This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

TopNet: Transformer-based Object Placement Network for Image Compositing

Sijie Zhu[1], Zhe Lin[2], Scott Cohen[2], Jason Kuen[2], Zhifei Zhang[2], Chen Chen[1]
[1]Center for Research in Computer Vision, University of Central Florida [2]Adobe Research
sizhu@knights.ucf.edu,{zlin,scohen,kuen,zzhang}@adobe.com,chen.chen@crcv.ucf.edu

Abstract

We investigate the problem of automatically placing an object into a background image for image compositing. Given a background image and a segmented object, the goal is to train a model to predict plausible placements (location and scale) of the object for compositing. The quality of the composite image highly depends on the predicted location/scale. Existing works either generate candidate bounding boxes or apply sliding-window search using global representations from background and object images, which fail to model local information in background images. However, local clues in background images are important to determine the compatibility of placing the objects with certain locations/scales. In this paper, we propose to learn the correlation between object features and all local background features with a transformer module so that detailed information can be provided on all possible location/scale configurations. A sparse contrastive loss is further proposed to train our model with sparse supervision. Our new formulation generates a 3D heatmap indicating the plausibility of all location/scale combinations in one network forward pass, which is $> 10\times$ faster than the previous sliding-window method. It also supports interactive search when users provide a pre-defined location or scale. The proposed method can be trained with explicit annotation or in a self-supervised manner using an off-the-shelf inpainting model, and it outperforms state-of-the-art methods significantly. User study shows that the trained model generalizes well to real-world images with diverse challenging scenes and object categories.

1. Introduction

Object compositing [15, 25] is a common and important workflow for image editing and creation. The goal is to insert an object from an image into a given background image such that the resulting image appears visually pleasing and realistic. Conventional workflows in object compositing rely on manual object placement, *i.e*. manually determining where the object should be placed (location) and in what size the object is placed (scale). However, manual placement does not fulfill the growing need for image creation for social sharing, advertising, education, etc., and AI-assisted compositing with automatic object placement is more desirable for future image creation applications. While there have been several works on learning-based object placement for specific scenes, general object placement with diverse scenes and objects still remains challenging with limited exploration, as it involves a deeper understanding of common sense, objects, and local details of scenes. Inaccurate object placement could lead to poor compositing results, *e.g*. a person floating in the sky, a dog larger than buildings, etc.

Existing works [7,10,12,26,29] formulate the problem in very different ways, as shown in Fig. 1. [10,26] directly predict multiple transformations or bounding boxes indicating the location and scale of the given objects. Such sparse predictions recommend the top candidate placements for users, but they do not provide any information about other possible locations and scales. They also fail to leverage the local clues in background images, as the bounding boxes are generated based on only global features. Another thread of works [12, 24] considers object placement as binary classification, which evaluates the plausibility of input images and placement of bounding boxes instead of generating candidate placements directly from input images. One recent work [29] utilizes a retrieval model to assess the plausibility of a given placement and evaluates a grid of locations and scales in a sliding-window manner. However, it requires multiple network forward passes to generate dense evaluation for one image, resulting in a slow inference speed.

In this paper, we propose TopNet, a Transformer-based Object Placement Network for real-world object compositing applications. Different from previous works, TopNet formulates object placement as a dense prediction problem: *generating evaluation for a dense grid of locations and scales in one network forward pass.* Given a background image and an object, TopNet directly generates a 3D heatmap indicating the plausibility score of object location and scale, which is $> 10\times$ faster than the previous sliding-window method [29]. Previous works [26, 29] com-

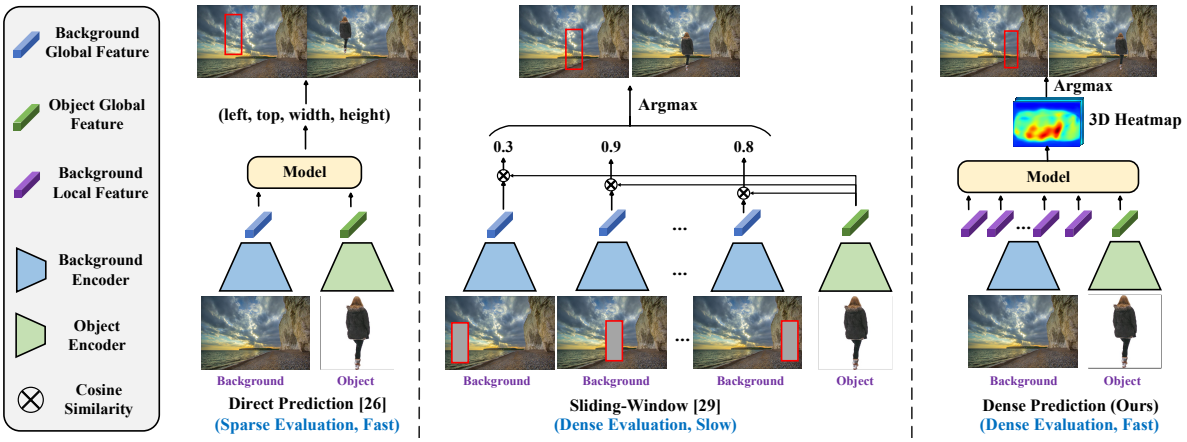

Figure 1. Comparison between different formulations for object placement. Our method provides a dense evaluation of possible locations/scales in one network forward pass.

