes. Each filtered bounding box is assigned to a ground truth box with an Intersection over Union (IoU) of at least 0.5, or classified as background.

Testing. Our GDCC framework preserves the original architectures of all the L2I and OD models, as well as the layout encoding approach of L2I models, ensuring that the inference speed of each model remains unchanged.

Following GeoDiffusion [6], fidelity is assessed using a Mask R-CNN [20] pre-trained on the NuImages training set for NuImages dataset [4]. A YOLOv4 [1] model pre-trained on COCO 2017 training set is used to derive YOLO score. The pre-trained detector first performs inference on the generated images, and the resulting predictions are then compared with the corresponding ground truth annotations. FID is achieved by computing the similarity between generated and real samples following [6, 36]. To assess the trainability, we augment the original training data with generated images and their corresponding layouts, creating a unified dataset. We subsequently train Faster R-CNN [51] on this unified dataset using the standard 1× schedule. Related details are shown in Appendix §C.

Method	mAP ↑	AP_{50} ↑	AP_{75} ↑	AP^m ↑	AP^l ↑
– Detection Fine-tuning –					
Faster R-CNN [51] [NIPS 15]	37.3	58.2	40.8	40.7	48.2
+ plain fine-tuning	37.5 ↑ 0.2	58.4	40.9	40.8	48.4
+ **GDCC**	**38.5** ↑ 1.2	**58.7**	**42.2**	**41.7**	**49.4**
– Generation Trainability –					
L.Diffusion [82] [CVPR 23]	36.5 ↓ 0.8	57.0	39.5	39.7	47.5
L.Diffuse [9] [ArXiv 23]	36.6 ↓ 0.7	57.4	39.5	40.0	47.4
GLIGEN [35] [CVPR 23]	36.8 ↓ 0.5	57.6	39.9	40.3	47.9
ControlNet [79] [ICCV 23]	36.9 ↓ 0.4	57.8	39.6	40.4	49.0
GeoDiffusion [6] [ICLR 24]	38.4 ↑ 1.1	58.5	42.4	42.1	50.3
+ plain fine-tuning	38.5 ↑ 1.2	58.6	42.4	42.2	50.3
+ **GDCC**	**39.0** ↑ 1.7	**58.9**	**43.1**	**42.6**	**50.7**

Table 2. **Quantitative results of detection fine-tuning and generation trainability on COCO 2017** [38]. Detection fine-tuning refers to fine-tuning the detector for 2 epochs during the training of GDCC, while generative trainability denotes the re-training of the detector on generated and real samples. "plain fine-tuning" refers to continuing training OD or L2I model for the same extra epochs as GDCC. A Faster R-CNN pre-trained on COCO 2017 is employed as the baseline. Detectors are evaluated on COCO 2017 validation set after training. The input resolution is set to 800×456 following [6]. See §4.2 for details.

Setting	# Training Data	Generation Trainability ↑	Generation Fidelity FID ↓	Generation Fidelity YOLO score ↑	Detection Score ↑
Baseline	75k	37.3	20.16	29.1	37.3
real	75k	39.0	18.02	31.4	38.5
real+syn.	75k+75k	39.6	17.54	32.0	38.9
real+syn.	75k+150k	**40.1**	**17.16**	**32.5**	**39.2**

Table 3. **Quantitative results of using extra auto-synthesized training data on COCO 2017** [38]. "syn." denotes synthetic layouts and corresponding images generated by GDCC. The "Baseline" for *Detection Score* and *Generation Trainability* is a Faster R-CNN [51] pre-trained on COCO 2017, while the "Baseline" for *Generation Fidelity* is GeoDiffusion [6]. See §3.2.3 and §4.2.

4.2. Quantitative Results

Generation Fidelity on COCO 2017 [38]. For generation fidelity, as shown in Table 1, GDCC learning framework significantly improves existing L2I generation methods in terms of both image fidelity, as measured by FID, and control fidelity, as evaluated by YOLO score, by a large degree.

