Semantic Object Accuracy for Generative Text-to-Image Synthesis

Tobias Hinz, Stefan Heinrich, and Stefan Wermter

Abstract—Generative adversarial networks conditioned on textual image descriptions are capable of generating realistic-looking images. However, current methods still struggle to generate images based on complex image captions from a heterogeneous domain. Furthermore, quantitatively evaluating these text-to-image models is challenging, as most evaluation metrics only judge image quality but not the conformity between the image and its caption. To address these challenges we introduce a new model that explicitly models individual objects within an image and a new evaluation metric called *Semantic Object Accuracy* (SOA) that specifically evaluates images given an image caption. The SOA uses a pre-trained object detector to evaluate if a generated image contains objects that are mentioned in the image caption, e.g., whether an image generated from *"a car driving down the street"* contains a car. We perform a user study comparing several text-to-image models and show that our SOA metric ranks the models the same way as humans, whereas other metrics such as the Inception Score do not. Our evaluation also shows that models which explicitly model objects outperform models which only model global image characteristics.

Index Terms—Text-to-image synthesis, generative adversarial network (GAN), evaluation of generative models, generative models

1 Introduction

Generative adversarial networks (GANs) [1] are capable of generating realistic-looking images that adhere to characteristics described in a textual manner, e.g., an image caption. For this, most networks are conditioned on an embedding of the textual description. Often, the textual description is used on multiple levels of resolution, e.g., first to obtain a course layout of the image at lower levels and then to improve the details of the image on higher resolutions. This approach has led to good results on simple, well-structured data sets containing a specific class of objects (e.g., faces, birds, or flowers) at the image center.

Once images and textual descriptions become more complex, e.g., by containing more than one object and having a large variety in backgrounds and scenery settings, the image quality drops drastically. This is likely because, until recently, almost all approaches only condition on an embedding of the complete textual description, without paying attention to individual objects. Recent approaches have started to tackle this by either relying on specific scene layouts [2] or by explicitly focusing on individual objects [3], [4]. In this work, we extend this approach by additionally focusing specifically on salient objects within the generated image. However, generating complex scenes containing multiple objects from a variety of classes is still a challenging problem.

The most commonly used evaluation metrics for GANs, the Inception Score (IS) [5] and the Fréchet Inception Distance (FID) [6], are not designed to evaluate images that contain multiple objects and depict complex scenes. In fact, both of these metrics depend on an image classifier (the Inception-Net), which is pre-trained on ImageNet, a data set whose images almost always contain only a single object at the image center. They also do not evaluate the consistency between image description and generated image and, therefore, can not evaluate whether a model generates images that actually depict what is described in the caption. Even evaluation metrics specifically designed for text-to-image synthesis evaluation such as the R-precision [7] often fail to evaluate more detailed aspects of an image, such as the quality of individual objects.

As such, our contributions are twofold: first, we introduce a novel GAN architecture called *OP-GAN* that focuses specifically on individual objects while simultaneously generating a background that fits with the overall image description. Our approach relies on an object pathway similar to [3], which iteratively attends to all objects that need to be generated given the current image description. In parallel, a global pathway generates the background features which later on get merged with the object features. Second, we introduce an evaluation metric specifically for text-to-image synthesis tasks which we call *Semantic Object Accuracy* (SOA). In contrast to most current evaluation metrics, our metric focuses on individual objects and parts of an image and also takes the caption into consideration when evaluating an image. Image descriptions often explicitly or implicitly mention what kind of objects are seen in an image, e.g., an image described by the caption *"a person holding a cell phone"* should depict both a person and a cell phone. To evaluate this, we sample all image captions from the COCO validation set that explicitly mention one of the 80 main object categories (e.e. "person", "dog", "car", etc.) and use them to generate images. We then use a pre-trained object detector [8] and check whether it detects the explicitly mentioned objects

• *The authors are with the Knowledge Technology Group, University of Hamburg, 22527 Hamburg, Germany. E-mail: {hinz, heinrich, wermter}@informatik.uni-hamburg.de.*

Manuscript received 29 Oct. 2019; revised 30 Apr. 2020; accepted 28 Aug. 2020. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Tobias Hinz.)
Recommended for acceptance by T. Berg.
Digital Object Identifier no. 10.1109/TPAMI.2020.3021209

within the generated images. We perform a user study over several current text-to-image models and show that SOA is highly compatible with human evaluation whereas other metrics, such as the Inception Score, are not.

We evaluate several variations of our proposed model as well as several state-of-the-art approaches that provide pretrained models. Our results show that current architectures are not able to generate images that contain objects of the same quality as the original images. While some models already achieve results close to or better than real images on scores such as the IS and R-precision, none of the models comes close to generating images that achieve SOA scores close to the real images. However, our results and user study also show that models that attend to individual objects in one way or another tend to perform better than models, which only focus on global image semantics.

2 Related Work

Modern architectures are able to synthesize realistic, high-resolution images of many domains. In order to generate images of high resolution many GAN [1] architectures use multiple discriminators at various resolutions [9]. Additionally, most GAN architectures use some form of attention for improved image synthesis [7] as well as matching aware discriminators [10] which identify whether images correspond to a given textual description.

Originally, most GAN approaches for text-to-image synthesis encoded the textual description into a single vector which was used as a condition in a conditional GAN (cGAN) [9], [10]. However, this faces limitations when the image content becomes more complex as e.g., in the COCO data set [11]. As a result, many approaches now use attention mechanisms to attend to specific words of the sentence [7], use intermediate representations such as scene layouts [2], condition on additional information such as object bounding boxes [3] or perform interactive image refinement [12]. Other approaches generate images directly from semantic layouts without additional textual input [13], [14]or perform a translation from text to images and back [15], [16].

Direct Text-to-Image Synthesis. Approaches that do not use intermediate representations such as scene layouts use only the image caption as conditional input. [10] use a GAN to generate images from captions directly and without any attention mechanism. Captions are embedded and used as conditioning vector and they introduce the widely adopted matching aware discriminator. The matching aware discriminator is trained to distinguish between real and matching caption-image pairs (“real”), real but mismatching caption-image pairs (“fake”), and matching captions with generated images (“fake”). [17] modify the sampling procedure during training to obtain a curriculum of mismatching caption-image pairs and introduce an auxiliary classifier that specifically predicts the semantic consistency of a given caption-image pair. [9], [18] use multiple generators and discriminators and are one of the first ones to achieve good image quality at resolutions of 256×256 on complex data sets. [19] have a similar architecture as [18] with multiple discriminators but only use one generator while [20] generate realistic high-resolution images from text with a single discriminator and generator.

[7] extend [9] and are the first ones to introduce an attention mechanism to the text-to-image synthesis task with GANs. The attention mechanism attends to specific words in the caption and conditions different image regions on different words to improve the image quality. [21] extend this and also consider semantics from the text description during the generation process. [22] introduce a dynamic memory part that selects “bad” parts of the initial image and tries to refine them based on the most relevant words. [23] refine the attention module by having spatial and channel-wise word-level attention and introduce a word-level discriminator to provide fine-grained feedback based on individual words and image regions. [24] decompose the text-to-image process into three distinct phases by first learning a prior over the text-image space, then sampling from this prior, and lastly using the prior to generate the image.

Text-to-Image Synthesis with Layouts. When using more complex data sets that contain multiple objects per image, generating an image directly becomes difficult. Therefore, many approaches use additional information such as bounding boxes for objects or intermediate representations such as scene graphs or scene layouts which can be generated automatically [25], [26], [27]. [28] and [29] build on [10] by additionally conditioning the generator on bounding boxes or keypoints of relevant objects. [30] decomposition textual descriptions into basic visual primitives to generate images in a compositional manner. [2] introduce the concept of generating a scene graph based on a caption. This scene graph is then used to generate an image layout and finally the image. Similar to [2], [31] use the caption to infer a scene layout which is used to generate images. [32] predict convolution kernels conditioned on the semantic layout, making it possible to control the generation process based on semantic information at different locations.

Given a coarse image layout (bounding boxes and object labels) [33] generate images by disentangling each object into a specified part (e.g., object label) and unspecified part (appearance). [3] generate images conditioned on bounding boxes for the individual foreground objects by introducing an object pathway that generates individual objects. [4] update the grid-based attention mechanism [7] by combining attention with scene layouts. Additionally, an object discriminator is introduced which focuses on individual objects and provides feedback whether the object is at the right location. [34] refine the grid-based attention mechanism between word phrases and specific image regions of various sizes based on an initial set of bounding boxes. [35] introduce a new feature normalization method and fine-grained mask maps to generate visually different images from a given layout. [36] generate images from scene graphs and allow the model to crop objects from other images to paste them into the generated image. [37] generate a visual-relation scene layout based on the caption. For this, they introduce a dedicated module which generates bounding boxes for objects at a given caption in order to condition the network during the image generation process.

