Hallucinating Visual Instances in Total Absentia

Jiayan Qiu[1], Yiding Yang[2], Xinchao Wang[2], and Dacheng Tao[1]

[1] UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering, The University of Sydney, Darlington, NSW 2008, Australia
{jqiu3225@uni.sydney.edu.au,dacheng.tao@sydney.edu.au}
[2] Stevens Institute of Technology, Hoboken, NJ 07030, USA
{yyang99,xinchao.wang}@stevens.edu

Abstract. In this paper, we investigate a new visual restoration task, termed as hallucinating visual instances in total absentia (HVITA). Unlike conventional image inpainting task that works on images with only part of a visual instance missing, HVITA concerns scenarios where an object is completely absent from the scene. This seemingly minor difference in fact makes the HVITA a much challenging task, as the restoration algorithm would have to not only infer the category of the object in total absentia, but also hallucinate an object of which the appearance is consistent with the background. Towards solving HVITA, we propose an end-to-end deep approach that explicitly looks into the global semantics within the image. Specifically, we transform the input image to a semantic graph, wherein each node corresponds to a detected object in the scene. We then adopt a Graph Convolutional Network on top of the scene graph to estimate the category of the missing object in the masked region, and finally introduce a Generative Adversarial Module to carry out the hallucination. Experiments on COCO, Visual Genome and NYU Depth v2 datasets demonstrate that the proposed approach yields truly encouraging and visually plausible results.

1 Introduction

If we are given a masked image as the one shown in Fig. 1 (a) and are asked to hallucinate the object that is entirely absent, we can, without much effort,

Fig. 1: Hallucinating instances in total absentia. Given a masked image with a scene object completely absent like (a), our approach hallucinates a contextually and visually plausible result, in this case the back cover of a cell phone shown in (b), by explicitly accounting for the global semantics. State-of-the-art inpainting approach [87], in this case where the target object is completely absent, fills the blank region with the background texture, as depicted in (c). The ground truth is shown in (d).

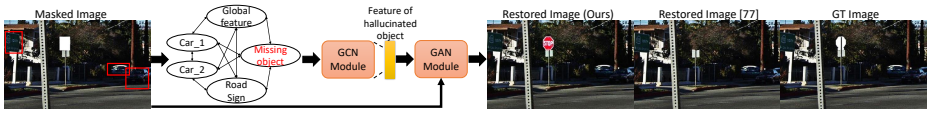


Fig. 2: Illustration of our framework. Given a masked image, we first detect the objects in the un-masked region and transform the image to a graph. We then use a GCN module to predict the semantic information of the masked part. Finally, conditioned on the masked image and the predicted semantic information of the missing region, we introduce a GAN module to produce our hallucination.

tell the type of the absent object and further imagine a fairly reasonable overall appearance of the image content in the masked region. In this regard, we identify this image to be an indoor scene featuring a desktop, on top of which we see a monitor, a keyboard, and a lamp, and then based on their relative spatial layouts, we may safely draw the conclusion that, the missing image content concerns about a wireless phone, which often appears in the vicinity of these visual instances.

We term this task, which has been barely explored in prior works, as *hallucinating visual instances in total absentia* (HVITA). Despite human brains are able to accomplish this task in a considerably effortless way, existing image generation methods that omit the explicit reasoning on the inter-object semantics, produce visually inferior results. In the case of Fig. 1 (c), a state-of-the-art image inpainting approach [87], without analyzing the instance-level contexts, fills the masked region using the background texture. In fact, all existing image inpainting methods, to our best knowledge, rely on first observing a considerable fraction of the target instance before filling the absent part, and are thus incompetent of tackling HVITA.

Towards solving the proposed HVITA task, we introduce an interesting approach that, to some degree, imitates the human reasoning process for hallucinating the missing content. Unlike inpainting schemes that rely upon a partial instance being present, our approach works on scenarios where objects are in total absentia. This means our approach attempts to infer not only the category of the missing object but also a legitimate appearance consistent with the overall ambiance of the input image.

Specifically, our proposed approach follows a end-to-end trainable architecture that comprises three stages as illustrated in Fig. 2. In the first stage, we transform the input image to a semantic graph wherein each node corresponds to a detected object in the scene, obtained from an off-the-shelf detector. The masked region is also modeled as a node, but with an unknown label to be predicted. In the second stage, we introduce a Graph Convolutional Network (GCN) to regress the semantic information of the masked area, during which process the global-level inter-object semantics are explicitly accounted for. Finally, we stack a Generative Adversarial Netowrk (GAN) to restore the masked region with the predicted semantic information and to generate visually realistic results.

Our contribution is therefore a novel framework designated to handle the proposed HVITA task, which is to our best knowledge the first attempt. This is accomplished by modeling each input image as a semantic graph, and then estimating the semantic information using a GCN module, followed by feeding the predictions into a GAN module so as to produce the final hallucination. The whole pipeline is end-to-end trainable. We evaluate our method on COCO [51], Visual Genome [41], and NYU Depth v2 [61], whose images comes from real world scenes with complex structural information, and obtain very encouraging qualitative and quantitative results.

2 Related work

We briefly review here prior works related to ours, including graph convolutional network, image inpainting, image generation, and image insertion.

Graph Convolutional Network. Earlier works on graph-related tasks either assume the node features to be fixed [66, 81, 80, 55, 54, 43], or apply iterative schemes to learn node representations, which are computationally expensive and at times unstable [17, 71, 21, 75]. Inspired by the progress of deep learning [42, 72, 27], two categories graph convolutional neural networks are proposed: spectral-based approaches, which aims to develop graph convolution based on the spectral theory [47, 45, 39, 79, 12, 28, 9], and spatial-based ones, which investigates information mutual dependency [78, 86, 31, 10, 4, 18, 60, 23, 19, 62, 3]. More recently, the attention-based methods are introduced to GCN and have achieved very promising results [1, 44, 53, 85, 84, 90, 77].