At a 256 × 256 input resolution, for the GeoDiffusion [6] method, our GDCC framework achieves improvements of **2.3%**/**2.3%**/**2.8%** in mAP, mAP_{50}, and mAP_{75}, reaching **31.4%**/**41.2%**/**36.4%**, even surpassing the performance of original GeoDiffusion at a 512 × 512 resolution. Additionally, GDCC achieves a **2.14%** improvement in FID. It is worth noting that, despite DetDiffusion [70]

Method	Res.	Epoch	FID ↓	Average Precision↑							
				mAP	AP_{50}	AP_{75}	AP^m	AP^l	trailer	ped.	car
Oracle	-	-	-	48.2	75.0	52.0	46.7	60.5	17.8	48.5	64.9
LostGAN [60] [ICCV 19]	256^2	256	59.95	4.4	9.8	3.3	2.1	12.3	0.3	2.7	12.2
LAMA [36] [ICCV 21]		256	63.85	3.2	8.3	1.9	2.0	9.4	1.4	1.3	8.8
Taming [27] [ArXiv 21]		256	32.84	7.4	19.0	4.8	2.8	18.8	6.0	3.0	17.3
GeoDiffusion [6] [ICLR 24]		64	14.58	15.6	31.7	13.4	6.3	38.3	13.3	6.5	26.3
+ plain fine-tuning		3	14.31	15.8	31.8	13.5	6.3	38.5	13.6	6.7	26.3
+ **GDCC**		3	**12.54**	**17.5**	**33.5**	**15.6**	**8.3**	**40.3**	**15.0**	**8.1**	**28.5**
ReCo [77] [CVPR 23]	512^2	64	27.10	17.1	41.1	11.8	10.9	36.2	8.0	7.6	31.8
GLIGEN [35] [CVPR 23]		64	16.68	21.3	42.1	19.1	15.9	40.8	8.5	14.7	38.7
ControlNet [79] [ICCV 23]		64	23.26	22.6	43.9	20.7	17.3	41.9	10.5	16.7	40.7
GeoDiffusion [6] [ICLR 24]		64	9.58	31.8	62.9	28.7	27.0	53.8	21.2	18.2	46.0
+ plain fine-tuning		3	9.32	32.0	63.1	28.8	27.1	54.1	21.4	18.3	46.0
+ **GDCC**		3	**7.97**	**33.6**	**64.7**	**30.7**	**28.6**	**55.9**	**29.5**	**20.2**	**47.6**

Method	mAP ↑
– Detection Fine-tuning –	
Faster R-CNN [51] [NIPS 15]	36.9
+ plain fine-tuning	37.2 ↑ 0.3
+ **GDCC**	**37.9** ↑ 1.0
– Generation Trainability –	
LostGAN [60] [ICCV 19]	35.6 ↓ 1.3
LAMA [36] [ICCV 21]	35.6 ↓ 1.3
Taming [27] [ArXiv 21]	35.8 ↓ 1.1
ReCo [77] [CVPR 23]	36.1 ↓ 0.8
GLIGEN [35] [CVPR 23]	36.3 ↓ 0.6
ControlNet [79] [ICCV 23]	36.4 ↓ 0.5
GeoDiffusion [6] [ICLR 24]	38.3 ↑ 1.4
+ plain fine-tuning	38.3 ↑ 1.4
+ **GDCC**	**38.9** ↑ 2.0

Table 4. **Quantitative results of generation fidelity (left), detection fine-tuning, and generation trainability (right) on NuImages**[4]. GDCC is fune-tuned for 3 epochs on pre-trained L2I and OD models. "plain fine-tuning" refers to continuing training OD or L2I model for the same extra epochs as GDCC. "ped." denotes pedestrian. For generation fidelity, YOLO score is reported as AP metrics. See §4.2.

Figure 3. **Generation visual results on COCO 2017** [38]. GDCC is fine-tuned on pre-trained GeoDiffusion [6] for 2 epochs. For fair comparisons, same seed is employed for sampling. See §4.3 for details. For more visualizations, please refer to Appendix §F-G.

employing additional and detailed mask annotations for supervision while GDCC only uses bounding box label, our method still outperforms it. For a 512 × 512 input, GDCC also achieves **2.0%**/**1.9%**/**2.4%** mAP and **1.74%** FID enhancement compared with initial model, demonstrating the **state-of-the-art** performance in L2I generation realm. Based on the classic controllable generator ControlNet [79], GDCC also achieves notable enhancements.