Semantic Image Manipulation. Finally, there are methods that allow humans to directly describe the image in an iterative process or that allow for direct semantic manipulation of images. [12] condition generation process on a dialogue describing the image instead of a single caption. [38] facilitate

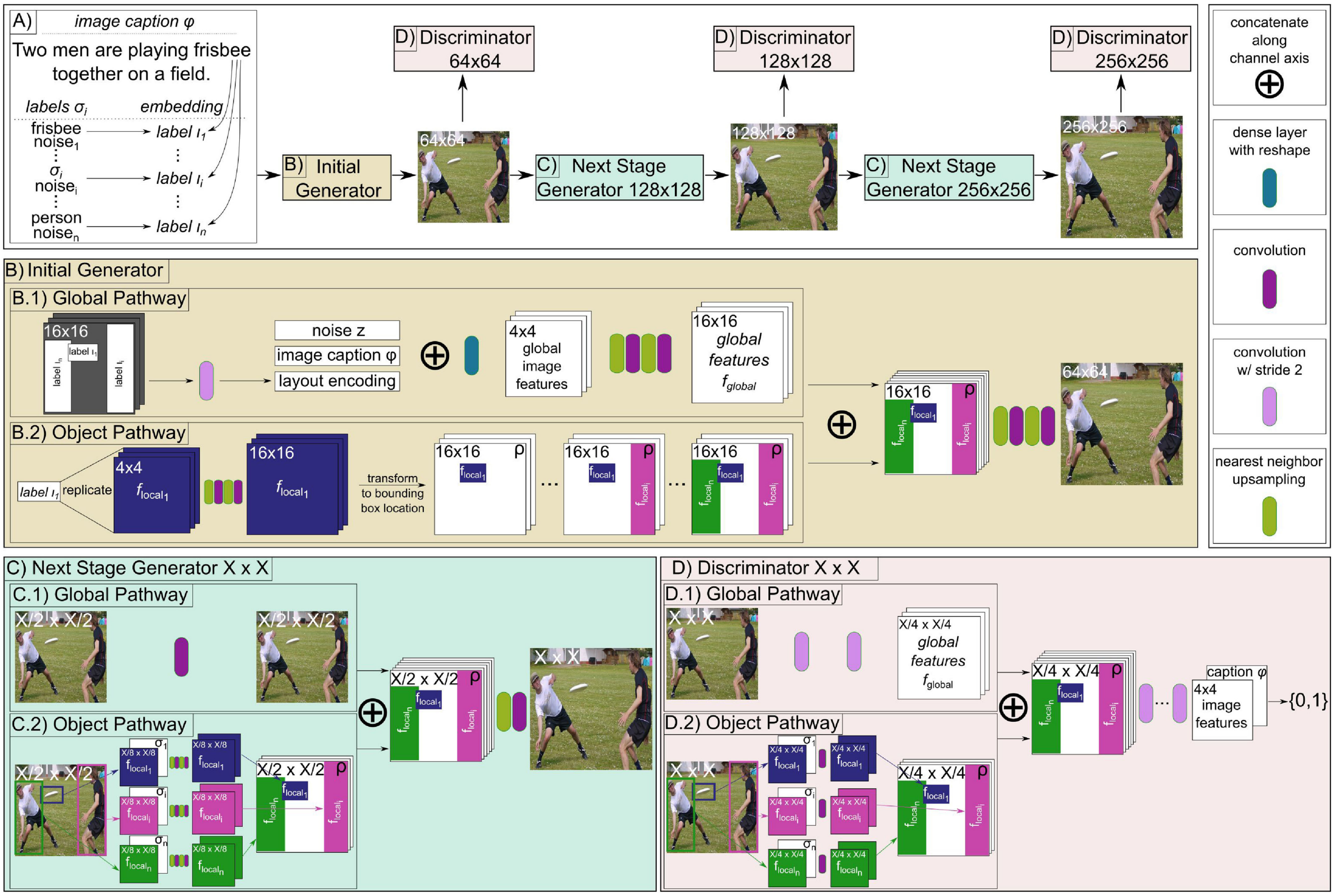

Fig. 1. Overview of our model architecture called *OP-GAN*. The top row shows a high-level summary of our architecture, while the bottom two rows show details of the individual generators and discriminators.

semantic image manipulation by allowing users to modify image layouts which are then used to generate images. [39] allow users to input object instance masks into an existing image represented by a semantic layout. [40] generate images iteratively from consecutive textual commands, [41] provide interactive image editing based on a current image and instructions on how to update the image, and [42] generate individual images for a sequence of sentences. [43] do interactive image generation but do not use text as direct input but instead update a scene graph from text over the course of the interaction. [44], [45], and [46] modify visual attributes of individual objects in an image while leaving text irrelevant parts of the image unchanged.

3 Approach

A traditional generative adversarial network (GAN) [1] consists of two networks: a generator G which generates new data points from randomly sampled inputs, and a discriminator D which tries to distinguish between generated and real data samples. In conditional GANs (cGANs) [47] both the discriminator and the generator are conditioned on additional information, e.g., a class label or textual information. This has been shown to improve performance and leads to more control over the data generating process. For a conventional cGAN with generator G, discriminator D, condition c (e.g., a class label), data point x, and a randomly sampled noise vector z the training objective V is:

$$\min_G \max_D V(D,G) = \mathbb{E}_{(x,c)\sim p_{\text{data}}}[log\ D(x,c)] + \mathbb{E}_{(z)\sim p_z,(c)\sim p_{\text{data}}}[log(1-D(G(z,c),c))]. \quad (1)$$

We use the AttnGAN [7] as our baseline architecture and add our object-centric modifications to it. The AttnGAN is a conditional GAN for text-to-image synthesis that uses attention and a novel additional loss to improve the quality of the generated images. It consists of a generator and three discriminators as shown in the top row of Fig. 1. Attention is used such that different words of the caption have more or less influence on different regions of the image. This means that, for example, the word "sky" has more influence on the generation of the top half of the image than the word "grass" even if both words are present in the image caption.

[7] also introduce the Deep Attentional Multimodal Similarity Model (DAMSM) which computes the similarity between images and captions. This DAMSM is used during training to provide additional, fine-grained feedback to the generator about how well the generated image matches its caption. We adapt the AttnGAN architecture with multiple object pathways which are learned end-to-end in both the discriminator and the generator, see *B* and *C* in Fig. 1.

These object pathways are conditioned on individual object labels (e.g., "person", "car", etc.) and the same object pathway is applied multiple times at a given image resolution at different locations and for different objects. This is similar to the approach introduced by [3]. However, [3] only use one object pathway in the generator at a small resolution and only one discriminator was equipped with an object pathway. In our approach, the generator contains three object pathways at various resolutions (16×16, 64×64, and 128×128) to further refine object features at higher resolutions and each of our three discriminators is equipped with its own object pathway, see D in Fig. 1.

For a given image caption φ we have several objects which are associated with this caption and which we represent with one-hot vectors $\sigma_i, i = 1...n$ (e.g., $\sigma_0 = \text{person}$, $\sigma_1 = \text{car}$, etc.). Each object pathway at a given resolution is applied iteratively for each of the objects σ_i. The location is determined by a bounding box describing the object's location and size. Each object pathway starts with an "empty" zero-tensor ρ and the features that are generated (generator) or extracted (discriminator) are added onto ρ at the location of the specific object's bounding box. After the object pathway has processed each object, ρ contains features at each object location and is zero everywhere else.

For the generator, we first concatenate the image caption's embedding φ, the one-hot label σ_i, and a randomly sampled noise vector z. We use this concatenated vector to obtain the final conditioning label ι_i for the current object σ_i:

$$\iota_i = \mathrm{F}(\varphi, z, \sigma_i), \tag{2}$$

where F is a fully connected layer followed by a non-linearity (A in Fig. 1).

The generator's first object pathway (*B.2* in Fig. 1) takes this conditioning label ι_i and uses it to generate features for the given object at a spatial resolution of 16×16. The features are then transformed onto ρ into the location of the respective bounding box with a spatial transformer network (STN) [48]. This procedure is repeated for each object σ_i associated with the given caption φ.

The global pathway in the first generator also gets the locations and labels ι_i for the individual objects. It spatially replicates these labels at the locations of the respective bounding boxes and then applies convolutional layers to the resulting layout to obtain a layout encoding (*B.1* in Fig. 1). This layout encoding, the image caption φ, and the noise vector z are used to generate coarse features for the image at a low resolution.

At higher levels in the generator, the object pathways are conditioned on the object features of the current object and the one-hot label σ_i for that object (*C.2* in Fig. 1). For this, we again use an STN to extract the features at the bounding box location of the object σ_i and resize the features to a spatial resolution of 16×16 (second object pathway) or 32×32 (third object pathway). We obtain a conditioning label in the same manner as for the first object pathway (Equation (2)), replicate it spatially to the same dimension as the extracted object features, and concatenate it with the object features along the channel axis. Following this, we apply multiple convolutional layers and upsampling to update the features of the given object. Finally, as in the first object pathway, we use an STN to transform the features into the bounding box location and add them onto ρ. The global pathway in the higher layers (*C.1* in Fig. 1) stays unchanged from the baseline architecture [7].

