MIGS: Meta Image Generation from Scene Graphs

Azade Farshad*
azade.farshad@tum.de

Sabrina Musatian*
sabrina.musatian@tum.de

Helisa Dhamo
helisa.dhamo@tum.de

Nassir Navab
nassir.navab@tum.de

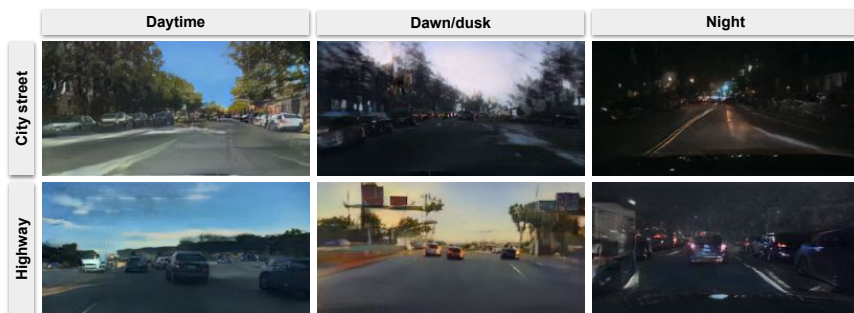Technical University of Munich
Munich, Germany

Figure 1: Sample of images generated by MIGS over a variety of scene attributes from BDD dataset [51]. The model is trained on 10-shot data for generating the images.

Abstract

Generation of images from scene graphs is a promising direction towards explicit scene generation and manipulation. However, the images generated from the scene graphs lack quality, which in part comes due to high difficulty and diversity in the data. We propose MIGS (Meta Image Generation from Scene Graphs), a meta-learning based approach for few-shot image generation from graphs that enables adapting the model to different scenes and increases the image quality by training on diverse sets of tasks. By sampling the data in a task-driven fashion, we train the generator using meta-learning on different sets of tasks that are categorized based on the scene attributes. Our results show that using this meta-learning approach for the generation of images from scene graphs achieves state-of-the-art performance in terms of image quality and capturing the semantic relationships in the scene. Project Website: https://migs2021.github.io/

1 Introduction

The task of high-quality image generation and manipulation has been capturing the attention of researchers for many years. Recent advances in deep learning for unconditional image synthesis [19, 20, 21, 22] has led to producing high-quality images that are often indistingu-

---

*The first two authors contributed equally to this work.

ishable from the real ones by a human. Some impressive results have been also achieved for the class-conditional image generation [3] and generation from semantic segmentation masks [28]. While the latter method allows for pixel-level control on the image, it requires a segmentation map which may be practically hard to obtain. An easier alternative to represent image semantics is a scene graph [16], *i.e.* a graph where nodes correspond to the objects and edges define the relationships between them. Image generation from such graphs is attractive as they allow full control over the semantics and it is easy to modify them in case of scene editing. Image generation from scene graphs has been introduced in [18]. While the results are encouraging, the images generated by this model lack quality when trained on datasets with diverse scenes, as the network struggles to learn meaningful representations to accommodate such discrepancies. To counterpart this problem we suggest using meta-learning in order to help the network focus its attention on specific tasks during training. Yet such a model is able to quickly adjust to a wide set of tasks during testing with only a few training samples available. Additionally, our approach allows us to introduce the task of few-shot learning for scene graph to image generation problem, which is an attractive scenario as it can help to generate semantically meaningful images in applications with a scarce amount of data. The contributions of our work can be summarized as follows:

- A novel meta-learning approach for the generation of images from scene graphs, which achieves state-of-the-art results compared to previous work

- Introduction of the few-shot learning problem for scene graph to image generation

- A novel task sampling method for scenes in the wild that can benefit other image generation with meta-learning scenarios.

We evaluate our proposed method on automatically generated scene graphs for Berkeley Deep Drive [51], Action Genome [15] and Visual Genome [23] datasets, showing superior results compared to the baselines, qualitatively and quantitatively. This is also verified by performing a user study on the quality of images. The source code of this work is provided in the supplement and will be publicly released upon its acceptance.

2   Related Work

Image generation   Recent advances in generative models, in particular, Generative Adversarial Networks (GANs) [10] have boosted the quality of image generation. A line of works explore generative models for unconditional image generation [19, 20, 21, 22]. Image generation models have also been explored conditionally, with a diverse set of priors such as semantic segmentation maps [4, 28, 36], natural language descriptions [24, 29, 38, 40] or translating from one image domain to another using paired [14] or unpaired data [42]. Most related to our approach are methods that generate images from scene graphs [18].

Image generation from scene graphs   Scene graphs [16] refer to representations that describes images, where nodes are objects, and the edges represent relationships between them. With the recent rise of large-scale scene graph datasets, such as Visual Genome [23] a diverse set of scene graph related tasks were explored. A line of works propose strategies for scene graph generation from images [11, 26, 37]. Johnson *et al.* [18] introduced the reverse task of image generation from scene graphs, using a 2D layout as an intermediate representation between graphs and images, where layouts are decoded to images using a Cascade

Refinement Network (CRN) [4] architecture. Later, a similar architecture was explored for image generation in an interactive form [1] as well as for semantic image manipulation [6]. Herzig *et al.* [12] proposed a model that uses canonical scene graphs, to improve robustness in terms of graph size and noise. Recently, Garg *et al.* [9] generate scene graphs unconditionally and later synthesize images from the resulting graphs. Other related works explore image generation directly from layouts [32, 33, 41] or investigate 3D scene graphs [7, 35].