The enhanced FID and YOLO score achieved with GDCC demonstrate its effectiveness. GDCC not only enables precise layout control in generation but also enhances quality of the generated images, improving their resemblance to real-world data. Additionally, the improvements across different controllable generation methods demonstrate that GDCC is not dependent on any specific approach, highlighting its robustness and extensibility. Furthermore, compared with plain fine-tuning with same epochs, GDCC achieves significant improvement.

Detection Performance and Generation Trainability on COCO 2017 [38]. A Faster R-CNN detector [51] trained on the COCO 2017 training set is employed for detection fine-tuning. To begin with, we set the performance of the detector on COCO 2017 validation set as our baseline.

As can be seen in Table 2, fine-tuning the detector at GDCC training process in an end-to-end manner leads to performance improvements with **1.2%** on the validation set. **For the first time**, we demonstrate that the L2I generation model can be advantageous to the object detector during training in an end-to-end manner, while previous works [6, 70] only use generated images to re-train the detector in the data augmentation manner. To make a comparison of generation trainability, we also re-train the detector with generated and real data with ImageNet [12] pre-trained weights. As shown, GeoDiffusion fine-tuned with

Components	Detection Score ↑	Generation Fidelity FID ↓	YOLO score ↑
Baseline	37.3	20.16	29.1
+ $\mathcal{L}_{\text{dm}}$	37.3	20.13	29.3
+ $\mathcal{L}_{\text{gen}}$	37.7	18.94	30.6
+ $\mathcal{L}_{\text{pred}}$	37.5	20.16	29.1
+ $\mathcal{L}_{\text{det}}$	38.0	19.28	29.9
+ **GDCC**	**38.5**	**18.02**	**31.4**

(a) essential components

t_{thre}	w	Epoch	Hours	mAP ↑	FID ↓
0	0	2	1.9	37.5	20.13
50	0	2	2.4	37.6	19.57
50	0	6	7.2	38.5	18.11
50	3	2	2.5	38.0	18.98
50	6	2	2.6	**38.5**	**18.02**
50	9	2	2.7	38.3	18.85
100	6	2	2.9	37.9	19.29

(b) reward strategy

Detectors	Detection Score ↑ original / fine-tuning	Generation Fidelity FID ↓	YOLO score ↑
Faster R-CNN [51]	37.3 / 38.5 ↑ 1.2	18.02	31.4
Mask R-CNN [20]	38.2 / 40.0 ↑ 0.8	17.86	31.6
Cascade R-CNN [5]	40.3 / 41.3 ↑ 1.0	17.64	31.8
YOLOX-S [17]	40.5 / 41.8 ↑ 1.3	17.60	31.9
DINO [78]	49.0 / 50.1 ↑ 1.1	17.12	32.5
CO-DETR [84]	52.0 / 52.9 ↑ 0.9	**16.93**	**32.8**

(c) different detectors

Table 5. **A set of ablative experiments on COCO 2017** [38]. GeoDiffusion [6] pre-trained on COCO [3, 38] is employed as L2I baseline. L2I and detection models are fine-tuned for 2 epochs. In (c), "fine-tuning" indicates optimizing the detector using GDCC. See §4.4.

GDCC achieves **1.7**%/**0.7**%/**2.3**% AP improvement over the baseline, demonstrating superior generation trainability.

Generation Fidelity on NuImages [4]. To illustrate the generalizability of GDCC with respect to dataset, more experiments are conducted on NuImages. As presented in Table 4, GDCC outperforms all baselines significantly in FID and YOLO score after three epochs of fine-tuning.

Detection Performance and Generation Trainability on NuImages [4]. As can be seen in Table 4, GDCC achieves improvement on NuImages validation set after fine-tuning Faster-RCNN which is pre-trained on training set. In a data augmentation manner, GDCC demonstrates an accuracy improvement of **2.0**% compared to the baseline.

Performance of Using Extra Auto-Synthesized Data on COCO 2017 [38]. Table 1, 2 use original paired COCO data, while Table 3 explores the use of synthetic data. As shown, using extra annotation-free auto-synthesized data boosts the performance of both generator and detector. As more synthetic data is incorporated, the performance further improves, highlighting the great potential of leveraging synthetic data. With the usage of 150k synthetic layouts and images, the detector mAP improves by **2.8**% through augmentation and **1.9**% in an end-to-end manner, respectively, while the generator FID achieves a **3.0**% gain.