Our final loss function for the generator is the same as in the original AttnGAN and consists of an unconditional, a conditional, and a caption-image matching part. The unconditional loss is

$$\mathcal{L}_G^{\text{uncon}} = -\mathbb{E}_{(\hat{x})\sim p_G}[log\ D(\hat{x}))], \tag{3}$$

the conditional loss is

$$\mathcal{L}_G^{\text{con}} = -\mathbb{E}_{(\hat{x})\sim p_G,(c)\sim p_{\text{data}}}[log\ D(\hat{x}, c))], \tag{4}$$

and the caption-image matching loss is $\mathcal{L}_G^{\text{DAMSM}}$ [7] which measures text-image similarity at the word level and is calculated with the pre-trained models provided by [7]. The complete loss for the generator then is:

$$\mathcal{L}_G = \mathcal{L}_G^{\text{uncon}} + \mathcal{L}_G^{\text{con}} + \lambda\mathcal{L}_G^{\text{DAMSM}}, \tag{5}$$

where we set $\lambda = 50$ as in the original implementation.

As in our baseline architecture, we employ three discriminators at three spatial resolutions: 64×64, 128×128, and 256×256. Each discriminator possesses a global and an object pathway which extract features in parallel (D in Fig. 1). In the object pathway we use an STN to extract the features of object σ_i and concatenate them with the one-hot vector σ_i describing the object. The object pathway then applies multiple convolutional layers before adding the extracted features onto ρ at the location of the bounding box.

The global pathway in each of the discriminators works on the full input image and applies convolutional layers with stride two to decrease the spatial resolution (*D.1*). Once the spatial resolution reaches that of the tensor ρ we concatenate the two tensors (full image features and object features ρ) along the channel axis and use convolutional layers with stride two to further reduce the spatial dimension until we reach a resolution of 4×4.

We calculate both a conditional (image and image caption) and an unconditional (only image) loss for each of the discriminators. The conditional input c during training consists of the image caption embedding φ and the information about objects σ_i (bounding boxes and object labels) associated with the image x, i.e. $c = \{\varphi, \sigma_i\}$. In the unconditional case the discriminators are trained to classify images as real or generated without any influence of the image caption by minimizing the following loss:

$$\mathcal{L}_{D_i}^{\text{uncon}} = -\mathbb{E}_{(x)\sim p_{\text{data}}}[log\ D(x)] - \mathbb{E}_{(\hat{x})\sim p_G}[log(1 - D(\hat{x}))]. \tag{6}$$

In order to optimize the conditional loss we concatenate the extracted features with the image caption embedding φ along the channel axis and minimize

$$\begin{aligned}\mathcal{L}_{D_i}^{\text{con}} = &-\mathbb{E}_{(x,c)\sim p_{\text{data}}}[log\ D(x, c)] \\ &- \mathbb{E}_{(\hat{x})\sim p_G,(c)\sim p_{\text{data}}}[log(1 - D(\hat{x}, c))].\end{aligned} \tag{7}$$

for each discriminator. Finally, to specifically train the discriminators to check for caption-image consistency we use the matching aware discriminator loss [10] with mismatching caption-image pairs and minimize

$$\mathcal{L}_{D_i}^{\text{cls}} = -\mathbb{E}_{(x,\sigma)\sim p_{\text{data}},(\varphi)\sim p_{\text{data}}}[log(1 - D(x, c))], \tag{8}$$

where image x and caption φ are sampled individually and randomly from the data distribution and are, therefore, unlikely to align.

We introduce an additional loss term similar to the matching aware discriminator loss $V_{\text{cls}}(D)$ which works on individual objects. Instead of using mismatching image-caption pairs, we use correct image-caption pairs, but with incorrect bounding boxes and minimize:

$$\mathcal{L}^{\text{obj}}_{D_i} = -\mathbb{E}_{(x,\varphi)\sim p_{\text{data}},(\sigma)\sim p_{\text{data}}}[log(1 - D(x,c))]. \quad (9)$$

Thus, the complete objective we minimize for each individual discriminator is:

$$\mathcal{L}_{D_i} = \mathcal{L}^{\text{uncon}}_{D_i} + \mathcal{L}^{\text{con}}_{D_i} + \mathcal{L}^{\text{cls}}_{D_i} + \mathcal{L}^{\text{obj}}_{D_i}. \quad (10)$$

We leave all other training parameters as in the original implementation [7] and the training procedure itself also stays the same.

4 Evaluation of Text-to-Image Models

Quantitatively evaluating generative models is difficult [49]. While there are several evaluation metrics that are commonly used to evaluate GANs, many of them have known weaknesses and are not designed specifically for text-to-image synthesis tasks. In the following, we first discuss some of the common evaluation metrics for GANs, their weaknesses, and why they might be inadequate for evaluating text-to-image synthesis models. Following this, we introduce our novel evaluation metric, Semantic Object Accuracy (SOA), and describe how it can be used to evaluate text-to-image models in more detail.

4.1 Current Evaluation Metrics

Inception Score and Fréchet Inception Distance. Most GAN approaches are trained on relatively simple images which only contain one object at the center (e.g., ImageNet, CelebA, etc). These methods are evaluated with metrics such as the Inception Score (IS) [5] and Fréchet Inception Distance (FID) [6], which use an Inception-Net usually pre-trained on ImageNet. The IS evaluates roughly how distinctive an object in each image is (i.e. ideally the classification layer of the Inception-Net has small entropy) and how many different objects the GAN generates overall (i.e. high entropy in the output of different images). The FID measures how similar generated images are to a control set of images, usually the validation set by calculating the distance in feature space between generated and real images. Consequently, the IS should be as high as possible, while the FID should be as small as possible.

Both evaluation metrics have known weaknesses [50], [51]. For example, the IS does not measure the similarity between objects of the same class, so a network that only generates one "perfect" sample for each class can achieve a very good IS despite showing an intra-class mode dropping behavior. Li *et al.* [4] also note that the IS overfits within the context of text-to-image synthesis and can be "gamed" by increasing the batch size at the end of the training. Furthermore, the IS uses the output of the classification layer of an Inception-Net pre-trained on the ImageNet data set. This might not be the best approach for a more complex data set in which each image contains multiple objects at distinct locations throughout the image, as opposed to the ImageNet data set which consists of images usually depicting one object in the image center. Fig. 2 shows some exemplary failure cases of the IS on images sampled from the COCO data set.

Fig. 2. Examples when IS fails for COCO images. The top row shows images for which the Inception-Net has very high entropy in its output layer, possibly because the images contain more than one object and are often not centered. The second row shows images containing different objects and scenes which were nonetheless all assigned to the same class by the Inception-Net, thereby negatively affecting the overall predicted diversity in the images.

The FID relies on representative ground truth data to compare the generated data against and also assumes that features are of Gaussian distribution, which is often not the case. For more complex data sets the FID also still suffers from the problem that the image statistics are obtained with a network pre-trained on ImageNet which might not be a representative data set. Finally, neither the IS nor the FID take the image caption into account during their evaluation.

VS Similarity and R-Precision. [19] introduce the visual-semantic similarity (VS similarity) metric which measures the distance between a generated image and its caption. Two models are trained to embed images and captions respectively and then minimize the cosine distance between embeddings of matching image-caption pairs while maximizing the cosine distance between mismatching image-caption pairs. A good model then achieves high VS similarity between a generated image and its associated caption.

[7] use the R-precision metric to evaluate how well an image matches a given description or caption. The R-precision score is similar to VS similarity, but instead of scoring the VS similarity between a given image and caption it instead performs a ranking of the similarity between the real caption and randomly sampled captions for a given generated image. For this, first, an image is generated conditioned on a given caption. Then, another 99 captions are chosen randomly from the data set. Both the generated images and the 100 captions are then encoded with the respective image and text encoder. Similar to VS similarity the cosine distance between the image embedding and each caption embedding is used as proxy for the similarity between the given image and caption. The 100 captions are then ordered in descending similarity and the top k (usually k=1) most similar captions are used to calculate the R-precision. Intuitively, R-precision calculates if the real caption is more similar to the generated image (in feature space) than 99 randomly sampled captions.

The drawback of both metrics is that they do not evaluate the quality of individual objects. For example the real caption

Fig. 3. Examples when R-precision fails for COCO images. The top row shows images from the COCO data set. The middle row shows the correct caption and the bottom row gives examples for characteristics of captions that are rated as being more similar than the original caption.

could state that *"a person stands on a snowy hill"* while the 99 random captions do not mention "snow" (which usually covers most of the background in the generated image) or "person" (but e.g., giraffe, car, bedroom, etc). In this case, an image with only white background (snow) would already make the real caption rank very highly in the R-precision metric and achieve a high VS similarity. See Fig. 3 for a visualization of this. As such, this metric does not focus on the quality of individual objects but rather concentrates on global background and salient features.

Classification Accuracy Score. [52] introduce the Classification Accuracy Score (CAS) to evaluate conditional image generation models, similar to [53]. For this, a classifier is trained on images generated by the conditional generative model. The classifier's performance is then evaluated on the original test set of the data set that was used to train the generative model. If the classifier achieves high accuracy on the test set, this indicates that the data it was trained on is representative of the real distribution. The authors find that neither the IS, the FID, nor combinations thereof are predictive of the CAS, further indicating that the IS and FID are only of limited use for evaluating image quality.

Caption Generation. [31] suggest evaluating text-to-image models by comparing original captions with captions obtained from generated images. The intuition is that if the generated image is relevant to its caption, then it should be possible to infer the original text from it. To this end, [31] use a pre-trained caption generator [54] to generate captions for each synthesized image and compare these to the original ones through standard language similarity metrics, i.e. BLEU, METEOR, and CIDEr. Except for CIDEr, these metrics were originally developed to evaluate machine translation and text summarization methods and were only later adopted for the evaluation of image captions.