bine background and foreground object features only at the global level, which fails to capture local clues for determining the object location. We propose to learn the correlation between global foreground object feature and local background features with a multi-layer transformer, leading to a more efficient and accurate evaluation of all possible placements. To train TopNet with sparse supervision where only one ground-truth placement bounding box is provided, we propose a sparse contrastive loss to encourage the ground-truth location/scale combination to have a relatively high score, while only minimizing the other combinations with the lowest score or a score higher than the ground-truth with a certain margin, thus preventing large penalty on other reasonable locations/scales. Once the 3D heatmap is predicted, top candidate placement bounding boxes can be generated by searching the local maximum in the 3D heatmap. The 3D heatmap also provides guidance for other possible locations/scales which are not the best candidate. Experiments on a large-scale inpainted dataset (Pixabay [1]) and annotated dataset (OPA [12]) show the superiority of our approach over previous methods. Our contributions are summarized as follows:

- A novel transformer-based architecture to model the correlation between object image and local clues from the background image, and generate dense object placement evaluation $>$ $10\times$ faster than previous sliding-window method [29].
- A sparse contrastive loss to effectively train a dense prediction network with sparse supervision.
- Extensive experiments on a large-scale inpainted dataset and annotated dataset with state-of-the-art performance.

2. Related Work

Object Placement Prediction. There are several works learning to directly predict object placement as a bounding box or transformation. Tan *et al*. [19] propose to predict bounding boxes for person objects using background and layout images. Then retrieval is applied to find a specific person object for compositing. ST-GAN [10] models the compositing realism in geometric wrap parameter space and learns to predict geometric transformation using adversary training. Similarly, Tripathi *et al*. [22] train a synthesizer network to predict transformation along with a discriminator to tell if the composite image is real. Compositional-GAN [2] designs a self-consistent network for compositing so that composite images can be decomposed back into individual objects. Li *et al*. [9] focus on indoor scenes by simultaneously learning location and plausible human poses. Lee *et al*. [7] propose to predict object masks for certain categories and then determine suitable object instances to be inserted. PlaceNet [26] is close to our setting as it can also be trained on inpainted background images with the original foreground object. It predicts bounding boxes with global features and random inputs and trains a discriminator to determine the plausibility of the bounding boxes along with global features of background and object images. These works either deal with street scenes with limited object categories (*e.g*. persons, vehicles, traffic lights) or indoor scenes with human pose joints or furniture. *We focus on real-world object placement for general compositing with diverse scenes and object categories, which is more challenging and not well studied in this field.*

Compositing Evaluation. Several recent works focus on object placement evaluation for general compositing with diverse scenes and object categories. OPA [12] proposes to train a binary classifier to determine whether a certain placement is realistic using both the composite image and placement mask. It is trained with human annotation on a subset of COCO dataset. GALA [29] trains a retrieval network to find the best object given a background image with placement bounding box, which could be considered as a single evaluation on certain placement. It is trained with

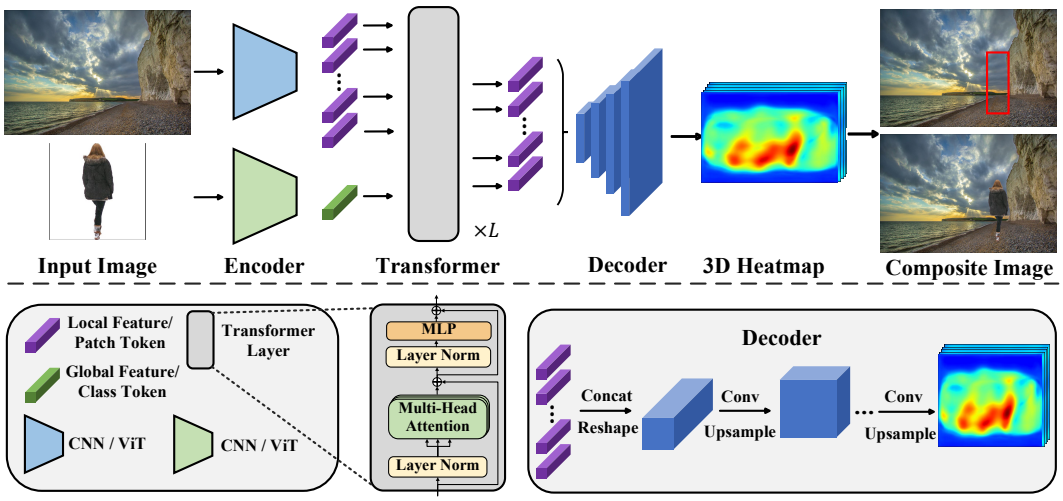

Figure 2. An overview of the proposed TopNet. Local background features and global object feature are extracted with two different encoders and fed into a transformer module to learn the correlation. A CNN-based upsampling decoder is then adopted to generate a 3D heatmap. Top-1/top-5 bounding boxes can be generated by finding global/local minimum in the 3D heatmap.

