Appendix

A. Derivation of Training Objective

Based on the assumption that the diffusion model can learn to utilize conditions during training, thereby generating images $\hat{\mathbf{x}} \sim p(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$, many studies integrate multiple conditions, e.g., text, depth, pose, into the training of diffusion model [6, 28, 31]:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim U(0,T)} \mathbb{E}_{\mathbf{x}_0,\mathbf{y}_1,\mathbf{y}_2\sim p(\mathbf{x}_0,\mathbf{y}_1,\mathbf{y}_2)} \mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{x}_0)}[\lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0)\|^2] \tag{9}$$

However, these studies simply incorporate multiple conditions into the diffusion model without modifying the optimization objective, making it unclear whether $\hat{\mathbf{x}} \sim p(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$ is actually achieved. To address this, we derive the two-conditional optimization objective as follows.

First, applying the forward diffusion process in Eq. 1, obtains the perturbed distribution $p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{y}_2)$, according to [1], the corresponding reverse-time SDE is given by:

$$d\mathbf{x} = [f(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{y}_2)]dt + g(t)d\bar{\mathbf{w}} \tag{10}$$

By simulating Eq. 10, we can generate samples from $p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{y}_2)$. To construct the reverse-time SDE, we need to estimate the conditional score. Similar to Eq. 3, the training objective is:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim U(0,T)} \mathbb{E}_{\mathbf{x}_t,\mathbf{y}_1,\mathbf{y}_2\sim p(\mathbf{x}_t,\mathbf{y}_1,\mathbf{y}_2)}[\lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{y}_2)\|^2] \tag{11}$$

However, the $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{y}_2)$ in Eq. 11 is hard to access. [3] provided a method to approximate $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$. By generalizing it to multi-conditional setting, we prove that the optimal solution of Eq. 11 is the same as the solution of Eq. 9:

Proposition 1. *The solution that minimizes*
$\mathbb{E}_{t\sim U(0,T)}\mathbb{E}_{\mathbf{x}_0,\mathbf{y}_1,\mathbf{y}_2\sim p(\mathbf{x}_0,\mathbf{y}_1,\mathbf{y}_2)}\mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{x}_0)}[\lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0)\|^2]$
is the same as the solution minimizes
$\mathbb{E}_{t\sim U(0,T)}\mathbb{E}_{\mathbf{x}_t,\mathbf{y}_1,\mathbf{y}_2\sim p(\mathbf{x}_t,\mathbf{y}_1,\mathbf{y}_2)}[\lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{y}_2)\|^2]$

Proof. Let $f(\mathbf{x}_t, \mathbf{x}_0, \mathbf{y}_1, \mathbf{y}_2) := \lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0)\|^2$, first, according to the Law of Iterated Expectations, we have:

$$\begin{aligned}
&\mathbb{E}_{t\sim U(0,T)}\mathbb{E}_{\mathbf{x}_0,\mathbf{y}_1,\mathbf{y}_2\sim p(\mathbf{x}_0,\mathbf{y}_1,\mathbf{y}_2)}\mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{x}_0)}[\lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0)\|^2] \\
=&\mathbb{E}_{t\sim U(0,T)}\mathbb{E}_{\mathbf{y}_1,\mathbf{y}_2\sim p(\mathbf{y}_1,\mathbf{y}_2)}\mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0|\mathbf{y}_1,\mathbf{y}_2)}\mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{x}_0)}[f(\mathbf{x}_t, \mathbf{x}_0, \mathbf{y}_1, \mathbf{y}_2)] \\
=&\mathbb{E}_{t\sim U(0,T)}\mathbb{E}_{\mathbf{y}_2\sim p(\mathbf{y}_2)}\mathbb{E}_{\mathbf{y}_1\sim p(\mathbf{y}_1|\mathbf{y}_2)}\mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0|\mathbf{y}_1,\mathbf{y}_2)}\mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{x}_0)}[f(\mathbf{x}_t, \mathbf{x}_0, \mathbf{y}_1, \mathbf{y}_2)]
\end{aligned} \tag{12}$$

