Supplementary for Learning Object Placement via Dual-path Graph Completion

Siyuan Zhou, Liu Liu, Li Niu, and Liqing Zhang

MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, China
{ssluvble,Shirlley,ustcnewly}@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn

In this supplementary file, we first introduce more implementation details of our GracoNet in Section 1. Then, we conduct extensive ablation studies in Section 2. We also provide additional visualizations to verify our method in Section 3. Finally, we discuss the limitations in Section 4.

1 Implementation Details

1.1 Image Pre-processing

All images are resized to 256×256 and normalized before they are fed into the network, *i.e.*, $H = 256$ and $W = 256$. Note that we maintain the relative aspect ratio between foreground and background before and after they are resized. For example, we use (fg^w, fg^h) and (bg^w, bg^h) to represent the original sizes of foreground and background, respectively. If $\mathrm{fg}^w/\mathrm{fg}^h > \mathrm{bg}^w/\mathrm{bg}^h$, we first resize foreground to $(256, (256 \cdot \mathrm{fg}^h \cdot \mathrm{bg}^w)/(\mathrm{fg}^w \cdot \mathrm{bg}^h))$, and then zero-pads it to $(256, 256)$ on the top side and bottom side evenly. If $\mathrm{fg}^w/\mathrm{fg}^h \leqslant \mathrm{bg}^w/\mathrm{bg}^h$, we first resize foreground to $((256 \cdot \mathrm{fg}^w \cdot \mathrm{bg}^h)/(\mathrm{fg}^h \cdot \mathrm{bg}^w), 256)$, and then zero-pads it to $(256, 256)$ on the left side and right side evenly. Meanwhile, background images and annotated composite images are directly resized to $(256, 256)$.

1.2 Transformation Function $\mathcal{F}_{\mathbf{t}}$ with Parameter $\mathbf{t}$

In our problem, we consider transformation parameters $\mathbf{t} = [t^r, t^x, t^y] \in \mathcal{R}^3$ with three degrees of freedom. We first define $t^r \in (0, 1)$ to represent the scaling ratio for the foreground object. After scaling, the height and the width of the new foreground region are $h = t^r H$ and $w = t^r W$. Since we do not change the aspect ratio of foreground after transformation, w and h are not independent. The scaled foreground object will be placed at a reasonable location (x, y) over the background, where (x, y) represents the background coordinate for the left top pixel point of the foreground region. We then define $t^x = \frac{x}{W-w} \in (0, 1)$ and $t^y = \frac{y}{H-h} \in (0, 1)$ to indicate the relative vertical and horizontal locations that the foreground object should be placed over the background scene. For the accessibility of back propagation for both $\mathbf{t}_u$ and $\mathbf{t}_s$ in our dual-path framework, we follow Spatial Transformer Network (STN) [2] to apply an affine transformation $\mathcal{A}$ with parameter Θ to transform the foreground region. With a simple derivation, the parameter Θ in our problem should be a function of $\mathbf{t}$:

$$\Theta(\mathbf{t}) = \begin{pmatrix} 1/t^r & 0 & (1-2t^x)(1/t^r-1) \\ 0 & 1/t^r & (1-2t^y)(1/t^r-1) \end{pmatrix}. \tag{1}$$

By applying affine transformation to foreground image I^{fg} and foreground mask M^{fg}, we obtain a transformed foreground image $\hat{\mathrm{I}}^{\mathrm{fg}} = \mathcal{A}(\mathrm{I}^{\mathrm{fg}}; \Theta)$ with a new mask $\mathrm{M}^{\mathrm{c}} = \mathcal{A}(\mathrm{M}^{\mathrm{fg}}; \Theta)$. The predicted composite image I^{c} is then calculated by $\mathrm{I}^{\mathrm{c}} = \mathrm{M}^{\mathrm{c}} * \hat{\mathrm{I}}^{\mathrm{fg}} + (1 - \mathrm{M}^{\mathrm{c}}) * \mathrm{I}^{\mathrm{bg}}$, in which $*$ means element-wise product. Since Θ is a function of $\mathbf{t}$, we could also describe I^{c} and M^{c} in function forms f^I and f^M conditioned on $\mathbf{t}$:

$$\begin{aligned} \mathrm{I}^{\mathrm{c}} &= f^I(\mathrm{I}^{\mathrm{bg}}, \mathrm{I}^{\mathrm{fg}}, \mathrm{M}^{\mathrm{fg}}; \mathbf{t}), \\ \mathrm{M}^{\mathrm{c}} &= f^M(\mathrm{M}^{\mathrm{fg}}; \mathbf{t}). \end{aligned} \tag{2}$$