Image Inpainting. Conventional image inpainting methods, both diffusion-based [7, 5, 46] and patch-based ones [6, 8, 11], use the intra-image information to fill the masked area. Thanks to the progress of deep learning especially generative adversial networks [20], many deep approaches have been proposed [69, 40, 25, 65, 88, 48, 74]. These methods ensure the semantic continuity, yet at times yield artifacts around the border. The more recent methods make use of both intra-image information and learning from large datasets, and gain significant improvement in terms of semantic continuity and visual authenticity [83, 87, 82, 89, 34, 93, 73, 88, 92]. However, image inpainting methods handles only parts of a missing object, of which the semantic label is known, and therefore can not be exploited to hallucinate completely-absent objects.

Image Generation. Image generation has gained unprecedented improvement since the introduction of GAN [20]. Following works focus on improving the capacity of the generator and the learning capability of the discriminator [68, 52, 58, 89, 70], designing more stable loss function [56, 76, 50, 2], or introducing semantic information into the generation process [59, 64, 35, 33, 91, 63]. In this work, we build our network based on the popular Conditional GAN [57] to produce the final hallucinated image.

Object Insertion. Object insertion aims to insert a human-chosen object into the target scene. Approaches of [24, 29, 14–16] focus on geometry for object modeling [67], and those of [22, 36] utilize user interactions for model generation.

Methods like [37, 38, 49], on the other hand, model the generation process as a self-adaptive algorithm. Unlike our task, object insertion concerns only on the insertion process, rather than inferring and further hallucinating the missing object using scene semantics.

3 Method

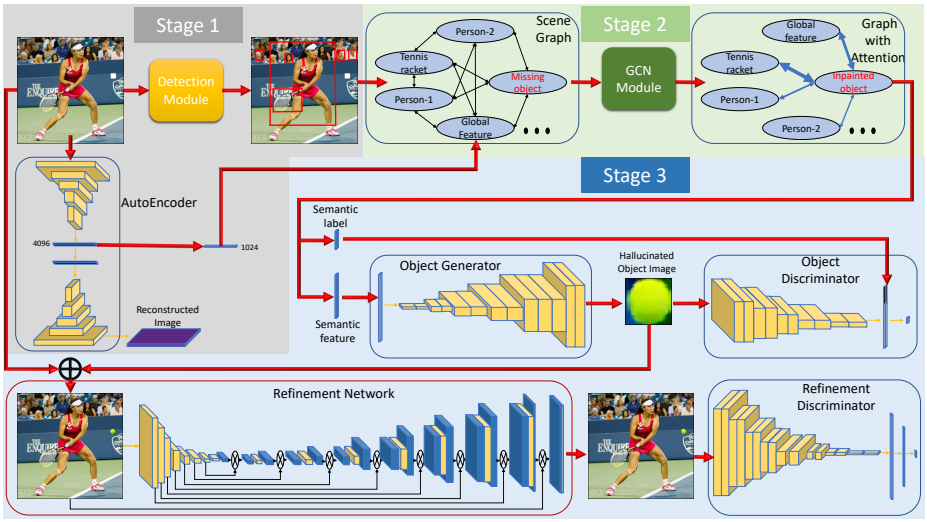


Fig. 3: Illustration of the proposed framework. The GCN module, object generator, and refinement network are trained end-to-end. $\bigoplus$ denotes the operation that fills the masked area with the restored object image, and $\bigotimes$ denotes concatenation. Note that, in the *Graph with Attention* of Stage 2, the thickness of an edge reflects its value of attention: a thicker edge indicates that the detected node is of higher importance to the hallucination inference.

In this section, we show the working scheme of the proposed framework in detail. As depicted in Fig. 3, our framework comprises three stages. In Stage 1, we utilize a detection module on the input image, where the object of interest is masked out, to detect the objects in the scene and extract their semantic and spatial information. Meanwhile, we adopt an Autoencoder to obtain the global feature of the masked image as a whole. In Stage 2, we model the masked image as a graph, in which each node corresponds to an object in the scene. We then insert two additional nodes, one for the absent object to be hallucinated and one for the global feature, into the graph. Afterwards, we apply a GCN module to estimate the features of the absent object. In Stage 3, we feed the hallucinated features of the absent object and the masked image into a GAN-based restoring module to achieve the restoration task.

3.1 Stage 1: Detection Module and Autoencoder

We adopt a detection module in Stage 1 to detect the objects in the masked image so as to derive *instance-level* features. Specifically, we train the Mask-RCNN model [26] on our training set, where masked images are used, and apply the trained detector on the test images. For each detected object, we construct a 1028-dimension feature vector that encodes both the semantic and spatial information. Features in first 1024 dimensions are taken directly from the last layer of Mask-RCNN to embrace the semantics, while features in the last four dimension, namely upper-left and lower-right coordinates of the detection bounding box, are adopted to encode its spatial information. Such 1028-dimension vectors are further fed to Stage 2, and taken to be the features of the corresponding node in the scene graph.