4.3. Qualitative Results

Fig. 3 shows representative generation visual results on COCO 2017, with the same random seed used during sampling to ensure fair comparison. L2I model [6] demonstrates stronger layout controllability (1st and 2nd columns) and superior image fidelity (2nd column) after fine-tuning with GDCC. More generation and detection visualizations are shown in Appendix §F and §G, respectively.

4.4. Diagnostic Experiments

To gain more insights into GDCC, we conduct a set of ablative studies on COCO 2017 [38] using GeoDiffusion [6].

Essential Components. As shown in Table5a, the diffusion training loss $\mathcal{L}_{\text{dm}}$ (*cf.* Eq. (2)) and the prediction loss $\mathcal{L}_{\text{pred}}$ lead to a slight improvement in generation fidelity and detection score, respectively, due to more iterations on training samples. When fine-tuning the generator with $\mathcal{L}_{\text{gen}}$ (*cf.* Eq.(10)) which contains both $\mathcal{L}_{\text{dm}}$ and layout translation cycle loss $\mathcal{L}_{\text{layoutTC}}$ (*cf.* Eq.(7)), there is a significant improvement in generation fidelity. Similarity, $\mathcal{L}_{\text{det}}$ (*cf.* Eq. (12)) with image translation cycle loss $\mathcal{L}_{\text{imageTC}}$ (*cf.* Eq.(11)) further improve detector's performance. GDCC, fine-tuning both the generator and detector in an end-to-end manner, achieves superior performance compared with each individual component. This indicates the duality of two tasks, and GDCC facilitates mutual enhancement during iterations.

Reward Strategy. A small t_{thre} facilitates the cycle reward process, while w not only increases the frequency of the cycle reward but also controls the balance between the cycle reward and the original $\mathcal{L}_{\text{dm}}$ or $\mathcal{L}_{\text{pred}}$. We aim to identify an appropriate value for w while constraining $\mathcal{L}_{\text{pred}}$ within a small range to ensure both the effectiveness and efficiency of the algorithm. Setting $t_{\text{thre}} = 0$ indicates that only $\mathcal{L}_{\text{dm}}$ and $\mathcal{L}_{\text{pred}}$ are active. Table 5b shows that: (i) a proper w enhances reward **training efficiency**, achieving better results in only 36% of training time without w; (ii) a large t_{thre} introduces noise into the reward process.

Different Detectors. GDCC is a general training framework independent of the generators and detectors. In our main experiments, we use Faster R-CNN [51] as the default detector. To evaluate the generalization ability of GDCC, we conduct experiments using different detectors. As shown in Table 5c, GDCC consistently improves both detection and generation scores across all tested detectors.

5. Conclusion

In this paper, we propose GDCC, an end-to-end framework that jointly optimizes L2I generation and OD tasks. By exploring the inherent duality between these two tasks, GDCC facilitates mutual enhancement of L2I and OD models through the layout and image translation cycle losses. Additionally, GDCC allows for more efficient use of auto-synthesized data, further enhancing data efficiency. Notably, our GDCC is computationally efficient thanks to the perturbative single-step sampling and priority timestep re-sampling strategies during training, while maintaining the same inference cost as the original L2I and OD models. Experiments confirm that GDCC improves both the controllability of L2I models and accuracy of OD.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (No. 62472222, 62306292), Natural Science Foundation of Jiangsu Province (No. BK20240080), Zhejiang Provincial Natural Science Foundation of China (No. LD25F020001), Fundamental Research Funds for the Central Universities (226-2025-00057), and CIPSC-SMP-Zhipu Large Model Cross-Disciplinary Fund.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 6, 2

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pages 18392–18402, 2023. 2

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 2, 5, 6, 8

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 5, 6, 7, 8, 2, 9

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 8

[6] Kai Chen, Enze Xie, Zhe Chen, Yibo Wang, Lanqing Hong, Zhenguo Li, and Dit-Yan Yeung. Geodiffusion: Text-prompted geometric control for object detection data generation. In *ICLR*, 2023. 1, 2, 3, 5, 6, 7, 8, 9