One challenge with this caption generation approach is that often many different captions are valid for a given image. Even if two captions are not similar, this does not necessarily imply that they do not describe the same image [55]. Furthermore, it has been shown that metrics such as BLEU, METEOR, and CIDEr are primarily sensitive to n-gram overlap which is neither necessary nor sufficient for two sentences to convey the same meaning [55], [56], [57] and do also not necessarily correlate with human judgments of captions [54], [58]. Finally, there is no requirement that captions, either real or generated, need to focus on specific objects. Instead, captions can also describe the general layout of a given scene (e.g., *a busy street with lots of traffic*) without explicitly mentioning specific objects. Some of these limitations might potentially be overcome in the future by novel image caption evaluation metrics that focus more on objects and semantic content in the scene [55], [57], [59].

Other Approaches. In contrast to the IS, which measures the diversity of a whole set of images, the diversity score [33] measures the perceptual difference between a pair of images in feature space. This metric can be useful when images are generated from conditional inputs (e.g., labels or scene layouts) to examine whether a model can generate diverse outputs for a given condition. However, the metric does not convey anything directly about the quality of the generated images or their congruence with any conditional information. [14], [60], [61] run a semantic segmentation network on generated images and compare the predicted segmentation mask to the ground truth segmentation mask used as input for the model. However, this metric needs a ground truth semantic segmentation mask and does not provide information about specific objects within the image.

4.2 Semantic Object Accuracy (SOA)

So far, most evaluation metrics are designed to evaluate the holistic image quality but do not evaluate individual areas or objects within an image. Furthermore, except for *Caption Generation* and *R-precision*, none of the scores take the image caption into account when evaluating generated images. To address the challenges and issues mentioned above we introduce a novel evaluation metric based on a pre-trained object detection network.[1] The pre-trained object detector evaluates images by checking if it recognizes objects that the image should contain based on the caption. For example, if the image caption is *"a person is eating a pizza"* we can infer that the image should contain both a person and a pizza and the object detector should be able to recognize both objects within the image. Since this evaluation measures directly whether objects specifically mentioned in the caption are recognizable in an image we call this metric *Semantic Object Accuracy* (SOA).

Some previous works have used similar approaches to evaluate the quality of the generated images. [3] evaluate how often expected objects (based on the caption) are detected by an object detector. However, only a subset of the captions is evaluated and the evaluated captions contain false positives (e.g., captions containing the phrase "hot dog" are evaluated based on the assumption that the image should contain a dog). [15] introduce a detection score that calculates (roughly) whether a pre-trained object detector detects an object in a generated image with high certainty. However, no information from the caption is taken into account, meaning any detection with high confidence is "good" even if the detected object does not make sense in the context of the caption. [62] use a pre-trained object detector to calculate the mean average precision and report precision-recall curves. However, the evaluation is done on synthetic data sets and without textual information as

1. Code for the evaluation metric and all experiments: https://github.com/tohinz/semantic-object-accuracy-for-generative-text-to-image-synthesis

conditional input. [33] use classification accuracy as an evaluation metric in which they report the object classification accuracy in generated images. For this, they use a ResNet-101 model which is trained on real objects cropped and resized from the original data. However, in order to calculate the score, the size and location of each object in the generated image must be known, so this evaluation is not directly applicable to approaches that do not use scene layouts or similar representations. [37] use recall and intersection-over-union (IoU) to evaluate the bounding boxes in their generated scene layout but do not apply these evaluations to generated images directly.

SOA. Since we work with the COCO data set we filter all captions in the validation set for specific keywords that are related to the available labels for objects (e.g., person, car, zebra, etc). For each of the 80 available labels in the COCO data set we find all captions that imply the existence of the respective object and generate three images for each of the captions. The supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.3021209, gives a detailed overview of how exactly the captions were chosen for each label. We then run the YOLOv3 network [8] pre-trained on the COCO data set on each of the generated images and check whether it recognizes the given object. We report the recall as a class average (SOA-C), i.e. in how many images per class the YOLOv3 on average detects the given object, and as an image average (SOA-I), i.e. on average in how many images a desired object was detected. Specifically, the SOA-C is calculated as

$$\text{SOA-C} = \frac{1}{|C|}\sum_{c\in C}\frac{1}{|I_c|}\sum_{i_c\in I_c}\text{YOLOv3}(i_c), \tag{11}$$

for object classes $c \in C$ and images $i \in I_c$ that are supposed to contain an object of class c. The SOA-I is calculated as

$$\text{SOA-I} = \frac{1}{\sum_{c\in C}|I_c|}\sum_{c\in C}\sum_{i_c\in I_c}\text{YOLOv3}(i_c), \tag{12}$$

and

$$\text{YOLOv3}(i_c) = \begin{cases} 1 & \text{if YOLOv3 detected an object of class } c \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Since many images can also contain objects that are not specifically mentioned (for example an image described by *"lots of cars are on the street"* could still contain persons, dogs, etc.) in the caption we do not calculate a false negative rate but instead only focus on the recall, i.e. the true positives.

SOA-Intersection Over Union. Several approaches (e.g., [3], [4], [31], [33], [37]) use additional conditioning information such as scene layouts or bounding boxes. For these approaches, our evaluation metric can also calculate the intersection over union (IoU) between the location at which different objects should be and locations at which they are detected, which we call SOA-IoU. To calculate the IoU we use every image in which the YOLOv3 network detected the respective object. Since many images contain multiple instances of a given object we calculate the IoU between each predicted bounding box for the given object and each ground truth bounding box. The final IoU for a given image and object is then the maximum of the values, i.e. the reported IoU is an upper bound on the actual IoU.

Overall this approach allows a more fine-grained evaluation of the image content since we can now focus on individual objects and their features. To get a better idea of the overall performance of a model we calculate both the class average recall/IoU (SOA-C/SOA-IoU-C) and image average recall/IoU (SOA-I/SOA-IoU-I). Additionally, we report the SOA-C for the forty most and least common labels (SOA-C-Top40 and SOA-C-Bot40) to see how well the model can generate objects of common and less common classes.

5 Experiments

We perform multiple experiments and ablation studies. In a first step, we add the object pathway (OP) on multiple layers of the generator and to each discriminator and call this model *OPv2*. We also train this model with the additional bounding box loss we introduced in Section 3. When the model is trained with the additional bounding box loss we refer to it as *BBL*.

Different approaches differ in how many objects per image are used during training. If an image layout is used, typically all objects (foreground and background) are used as conditioning information. Other approaches limit the number of objects during per training [2], [3]. To examine the effect of training with different numbers of objects per image we train our approach with either a maximum of three objects per image (standard) or with up to ten objects per image, which we refer to as many objects (*MO*). When training with a maximum of three objects per image we sample randomly from the training set at train time, i.e. each batch contains images which contain zero to three objects. If an image contains more than three objects we choose the three largest ones in terms of area of the bounding box. When training with up to ten objects per image we slightly change our sampling strategy so that each batch consists of images that contain the same amount of objects. This means that, e.g., each image in a batch contains exactly four objects, while in the next batch each image might contain exactly seven objects. This increases the training efficiency as most of the images contain fewer than five objects.

As a result of the different settings we perform the following experiments:

1) *OPv2*: apply the object pathway (OP) on multiple layers of the generator and on all discriminators, training without the bounding box loss and with a maximum of three objects per image.
2) *OPv2 + BBL*: same as *OPv2* but with the bounding box loss added to the discriminator loss term.
3) *OPv2 + MO*: same as *OPv2* but with a maximum of ten objects per image.
4) *OPv2 + BBL + MO* (*OP-GAN*): combination of all three approaches.

We train each model three times on the 2014 split of the COCO data set. At test time we use bounding boxes generated by a network [4] as the conditioning information. Therefore, except for the image caption no other ground truth information is used at test time.