Finally, our transformation function $\mathcal{F}_{\mathbf{t}}$ is defined by

$$\mathcal{F}_{\mathbf{t}}(\mathrm{I}^{\mathrm{bg}}, \mathrm{I}^{\mathrm{fg}}, \mathrm{M}^{\mathrm{fg}}) \triangleq (\, f^I(\mathrm{I}^{\mathrm{bg}}, \mathrm{I}^{\mathrm{fg}}, \mathrm{M}^{\mathrm{fg}}; \mathbf{t}),\ f^M(\mathrm{M}^{\mathrm{fg}}; \mathbf{t})\,). \tag{3}$$

As discussed in Section 3.2 in the main paper, given a labeled positive composite image $\mathrm{I}^{\mathrm{c}}_{\mathrm{pos}}$ with object mask $\mathrm{M}^{\mathrm{c}}_{\mathrm{pos}}$, we should calculate its ground-truth transformation parameters $\mathbf{t}_{\mathrm{gt}} = [t^r_{\mathrm{gt}}, t^x_{\mathrm{gt}}, t^y_{\mathrm{gt}}]$ for calculating reconstruction loss $\mathcal{L}^{rec}_s$. The procedure of obtaining $\mathbf{t}_{\mathrm{gt}}$ from annotation derives from the definition of transformation parameters, as described in the following. We first obtain the bounding box of the foreground region on $\mathrm{I}^{\mathrm{c}}_{\mathrm{pos}}/\mathrm{M}^{\mathrm{c}}_{\mathrm{pos}}$, denoted by $(x_{\mathrm{gt}}, y_{\mathrm{gt}}, w_{\mathrm{gt}}, h_{\mathrm{gt}})$. Then, the ground-truth transformation parameters $\mathbf{t}_{\mathrm{gt}}$ are calculated by $t^r_{\mathrm{gt}} = \max(\frac{w_{\mathrm{gt}}}{W}, \frac{h_{\mathrm{gt}}}{H})$, $t^x_{\mathrm{gt}} = \frac{x_{\mathrm{gt}}}{W - w_{\mathrm{gt}}}$, and $t^y_{\mathrm{gt}} = \frac{y_{\mathrm{gt}}}{H - h_{\mathrm{gt}}}$.

1.3 Reconstruction Loss $\mathcal{L}^{rec}_s$

Since we expect the supervised path to reconstruct $(\mathrm{I}^{\mathrm{c}}_s, \mathrm{M}^{\mathrm{c}}_s)$, the reconstruction loss $\mathcal{L}^{rec}_s$ is designed to force $\mathbf{t}_s$ to be close to the ground-truth $\mathbf{t}_{\mathrm{gt}}$. We define $\mathcal{L}^{rec}_s$ as a weighted mean squared error (*i.e.*, Weighted MSE) between $\mathbf{t}_s$ and $\mathbf{t}_{\mathrm{gt}}$:

$$\mathcal{L}^{rec}_s = \frac{\alpha^r (t^r_s - t^r_{\mathrm{gt}})^2 + \alpha^x (t^x_s - t^x_{\mathrm{gt}})^2 + \alpha^y (t^y_s - t^y_{\mathrm{gt}})^2}{3} \tag{4}$$

with weight $\boldsymbol{\alpha} = [\alpha^r, \alpha^x, \alpha^y]$. Specifically, we adopt dynamic weight in our implementation, where $\boldsymbol{\alpha}$ can be described as a set of functions determined by variable t^r_s, that is, $\alpha^r = f^r(t^r_s)$, $\alpha^x = f^x(t^r_s)$, and $\alpha^y = f^y(t^r_s)$. $f^r(t^r_s)$ should be a monotonically increasing function and $f^x(t^r_s), f^y(t^r_s)$ should be monotonically decreasing functions when $t^r_s \in (0, 1)$. The reason for this design is intuitive. When t^r_s is small and close to 0, we pay more attention to where the foreground object should be placed, instead of the scale of the foreground region. Conversely, when t^r_s is large and close to 1, the relative vertical and horizontal location t^x_s and t^y_s become less important and now the scale of foreground object becomes our main concern. Concretely, we define $\boldsymbol{\alpha}$ in the form of trigonometric functions: $\alpha^r = \sin(\frac{\pi}{2} t^r_s)$, $\alpha^x = \cos(\frac{\pi}{2} t^r_s)$, and $\alpha^y = \cos(\frac{\pi}{2} t^r_s)$. We have also explored other variants of reconstruction loss and compared them with our choice in Section 2.3.