Apart from the instance-level features, we also explicitly account for the global appearance of the entire image by extracting *image-level* features. To this end, we adopt a customized Autoencoder shown in Fig. 4a, where Batch Normalization (BN) [32] and ReLU are implemented in each layer except the last one. For the last layer, Tanh is adopted as the active function. We fill the masked area in the input image with a pixel value of $v = 0.5$ ($v \in [0, 1]$). We then extract the first 4096-dimension feature from the Autoencoder as the global feature of the masked image, and pass feature vector to Stage 2. Specifically, the loss for Autoencoder is taken to be pixel-level square loss between the reconstructed image and the input image:

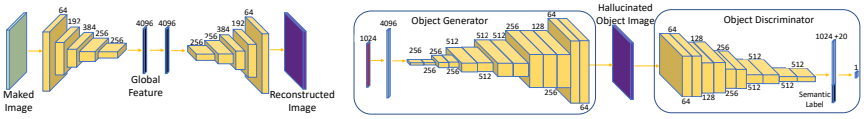
$$\mathcal{L}_{AE} = \frac{1}{N} \sum_{i=1}^{N} \|I_{in}^{i} - \hat{I}^{i}\|^2, \tag{1}$$

where I_{in}^{i} and $\hat{I}^{i}$ represent the i-th masked image and the reconstructed image respectively, and N denotes the number of samples.

3.2 Stage 2: GCN Module

The second stage of our framework takes as input both the instance- and image-level semantics obtained in Stage 1, and models the interplays between the scene objects using a scene graph. Let N denotes the number of detected objects in Stage 1. We construct a graph of $N+2$ nodes: N nodes that correspond to the N detected object instances, one node that encodes the image-level global features, and one node that denotes the missing object. We then link all the pairs of the $N + 2$ nodes to form a complete graph.

Each node in the graph holds a 1028-dimensions feature. For the N detected-object nodes, we directly take their *instance-level* features obtained in Stage 1 to be the node features. For the image node that accounts for the global semantics, we take the *image-level* feature of 4096 dimensions in Stage 1, and reduce it to 1024 dimensions using a learnable fully connected layer; we then pad 4 other dimensions, namely the origin coordinate $(0, 0)$ and the size of the image, to the 1024-dimension feature and form a 1028-dimension one. Finally for the absent-object node, we initialize its first 1024 dimensions using Gaussian noise and stack its 4-dimension location, i.e., the upper-left and lower-right coordinates of mask.



(a) (b)

Fig. 4: (a) shows the architecture of our Autoenocoder for extracting image-level semantics, in which the first 4096-dimension feature is used as the global feature. (b) shows the network architecture of the object generative adversarial network (OGAN) for hallucinating the absent object in in the mask.

Next, we adopt a graph convolutional network to reveal the intrinsic relationship between objects and derive the semantic features of the absent one to be hallucinated. Specifically, we adopt the popular graph attention network (GAT) model [77] to achieve this task. Given the input graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and the corresponding features of nodes $X = \{x_1, x_2, ...x_{|\mathcal{V}|}\}, x_i \in \mathbb{R}^F$, one layer of the adopted GCN model can be formulated as

$$x_i' = \sum_{j:j\to i\in\mathcal{E}} \frac{e^{\mathcal{A}_\theta(h_\phi(x_i),h_\phi(x_j))}}{\sum_{j:j\to i\in\mathcal{E}} e^{\mathcal{A}_\theta(h_\phi(x_i),h_\phi(x_j))}} * h_\phi(x_j), \tag{2}$$

where $\mathcal{A}$ is a function that takes a pair of nodes as input and outputs the attention score between them, and h is a non-linear mapping that takes the node features as input and maps them to a new space. The two functions $\mathcal{A}$ and h are controlled by parameters θ and ϕ, respectively. The new feature of the center node i is updated as the weighted sum of all the features from its neighbors. Thanks to the attention mechanism, the network learns to recover the feature of the absent object by interacting with other objects and global semantics of the entire image.

The loss of the GCN module comprises two parts, $\mathcal{L}_{CE}$ and $\mathcal{L}_{feat}$, defined as follows,

$$\mathcal{L}_{CE} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{H}(f(x_i'), f(x_i)), \quad \mathcal{L}_{feat} = \frac{1}{N}\sum_{i=1}^{N} ||x_i' - x_i||_2, \tag{3}$$

where f denotes the classifier of detector in Stage 1 and $f(\cdot)$ returns the classification probability, x_i denotes the ground truth feature of the hallucinated object in image i, and x_i' is the aggregated feature of the hallucinated object from GCN module. $\mathcal{H}$ denotes the cross entropy function. Specifically, $\mathcal{L}_{CE}$ accounts for the predicted label of the absent object, and $\mathcal{L}_{feat}$ enforces the similarity between the aggregated feature and the ground truth one. The loss for GCN module is thus taken as

$$\mathcal{L}_{GCN} = \mathcal{L}_{CE} + \mathcal{L}_{feat}. \tag{4}$$

The semantic feature of the hallucinated object, aggregated from its neighbouring objects and the entire scene by the GCN module, is then passed to the next stage.

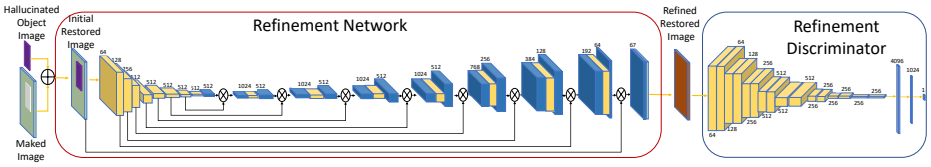


Fig. 5: Architecture of our global refinement adversarial network (GRAN). $\bigoplus$ denotes summation and the $\bigotimes$ denotes concatenation. GRAN tasks as input the initial restored image and outputs the refined one, ensuring the semantic continuity and visual authenticity.