TABLE 1
Inception Score (IS), Fréchet Inception Distance (FID), R-Precision, Caption Generation With CIDEr, and Semantic Object Accuracy on Class (SOA-C) and Image Average (SOA-I) on the MS-COCO Data Set

Model	IS $\uparrow$	FID $\downarrow$	R-precision (k=1) $\uparrow$	CIDEr $\uparrow$	SOA-C $\uparrow$	SOA-I $\uparrow$
Original Images	34.88 ± 0.01	6.09 ± 0.05	68.58 ± 0.08	0.795 ± 0.003	74.97	80.84
AttnGAN [7]†	23.61 ± 0.21	33.10 ± 0.11	83.80	0.695 ± 0.005	25.88	39.01
[34]	23.74 ± 0.36		86.44 ± 3.38			
ControlGAN [23]	24.06 ± 0.60		82.43			
AttnGAN + OP [3]†	24.76 ± 0.43	33.35 ± 1.15	82.44	0.689 ± 0.008	25.46	40.48
MirrorGAN [16]	26.47 ± 0.41		74.52			
Obj-GAN [4]†	24.09 ± 0.28	36.52 ± 0.13	87.84 ± 0.08	0.783 ± 0.002	27.14	41.24
HfGAN [20]	27.53 ± 0.25					
DM-GAN [22]†	32.32 ± 0.23	27.34 ± 0.11	$\mathbf{91.87 \pm 0.28}$	$\mathbf{0.823 \pm 0.002}$	33.44	48.03
SD-GAN [21]	$\mathbf{35.69 \pm 0.50}$					
OP-GAN (Best Model)	27.88 ± 0.12	$\mathbf{24.70 \pm 0.09}$	89.01 ± 0.26	0.819 ± 0.004	**35.85**	**50.47**
OPv2, 0 obj	26.80 ± 1.01	30.01 ± 1.81	83.87 ± 1.22	0.760 ± 0.004	26.04 ± 1.47	37.56 ± 1.27
OPv2, 1 obj	27.68 ± 0.47	26.18 ± 0.27	87.37 ± 0.60	0.798 ± 0.013		
OPv2, 3 obj	27.78 ± 0.50	26.45 ± 0.40	87.74 ± 1.08	0.805 ± 0.011		
OPv2, 10 obj	27.66 ± 0.34	26.52 ± 0.44	87.73 ± 0.98	0.806 ± 0.006	33.82 ± 0.69	48.39 ± 1.01
OPv2 + BBL, 0 obj	24.60 ± 1.25	33.03 ± 0.76	81.27 ± 1.45	0.735 ± 0.029	24.00 ± 2.13	34.01 ± 2.89
OPv2 + BBL, 1 obj	26.34 ± 0.55	26.59 ± 1.04	86.42 ± 0.60	0.783 ± 0.006		
OPv2 + BBL, 3 obj	26.52 ± 0.47	26.74 ± 1.08	87.08 ± 0.60	0.793 ± 0.013		
OPv2 + BBL, 10 obj	26.48 ± 0.58	26.83 ± 1.10	86.80 ± 0.56	0.794 ± 0.015	33.19 ± 0.40	48.24 ± 0.68
OPv2 + MO, 0 obj	24.32 ± 1.65	35.36 ± 1.95	79.75 ± 1.87	0.695 ± 0.015	21.15 ± 1.47	30.24 ± 2.36
OPv2 + MO, 1 obj	27.36 ± 0.49	25.06 ± 1.11	88.33 ± 0.81	0.789 ± 0.008		
OPv2 + MO, 3 obj	27.65 ± 0.37	24.96 ± 1.12	89.13 ± 0.42	0.807 ± 0.014		
OPv2 + MO, 10 obj	27.59 ± 0.43	$\mathbf{24.94 \pm 1.09}$	$\mathbf{89.14 \pm 0.41}$	0.805 ± 0.013	33.46 ± 1.01	47.93 ± 1.56
OPv2 + BBL + MO, 0 obj	21.84 ± 0.83	45.79 ± 1.16	72.71 ± 1.75	0.626 ± 0.025	16.55 ± 1.81	22.76 ± 2.17
OPv2 + BBL + MO, 1 obj	27.61 ± 0.67	26.19 ± 0.82	87.85 ± 0.25	0.791 ± 0.009		
OPv2 + BBL + MO, 3 obj	$\mathbf{28.04 \pm 0.65}$	25.91 ± 1.03	88.90 ± 0.24	0.810 ± 0.009		
OPv2 + BBL + MO, 10 obj	27.90 ± 0.79	25.80 ± 1.01	89.00 ± 0.17	$\mathbf{0.814 \pm 0.007}$	$\mathbf{34.51 \pm 1.12}$	$\mathbf{48.90 \pm 0.72}$

Results of our models are obtained with generated bounding boxes. Scores for models marked with † were calculated with a pre-trained model provided by the respective authors.

6 Evaluation and Analysis

Tables 1 and 2 give an overview of our results for the COCO data set. The first half of the table shows the results on the original images from the data set and from related literature while the second half shows our results. To make a direct comparison we calculated the IS, FID, CIDEr, and R-precision scores ourselves for all models which are provided by the authors. As such, the values from AttnGAN [7], AttnGAN+OP [3], Obj-GAN [4], and DM-GAN [22] are the ones most directly comparable to our reported values since they were calculated in the same way.

Note that there is some inconsistency in how the FID is calculated in prior works. Some approaches, e.g., [4], compare the statistics of the generated images only with the statistics of the respective "original" images (i.e. images corresponding to the captions that were used to generate a given image). We, on the other hand, generate 30,000 images from 30,000 randomly sampled captions and compare their statistics with the statistics of the full validation set. Many of the recent publications also do not report the FID or R-precision. This makes a direct comparison difficult as we show that the IS is likely the least meaningful score of the three since it easily overfits [4] and due to the reasons mentioned in Section 4. We calculate each of the reported values of our models three times for each trained model (nine times in total) and report the average and standard deviation. To calculate the SOA scores we generate three images for each caption in the given class, except for the "person" class, for which we randomly sample 30,000 captions (from over 60,000) and generate one image for each of the 30,000 captions.

6.1 Quantitative Results

Overall Results. As Table 1 shows, all our models outperform the baseline AttnGAN in all metrics. The IS is improved by 16-19 percent, the R-precision by 6-7 percent, the SOA-C by 28-33 percent, the SOA-I by 22-25 percent, the FID by 20-25 percent, and CIDEr by 15-18 percent. This was achieved by adding our object pathways to the baseline model without any further tuning of the architecture, hyperparameters, or the training procedure. Our approach also outperforms all other approaches based on FID, SOA-C, and SOA-I. While there are two approaches that report a IS higher than our models, it has previously been observed that this score is likely the least meaningful for this task and can be gamed to achieve higher numbers [4], [51]. Our user study also shows that the IS is the score that has the least predictive value for human evaluation.

We also calculated each score using the original images of the COCO data set. For the IS we sampled three times 30,000 images from the validation set and resized them to 256×256 pixels. These images were also used to calculate the CIDEr score. To calculate the FID we randomly sampled three times 30,000 images from the training set and compared them to

TABLE 2
Comparison of the Recall Values for the Different Models

Model	SOA-C / IoU	SOA-I / IoU	SOA-C-Top40 / IoU	SOA-C-Bot40 / IoU
Original Images	74.97 / 0.550	80.84 / 0.570	78.77 / 0.546	71.18 / 0.554
AttnGAN [7]	25.88 / −−	39.01 / −−	37.47 / −−	14.29 / −−
AttnGAN + OP [3]	25.46 / 0.236	40.48 / 0.311	39.77 / 0.308	11.15 / 0.164
Obj-GAN [4]	27.14 / 0.513	41.24 / 0.598	39.88 / 0.587	14.40 / 0.438
DM-GAN [22]	33.44 / −−	48.03 / −−	47.73 / −−	19.15 / −−
OPv2	33.82 (26.04) / 0.207	48.39 (37.56) / 0.270	48.34 (36.53) / 0.260	19.31 (15.55) / 0.152
OPv2 + BBL	33.19 (24.00) / 0.210	48.24 (34.01) / 0.270	47.96 (32.96) / 0.261	18.43 (15.04) / 0.159
OPv2 + MO	33.46 (21.15) / 0.214	47.93 (30.24) / 0.275	47.84 (28.15) / 0.264	19.07 (14.15) / 0.163
OPv2 + BBL + MO	34.51 (16.55) / 0.217	48.90 (22.76) / 0.278	49.70 (22.19) / 0.269	19.32 (10.91) / 0.165

We used generated bounding boxes to calculate the values. Numbers in brackets show scores when the object pathway was not used at test time.

the statistics of the validation set. The R-precision was calculated on three times 30,000 randomly sampled images and the corresponding caption from the validation set and the SOA-C and SOA-I were calculated on the real images corresponding to the originally chosen captions.

As we can see, the IS is close to the current state of the art models with a value of 34.88. It is possible to achieve a much higher IS on other, simpler data sets, e.g., IS > 100 on the ImageNet data set [63]. This indicates that the IS is indeed not a good evaluation metric, especially for complex images consisting of multiple objects and various locations. The difference between the R-precision on real and generated images is even larger. On the original images, the R-precision score is only 68.58, which is much worse than what current models can achieve (> 88).

One reason for this might be that the R-precision calculates the cosine similarity between an image embedding and a caption embedding and measures how often the caption that was used to generate an image is more similar than 99 other, randomly sampled captions. However, the same encoders that are used to calculate the R-precision are also used during training to minimize the cosine similarity between an image and the caption it was generated from. As a result, the model might already overfit to this metric through the training procedure. Our observation is that the models tend to heavily focus on the background to make it match a specific word in the caption (e.g., images tend to be very white when the caption mentions "snow" or "ski", very blue when the caption mentions "surf" or "beach", very green when the caption mentions "grass" or "savanna", etc.) This matching might lead to a high R-precision score since it leads, on average, to a large cosine similarity. Real images do not always reflect this, since a large part of the image might be occupied by a person or an animal, essentially "blocking out" the background information. We see a similar trend for the CIDEr evaluation where many models achieve a score similar to the score reached by real images. Regardless of what the actual reason is, the question remains whether evaluation metrics like the IS, R-precision, and CIDEr are meaning- and helpful when models that can not (as of now) generate images that would be confused as "real" achieve scores comparable to or better than real images.