contrastive learning on the masked background and the corresponding foreground objects without human annotation. GALA further demonstrates that such a single evaluation can be extended with a sliding-window grid search to predict object placement. Although it is annotation-free and performs well on small-scale datasets, its inference time is proportional to the search space which could be large in practical scenarios. *We tackle this problem efficiently by generating dense evaluation in one network forward pass.*

Object Detection/Segmentation. Objection detection [20] and segmentation [5,8,17] are related because they also generate dense prediction with bounding boxes or heatmaps. Recent segmentation methods [8, 28] generally follow the design of an encoder-decoder framework, where the encoder extracts high-level features from images and the decoder applies convolutional operations and upsampling to generate a segmentation map with high resolution. *However, these works cannot be directly applied to our problem.* First, detection and segmentation takes one background image as input, while our task has another object image as input. Second, detection and segmentation have dense supervision, *i.e.* ground-truth bounding boxes for all objects or pixel-level segmentation masks, but our task usually has only one ground-truth bounding box as sparse supervision. *The proposed novel architecture and loss function are specifically designed to tackle these two issues.*

3. Methodology

3.1. Formulation

Given a background image $I_b \in \mathbb{R}^{h_b \times w_b \times 3}$ and a foreground object image $I_o \in \mathbb{R}^{h_o \times w_o \times 3}$, typical object placement is represented as a bounding box $[l, t, w, h]$, indicating the left, top, width, and height of the object in final composite image $I_c \in \mathbb{R}^{h_b \times w_b \times 3}$. Since the aspect ratio of the object image is known, we assume that the aspect ratio of the object is kept as w_o/h_o so that the scale can be defined as one number $s = \sqrt{wh/w_b h_b}$. Then the dense evaluation model predicts a 3D heatmap $H \in \mathbb{R}^{h_b \times w_b \times c}$, where c is the number of pre-defined scales, indicating the plausibility of each location-scale combination. Each spatial location in the heatmap corresponds to the center of a placement bounding box. By default, we use $c = 16$ and each channel represents a scale value from 0.15 to 0.9 with an interval of 0.05, according to the distribution of detectable objects in their original background images.

During inference, we first normalize H with maximum and minimum value as $\hat{H} = (H - min(H))/(max(H) - min(H))$. The top-1 prediction can be easily generated as $\arg\max(\hat{H})$, the global maximum location in the 3D heatmap. Top-k candidate predictions can be generated by finding local maximum peaks where the heatmap value is larger than a threshold. We use double standard deviation over average value as the threshold in the experiments.

3.2. Architecture

Global and Local Features. As shown in Fig. 2, we adopt two encoder networks to learn different features for background and object images. The encoder could be CNN [6] (Convolutional Neural Network) or ViT [4] (Vision Transformer). To determine whether a specific location is suitable for the object with a certain scale, the local clues in background images could provide detailed information. We thus keep all the local features/tokens from the last convolutional or transformer layer of the background encoder. For foreground objects, the image is relatively simple, so

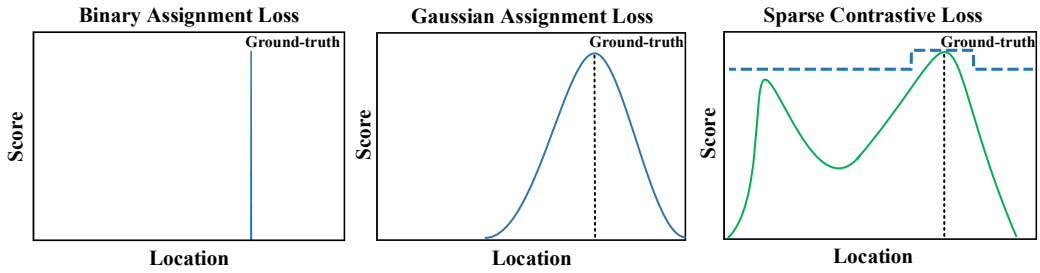

Figure 3. An example of different losses in 1D space. The solid blue curve denotes actual assignment, while the dash blue curve denotes upper bound assignment. The green solid curve illustrates one possible multi-peak score distribution, which is not allowed by the other assignment losses.

we only keep the global feature (*i.e*. the last feature before classification head), which encodes high-level information.