3.3 Stage 3: GAN module

Given the predicted class and features of the absent object, we conduct hallucination on the masked area to produce the final image output. This is achieved, in our implementation, via a GAN module. Specifically, our GAN module consists of two networks, an object generative adversarial network (OGAN) for generating the object and a global refinement adversarial network (GRAN) for refining restored image.

The structure of our OGAN is showed in Fig. 4b. It can be seen that the network is designed as a conditional GAN to generate the target object with the predicted class. In the Object Generator network, BN and Leaky ReLU are implemented in every layer except the last, where Tanh is used as the activation function. As for the Object Discriminator, BN and ReLU are applied to every layer but the last, where no activation function is applied and least squares loss [56] is implemented. To this end, we write,

$$\begin{aligned}\min_{OD} V_{OGAN}(OD) &= \frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}[(OD(x,y)-b)^2] + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[(OD(OG(z),y)-a)^2] \\ \min_{OG} V_{OGAN}(OG) &= \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[(OD(OG(z),y)-c)^2],\end{aligned} \tag{5}$$

where OD denotes the Object Discriminator network, and OG denotes the Object Generator network. a and b denote the ground truth fake and real labels, respectively. c denotes the value that OG wants OD to believe for the fake data, and y denotes the label of predicted class.

Once the image of the absent object is hallucinated, we resize and then insert it into the masked area of the input image, and obtain the initial restored image. Such restored images, despite semantically meaningful, turn out to be visually implausible, as the OGAN focuses on producing an object image of a specified class but overlooks the background content within the mask. This calls for another image generator that looks at the image as a whole and explicitly accounts for the visual continuity.

To this end, we introduce GRAN, a second GAN-based module that ensures the semantic continuity and visual authenticity of the restored image. The architecture of our GRAN is shown in Fig. 5. It comprises a Refinement Network and a Refinement Discriminator. For the Refinement Network, we implement it as an

U-net structure, which utilizes multi-scale features and avoids gradient vanishing. Specifically, in this network, BN and Leaky ReLU are implemented in every layer except the last, where again Tanh is adopted as active function. As for the Refinement Discriminator, we implement BN and ReLU in every layer except the last one. We also adopt the least square loss for the Refinement Disriminator:

$$\begin{aligned}\min_{RD} V_{GRAN}(RD) &= \frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}[(RD(x)-b)^2] + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[(RD(RN(z))-a)^2] \\ \min_{RN} V_{GRAN}(RN) &= \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[(RD(RN(z))-c)^2],\end{aligned} \tag{6}$$

where RD denotes the Refinement Discriminator network, and RN denotes the Refinement network. Moreover, we impose a l_1 loss on the Refinement network for the un-masked areas:

$$\mathcal{L}_{valid} = \frac{1}{N_{I_{valid}}}\|(1-M)\odot(I_{out}-I_{is})\|_1, \tag{7}$$

where $N_{I_{valid}}$ denotes the number of unmasked pixels, M denotes the binary mask (1 for holes), I_{out} denotes the output of the Refinement network, and I_{is} denotes the initial restored image.

In the first training step, our GAN module is pre-trained to ensure the ability of generating visual realistic images. The details will be explained in the experiment section.

3.4 End-to-end training

The GCN and GAN module in our framework are end-to-end trainable. In our implementation, we adopt the end-to-end training scheme of the two modules, as the visual authenticity loss of the GAN in Stage 3 may facilitate the GCN in Stage 2 to aggregates and encode more discriminant information into its obtained semantic feature. We shown in Fig. 6 a comparison between the results obtained by end-to-end training and by separate training, where the former yields to the more semantically plausible result.

In sum, the losses for GCN module, OGAN, and GRAN are summarized as follows,

$$\begin{aligned}\mathcal{L}_{GCN} &= \lambda_1\mathcal{L}_{RN_{GRAN}} + \lambda_2\mathcal{L}_{OG_{OGAN}} + \mathcal{L}_{CE} + \mathcal{L}_{feat}, \\ \mathcal{L}_{OG} &= \mathcal{L}_{RN_{GRAN}} + \lambda_3\mathcal{L}_{OG_{OGAN}}, \\ \mathcal{L}_{RN} &= \mathcal{L}_{RN_{GRAN}} + \lambda_4\mathcal{L}_{valid},\end{aligned} \tag{8}$$

where λ_1, λ_2, λ_3 and λ_4 denotes the balancing weights.

4 Experiments

In this section, we provide our experimental setups and show the results. Since we are not aware of any existing work that performs exactly the same task as we do here, we mainly focus on showing the promise of the proposed framework. We also compare part of our framework with other popular models. Our goal is, again, to show the possibility of hallucinating totally absent but reasonable

(a) (b) (c) (d)

Fig. 6: Comparison between training Stage 2 and Stage 3 separately and end-to-end. (a), (b), (c) and (d) denote, respectively, the masked image, restored image with end-to-end training, restored image with separately training, and the ground truth.

(a) (b) (c)

Fig. 7: An example that our method produces visually and structurally reasonable, yet distorted image completion, in this case the distorted car. Image (a) ,(b) and (c) respectively denote the input, our restoration image, and the ground truth.

objects in the masked image, rather than trying to beat the state-of-the-art GCN, GAN, and inpainting models. More complicated networks, as long as they are end-to-end trainable, can be adopted in our framework to achieve potentially better performances.