The FID and the SOA values are the only two evaluation metrics (that we used) for which none of the current state of the art models can come close to the values obtained with the original images. The FID is still much smaller on the real data (6.09) compared to what current models can achieve (> 24 for the best models). While the FID still uses a network pre-trained on ImageNet it compares activations of convolutional layers for different images and is, therefore, likely still more meaningful and less dependent on specific object settings than the IS. Similarly, the SOA-C (SOA-I) on real data is 74.97 (80.84), while current models achieve values of around $30 - 36$ $(40 - 50)$. Since the network used to calculate the SOA values is not part of the training loop the models can not easily overfit to this evaluation metric like they can for the R-precision. Furthermore, the results of the SOA evaluation confirm the impression that none of the models is able to generate images with multiple distinct objects of a quality similar to real images.

Impact of the Object Pathway. To get a clearer understanding of how the evaluation metrics might be impacted by the object pathway we calculate our scores for a different number of generated objects. More specifically, we only apply the object pathway for a maximum given number of objects (0, 1, 3, or 10) per image. Intuitively, we would assume that without the application of the object pathway the IS and FID should be decreased, since the object pathway is not used to generate any object features and the images should, therefore, consist mostly of background. Additionally, we can get an intuition of how important the object pathway is for the overall performance of the network by looking at how it affects the R-precision, SOA-C, and SOA-I.

As Table 1 shows, all models perform markedly worse when the object pathway is not used (0 obj). We find that the models trained with up to ten objects per image seem to rely more heavily on the object pathway than models trained with three objects per image. For models trained with only three objects per image (*OPv2* and *OPv2 + BBL*) the IS decreases by around $1 - 2$, the R-precision decreases by around $4 - 5$, the SOA-C (SOA-I) decreases by around $7 - 9$ $(11 - 14)$, CIDEr decreases by around 6-8 percent, and the FID increases by around $4 - 7$. On the other hand, models trained with up to 10 objects suffer much more when the object pathway is removed, with the IS decreasing by around $3 - 6$, the R-precision decreasing by around $9 - 15$, the SOA-C (SOA-I) decreasing by around $12 - 18$ $(17 - 28)$, CIDEr decreasing by around 16-30 percent, and the FID increasing by around $10 - 20$. These results indicate that the object pathways are an important part of the model and are responsible for at least some of the improvements compared to the baseline architecture.

Impact of Bounding Box Loss. Adding the bounding box loss to the object pathways has a small negative effect on all scores, but does slightly improve the IoU scores (see Table 2). Note that the weighting of the bounding box loss in the overall loss term was not optimized but simply weighted with the same strength as the matching aware discriminator loss $\mathcal{L}_D^{\text{cls}}$. It is possible that the positive effect of the bounding box loss could be increased by weighting it differently.

Impact of Training on Many Objects. Training the model with up to ten objects per image has only minor effects on the IS and SOA scores, but improves the FID and R-precision. However, we observe that the models trained with only three objects per image slightly decrease in their performance once the object pathway is applied multiple times. Usually, the models trained on only three objects achieve their best performance when applying the object pathway three times as at training time. Once the model is trained on up to ten objects though, we do not observe this behavior anymore and instead achieve comparable or even better results when applying the object pathway ten times per image.

SOA Scores. Table 2 shows the results for the SOA and SOA-IoU. The SOA-I values are consistently higher than the SOA-C values. Since the SOA-I is calculated on image average (instead of class average like the SOA-C) it is skewed by objects that often occur in captions and images (e.g., persons, cats, dogs, etc.). The SOA values for the most and least common 40 objects show that the models perform much better on the more common objects. Actually, most models perform about two times better on the common objects showing their problem in generating objects that are not often observed during training. For a detailed overview of how each model performed on the individual labels please refer to the supplementary material, available online.

When we look at the IoU scores we see that the Obj-GAN [4] achieves by far the best IoU scores (around 0.5), albeit at the cost of lower SOA scores. Our models usually achieve an IoU of around $0.2 - 0.3$ on average. Training with up to ten objects per image and using the bounding box loss slightly increases the IoU. However, similar to previous work [3], [4] we find that the AttnGAN architecture tends to place salient object features at many locations of the image which affects the IoU scores negatively.

When looking at the SOA for individual objects (see Fig. 5) we find that there are objects for which we can achieve very high SOA values (e.g., person, cat, dog, zebra, pizza, etc.). Interestingly, we find that all tested methods perform “good” or “bad” at the same objects. For example, all models perform reasonably well on objects such as *person* and *pizza* (many examples in the training set) as well as e.g., *plane* and *traffic light* (few examples in the training set). Conversely, all models fail on objects such as *table* and *skateboard* (many examples in the training set) as well as e.g., *hair drier* and *toaster* (few examples in the training set).

We found that objects need to have three characteristics to achieve a high SOA and the highest SOA scores are achieved when objects possess all three characteristics. The first important characteristic is easily predictable: the higher the occurrence of an object in the training data, the better (on average) the final performance on this object. Second, large objects, i.e. objects that usually cover a large part of the image (e.g., *bus* or *elephant*), are usually modeled better than objects that are usually small (*spoon* or *baseball glove*). The final and more subtle characteristic is the surface texture of an object. Objects with highly distinct surface textures (e.g., *zebra*, *giraffe*, *pizza*, etc.) achieve high SOA scores because the object detection network relies on these textures to detect objects. However, while the models are able to correctly match the surface texture (e.g., black and white stripes for a zebra) they are still not capable of generating a realistic-looking shape of many objects. As a result, many of these objects possess the “correct” surface texture but their shape is more a general “blob” consisting of the texture and not a distinct form (e.g., a snout and for legs for a zebra). See Fig. 6 for a visualization of this.

This is one of the weaknesses of the SOA score as it might give the wrong impression that an 80 percent object detection rate means in 80 percent of the cases the object is recognizable and of real-world quality. This is not the case, as the SOA scores are calculated with a pre-trained object detector which might focus more on texture and less on shapes of objects [64]. Consequently, the results of the SOA are more aptly interpreted as cases where a model was able to generate features that an independently pre-trained object detector would classify as a given object. The overall quality of the metric is, therefore, strongly dependent on the object detector and future improvements in this area might also lead to more meaningful interpretations of the SOA scores.

Fig. 4 shows images generated by our different models. All images shown in this paper were generated without ground truth bounding boxes but instead use generated bounding boxes [4]. The first column shows the respective image from the data set, while the next four columns show the generated images. We can see that all models are capable of generating recognizable foreground objects. It is often difficult to find qualitative differences in the images generated by the different models. However, we find that the models using the bounding box loss usually improve the generation of rare objects. Training with ten objects per image usually leads to a slightly better image quality overall, especially for images that contain many objects.

As we saw in the quantitative evaluation, the object pathway can have a large impact on the image quality. Fig. 7 shows what happens when (some of) the object pathways are not used in the full model (*OPv2 + BBL + MO*). Again, the first column shows the original image from the data set and the second column shows images generated without the use any of the object pathways. The next three columns show generated images when we consecutively use the object pathways, starting with the lowest object pathway and iteratively adding the next object pathway until we reach the full model. When no object pathway is used (first column) we clearly see that only background information is generated. Once the first object pathway is added we also get foreground objects and their quality gets slightly better by adding the higher-level object pathways.

User Study. In order to further validate our results, we performed a user study on Amazon Mechanical Turk. Similar to other approaches [9], [21], [31] we sampled 5,000 random captions from the COCO validation set. For each caption, we generated one image with each of the following models: our OP-GAN, the AttnGAN [7], the AttnGAN-OP

Fig. 4. Comparison of images generated by different variations of our models.

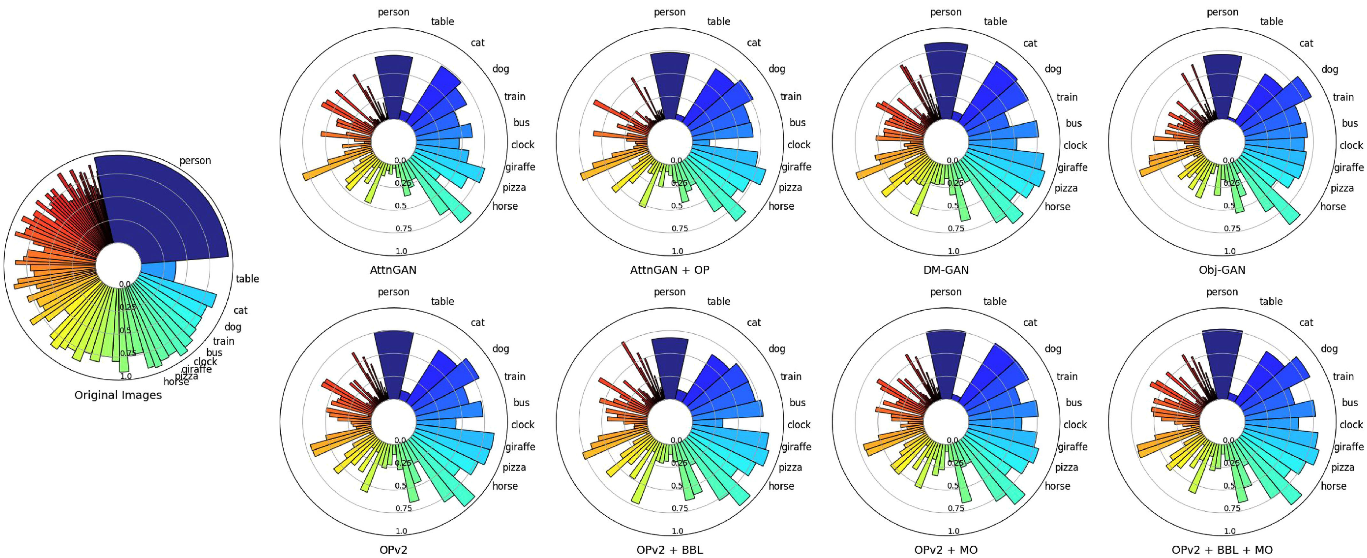

Fig. 5. Comparison of SOA scores: SOA per class with degree of a bin reflecting relative frequency of that class.