Background and Object Correlation. To learn the correlation between the background local features and global feature of the object image, we adopt a multi-layer transformer [23] module, which has been shown to model strong global attention well in vision tasks [4, 21]. We treat the background features as patch tokens [4] in ViT and add 2D version of sinusoidal positional embedding [23] accordingly. The class token is then replaced with the global feature of the object image. We take all the patch tokens of the last layer as output to feed into the upsampling decoder. A 1×1 convolutional layer is applied to the input features to convert the feature dimension to the dimension of transformer module. The inner architecture of a transformer layer is shown in Fig. 2, and the key component is the multi-head self-attention module. In a basic self-attention module, each input token is first converted into query, key and value, denoted as Q, K, V with dimension d, with three learnable linear projections. Then attention map is computed as $softmax(QK^T/d)V$. A multi-head attention module runs multiple basic self-attention modules in parallel and concatenates the outputs from all the heads.

Upsampling Decoder. Given the input background image with size of $h_b \times w_b \times 3$, there are typically $\frac{h_b}{2^k} \frac{w_b}{2^k}$ local features with dimension d, where 2^k is the downsampling ratio. As shown in Fig. 2, they are first concatenated and reshaped into size of $\frac{h_b}{2^k} \times \frac{w_b}{2^k} \times d$. Similar to the pyramid architecture in image segmentation [28], the decoder then applies k convolutional layers where each one is followed by a $2\times$ spatial upsampling block. Each convolution layer has a kernel of 3×3 and reduces the dimension by $2\times$. The last layer has an output dimension of c, which is the number of pre-defined scales. We set k as 4 in our experiments.

3.3. Loss Function

The main challenge for dense location and scale evaluation is the sparse supervision signal. Small-scale annotation in CAIS [27] or OPA [12] only provides one positive placement bounding box for each sample, without supervision on other locations/scales. For large-scale datasets without explicit annotation, one way to generate supervision is to mask out the original objects in background images and generate pure background images using off-the-shelf inpainting models. Then bounding box of the original object can be considered as the ground-truth placement, but the supervision is still sparse – only one location and scale.

One simple idea to supervise the model is to assign a ground-truth score for each location-scale combination, *i.e*. each data point in the 3D heatmap. Simple binary assignment considers the only ground-truth combination (GT data point) as 1, and all other location-scale combinations as 0. A smoother assignment is Gaussian assignment, which gives the score according to the distance between each data point and the ground-truth point in the 3D space. It considers locality of the score, *i.e*. locations/scales close to the ground-truth should still be good placement candidates. These assignments consider all locations/scales far away from the ground-truth as negative points with low scores. However, such an assumption does not hold in most cases. Given a specific background scene, certain objects could be compatible at many locations with different scales. One could drag the object in Fig. 4 along the ground toward the camera direction and increase the scale accordingly. However, considering local clues like light and aesthetic judgment, some location-scale combinations are still better than others. We thus assume that multiple good candidate bounding boxes exist in a background image, and propose to maximize the score at the ground-truth location/scale while allowing local peaks with high scores in other locations/scales. One toy example of different losses in 1D space is shown in Fig. 3. Assume that the ground-truth coordinate in the 3D heatmap $H \in \mathbb{R}^{h_b \times w_b \times c}$ is (x_{gt}, y_{gt}, z_{gt}). The first loss term is formulated as:

$$\mathcal{L}_{con} = \sum_{(x,y,z)} |H(x,y,z) - H(x_{gt}, y_{gt}, z_{gt}) + M(x,y,z)|^+, \quad (1)$$

where $(x, y, z) \in \mathbb{R}^{h_b \times w_b \times c}$ and $|\cdot|^+$ means $max(\cdot, 0)$. $M \in \mathbb{R}^{h_b \times w_b \times c}$ is the margin matrix, indicating how much $H(x_{gt}, y_{gt}, z_{gt})$ should be higher than $H(x, y, z)$ for any (x, y, z). The margin is set as 0 for the neighborhood of ground-truth location/scale if $|x - x_{gt}| \leq 20, |y - y_{gt}| \leq$

Figure 4. Example inpainted background images. If there are less than 3 objects, we add one additional randomly shifted object mask to prevent the model from learning inpainting artifacts.

$20, |z - z_{gt}| \leq 2$. Otherwise, it is empirically set as 0.1. We then apply the second term as:

$$\mathcal{L}_{range} = |1 - H(x_{gt}, y_{gt}, z_{gt})| + |min(H)|, \quad (2)$$

so that the scores fall in $[0, 1]$. It also encourages the lowest score to be 0, as there is always a bad location or scale for certain background and object images. This term prevents the model from predicting a high score for all locations/scales. The overall spare contrastive loss is defined as the summation of two terms $\mathcal{L} = \mathcal{L}_{con} + \mathcal{L}_{range}$.