4.1 Datasets and Implementation Details

We adopt three datasets, COCO [51], Visual Genome [41], and NYU Depth v2 [61] to validate the proposed object hallucination approach. Image from these datasets feature complex scenes with rich contextual information, from which we can draw discriminant information to infer the type and appearance of the absent object. Other datasets, such as Pascal VOC [30] and ImageNet [13], do not fit our purpose well, since images from Pascal VOC typically contain only one object, while those from ImageNet often feature objects of the same class.

Microsoft COCO 2017 Dataset [51]. It is a detection dataset that comprises 123k complex scene images from 81 object classes for classification. We perform HVITA on 20 out of 81 classes: we use 40k images for training, 5k for validation, and 32k images for testing. We leave out the other classes, as the state-of-the-art object detector, GCN, and GAN are still short of the capability to detect and generate objects under in-the-wild higher-order physical constraints, such as the 3D orientation of a scene object. In the case of Fig. 7, for example, even though our approach successfully encodes the type of the missing object, in this case a car, and further generates a visually reasonable image, the 3D orientation of car, however, deviates from ground truth.

Visual Genome [41]. It comprises 110k images and 3.8m object instances. We adopt this dataset because it features a large number of complex scenes.

Term	Ours	GNN[39]	Ours-NYU	GNN-NYU	Ours-Genome	GNN-Genome
Accuracy (%)	87.93	79.23	80.41	75.54	85.01	80.47
Balanced Acc (%)	85.62	74.82	76.83	69.31	82.56	77.51

Table 1: Comparative results of our GCN module and GNN on COCO and NYU v2.

This dataset, however, includes a total number of 38k fine-grained object classes, making it infeasible to train a classifier and detector that can perform well on all classes with the limited samples. To this end, we group the finer-grained classes into coarser ones, such as clustering *blue suitcase* and *black suitcase* into a *suitcase* class. Then, we use the grouped classes as ground truth to train our detection and GCN module. In our experiment, we implement our framework on the same 20 classes as done for COCO. Thus, we use 30k images for training, 5k for validation, and 20k for testing. We use exactly same network structures and training parameters as for the COCO dataset.

NYU Depth v2 [61]. It comprises 1449 images from real-world indoor scenes. We adopt this dataset for testing the generalization capability of our method, as it provides complex scene with abundant structural information.

Implementation Details. Our networks are implemented using PyTorch and with 4 Tesla V-100 SXM2 GPUs. In the pre-training stage, the batch size for Autoencoder, GCN module and GAN module are 192, 32 and 96, respectively. During the end-to-end training stage, the batch size is set to 16 for GCN and GAN module. The loss balancing weights λ_1, λ_2, λ_3 and λ_4 are set to be 0.01, 0.05, 5 and 0.1, respectively. We follow a two-step training strategy, which we find to be more efficient and effective than the single-step strategy that trans Stage 2 and Stage 3 all at once. In the first step, we supervisedly pre-training Stage 2 and Stage 3 independently. And in the second step, we end-to-end training these two stages with loss adjustment.

4.2 GCN Module

We compare our Graph Attention Network with GNN proposed by Kipf et al. [39] on the classification accuracy and balanced class accuracy, to demonstrate the its capability to infer the correct object class. It can be seen from Tab. 1 that our GCN model outperforms GNN on both metrics. This is because the GNN model aggregates the information of missing node from its neighbours averagely, thus unable to learn relationship importance embedded in the structural information.

For the GCN module, we use a three-layer neural network with residual connections. The number of attention heads for each layer are all set to be 6, and the feature dimension of the head for each layer are set to be 128, 256, 1024 respectively. For the last layer, we adopt a ReLU activation function to make the aggregated features comparable with the ground truth. The learning rate of GCN module is set to be 0.003 in the pre-trainig stage and 0.00001 in the end-to-end training stage.

Term	OG-only	RN-only	Full-OG	Full-RN
PSNR	26.23	29.06	27.14	29.17
SSIM	0.9002	0.9144	0.9078	0.9165

Table 2: Results of the GAN networks trained separately (OG-only/RN-only) and with end-to-end training (Full-OG/Full-RN).

4.3 GAN Module

We show here the pre-training process of our GAN module in detail. We feed the ground-truth object feature and class label to the GAN module, and use ground-truth object image and the scene image as the target output of OG and RN to compute a l_1 loss. An extra loss for OG is taken to be $\mathcal{L}_{OG}^{sup} = \frac{1}{N_{I_{Ogt}}} \| I_{OG} - I_{Ogt} \|_1$, where the $N_{I_{Ogt}}$ denotes the number of pixels in the ground truth, I_{OG} denotes the generated image from OG, and I_{Ogt} denotes the ground truth object image. We also define an extra loss for RN to be $\mathcal{L}_{RN}^{sup} = \frac{1}{N_{I_{hole}}} \| (M) \odot (I_{out} - I_{in}) \|_1$. For the OG network, a squared RGB image with size 128 is generated. For the RN network, however, a squared RGB image with size 512 is required for input and output. To end this, for ground truth scene image, we first resize and then crop it centered at the masked area.

Moreover, in the pre-training process, we randomly add Gaussian noises on the background area in image generated by OG. This operation helps RN improve the performance on ensuring semantic continuity and visual authenticity. Here, the loss for OG and RN are computed as

$$\begin{aligned} \mathcal{L}_{OG}^{pre} &= \mathcal{L}_{RN_{GRAN}} + \beta_1 \mathcal{L}_{OG_{OGAN}} + \beta_2 \mathcal{L}_{OG}^{sup}, \\ \mathcal{L}_{RN}^{pre} &= \mathcal{L}_{RN_{GRAN}} + \beta_3 \mathcal{L}_{valid} + \beta_4 \mathcal{L}_{RN}^{sup}, \end{aligned} \tag{9}$$

where β_1, β_2, β_3 and β_4 denotes the loss balancing weights and are set to 5, 0.1, 0.1 and 0.5, respectively.