The $\mathbf{y}_1$ and $\mathbf{y}_2$ are independent of each other. Given $\mathbf{x}_0$, $\mathbf{y}_1$ and $\mathbf{y}_2$ are independent of $\mathbf{x}_t$. Let $g(\mathbf{x}_t, \mathbf{x}_0, \mathbf{y}_1, \mathbf{y}_2) := \lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}_1, \mathbf{y}_2)\|^2$, Eq. 12 can be written as:

$$\begin{aligned}
&\mathbb{E}_{t\sim U(0,T)}\mathbb{E}_{\mathbf{y}_2\sim p(\mathbf{y}_2)}\mathbb{E}_{\mathbf{y}_1\sim p(\mathbf{y}_1)}\mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0|\mathbf{y}_1,\mathbf{y}_2)}\mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{x}_0)}[f(\mathbf{x}_t, \mathbf{x}_0, \mathbf{y}_1, \mathbf{y}_2)] \\
=&\mathbb{E}_{t\sim U(0,T)}\mathbb{E}_{\mathbf{y}_2\sim p(\mathbf{y}_2)}\mathbb{E}_{\mathbf{y}_1\sim p(\mathbf{y}_1)}\mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0|\mathbf{y}_1,\mathbf{y}_2)}\mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{x}_0,\mathbf{y}_1,\mathbf{y}_2)}[g(\mathbf{x}_t, \mathbf{x}_0, \mathbf{y}_1, \mathbf{y}_2)]
\end{aligned} \tag{13}$$

Let t, $\mathbf{y}_1$ and $\mathbf{y}_2$ be arbitrary fixed values, then we can define $h(\mathbf{x}_t) := s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta})$, $q(\mathbf{x}_0) := p(\mathbf{x}_0|\mathbf{y}_1, \mathbf{y}_2)$ and $q(\mathbf{x}_t|\mathbf{x}_0) := p(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}_1, \mathbf{y}_2)$, applying the Law of Iterated Expectations, we have:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0|\mathbf{y}_1,\mathbf{y}_2)}\mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{x}_0,\mathbf{y}_1,\mathbf{y}_2)}[\lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0, \mathbf{y}_1, \mathbf{y}_2)\|^2] \\
=&\mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0)}\mathbb{E}_{\mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0)}[\lambda(t)\|h(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0)\|^2] \\
=&\mathbb{E}_{\mathbf{x}_t\sim q(\mathbf{x}_t)}[\lambda(t)\|h(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)\|^2] \\
=&\mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{y}_1,\mathbf{y}_2)}[\lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{y}_2)\|^2]
\end{aligned} \tag{14}$$

Since t, $\mathbf{y}_1$, $\mathbf{y}_2$ are arbitrary, Eq. 14 is true for all t, $\mathbf{y}_1$, $\mathbf{y}_2$, via Eq. 14 and the Law of Iterated Expectations, we can easily rewrite Eq. 13 as:

$$\begin{aligned}
&\mathbb{E}_{t\sim U(0,T)}\mathbb{E}_{\mathbf{y}_2\sim p(\mathbf{y}_2)}\mathbb{E}_{\mathbf{y}_1\sim p(\mathbf{y}_1)}\mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{y}_1,\mathbf{y}_2)}[\lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{y}_2)\|^2] \\
=&\mathbb{E}_{t\sim U(0,T)}\mathbb{E}_{\mathbf{x}_t,\mathbf{y}_1,\mathbf{y}_2\sim p(\mathbf{x}_t,\mathbf{y}_1,\mathbf{y}_2)}[\lambda(t)\|s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{y}_2)\|^2]
\end{aligned} \tag{15}$$

□

With this Proposition, we have established that the optimal solution $s(\mathbf{x}_t, \mathbf{y}_1, \mathbf{y}_2, t; \boldsymbol{\theta}^*)$ of Eq. 9 is able to approximate the multi-conditional score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{y}_2)$.

B. Algorithm

Algorithm 1 Bounding Boxes Arrangement

Input: Adjacency matrix A of bounding boxes, the number of bounding boxes n.
Output: Array $channels$ containing the assigned channel for each bounding box.