1.4 Evaluation Metrics

As introduced in Section 4.1 in the main paper, we adopt user study, accuracy, and FID [1] to evaluate generation plausibility, and adopt LPIPS [6] to evaluate generation diversity during inference. In the following, we will discuss about more details in these four metrics.

User Study. The user study is conducted with 20 voluntary participants. We compare the object placement generation results of TERSE, PlaceNet, and our proposed method. For a given pair of foreground and background during inference (*i.e.*, a test sample), each method produces one composite image. Then, each participant chooses the most reasonable one from these three composite images. Each method is then scored by the proportion of participants who choose it (*w.r.t.* this test sample). Finally, we average the score among all test samples to obtain the user study score for each method.

Accuracy. We extend SimOPA [3] model to check the accuracy of object placement generation results. We omit the details of extended model here. The extended model functions as a binary classifier that distinguishes between reasonable and unreasonable object placements. We define accuracy as the proportion of the generated composite images that are classified as positive by the binary classifier during inference. We have released the code and model of binary classifier for evaluation.

FID. Fréchet Inception Distance (FID) [1] is a measure of similarity between two datasets of images. It was shown to correlate well with human judgement of visual quality and is most often used to evaluate the quality of samples of Generative Adversarial Networks. We calculate FID score between one set of composite images generated by the network and another set of ground-truth positive composite images in the OPA *test* set.

LPIPS. In Generative Adversarial Networks, LPIPS [6] is commonly used to measure the perceptual similarity between two images. In this work, we adopt LPIPS to measure the generation diversity of models. For a given pair of foreground and background during inference (*i.e.*, a test sample), we generate 10 different composite images by sampling the random vector for 10 times. We first compute LPIPS for all pairs of composite images among 10 generation results for each test sample, and then calculate the averaged LPIPS among all test samples. Since LPIPS reveals the difference between two images, a larger LPIPS score corresponds to a better generation diversity.

2 Ablation Studies

2.1 Degree of Annotation.

Table 1 shows an ablation study on the degree of annotation we use. Without the supervised path, the model witnesses a sharp decrease in performance and falls

Table 1. Ablation study on degree of annotation

Annotation Degree	*Plausibility*		*Diversity*
	acc.↑	FID↓	LPIPS↑
$\mathcal{P}_u$	0.637	68.17	0
$\mathcal{P}_u + \mathcal{L}_s^{cls}$	0.754	34.80	0.130
$\mathcal{P}_u + \mathcal{P}_s$	0.847	27.75	0.206

Table 2. Ablation study on using negative training samples

Method	Pos	Neg	*Plausibility*		*Diversity*
			acc.↑	FID↓	LPIPS↑
TERSE [4]	✓		0.588	49.35	0
	✓	✓	0.679	46.94	0
PlaceNet [5]	✓		0.619	32.50	0.101
	✓	✓	0.683	36.69	0.160
GracoNet	✓		0.808	27.15	0.206
	✓	✓	0.847	27.75	0.206

into mode collapse. After we add the classification loss $\mathcal{L}_s^{cls}$ to assist with the discriminator, the model works better in plausibility and relieves mode collapse. Adopting the complete supervised path brings another performance jump in all metrics. These experiments prove that fully exploiting supervision is crucial in object placement learning. Our supervised path is just designed under this guidance. By constructing a bijection between the latent vector and the predicted composite image, the model could effectively overcome the mode collapse problem. The supervised path successfully guides the unsupervised path to generate more reasonable and diversified object placements.

2.2 Using Negative Samples for Training

As introduced in Section 1 in the main paper, OPA dataset [3] is the first object placement assessment dataset that contains composite images and their binary rationality labels indicating whether they are reasonable (positive sample) or not (negative sample) in terms of foreground object placement. As discussed in Section 4.2 in the main paper, baseline TERSE [4] and PlaceNet [5] did not include negative samples in their method because they had been proposed before OPA dataset was released. For most experiments on OPA dataset, we fairly use positive samples and negative samples together for both baselines and our method (*e.g.*, experiments in Section 4.2 in the main paper). In this section, we aim to investigate whether introducing negative samples is necessary or not.