[3], the Obj-GAN [4], and the DM-GAN [22]. We showed each user a given caption and the respective five images (without time limit) from the models in random order and asked them to choose the image that depicts the given caption best. We evaluated each image caption twice, for a total of 10,000 evaluations with the help of 200 participants.

Table 3 shows how often each model was chosen as having produced the best image given a caption (variance was estimated by bootstrap [65]). This evaluation reveals that

Fig. 6. Generated images and objects recognized by the pre-trained object detector (YOLOv3) which was used to calculate the SOA scores. The results highlight that, like most other CNN based object detectors, YOLOv3 focuses much more on texture and less on actual shapes.

TABLE 3
Human Evaluation Results (Ratio of 1st by Human Ranking) of Five Models on the MS-COCO Data Set Given a Caption

AttnGAN-OP [3]	$14.65\% \pm 0.35$
AttnGAN [7]	$16.80\% \pm 0.43$
Obj-GAN [4]	$20.96\% \pm 0.33$
DM-GAN [22]	$22.42\% \pm 0.41$
OP-GAN (ours)	**$25.17\% \pm 0.43$**

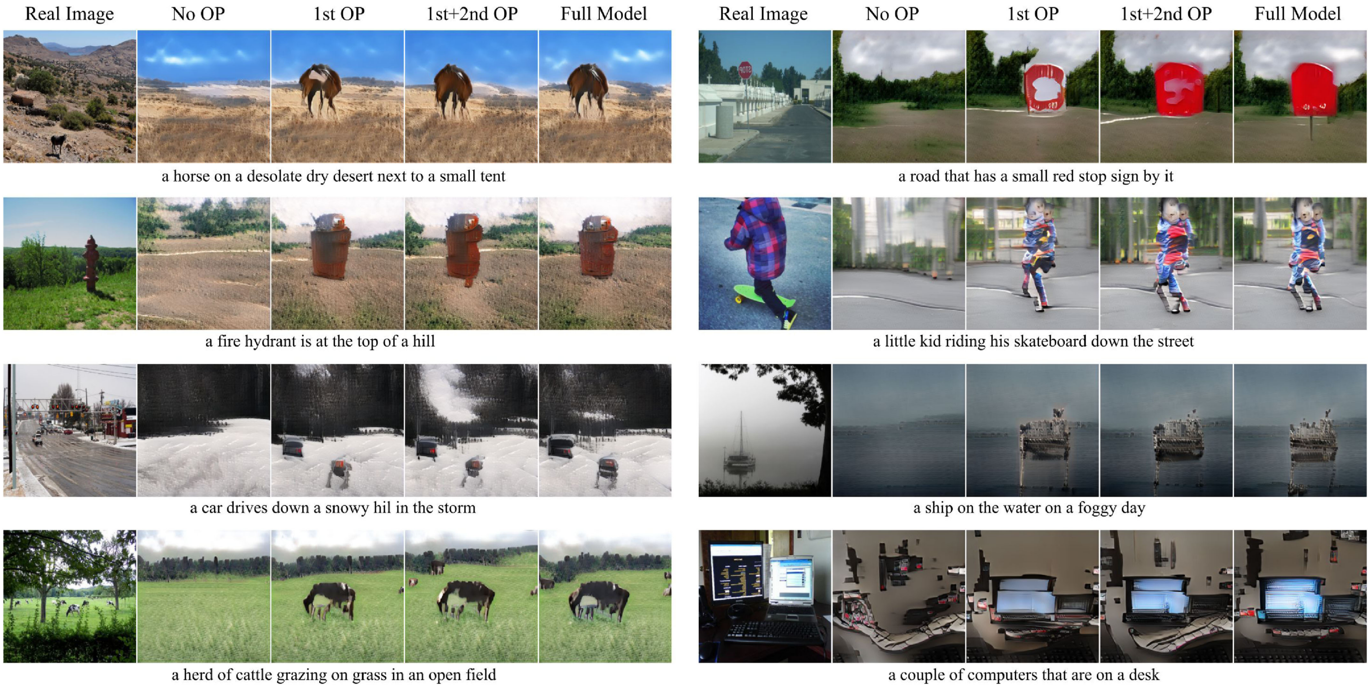

Fig. 7. Comparison of images generated by our model (*OP-GAN*) with OPs switched on and off.

Fig. 8. Comparison of images generated by our model (*OP-GAN*) with images generated by other current models.

the human ranking closely reflects the ranking obtained through the SOA and FID scores. One notable exception are the two worst performing models (AttnGAN and Attn-GAN-OP), which we measure to perform similar according to the SOA and FID scores, but obtain different results in the user study. We find that the IS score is not predictive of the performance in the user study. The R-precision and CIDEr are somewhat predictive, but predict a different ranking of the top-three performing models. Overall, we find that our OP-GAN performs best according to both the SOA scores and the human evaluation. As hypothesized in Section 4 we also observe that the FID and SOA scores are the best predictors for a model's performance in a human user evaluation.

6.2 Qualitative Results

Fig. 8 shows examples of images generated by our model (*OPv2 + BBL + MO*) and those generated by several other models [3], [4], [7], [22]. We observe that our model often generates images with foreground objects that are more recognizable than the ones generated by other models. For more common objects (e.g., person, bus or plane) all models manage to generate features that resemble the object but in most cases do not generate a coherent representation from these features and instead distribute them throughout the image. As a result, we notice features that are associated with an object but not necessarily form one distinct and coherent appearance of that object. Our model, on the other hand, is often able to generate one (or multiple) coherent object(s) from the features, see e.g., the generated images containing a bus, cattle, or the plane.

When generating rare objects (e.g., cake or hot dog) we observe that our model generates a much more distinct object than the other models. Indeed, most models fail completely to generate rare objects and instead only generate colors associated with these objects. Finally, when we inspect more complex scenes we see that our model is also capable of generating multiple diverse objects within an image. As opposed to the other images for "*room showing a sink and some drawers*" we can recognize a sink-like shape and drawers in the image generated by our model. Similarly, our model can also generate an image containing a reasonable shape of a banana and a cup of coffee, whereas the other models only seem to generate the texture of a banana without the shape and completely ignore the cup of coffee.

7 Conclusion

In this paper, we introduced a novel GAN architecture (*OP-GAN*) that specifically models individual objects based on some textual image description. This is achieved by adding object pathways to both the generator and discriminator which learn features for individual objects at different resolutions and scales. Our experiments show that this consistently improves the baseline architecture based on quantitative and qualitative evaluations.

We also introduce a novel evaluation metric named *Semantic Object Accuracy* (SOA) which evaluates how well a model can generate individual objects in images. This new SOA evaluation allows to evaluate text-to-image synthesis models in more detail and to detect failure and success modes for individual objects and object classes. A user study with 200 participants shows that the SOA score is consistent with the ranking obtained by human evaluation, whereas other scores such as the Inceptions Score are not. Evaluation of several state-of-the-art approaches using SOA shows that no current approach is able to generate realistic foreground objects for the 80 classes in the COCO data set. While some models achieve high accuracy for several of the most common objects, all of them fail when it comes to modeling rare objects or objects that do not have an easily recognizable surface structure. However, using the SOA as an evaluation metric on text-to-image models provides more detailed information about how well they perform for different object classes or image captions and is well aligned with human evaluation.

Acknowledgments

The authors would like to thank the partial support from the German Research Foundation DFG under project CML (TRR 169). They would also like to thank the NVIDIA Corporation for their support through the GPU Grant Program.