$channels \leftarrow$ array of length n initialized with 0
for $i = 1$ to n **do**
  $C_i \leftarrow \emptyset$
  for $j = 1$ to i **do**
    if $A[i][j] = 1$ and $channels[j] \neq 0$ **then**
      Add $channels[j]$ to C_i
    end if
  end for
  Assign the smallest channel not in C_i to $channels[i]$
end for
Return $channels$

C. Implementation details

Dataset preparation. In the GWHD, the images have high resolution but are relatively few in number. Therefore, we divided the original 1024×1024 images into 9 images of size 512×512 with step size 256. After splitting, there are a total of 58,635 images in GWHD. For the COCO 2017 dataset, we train with the official training set and test the proposed L2I method on the validation set.

Domain adaption. To evaluate the effectiveness of DODA for domain adaptation, we focus on the domains within the GWHD test set where AP_{50} lower than 0.8. For each domain, we use DODA to generate a 200 image dataset, then fine-tune the YOLOX-L on this synthetic data for one epoch.

L2I on COCO. Following the setting of [6, 7], we apply the proposed LI2I method to Stable Diffusion [39] v1.5. To preserve the knowledge learned from billions of images [44], we use the encoder of U-Net as the layout encoder, and following [64] initialize it with the weight of the diffusion model. Since Stable Diffusion is a T2I model, we constructed a simple text prompt for our method: "a photograph with $(N^1_{\text{cls}})(Cls^1), \ldots, (N^i_{cls})(Cls^i)$", where Cls^i is the category, and N^i_{cls} denotes the number of objects belonging to that category. Since COCO contains multiple categories, we design a layout coding method that different from Sec. 3.3.2, objects of the same category are depicted with the same hue but weaker brightness, and the bounding box of each object is drawn in descending order of area.

Hyperparameters. By default, we use 4 NVIDIAA-V100-32GB, but all models in this paper can be trained on one single V100, and the GPU Memory usage and approximate computational requirements for one GPU are provided in the last two rows of Table 9 and Table 10. When training with multiple cards, all parameters including Learning Rate are the same, except Iterations.

D. Evaluation metrics

Fréchet Inception Distance (FID) [19] reflects the quality of the generated image. FID measures similarity of features between two image sets and the features extracted by the pre-trained Inception-V3 [48].

Inception Score (IS) [41] uses a pre-trained Inception-V3 [48] to classify the generated images, reflecting the diversity and quality of the images. When calculating the IS for Table 3, as in the original paper, we divided the data into 10 splits. The error bar for IS is the standard deviation between the splits.

COCO Metrics refers to fine-tuning detectors using synthetic data, and then calculating AP according to the official COCO.

YOLO Score uses a pre-trained YOLOX-L [15] to detect the generated image, and calculates the AP between the detection result and the input label, which reflects the ability of the generated model to control the layout.

Feature Similarity (FS). As discussed in Sec.3.3.1, domain shift manifests in feature differences, the domain encoder should guide diffusion to generate images aligned with reference images' features. Here we use DINO-V2 [36] to extract features

Table 9. Hyperparameters for pre-training DODA. DODA leverages latent diffusion (LDM) [39] as the base diffusion model, which uses variational autoencoder (VAE) [25] to encode the image into the latent space and thus reduces the computation, so the pre-training of DODA is divided into two stages: the VAE and LDM.

		VAE	LDM
Dataset		All images in GWHD	All images in GWHD
Target Image Shape		$256 \times 256 \times 3$	$256 \times 256 \times 3$
Domain Reference Image Shape		-	$224 \times 224 \times 3$
Data Augmentation	Target Image	Random Rotation Random Crop Random Flip	Random Rotation Random Crop Random Flip
	Reference Image	-	Random Crop
f		4	4
Channels		128	224
Channel Multiplier		1,2,4	1,2,4
Attention Resolutions		-	2,4
Number of Heads		-	8
Learning Rate		2.5e-6	2.5e-5
Iterations		480k	600k
Batch Size		8	16
GPU Memory usage		32 GB	16 GB
Computational consumption		20 v100-days	14 v100-days

Table 10. Hyperparameters for layout-to-image.