We compare the results of two training schemes for GAN module in pre-training step: training OGAN and GRAN separately, and training them jointly end-to-end. It can be seen from Tab. 2 that the end-to-end training improves the performance of OG network by a large margin. This can be in part explained by that, when training with GRAN, the supervision of background refinement are passed to OGAN, allowing for the OG network to ensure both the visual authenticity and the background continuity. The learning rate for GAN module is set to 10^{-5} during the pre-training stage. During the end-to-end training stage, the learning rate is reduced from 10^{-5} to 10^{-6}.

4.4 User Study

To validate the authenticity of the hallucinated image, we conduct two user-study experiments, where 166 users are involved to evaluate the visual quality of our produced images. In the first experiment, we send each user 30 randomly selected image pairs, where one of them is the ground-truth image and the other

Term	COCO -Uexp1	Visual Genome -Uexp1	NYU -Uexp1	COCO -Uexp2	Visual Genome -Uexp2	NYU -Uexp2
Score	0.406	0.399	0.373	3.38	3.31	3.17
Std	0.049	0.052	0.075	0.098	0.092	0.15

Table 3: Statistical results of user study experiments. *std* here denotes the standard deviation of people.

Term -full	Ours -full	GNN -full	Ours -L_{KL}	GNN -L_{KL}	Ours -without-G	GNN -without-G	Ours -without-S	GNN -without-S
Accuracy (%)	87.93	79.23	85.46	73.01	86.55	75.67	84.86	77.64
Balanced Acc (%)	85.62	74.82	80.12	70.56	84.19	71.78	81.21	72.58

Table 4: Results of Ours and GNN under different setups. We compare the performances of the two networks trained with full settings in our paper (Ours-full/GNN-full), the performances of the two networks trained using Kullback–Leibler divergence loss L_{KL} to replace the cross-entropy loss L_{class} (Ours-L_{KL}/GNN-L_{KL}), the performance of the two networks trained without global feature from Autoencoder (Ours-without-G/GNN-without-G), and those of the two trainings without spatial information (Ours-without-S/GNN-without-S).

is our hallucinated one, and ask the user to pick which image of the two is the real one. Finally, the proposed method achieves 40.6% real chosen on COCO, 39.9% on Visual Genome, and 37.3% on NYU v2.

For the second experiment, we send each user 30 randomly selected image triplets: the original image, the image with an instance masked out, and the hallucinated image. We then ask the user to give a grade for each hallucinated image: very visually real (4 points), fairly visually real (3 points), borderline (2 points), fairly visually fake (1 points), very visually fake (0 points). The obtained average score is 3.38 on COCO, 3.31 on Visual Genome and 3.17 on NYU V2. The results of these two experiments are summarized in Tab. 3, where our proposed approach indeed achieves promising and stable performances on the user study.

4.5 Ablation Studies

More visual samples. We show more visual examples in Fig. 8 for COCO and Visual Genome and NYU v2 datasets, where our methods generates visually pleasing results.

Results using Kullback–Leibler divergence loss in GCN module. We replace the cross-entropy loss $\mathcal{L}_{CE}$ of GCN module by the Kullback–Leibler divergence loss $\mathcal{L}_{KL}$, which regresses the probabilities of the predicted classes. The supervised label is the ground-truth probability from the detection module. It can be seen from Tab. 4 that the $\mathcal{L}_{KL}$ decreases the prediction accuracies of both models. This shows when the hallucinated object's semantic feature contains multi-class information, its relationship with other objects is hard to determine, thus increasing the complexity of semantic understanding.

Fig. 8: Results on the COCO (Rows 1,2,4), NYU v2 (Row 3) and Visual Genome (Rows 5,6) dataset images. For each group of images, the first one is the masked image, the second one is our restored image and the third one is the original image.

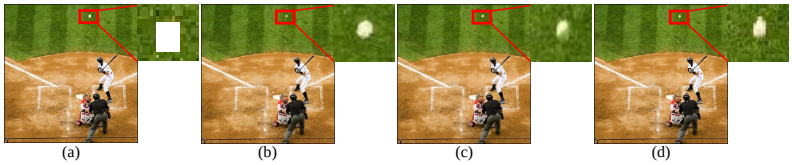


Fig. 9: Comparing training GAN module with and without global scene features from Autoencoder. The image (a), (b) ,(c) and (d) denotes the masked image, the restored image with global feature from Autoencoder, the restored image without global feature, and the ground truth image, respectively. Ours GCN module is used to hallucinate the semantic feature of the missing object.

Results without using global feature in GCN module. In this experiment, we train our GCN module and GNN without the global information from the Autoencoder. This means the GCN can only extract structural information from the remaining objects. As can be seen from Tab. 4, the performance decreased. This is because the missing objects are with strong relationship with the scene style. An visual example is shown in Fig. 9: when the semantic feature of the

(a) (b) (c)

Fig. 10: An example that our framework produces visual realistic and structurally reasonable restoration but with a incorrectly predicted label.

baseball is not trained with global feature, a tennis ball is hallucinated. One reason for this error is that, COCO contains both types of balls in the same class. After training with the global feature, however, a baseball is hallucinated, showing effect of the global feature for the image generating process.

Results without using spatial information in GCN module. We train our GCN module and GNN without the spatial information, and thus the feature of an object is shortened to be 1024 dimensions. It can be seen from Tab. 4 that the performances of both methods decrease significantly, which demonstrates the important role played by the spatial coordinates of the defections.