4. Experiment

4.1. Datasets and Implementation Detail

Pixabay [1] is a dataset collected from "pixabay.com", a free stock image site. It contains millions of high-quality, diverse, free-to-use photos which are perfect for building and evaluating models for real-world object compositing. We follow [29] to collect $928,018$ images and apply object detection [20] and segmentation [8] to generate foreground objects and background images. This results in $5,771,912$ objects and $928,018$ background images. We then filter out tiny objects that are unlikely to be used for composting, and background images with overly large object mask where no background information is left. We keep objects with high confidence detection scores and proper bounding box sizes, resulting in $833,964$ foreground and background pairs with 914 non-zero categories. Finally, they are randomly split into training/evaluation sets with $90\%/10\%$ ratio.

Since there is no annotation for this large-scale dataset, we generate pure background images by removing objects from each image. The original bounding box of the object would always be a good placement candidate and we consider it as ground-truth. We adopt the off-the-shelf model of LAMA [18] for inpainting. For images with three or more objects, we randomly select three object masks for inpainting. If there are fewer than three objects, we use all the object masks and add an additional mask by randomly shifting one of the object masks. In this way, multiple locations contain inpainting artifacts, which reduces the risk of model overfitting to inpainting artifacts. As shown in Fig. 4, the inpainting model [18] works well on recovering the background sea and grass. Furthermore, we observe that images with too many objects have low inpainting quality due to strong occlusion. Therefore, we remove the images with more than 5 objects, resulting in $367,384$ pairs for training and $41,166$ pairs for evaluation. Each pair contain one inpainted background and the original foreground object.

OPA [12] dataset is proposed for object placement evaluation with human annotation for each composite image. In total, it contains $62,074$ training images and $11,396$ test images without overlap. All the images are collected from COCO [11] dataset and each composite image is generated with one background, one object, and one placement bounding box. We adopt OPA for our object placement prediction experiment, using only the positive samples. We consider the positive bounding box as ground-truth, resulting in $21,350$ image pairs for training and $3,566$ pairs for testing.

Implementation Detail. Our method is implemented based on PyTorch [16] and trained on one RTX A5000 GPU. The batch size is set to 64 for all methods. By default, we use ViT-small [4] pre-trained weights [21] on ImageNet [3] as encoder backbone. Object images are first padded with white pixels as square images before being fed into the encoder. Then both the object and background images are resized to 224×224 and normalized with RGB average values. We adopt AdamW [14] optimizer with a learning rate of 0.00001 and weight decay of 0.03. The learning rate is adjusted with cosine scheduling [13].

4.2. Evaluation Metric

Top-k IOU. Given ground-truth bounding box and the predicted bounding boxes, one simple way to evaluate is to compute the IOU (Intersection over Union) between the ground-truth and top-1 predicted box. However, such evaluation tends to have 0 IOU for lots of samples, as there are usually multiple good candidate locations and the top-1 prediction might be reasonable but not exactly at the ground-truth location. In practice, it would also be better to provide multiple candidates for users to select. We thus apply top-5 IOU as evaluation metric, *i.e*. the best IOU between ground-truth and top-5 predicted bounding boxes.

Normalized Score. For methods with heatmap scores, the ground-truth may not be necessary to have the highest score, but it should be among the best ones. We thus apply the normalized score as one of the evaluation metrics. The heatmap score is first normalized with the minimum and maximum value as $\hat{H}$ in Sec. 3.1, which is referred to as Normalized

Method	Infer. Time (s)	Pixabay		OPA	
		$IOU > 0.5$	Mean IOU	$IOU > 0.5$	Mean IOU
Regression [26]	0.08	48.23	0.448	7.24	0.178
†Retrieval [29]	1.69	11.91	0.220	2.08	0.112
Classifier [12]	0.55	6.82	0.147	2.54	0.115
PlaceNet [26]	0.16	19.44	0.308	10.09	0.225
Ours	0.11	**74.74**	**0.620**	**15.95**	**0.241**

Table 1. Evaluation on top-5 predictions in terms of maximum IOU between the top-5 predicted bounding boxes and the ground-truth.

Method	Infer. Time (s)	NS			Mean $NS(\uparrow)$
		> 0.95	> 0.9	> 0.75	
†Retrieval [29]	1.35	8.00	20.32	57.84	0.75
Classifier [12]	0.41	21.21	32.15	53.71	0.70
Ours	**0.11**	**47.53**	**67.70**	**90.30**	**0.90**

Table 2. Evaluation on location prediction given ground-truth scale on Pixabay. NS (normalized score) denotes the predicted score at the ground truth location normalized by the maximum and minimum values across all locations.