Table 2 shows an ablation study on whether to use negative training samples for different methods in the training stage. As illustrated, accuracy drops without the assistance of negative samples in all methods. It is because simultaneously using positive samples and negative samples balances the process of adversarial training and enables the discriminator to learn from a wider range of data

Table 3. Ablation study on different reconstruction losses

Type	α^r	α^x & α^y	*Plausibility*		*Diversity*
			acc.↑	FID↓	LPIPS↑
L1	1	1	0.820	29.38	0.063
L2	1	1	0.833	29.27	0.069
L2	t_s^r	$1 - t_s^r$	0.836	27.92	0.190
L2	$\sin(\frac{\pi}{2}t_s^r)$	$\cos(\frac{\pi}{2}t_s^r)$	0.847	27.75	0.206

distribution. By comparing different methods, we find that removing negative samples from training process has the largest impact on TERSE (about 0.09 accuracy drop) and the smallest impact on our method (about 0.04 accuracy drop). This proves the robustness of our method because only our method still performs reasonably with the absence of negative samples.

2.3 Different Types of Reconstruction Losses

As discussed in Section 1.3, we use a weighted MSE with trigonometric dynamic weights as the reconstruction loss $\mathcal{L}_s^{rec}$ between $\mathbf{t}_s$ and $\mathbf{t}_{\mathrm{gt}}$. In Table 3, we explore different types of reconstruction losses, including L1-loss, L2-loss, L2-loss with linear dynamic weights, and L2-loss with trigonometric dynamic weights (our method). L1-loss and L2-loss both perform badly in generation diversity, because the model can not pay more attention to learning location (*resp.*, size) information when t_s^r is small (*resp.*, large). L2-loss with linear/trigonometric dynamic weights overcomes this weakness and dynamically changes its attention during training. Comparably, trigonometric weights work slightly better than linear weights, so the former type becomes our final choice.

2.4 Advantages Against Baselines

TERSE, PlaceNet, and our method all adopt an adversarial training strategy. The generator functions to produce transformation parameters that reasonably places the foreground object over the background scene to form a composite image. The discriminator works by distinguishing between reasonable composite images and unreasonable composite images. The most important difference between baselines and our method lies in two aspects: the generator design and the usage of positive composite images.

Generator Design. TERSE uses a shared backbone followed by two separate branches to encode heterogeneous features for foreground and background. Then these two kinds of features are concatenated and regressed to predict transformation parameters. PlaceNet uses two independent encoders to extract features for foreground and background, respectively. Foreground and background features are concatenated with different random vectors to predict various object placements via a shared decoder network. Meanwhile, a diversity loss is designed to preserve

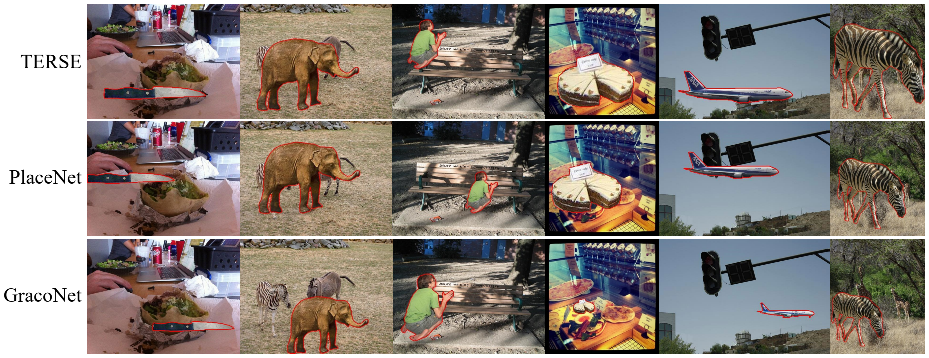

Fig. 1. Visualization of object placement results for different foreground objects and background scenes on OPA *test* set. Foreground is outlined in red