[7] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *WACV*, pages 5343–5353, 2024. 2

[8] Weifeng Chen, Jiacheng Zhang, Jie Wu, Hefeng Wu, Xuefeng Xiao, and Liang Lin. Id-aligner: Enhancing identity-preserving text-to-image generation with reward feedback learning. *arXiv preprint arXiv:2404.15449*, 2024. 3

[9] Jiaxin Cheng, Xiao Liang, Xingjian Shi, Tong He, Tianjun Xiao, and Mu Li. Layoutdiffuse: Adapting foundational diffusion models for layout-to-image generation. *arXiv preprint arXiv:2302.08908*, 2023. 2, 6

[10] Guillaume Couairon, Marlene Careil, Matthieu Cord, Stéphane Lathuiliere, and Jakob Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. In *ICCV*, pages 2174–2183, 2023. 2

[11] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE TPAMI*, 45(9):10850–10869, 2023. 2, 3

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 7, 2

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, pages 8780–8794, 2021. 2, 3

[14] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *CVPR*, pages 1801–1810, 2019. 3

[15] Wan-Cyuan Fan, Yen-Chun Chen, DongDong Chen, Yu Cheng, Lu Yuan, and Yu-Chiang Frank Wang. Frido: Feature pyramid diffusion for complex scene image synthesis. In *AAAI*, pages 579–587, 2023. 6

[16] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2

[17] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 8, 5, 11

[18] Longteng Guo, Jing Liu, Peng Yao, Jiangwei Li, and Hanqing Lu. Mscap: Multi-style image captioning with unpaired stylized text. In *CVPR*, pages 4204–4213, 2019. 3

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1, 6, 8, 2

[21] Sen He, Wentong Liao, Michael Ying Yang, Yongxin Yang, Yi-Zhe Song, Bodo Rosenhahn, and Tao Xiang. Context-aware layout to image generation with enhanced object appearance. In *CVPR*, pages 15049–15058, 2021. 6

[22] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2

[23] 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 *NeurIPS*, 2017. 5, 1, 2

[24] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2, 3

[26] Loshchilov Ilya and Hutter Frank. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[27] Manuel Jahn, Robin Rombach, and Björn Ommer. High-resolution complex scene synthesis with transformers. *arXiv preprint arXiv:2105.06458*, 2021. 2, 6, 7

[28] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, pages 1219–1228, 2018. 2

[29] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *CVPR*, pages 6007–6017, 2023. 2

[30] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, pages 1857–1865, 2017. 3

[31] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *NeurIPS*, 34, 2021. 2

[32] Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *CVPR*, pages 1952–1961, 2023. 2

[33] Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. Controlnet++: Improving conditional controls with efficient consistency feedback. *arXiv preprint arXiv:2404.07987*, 2024. 1, 2, 3, 4

[34] Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, and Ming Zhou. Visual question generation as dual task of visual question answering. In *CVPR*, pages 6116–6124, 2018. 3

[35] 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 *CVPR*, pages 22511–22521, 2023. 6, 7

[36] Zejian Li, Jingyu Wu, Immanuel Koh, Yongchuan Tang, and Lingyun Sun. Image synthesis from layout with locality-aware mask adaption. In *ICCV*, pages 13819–13828, 2021. 1, 2, 5, 6, 7

[37] Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, et al. Rich human feedback for text-to-image generation. In *CVPR*, pages 19401–19411, 2024. 3

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 5, 6, 7, 8, 2, 4, 10, 11, 12

[39] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 2

[40] Chang Liu, Xiangtai Li, and Henghui Ding. Referring image editing: Object-level image editing via referring expressions. In *CVPR*, pages 13128–13138, 2024. 2

[41] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022. 2

[42] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 2

[43] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J Crandall, and Steven CH Hoi. Learning video object segmentation from unlabeled videos. In *CVPR*, pages 8960–8970, 2020. 3

[44] Jian Ma, Wenguan Wang, Yi Yang, and Feng Zheng. Mutual learning for acoustic matching and dereverberation via visual scene-driven diffusion. In *European Conference on Computer Vision*, pages 331–349, 2024. 3