Method	Infer. Time (s)	IOU			Mean Error(↓)
		> 0.95	> 0.9	> 0.75	
†Retrieval [29]	0.32	15.71	31.50	68.05	0.105
Classifier [12]	0.13	16.02	30.15	61.25	0.108
Ours	**0.11**	**27.04**	**50.70**	**89.65**	**0.052**

Table 3. Evaluation on scale prediction given ground-truth location on Pixabay. IOU is computed on the ground-truth location.

Score (NS). NS at the ground-truth location/scale may not be 1, but should be relatively high as compared with other locations and scales. Therefore, we compute the mean NS and portion of NS above a certain threshold (*e.g.* 0.9). The NS is more reasonable than IOU when only location is evaluated, as a small spatial shift could lead to a 0 IOU.

4.3. Comparison with State-of-the-art

We compare the proposed method with state-of-the-art and several baseline methods: 1) "Regression" [26] simply trains the network to predict the ground-truth bounding box with MSE (Mean Square Error) loss. The regression head (MLP) is adopted on the concatenated global features of background and object images. 2) "†Retrieval" follows the pipeline in [29], except we modify the initial bounding box size as the average size of our new dataset. † denotes that the retrieval model has a different setting than other methods. For Pixabay, the model is directly obtained from [29] which was trained on a different version. It does not apply inpainting and has about $2\times$ images as compared with our filtered version. The original retrieval model is trained with the masked background and the corresponding foreground objects, thus does not require inpainting on background images. However, the OPA dataset does not provide such data. Therefore, we use the ground-truth object for training instead of the original object. 3) "Classifier" follows [12] to train a binary classifier and predicts whether a composite image is reasonable. It is further extended with the sliding-window method in [29] to generate location/scale, *i.e.* generating composite images and masks with a grid of location/scale combinations and selecting the one with the maximum score. 4) PlaceNet [26] follows the implementation in [26] with adversary training. The bounding box is predicted based on the global features of background and object images along with a random vector, and a discriminator is trained to tell whether the bounding box is reasonable, conditioned on the global features.

In Table 1, we show the top-5 IOU (Sec. 4.2) based evaluation on both Pixabay and OPA datasets. Top-1 IOU is included in **supplementary material**. The proposed method achieves significant improvements over state-of-the-art methods on both inpainted and manually annotated datasets, indicating the superiority of the dense prediction as compared with sparse prediction or sliding-window formulation. Remarkably, the inference speed of the proposed method is on-par with the single-step prediction ("Regression") and is over $> 10\times$ **faster** than the previous sliding-window based method, *i.e.* "†Retrieval".

To show the effectiveness of location and scale prediction separately, we conduct experiments on interactive search where the users have provided the ground-truth location or scale. In our 3D heatmap, we simply set the predefined dimensions as the desired numbers and search for the other dimensions. For sliding-window-based methods ("†Retrieval", "Classifier"), the sliding windows are applied only on the dimensions to be searched. For location prediction, we generate a 2D heatmap using all the methods and evaluate using the NS (Sec. 4.2). The sparse prediction methods like "Regression" are not feasible in this case. For scale prediction, we fix the location as the ground-truth location and compute the IOU between the ground-truth and prediction. The prediction of sparse prediction methods would not change in this case, while other methods can be improved using the additional information of ground-truth location. Tables 2 and 3 show that the proposed method still significantly outperforms the previous methods.

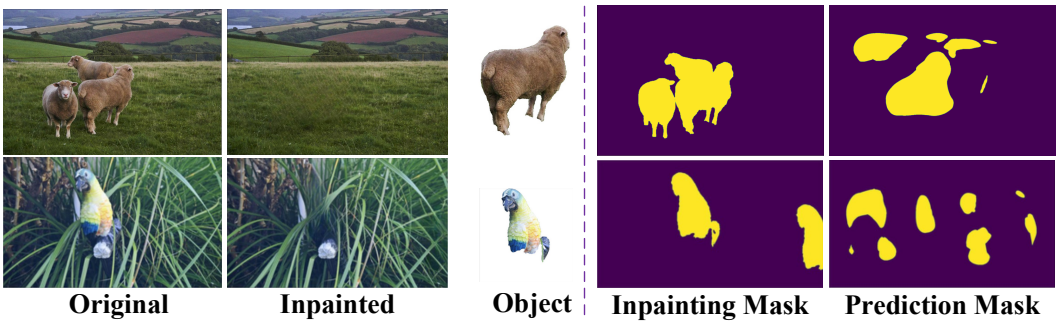

Figure 5. Example of inpainting mask and prediction mask. They usually do not have large overlapping regions.

Figure 6. Qualitative results on real-world images. Best viewed with zoom-in. Tree branch and indoor room scenarios are very challenging.