Dataset		COCO 2017 training	COCO 2017 training	GWHD training
Target/Layout Image Shape		$256 \times 256 \times 3$	$512 \times 512 \times 3$	$256 \times 256 \times 3$
Domain Reference Image Shape		-	-	$224 \times 224 \times 3$
Data Augmentation	Target Image	Random Flip	Random Flip	Random Rotation Random Crop Random Flip
	Reference Image	-	-	Random Crop
Base Model		SD1.5	COCO 256	LDM in Table 9
f		8	8	4
Channels		320	320	224
Channel Multiplier		1,2,4,4	1,2,4,4	1,2,4
Attention Resolutions		1,2,4	1,2,4	2,4
Number of Heads		8	8	8
Learning Rate		2.5e-5	2.5e-5	1e-5
Iterations		100K	30K	80K
Batch Size		16	8	16
GPU Memory usage		27 GB	25 GB	20 GB
Computational consumption		40 v100-hours	56 v100-hours	40 v100-hours

from the generated images and their corresponding reference images, calculate the cosine similarity for each pair, and then compute the average similarity across multiple image pairs. Compared with FID, FS provides more fine-grained information.

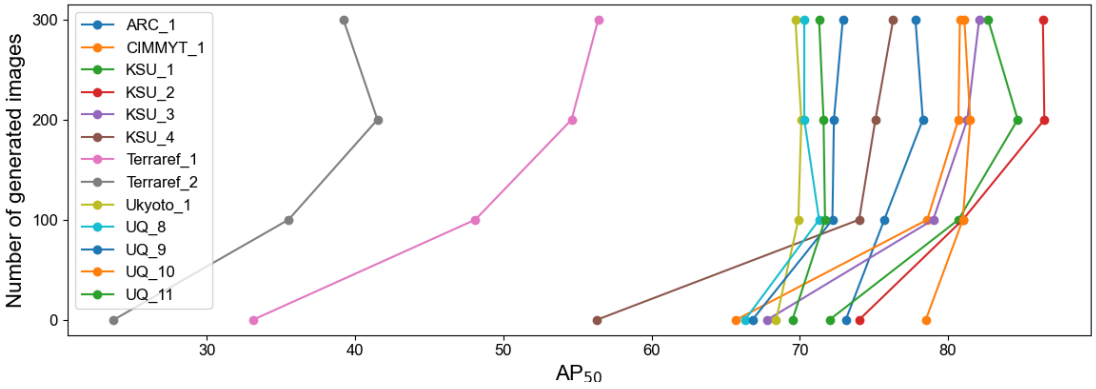

Figure 4. Ablations on the number of generated images. For most domains, 200 generated images are sufficient.

E. More Ablation

Channel coding. Table 11 evaluates the effectiveness of our channel coding component. By representing overlapping instances through different color channels, the model better distinguish overlapped instances and thus more accurately control the layout.

Reference image selection method. In the main experiments, we randomly sample reference images x_{ref} from the entire set of target domain images x. In this section, we investigate how the choice of reference images affects the generated data. We first sample different numbers of images from x to create reference pools x^i_{pool} of varying sizes, and then randomly sample x_{ref} from each x^i_{pool}. For each x^i_{pool}, we repeat the sampling process 5 times to compute the standard deviation. As shown in Table 12, when the reference pool is extremely small, the diversity of x_{ref} is low, resulting in low AP scores, and the standard deviation is large because the sampling bias is amplified. Once the size of the x^i_{pool} exceeds 100, the AP stabilizes.

Number of generated images. We investigate changes in the performance of using different amounts of generated data. As shown in Fig. 4, for most domains, a dataset consisting of 200 synthetic images is sufficient to convey the characteristics of the target domain. Increasing the number of images does not significantly improve performance. To ensure consistency across experiments, we use 200 images by default for all domains.

Table 11. Ablations on the layout channel coding. Channel coding can help the model more accurately control layout.

Channel coding	YOLO Score↑					
	mAP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l
✗	26.4	67.8	14.5	20.0	31.3	28.6
✓	27.4	70.0	15.3	20.8	32.7	29.9

Table 12. Ablations on the selecting of reference images.