[45] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2

[46] Alex 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. *arXiv preprint arXiv:2112.10741*, 2021. 2

[47] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 2

[49] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831, 2021. 2

[50] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[51] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 6, 7, 8, 2, 5, 10

[52] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2

[53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 3

[54] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH*, pages 1–10, 2022. 2

[55] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, pages 36479–36494, 2022. 2

[56] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *CVPR*, pages 6649–6658, 2019. 3

[57] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265, 2015. 2

[58] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[59] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[60] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *ICCV*, pages 10531–10540, 2019. 2, 6, 7

[61] Saksham Suri, Fanyi Xiao, Animesh Sinha, Sean Chang Culatana, Raghuraman Krishnamoorthi, Chenchen Zhu, and Abhinav Shrivastava. Gen2det: Generate to detect. *arXiv preprint arXiv:2312.04566*, 2023. 3

[62] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *AAAI*, pages 2647–2655, 2021. 2

[63] Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*, 2017. 3

[64] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, pages 1921–1930, 2023. 2

[65] Anwaar Ulhaq, Naveed Akhtar, and Ganna Pogrebna. Efficient diffusion models for vision: A survey. *arXiv preprint arXiv:2210.09292*, 2022. 2

[66] Hanqing Wang, Wei Liang, Jianbing Shen, Luc Van Gool, and Wenguan Wang. Counterfactual cycle-consistent learning for instruction following and generation in vision-language navigation. In *CVPR*, pages 15471–15481, 2022. 3

[67] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, pages 1308–1317, 2019. 3

[68] Ning Wang, Jiajun Deng, and Mingbo Jia. Cycle-consistency learning for captioning and grounding. In *AAAI*, pages 5535–5543, 2024. 3

[69] Wenguan Wang, Yi Yang, and Yunhe Pan. Visual knowledge in the big model era: Retrospect and prospect. *Frontiers of Information Technology & Electronic Engineering*, 26(1):1–19, 2025. 2

[70] Yibo Wang, Ruiyuan Gao, Kai Chen, Kaiqiang Zhou, Yingjie Cai, Lanqing Hong, Zhenguo Li, Lihui Jiang, Dit-Yan Yeung, Qiang Xu, et al. Detdiffusion: Synergizing generative and perceptive models for enhanced data generation and perception. In *CVPR*, pages 7246–7255, 2024. 1, 2, 3, 6, 7

[71] Qiao Wu, Jiaqi Yang, Kun Sun, Chu'ai Zhang, Yanning Zhang, and Mathieu Salzmann. Mixcycle: Mixup assisted semi-supervised 3d single object tracking with cycle consistency. In *ICCV*, pages 13956–13966, 2023. 3

[72] 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 *ICCV*, pages 7452–7461, 2023. 2

[73] Jinheng Xie, Kai Ye, Yudong Li, Yuexiang Li, Kevin Qinghong Lin, Yefeng Zheng, Linlin Shen, and Mike Zheng Shou. Learning visual prior via generative pre-training. In *NeurIPS*, 2024. 5, 4

[74] Sihan Xu, Ziqiao Ma, Yidong Huang, Honglak Lee, and Joyce Chai. Cyclenet: Rethinking cycle consistency in text-guided diffusion for image manipulation. *NeurIPS*, 36, 2024. 3

[75] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4): 1–39, 2023. 2

[76] Zuopeng Yang, Daqing Liu, Chaoyue Wang, Jie Yang, and Dacheng Tao. Modeling image composition for complex scene generation. In *CVPR*, pages 7764–7773, 2022. 2, 6

[77] 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 *CVPR*, pages 14246–14255, 2023. 1, 2, 6, 7

[78] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 8

[79] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 1, 5, 6, 7, 2

[80] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *CVPR*, pages 8584–8593, 2019. 2

[81] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, et al. X-paste: Revisiting scalable copy-paste for instance segmentation using clip and stablediffusion. In *ICML*, pages 42098–42109, 2023. 3

[82] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *CVPR*, pages 22490–22499, 2023. 2, 6

[83] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 3

[84] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *ICCV*, pages 6748–6758, 2023. 8, 5, 12