References

[1] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Advances Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[2] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1219–1228.
[3] T. Hinz, S. Heinrich, and S. Wermter, "Generating multiple objects at spatially distinct locations," in *Proc. Int. Conf. Learn. Representations*, 2019.
[4] W. Li *et al.*, "Object-driven text-to-image synthesis via adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12 174–12 182.
[5] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
[6] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.
[7] T. Xu *et al.*, "AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1316–1324.
[8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018, *arXiv: 1804.02767*.
[9] H. Zhang *et al.*, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.
[10] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1060–1069.
[11] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Proc. Europ. Conf. Comput. Vis.*, 2014, pp. 740–755.
[12] S. Sharma, D. Suhubdy, V. Michalski, S. E. Kahou, and Y. Bengio, "Chatpainter: Improving text to image generation using dialogue," in *Proc. Int. Conf. Learn. Representations Workshop*, 2018.
[13] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," 2016, *arXiv:1612.00215*.
[14] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2337–2346.
[15] S. Sah *et al.*, "Semantically invariant text-to-image generation," in *Proc. IEEE Conf. Image Process.*, 2018, pp. 3783–3787.
[16] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Mirrorgan: Learning text-to-image generation by redescription," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1505–1514.
[17] M. Cha, Y. L. Gwon, and H. Kung, "Adversarial learning of semantic relevance in text to image synthesis," in *Proc. AAAI Conf. Artificial Intell.*, 2019, pp. 3272–3279.
[18] H. Zhang *et al.*, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5907–5915.
[19] Z. Zhang, Y. Xie, and L. Yang, "Photographic text-to-image synthesis with a hierarchically-nested adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6199–6208.
[20] X. Huang, M. Wang, and M. Gong, "Hierarchically-fused generative adversarial network for text to realistic image synthesis," in *Proc. IEEE Conf. Computer and Robot Vis.*, 2019, pp. 73–80.
[21] G. Yin, B. Liu, L. Sheng, N. Yu, X. Wang, and J. Shao, "Semantics disentangling for text-to-image generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2327–2336.
[22] M. Zhu, P. Pan, W. Chen, and Y. Yang, "DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5802–5810.
[23] B. Li, X. Qi, T. Lukasiewicz, and P. Torr, "Controllable text-to-image generation," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 2065–2075.
[24] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Learn, imagine and create: Text-to-image generation from prior knowledge," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 885–895.
[25] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "LayoutGAN: Generating graphic layouts with wireframe discriminators," *Int. Conf. Learn. Representations*, 2019.
[26] A. A. Jyothi, T. Durand, J. He, L. Sigal, and G. Mori, "LayoutVAE: Stochastic scene layout generation from a label set," *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9894–9903.
[27] B. Li, B. Zhuang, M. Li, and J. Gu, "Seq-SG2SL: Inferring semantic layout from scene graph through sequence to sequence learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 7434–7442.
[28] S. Reed, A. van den Oord, N. Kalchbrenner, V. Bapst, M. Botvinick, and N. de Freitas, "Generating interpretable images with controllable structure," 2016. [Online]. Available: https://openreview.net/forum?id=Hyvw0L9el&noteId=Hyvw0L9el
[29] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 217–225.

[30] A. Raj, C. Ham, H. Alamri, V. Cartillier, S. Lee, and J. Hays, "Compositional generation of images," in *Proc. Advances Neural Inf. Process. Syst. ViGIL*, 2017.
[31] S. Hong, D. Yang, J. Choi, and H. Lee, "Inferring semantic layout for hierarchical text-to-image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7986–7994.
[32] X. Liu, G. Yin, J. Shao, X. Wang, and H. Li, "Learning to predict layout-to-image conditional convolutions for semantic image synthesis," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 568–578.
[33] B. Zhao, L. Meng, W. Yin, and L. Sigal, "Image generation from layout," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8584–8593.
[34] W. Huang, R. Y. D. Xu, and I. Oppermann, "Realistic image generation using region-phrase attention," in *Proc. Asian Conf. Mach. Learn.*, 2019, pp. 284–299.
[35] W. Sun and T. Wu, "Image synthesis from reconfigurable layout and style," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 10530–10539.
[36] Y. Li, T. Ma, Y. Bai, N. Duan, S. Wei, and X. Wang, "PasteGAN: A semi-parametric method to generate image from scene graph," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 3948–3958.
[37] D. M. Vo and A. Sugimoto, "Visual-relation conscious image generation from structured-text," 2019, *arXiv: 1908.01741*.
[38] S. Hong, X. Yan, T. Huang, and H. Lee, "Learning hierarchical semantic image manipulation through structured representations," in *Proc. Advances Neural Inf. Process. Syst.*, 2018, pp. 2712–2722.
[39] D. Lee, M.-Y. Liu, M.-H. Yang, S. Liu, J. Gu, and J. Kautz, "Context-aware synthesis and placement of object instances," in *Proc. Advances Neural Inf. Process. Syst.*, 2018, pp. 10 413–10 423.
[40] A. El-Nouby *et al.*, "Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, 10304–10312.
[41] Y. Cheng, Z. Gan, Y. Li, J. Liu, and J. Gao, "Sequential attention GAN for interactive image editing via dialogue," 2018, *arXiv: 1812.08352*.
[42] Y. Li *et al.*, "Storygan: A sequential conditional gan for story visualization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6329–6338.
[43] G. Mittal, S. Agrawal, A. Agarwal, S. Mehta, and T. Marwah, "Interactive image generation using scene graphs," in *Proc. Int. Conf. Learn. Representations Workshop*, 2019.
[44] S. Nam, Y. Kim, and S. J. Kim, "Text-adaptive generative adversarial networks: manipulating images with natural language," in *Proc. Advances Neural Inf. Process. Syst.*, 2018, pp. 42–51.
[45] X. Zhou, S. Huang, B. Li, Y. Li, J. Li, and Z. Zhang, "Text guided person image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3663–3672.
[46] S. Yu, H. Dong, F. Liang, Y. Mo, C. Wu, and Y. Guo, "Simgan: Photo-realistic semantic image manipulation using generative adversarial networks," in *Proc. IEEE Conf. Image Process.*, 2019, pp. 734–738.
[47] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[48] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[49] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," in *Proc. Int. Conf. Learn. Representations*, 2016.
[50] A. Borji, "Pros and cons of GAN evaluation measures," *Comput. Vis. Image Understanding*, vol. 179, pp. 41–65, 2019.
[51] S. Barratt and R. Sharma, "A note on the inception score," in *Proc. Int. Conf. Mach. Learn. Workshop*, 2018.
[52] S. Ravuri and O. Vinyals, "Classification accuracy score for conditional generative models," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 1268–12279.
[53] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my GAN?" in *Proc. Europ. Conf. Comput. Vis.*, 2018, pp. 213–229.
[54] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 652–663, Apr. 2017.
[55] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *Proc. Europ. Conf. Comput. Vis.*, 2016, pp. 382–398.
[56] J. Giménez and L. Màrquez, "Linguistic features for automatic evaluation of heterogenous mt systems," in *Proc. ACL Workshop Statist. Mach. Transl.*, 2007, pp. 256–264.
[57] P. S. Madhyastha, J. Wang, and L. Specia, "VIFIDEL: Evaluating the visual fidelity of image descriptions," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 6539–6550.
[58] M. Kilickaya, A. Erdem, N. Ikizler-Cinbis, and E. Erdem, "Re-evaluating automatic metrics for image captioning," in *Proc. Conf. Europ. Chapter Assoc. Comput. Linguistics*, 2017, pp. 199–209.
[59] P. Agarwal, A. Betancourt, V. Panagiotou, and N. Díaz-Rodríguez, "Egoshots, an ego-vision life-logging dataset and semantic fidelity metric to evaluate diversity in image captioning models," in *Proc. ICLR Workshop Mach. Learn. Real Life*, 2020.
[60] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1520–1529.
[61] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.
[62] Z. Deng, J. Chen, Y. Fu, and G. Mori, "Probabilistic neural programmed networks for scene generation," in *Proc. Advances Neural Inf. Process. Syst.*, 2018, pp. 4028–4038.
[63] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. Int. Conf. Learn. Representations*, 2019.
[64] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in *Proc. Int. Conf. Learn. Representations*, 2019.
[65] B. Efron, "Bootstrap methods: another look at the jackknife," in *Breakthroughs in Statistics*. Berlin, Germany: Springer, 1992, pp. 569–593.

Tobias Hinz received the bachelor's degree in business informatics from the University of Mannheim, in 2014, and the master's degree in intelligent adaptive systems from the University of Hamburg, Germany, in 2016. He is currently working toward the PhD degree at the Knowledge Technology Group, University of Hamburg and since 2019 he is a research associate with the international research centre Crossmodal Learning (TRR-169). His current research interests include generative models, computer vision, and scene understanding. In particular, he is interested in how to learn representations of complex visual scenes in an unsupervised manner.

Stefan Heinrich received the Diplom (German MSc) degree in computer science and cognitive psychology from the University of Paderborn, and the PhD degree in computer science from the Universität Hamburg, Germany. He is a currently postdoctoral researcher at the International Research Center for Neurointelligence of the University of Tokyo and previously was appointmened as a postdoctoral research associate with the international collaborative research centre Crossmodal Learning (TRR-169). His research interests include located in between artificial intelligence, cognitive psychology, and computational neuroscience. Here, he aims to explore computational principles in the brain, such as timescales, compositionality, and uncertainty, to foster our fundamental understanding of the brain's mechanisms but also to exploit them in developing machine learning methods for intelligent systems.

Stefan Wermter is currently a full professor at the University of Hamburg, Germany, and director of the Knowledge Technology Research Group. His main research interests include the fields of neural networks, hybrid knowledge technology, neuroscience-inspired computing, cognitive robotics, and human-robot interaction. He has been an associate editor of the journal 'Transactions on Neural Networks and Learning Systems', is associate editor of 'Connection Science' and 'International Journal for Hybrid Intelligent Systems', and is on the editorial board of the journals 'Cognitive Systems Research', 'Cognitive Computation' and 'Journal of Computational Intelligence.' Currently, he serves as a co-coordinator of the International Collaborative Research Centre on Crossmodal Learning (TRR-169) and is the coordinator of the European Training Network SECURE on safety for cognitive robots. He is also the elected President for the European Neural Network Society for 2020–2022.