An interesting case. We show a case in Fig. 10, where an incorrect object label is predicted, yet a visually plausible and contextually reasonable object is hallucinated. This highlights that, our method can utilize the structural information to produce realistic object hallucinations, even with a wrong label.

5 Conclusion

In this paper, we introduce a new image restoration task, named hallucinating visual instances in total absentia (HAVITA). Unlike image inpainting that aims to fill the missing part of a visual instance present in the scene, HAVITA concerns about hallucinating an completely-absent object. To this end, we propose an end-to-end network that models the input image as a semantic graph, and then predicting the semantic information using a GCN module, followed by feeding the semantic information into our GAN module to generate the final hallucination. Results on three datasets demonstrate the encouraging potential of our approach. In our future work, we will study hallucinating objects with more complex physical constraints, such as the 3D orientation of a running car following that of the road.

Acknowledgement This research was supported by Australian Research Council Projects FL-170100117, DP-180103424, LE-200100049 and the startup funding of Stevens Institute of Technology.

References

1. Abu-El-Haija, S., Perozzi, B., Al-Rfou, R., Alemi, A.A.: Watch your step: Learning node embeddings via graph attention. In: Advances in Neural Information Processing Systems. pp. 9180–9190 (2018)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
3. Atwood, J., Towsley, D.: Diffusion-convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1993–2001 (2016)
4. Bacciu, D., Errica, F., Micheli, A.: Contextual graph markov model: A deep and generative approach to graph processing. In: ICML (2018)
5. Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., Verdera, J.: Filling-in by joint interpolation of vector fields and gray levels. IEEE transactions on image processing **10**(8), 1200–1211 (2001)
6. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. In: ACM Transactions on Graphics (ToG). vol. 28, p. 24. ACM (2009)
7. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 417–424. ACM Press/Addison-Wesley Publishing Co. (2000)
8. Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. IEEE transactions on image processing **12**(8), 882–889 (2003)
9. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
10. Chen, J., Zhu, J., Song, L.: Stochastic training of graph convolutional networks with variance reduction. arXiv preprint arXiv:1710.10568 (2017)
11. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. vol. 2, pp. II–II. IEEE (2003)
12. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. pp. 3844–3852 (2016)
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
14. Furukawa, Y., Hernández, C., et al.: Multi-view stereo: A tutorial. Foundations and Trends® in Computer Graphics and Vision **9**(1-2), 1–148 (2015)
15. Fyffe, G., Jones, A., Alexander, O., Ichikari, R., Graham, P., Nagano, K., Busch, J., Debevec, P.: Driving high-resolution facial blendshapes with video performance capture. In: ACM SIGGRAPH 2013 Talks, pp. 1–1 (2013)
16. Fyffe, G., Nagano, K., Huynh, L., Saito, S., Busch, J., Jones, A., Li, H., Debevec, P.: Multi-view stereo on consistent face topology. In: Computer Graphics Forum. vol. 36, pp. 295–309. Wiley Online Library (2017)
17. Gallicchio, C., Micheli, A.: Graph echo state networks. In: The 2010 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2010)
18. Gao, H., Wang, Z., Ji, S.: Large-scale learnable graph convolutional networks. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1416–1424. ACM (2018)
19. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1263–1272. JMLR. org (2017)

20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
21. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. vol. 2, pp. 729–734. IEEE (2005)
22. Grosse, R., Johnson, M.K., Adelson, E.H., Freeman, W.T.: Ground truth dataset and baseline evaluations for intrinsic image algorithms. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 2335–2342. IEEE (2009)
23. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. pp. 1024–1034 (2017)
24. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
25. Hays, J., Efros, A.A.: Scene completion using millions of photographs. Communications of the ACM **51**(10), 87–94 (2008)
26. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
28. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 (2015)
29. Hernandez, C., Vogiatzis, G., Cipolla, R.: Multiview photometric stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(3), 548–554 (2008)
30. Hoiem, D., Divvala, S.K., Hays, J.H.: Pascal voc 2008 challenge. In: PASCAL challenge workshop in ECCV. Citeseer (2009)
31. Huang, W., Zhang, T., Rong, Y., Huang, J.: Adaptive sampling towards fast graph representation learning. In: Advances in Neural Information Processing Systems. pp. 4558–4567 (2018)
32. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
33. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
34. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in neural information processing systems. pp. 2017–2025 (2015)
35. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
36. Karsch, K., Hedau, V., Forsyth, D., Hoiem, D.: Rendering synthetic objects into legacy photographs. ACM Transactions on Graphics (TOG) **30**(6), 1–12 (2011)
37. Karsch, K., Liu, C., Kang, S.B.: Depth transfer: Depth extraction from video using non-parametric sampling. IEEE transactions on pattern analysis and machine intelligence **36**(11), 2144–2158 (2014)
38. Karsch, K., Sunkavalli, K., Hadap, S., Carr, N., Jin, H., Fonte, R., Sittig, M., Forsyth, D.: Automatic scene inference for 3d object compositing. ACM Transactions on Graphics (TOG) **33**(3), 1–15 (2014)
39. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