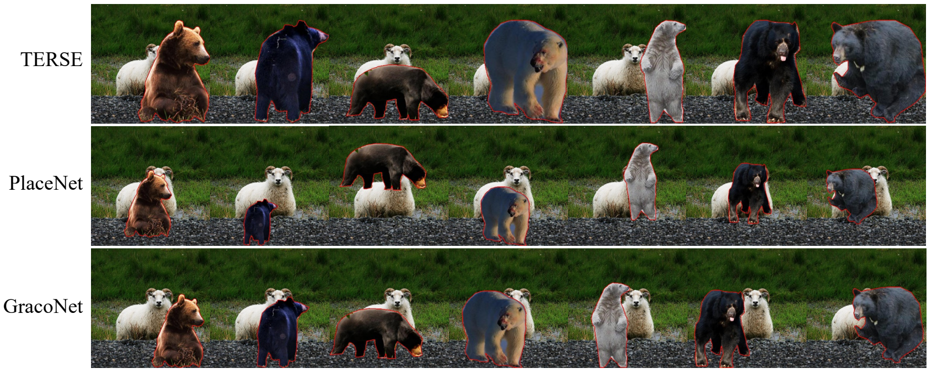

Fig. 2. Visualization of object placement results for the same background scene with different foreground objects on OPA *test* set. Foreground objects are outlined in red

the pairwise distance between the predicted placements and the corresponding random vectors. Compared with generators in these baselines, our proposed generator contains a novel GCM module that treats the object placement task as a graph completion problem. The background is considered as different nodes with different locations/sizes, whereas the foreground is considered as unique node lacking for location/size. GCM aims to reasonably place the foreground node among different background nodes to complete the graph. As introduced in Section 3.1 in the main paper, GCM mainly consists of Node Extraction Head (NEH) and Placement Seeking Network (PSN). We have investigated the functionality of NEH and PSN via an ablation study in Table 4 in the main paper, which shows that both NEH and PSN are crucial in GCM. Besides, according to Table 1, without using the supervised path, our simplified version $\mathcal{P}_u + \mathcal{L}_s^{cls}$ containing GCM can already beat TERSE and PlaceNet in generation plausibility, which proves the advantage of our GCM design.

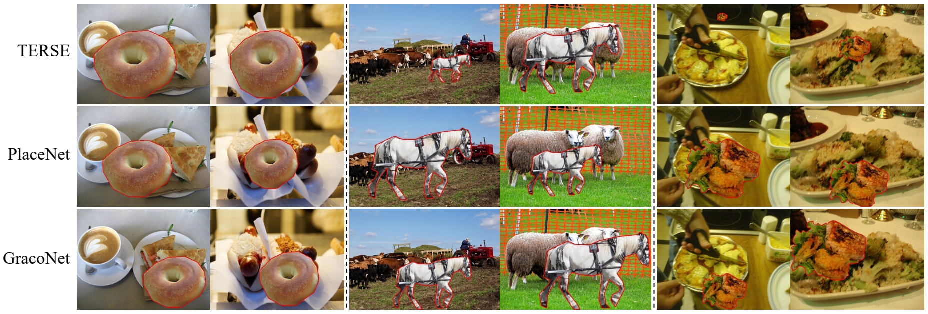

Fig. 3. Visualization of object placement results for the same foreground object with different background scenes on OPA *test* set. Foreground objects are outlined in red

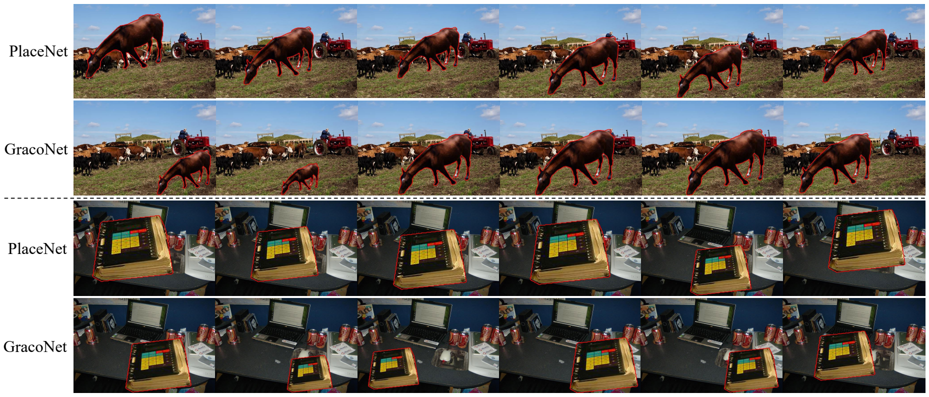

Fig. 4. Visualization of object placement diversity on OPA *test* set by sampling different random vectors. Foreground objects are outlined in red