4.4. Overfitting to Inpainting Artifact?

Although we randomly apply inpainting on multiple regions in addition to the ground-truth location, the model may still overfit to the inpainting artifact by predicting only the inpainted regions. To understand whether the model is overfitting on the inpainting artifacts, we compare the inpainting mask and our prediction heatmap. We first select the highest score at each location of the predicted heatmap, resulting in a 2D heatmap. Then we binarize the 2D heatmap with a threshold so that the generated mask highlights the same number of pixels as the inpainting mask. We compute the IOU between the prediction mask and the inpainting mask to see if there is a strong correlation. Since the location of the original object is also inpainted and trained as ground-truth, even a reasonable model is likely to predict high scores for the ground-truth inpainted regions, resulting in overlapping between the prediction mask and the inpainting mask. We observe that only a small portion ($\sim 14\%$) of predictions highlight a similar region ($IOU > 0.5$) as the inpainting mask, which means the model is not always highlighting all the inpainted regions. We further show qualitative results in Fig. 5 to illustrate that the model is not overfitting to the inpainting artifacts, which is consistent with the observation in [26].

4.5. Generalization to Real-world Images

To evaluate the generalization ability on real-world images, we collect a diverse set of pure background images with compatible foreground object images. The background images are selected by searching for keywords in Pixabay engine, including mountain road, city street, beach, island, indoor room, food, tree branch, etc. Some of them are very challenging, *e.g.* inserting birds on branches as shown in Fig. 6. In total, we select 24 background images with $3 \sim 6$ objects images for each, resulting in 100 pairs for evaluation. Each of the 5 methods generates 100 results, leading to 500 results for evaluation. We then shuffle the relative order of methods and ask users to rate all the 500 samples in three levels: 0) "Unsatisfactory": The location and scale are clearly wrong. 1) "Borderline": The location and scale are OK but somewhat unrealistic. 2) The location and scale are clearly reasonable. Each sample is rated by at least five individuals and we report the average portion of each level.

Method	Unsatisfactory ↓	Borderline	Satisfactory ↑
Regression [26]	46.8	17.4	35.8
†Retrieval [29]	45.4	22.6	32.0
Classifier [12]	72.0	9.6	18.4
PlaceNet [26]	69.0	12.6	18.4
Ours	**42.8**	17.8	**39.4**

Table 4. Human evaluation on location and scale prediction for real-world images.

As shown in Table 4, our method achieves a higher satisfactory (True Positive) rate and a lower unsatisfactory (False Positive) rate than all the other methods. Note that there is a domain gap between inpainted images and natural images, the performance of methods trained on inpainted images is

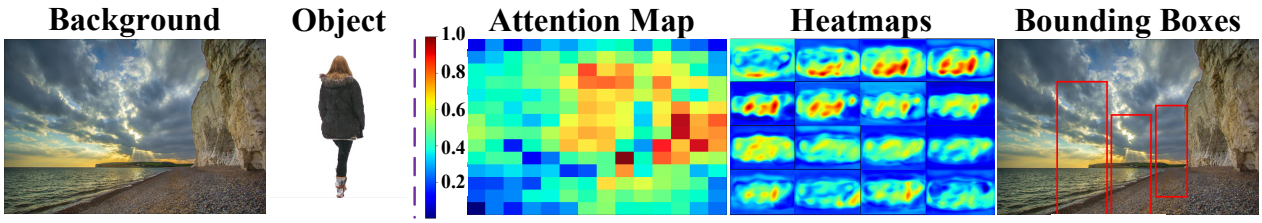

Figure 7. Example attention map between the object token and background local tokens, which highlights multiple candidate regions.

Ablations	$IOU > 0.5$	Mean IOU
Binary Assignment	0.00	0.050
Gaussian Assignment	9.50	0.252
Sparse Contrastive	**74.74**	**0.620**

Table 5. Effect of different loss functions.

Ablations	$IOU > 0.5$	Mean IOU
Global Only	0.78	0.069
Local Concat.	48.16	0.466
Local Atten.	**74.74**	**0.620**

Table 6. Effect of feature concatenations.

somewhat degraded. The performance of our method could be further improved by finetuning with small-scale human annotation on real-world images. Qualitative results are provided in Fig. 6 to show that the proposed method performs better when local clue is critical for object placement.

4.6. Ablation Study

Effect of Loss Functions. We compare the proposed sparse contrastive loss with binary and Gaussian assignment loss (Sec. 3.3) and report the top-5 IOU (Sec. 4.2) on Pixabay dataset in Table 5. "Binary Assignment" and "Gaussian Assignment" both encourage a single-peak heatmap, but their performance is much lower than the proposed "Sparse Contrastive", indicating the margin-based contrastive design is necessary for this task. A multi-peak distribution is usually observed in the prediction of the proposed method, which aligns with the fact that there are multiple reasonable location/scale combinations to insert an object.