References pool size	AP_{50}
0	30.5
10	37.22±8.89
100	48.00±1.92
400	49.36±1.55
1600	48.58±1.95

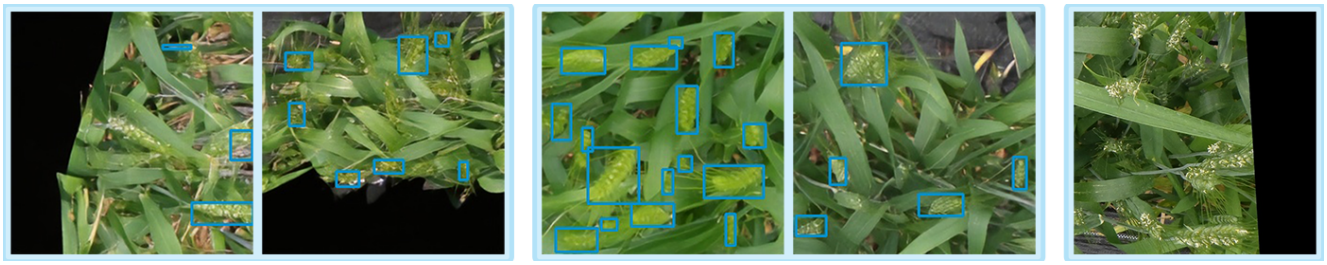

Figure 5. Examples of generated images in domain "Ukyoto_1". Left, many generated images have unnatural black edges. Middle, normal generated images, which are better aligned with the input layout (blue bounding boxes) than the images with black edges. Right, some real images used for pre-training also have black edges.

F. Influence of Noisy Samples in Pre-training Data

On the "Ukyoto_1" domain, the improvement in mAP is only marginal. As shown in Fig. 5 left, we observed many images with an unusual black area. Compared to images without black edges (Fig. 5 middle), these images also show poorer alignment with the given layout. The black regions and the poorer alignment degrade the quality of generated data. Upon further inspection, we found that these black regions originate from the real images used for pre-training (as illustrated in Fig. 5 right). When preparing the pre-training dataset, it necessary to filter out such images to improve data quality.

G. Qualitative Results of Agricultural Object Detection Domain Adaptation

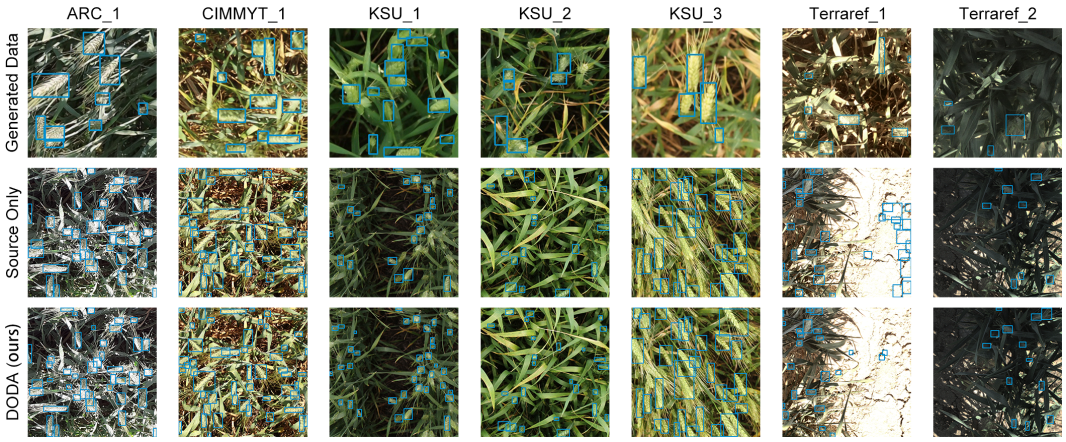

Figure 6. Visualization of generated data and detection results on target domains.

H. Qualitative comparisons with previous L2I methods on COCO

Figure 7. Visualization of comparisons between our proposed LI2I method and previous LT2I methods on COCO. LI2I generates images with more detail and greater control over layout, especially for small objects.