Usage of Positive Composite Images. Baseline methods simply use positive composite images to train the discriminator, which wastes a lot of useful information. In contrast, we introduce a dual-path framework to effectively investigate the positive composite images. As introduced in Section 3.2 in the main paper, we establish a bijection between the latent vector and the predicted object placement in the supervised path. Specifically, we draw information from positive composite images to predict the latent vector, which is then utilized to reconstruct the positive composite image via a regression block and a transformation function. Since the two paths share weights in most network layers, the supervised path could gradually guide the unsupervised path in the training stage. Under this design, our model successfully discovers the underlined object placement knowledge and produces satisfactory composite images during inference. By comparing $\mathcal{P}_u + \mathcal{L}_s^{cls}$ and $\mathcal{P}_u + \mathcal{P}_s$ in Table 1, we can see that both generation plausibility and diversity are greatly enhanced after adding the supervised path, which demonstrates the power of supervised path in utilizing the positive composite images.

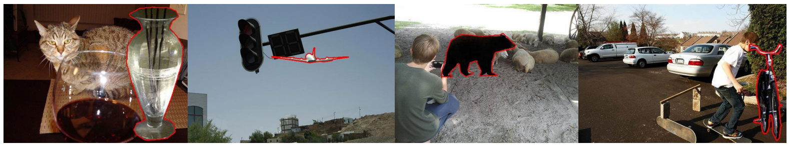

Fig. 5. Failure cases in terms of occlusion between foreground and background

3 Visualization of Object Placement

Figure 1, Figure 2, Figure 3, and Figure 4 visualize more object placement results for different methods on OPA *test* set, which supplement Figure 3 in the main paper. On the one hand, Figure 1, Figure 2 and Figure 3 focus on generation plausibility of TERSE [4], PlaceNet [5], and our method. Figure 1 displays the object placement results from different foreground objects and different background scenes. Figure 2 displays the combination of an identical background scene with different foreground objects. Figure 3 displays the combination of an identical foreground object with different background scenes. From these three figures, we find that our method not only adapts better to different foreground objects with reasonable locations and sizes conditioned on a given background, but also predicts more robust foreground objects under diverse background scenes.

On the other hand, Figure 4 shows the generation diversity of PlaceNet [5] and our method by sampling different random vectors conditioned on the same pair of foreground and background. Our method outperforms PlaceNet by discovering more possible reasonable locations on the background, as well as changing spatial sizes of foreground correspondingly. In contrast, PlaceNet tends to meet mode collapse problems in some situations. These visualized examples effectively prove that our method simultaneously achieves better generation plausibility and diversity, and reaches a satisfactory balance between these two aspects.

4 Discussion on Limitation

In object placement, an important concern is occlusion between foreground and background. The generated composite images of a satisfactory object placement model should be reasonable when the foreground object is placed over the background scene. In TERSE and PlaceNet, we find that the foreground sometimes covers a counterintuitive background region, *e.g.*, a sandwich wrongly covers the hand in row 2 and column 5 of Figure 3. In our method, this phenomenon has been alleviated, but it still exists in some generated composite images, as shown in Figure 5. This is probably because our model lacks a module that explicitly detects the occlusion relationship between foreground and background. GCM leverages the graph completion strategy to deal with the occlusion problem in an implicit way, which makes it surpass the baseline methods. However, incorporating a more explicit module to specifically address this problem may lead to better results. This provides a guiding direction for the optimization of future models on object placement.

References

1. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017)
2. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. NIPS (2015)
3. Liu, L., Zhang, B., Li, J., Niu, L., Liu, Q., Zhang, L.: OPA: Object placement assessment dataset. arXiv preprint arXiv:2107.01889 (2021)
4. Tripathi, S., Chandra, S., Agrawal, A., Tyagi, A., Rehg, J.M., Chari, V.: Learning to generate synthetic data via compositing. In: CVPR (2019)
5. Zhang, L., Wen, T., Min, J., Wang, J., Han, D., Shi, J.: Learning object placement by inpainting for compositional data augmentation. In: ECCV (2020)
6. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)