Effect of Feature Concatenation. The are several ways to concatenate the local background features with the global object feature. "Global Only" simply concatenates the global features of both background and object images as the input of the upsampling decoder, which does not involve any local information. "Local Concat." directly concatenates the global object feature to every local feature of the background image as the input of the upsampling decoder. In this case, every local feature has information from the object image, but there is no long-range correlation between different local features. "Local Atten." combines the local background features and global object feature with a transformer, which enables the learning of correlation between any of these features with self-attention. In Table 6, "Local Atten." achieves much better performance than other configurations in terms of top-5 IOU (Sec. 4.2) on Pixabay dataset, indicating the importance of learning correlation between local background features and the global object feature.

Visualization. In Fig. 7, we show an example of the learned attention map between the foreground object token and the background local tokens. We observe that it focuses on important local regions, which can not be learned by previous methods with only global features. This supports the motivation of our design to learn local clues for object placement. One complete view of 16 heatmaps for all scales is also provided. Different locations could be recommended for different scales.

5. Discussion and Conclusion

We propose a novel object placement method for real-world compositing, which generates dense evaluation on all pre-defined placements (location/scale of the bounding box) in a single network forward pass. It learns the correlation between object feature and local background features using a transformer module so that local clues in background images can be leveraged to determine whether a particular placement is plausible. Experiments on both manually annotated dataset and large-scale inpainted dataset show significant improvements over previous state-of-the-art methods. It also generalizes well to challenging real-world cases.

Limitation and Broader Impact. One limitation is that our model only handles 2D image without considering 3D information, *e.g.* lighting, shadow, and occlusion. We could add 3D modeling for both the object and background in the future to achieve more realistic object placement. Another limitation is that we rely on an off-the-shelf inpainting model for creating our training dataset, and there is a domain gap between inpainted images and natural ones. We can add semi-supervised fine-tuning on a small set of manually annotated images to improve the generalization ability. One possible negative societal impact is the bias of our system's performance on certain combinations of object categories and background types. More Discussion is included in supplementary material.

References

[1] *https://pixabay.com/*. 2, 5

[2] Samaneh Azadi, Deepak Pathak, Sayna Ebrahimi, and Trevor Darrell. Compositional gan: Learning image-conditional binary composition. *International Journal of Computer Vision*, 128(10):2570–2585, 2020. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3, 4, 5

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[7] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. *arXiv preprint arXiv:1812.02350*, 2018. 1, 2

[8] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020. 3, 5

[9] Xueting Li, Sifei Liu, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Putting humans in a scene: Learning affordance in 3d indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12368–12376, 2019. 2

[10] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9455–9464, 2018. 1, 2

[11] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[12] Liu Liu, Bo Zhang, Jiangtong Li, Li Niu, Qingyang Liu, and Liqing Zhang. Opa: Object placement assessment dataset. *arXiv preprint arXiv:2107.01889*, 2021. 1, 2, 4, 5, 6, 7

[13] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[15] Li Niu, Wenyan Cong, Liu Liu, Yan Hong, Bo Zhang, Jing Liang, and Liqing Zhang. Making images real again: A comprehensive survey on deep image composition. *arXiv preprint arXiv:2106.14490*, 2021. 1

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 5

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[18] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2149–2159, 2022. 5

[19] Fuwen Tan, Crispin Bernier, Benjamin Cohen, Vicente Ordonez, and Connelly Barnes. Where and who? automatic semantic-aware person composition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1519–1528. IEEE, 2018. 2

[20] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 3, 5

[21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 4, 5

[22] Shashank Tripathi, Siddhartha Chandra, Amit Agrawal, Ambrish Tyagi, James M Rehg, and Visesh Chari. Learning to generate synthetic data via compositing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 461–470, 2019. 2

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4

[24] Anna Volokitin, Igor Susmelj, Eirikur Agustsson, Luc Van Gool, and Radu Timofte. Efficiently detecting plausible locations for object placement using masked convolutions. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 252–266, Cham, 2020. Springer International Publishing. 1

[25] He Zhang, Jianming Zhang, Federico Perazzi, Zhe Lin, and Vishal M Patel. Deep image compositing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 365–374, 2021. 1

[26] Lingzhi Zhang, Tarmily Wen, Jie Min, Jiancong Wang, David Han, and Jianbo Shi. Learning object placement by inpainting for compositional data augmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow,*

UK, August 23–28, 2020, Proceedings, Part XIII 16, pages 566–581. Springer, 2020. 1, 2, 6, 7

[27] Hengshuang Zhao, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Brian Price, and Jiaya Jia. Compositing-aware image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 502–516, 2018. 4

[28] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 3, 4

[29] Sijie Zhu, Zhe Lin, Scott Cohen, Jason Kuen, Zhifei Zhang, and Chen Chen. Gala: Toward geometry-and-lighting-aware object search for compositing. *arXiv preprint arXiv:2204.00125*, 2022. 1, 2, 5, 6, 7