40. Köhler, R., Schuler, C., Schölkopf, B., Harmeling, S.: Mask-specific inpainting with deep neural networks. In: German Conference on Pattern Recognition. pp. 523–534. Springer (2014)
41. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision **123**(1), 32–73 (2017)
42. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
43. Lan, L., Wang, X., Zhang, S., Tao, D., Gao, W., Huang, T.S.: Interacting tracklets for multi-object tracking. IEEE Transactions on Image Processing **27**(9), 4585–4597 (2018)
44. Lee, J.B., Rossi, R., Kong, X.: Graph classification using structural attention. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1666–1674. ACM (2018)
45. Levie, R., Monti, F., Bresson, X., Bronstein, M.M.: Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Transactions on Signal Processing **67**(1), 97–109 (2018)
46. Levin, A., Zomet, A., Weiss, Y.: Learning how to inpaint from global image statistics. In: null. p. 305. IEEE (2003)
47. Li, R., Wang, S., Zhu, F., Huang, J.: Adaptive graph convolutional neural networks. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
48. Li, Y., Liu, S., Yang, J., Yang, M.H.: Generative face completion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3911–3919 (2017)
49. Liao, Z., Karsch, K., Zhang, H., Forsyth, D.: An approximate shading model with detail decomposition for object relighting. International Journal of Computer Vision **127**(1), 22–37 (2019)
50. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
51. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
52. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. In: Advances in neural information processing systems. pp. 469–477 (2016)
53. Liu, Z., Chen, C., Li, L., Zhou, J., Li, X., Song, L., Qi, Y.: Geniepath: Graph neural networks with adaptive receptive paths. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4424–4431 (2019)
54. Maksai, A., Wang, X., Fleuret, F., Fua, P.: Non-markovian globally consistent multi-object tracking. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
55. Maksai, A., Wang, X., Fua, P.: What players do with the ball: A physically constrained interaction modeling. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
56. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2794–2802 (2017)
57. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
58. Miyato, T., Koyama, M.: cgans with projection discriminator. arXiv preprint arXiv:1802.05637 (2018)

59. Mo, S., Cho, M., Shin, J.: Instagan: Instance-aware image-to-image translation. arXiv preprint arXiv:1812.10889 (2018)
60. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5115–5124 (2017)
61. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
62. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: International conference on machine learning. pp. 2014–2023 (2016)
63. Park, E., Yang, J., Yumer, E., Ceylan, D., Berg, A.C.: Transformation-grounded image generation network for novel 3d view synthesis. In: Proceedings of the ieee conference on computer vision and pattern recognition. pp. 3500–3509 (2017)
64. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Gaugan: semantic image synthesis with spatially adaptive normalization. In: ACM SIGGRAPH 2019 Real-Time Live! p. 2. ACM (2019)
65. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
66. Qiu, J., Wang, X., Fua, P., Tao, D.: Matching seqlets: An unsupervised approach for locality preserving sequence matching. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
67. Qiu, J., Wang, X., Maybank, S.J., Tao, D.: World from blur. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
68. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
69. Ren, J.S., Xu, L., Yan, Q., Sun, W.: Shepard convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 901–909 (2015)
70. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in neural information processing systems. pp. 2234–2242 (2016)
71. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**(1), 61–80 (2008)
72. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
73. Song, Y., Yang, C., Lin, Z., Liu, X., Huang, Q., Li, H., Jay Kuo, C.C.: Contextual-based image inpainting: Infer, match, and translate. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
74. Song, Y., Yang, C., Shen, Y., Wang, P., Huang, Q., Kuo, C.C.J.: Spg-net: Segmentation prediction and guidance network for image inpainting. arXiv preprint arXiv:1805.03356 (2018)
75. Sperduti, A., Starita, A.: Supervised neural networks for the classification of structures. IEEE Transactions on Neural Networks **8**(3), 714–735 (1997)
76. Tran, D., Ranganath, R., Blei, D.: Hierarchical implicit models and likelihood-free variational inference. In: Advances in Neural Information Processing Systems. pp. 5523–5533 (2017)
77. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
78. Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. arXiv preprint arXiv:1809.10341 (2018)

79. Wang, X., Li, Z., Tao, D.: Subspaces indexing model on grassmann manifold for image search. IEEE Transactions on Image Processing **20**(9), 2627–2635 (2011)
80. Wang, X., Türetken, E., Fleuret, F., Fua, P.: Tracking interacting objects using intertwined flows. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(11), 2312–2326 (2016)
81. Wang, X., Türetken, E., Fleuret, F., Fua, P.: Tracking interacting objects optimally using integer programming. In: European Conference on Computer Vision and Pattern Recognition (ECCV). pp. 17–32 (2014)
82. Yan, Z., Li, X., Li, M., Zuo, W., Shan, S.: Shift-net: Image inpainting via deep feature rearrangement. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 1–17 (2018)
83. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6721–6729 (2017)
84. Yang, Y., Qiu, J., Song, M., Tao, D., Wang, X.: Distilling knowledge from graph convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
85. Yang, Y., Wang, X., Song, M., Yuan, J., Tao, D.: SPAGAN: shortest path graph attention network. In: International Joint Conference on Artificial Intelligence (IJCAI) (2019)
86. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. In: Advances in Neural Information Processing Systems. pp. 4800–4810 (2018)
87. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5505–5514 (2018)
88. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4471–4480 (2019)
89. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. arXiv preprint arXiv:1805.08318 (2018)
90. Zhang, J., Shi, X., Xie, J., Ma, H., King, I., Yeung, D.Y.: Gaan: Gated attention networks for learning on large and spatiotemporal graphs. arXiv preprint arXiv:1803.07294 (2018)
91. Zheng, C., Cham, T.J., Cai, J.: T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 767–783 (2018)
92. Zheng, C., Cham, T.J., Cai, J.: Pluralistic image completion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1438–1447 (2019)
93. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: European conference on computer vision. pp. 286–301. Springer (2016)