Meta-learning   Meta-learning or learning to learn was initially introduced for the few-shot classification problem. Model-agnostic Meta-Learning [8] (MAML) is one of the most well-known works for few-shot image classification. MAML aims to optimize a model on a set of tasks using second-order gradient computation to obtain a model that adapts fast to newly seen tasks. Due to the extensive computation demands of MAML, a first-order approximation of it was proposed in Reptile [27] with similar performance. These approaches have been mainly adopted in image classification and segmentation problems. A combination of meta-learning with GANs was introduced in [39] which employs adversarial training for few-shot image classification. However, the employment of meta-learning for the few-shot image generation task has been rarely explored. FIGR [5] uses meta-learning for few-shot image generation for small-scale datasets of black and white images, while few-shot image to image translation [25] generates images by adapting input from a source domain to a target domain. Despite the previous use of meta-learning for image generation, it has been only used on data with limited diversity and low resolution. To our knowledge, this is the first work on the few-shot generation of high-resolution scenes in the wild.

3   Methodology

3.1   Problem Definition

A task in meta-learning for few-shot image classification is defined as a set of image, label pairs. In our work, we define the task as scene graph and image pairs. Given a set $\mathcal{D}$ of image $\mathcal{I}$ and scene graph $\mathcal{G}$ pairs, we define our initial dataset $\mathcal{D} = \{\mathcal{I}, \mathcal{G}\}$. The dataset is divided into different tasks based on the predefined task definition. In each iteration of the training phase, a task is randomly sampled and the scene graph to image model parameters are optimized on the selected task. In the test phase, the trained parameters are then used to fine-tune the model on specific target tasks. The main components of our method which are the image generation and meta-learning are described over the following sections.

3.2   Image generation

To tackle the task of scene graph to image generation we build upon SG2Im [18] architecture as a foundation. SG2Im takes as input a scene graph, where the nodes correspond to the objects and the edges define relationships between them. The scene graph is processed by graph convolutional network (GCN), which operates on triplets subject-predicate-object, to propagate information along the edges. This results in processed per-node features, where each object has its own embedding vector that encodes information about itself as well as relationships with connecting objects. These embeddings are then used to predict a set of bounding boxes and segmentation masks for each object. The predicted boxes and masks are combined to project the GCN features to image space and obtain a scene layout. The next
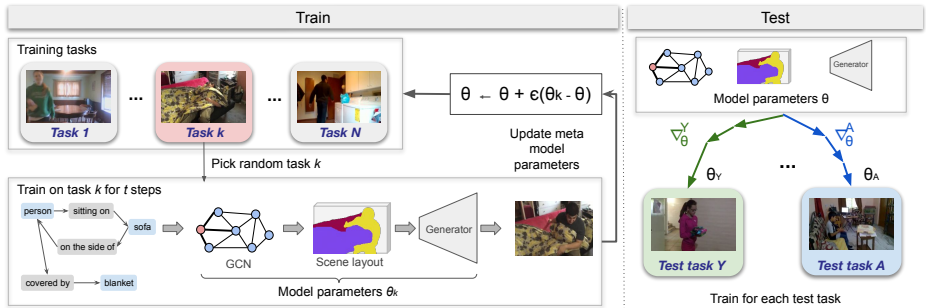
Figure 2: **Method Overview**. Our method consists of two phases: Meta-training and Testing. In the meta-training phase, the model parameters θ are updated on a sampled task l in each iteration. One task is a set of image and scene graph pairs that are unified in one group by some criteria. We pass a scene graph through a GCN to produce features from the nodes embeddings for creating a scene layout. The layout is passed to the generator, which synthesises the final image. In the testing phase, we fine-tune the model θ for a defined number of shots on each specific task which results in the generation of our final images.

step in this pipeline is an image generator that receives a scene layout and produces an image corresponding to the given semantic definition. In order to force the network to produce both realistically looking and semantically correct images, two image discriminators are used on top of the generated image. The first one discriminates individual objects in a local context, while the second one classifies a picture as a whole. We use different loss terms for training the model. Most of the losses are adopted from [16], but some extra loss terms are also used which we mention below. To prevent the model from generating trivial solutions, we employ the perceptual loss [17] $\lambda_p \mathcal{L}_p$ using the VGG network. There are two GAN losses defined, one for the whole image and one to make individual objects look realistic. These are defined as $\mathcal{L}_{\text{GAN,global}}$ and $\mathcal{L}_{\text{GAN,obj}}$ respectively. To ensure the quality of generated objects, the auxiliary classification loss $\mathcal{L}_{\text{aux,obj}}$ is used. The loss for predicting the bounding boxes is defined by $\mathcal{L}_{box}$ which is calculated using the $\mathcal{L}_1$ loss between the predicted and ground truth bounding boxes. Finally, the image loss $\mathcal{L}_{\text{im}}$ which is the $\mathcal{L}_1$ distance between the predicted image and the ground truth image is used. Equation 1 shows the task loss $\mathcal{L}_\tau$ definition.

$$
\begin{aligned}
\mathcal{L}_\tau = \lambda_b \mathcal{L}_{box} &+ \lambda_g \min_G \max_D \mathcal{L}_{\text{GAN,global}} \\
&+ \lambda_o \min_G \max_D \mathcal{L}_{\text{GAN,obj}} + \lambda_a \mathcal{L}_{\text{aux,obj}} \\
&+ \lambda_p \mathcal{L}_p + \lambda_{\text{im}} \mathcal{L}_{\text{im}},
\end{aligned}
\tag{1}
$$

where $\lambda_b, \lambda_g, \lambda_o, \lambda_a, \lambda_p, \lambda_{\text{im}}$ are weighting values and

$$
\mathcal{L}_{\text{GAN}} = \mathbb{E}_{q \sim p_{\text{real}}} \log D(q) + \mathbb{E}_{q \sim p_{\text{fake}}} \log(1 - D(q)),
\tag{2}
$$

where p_{real} refers to the real data distribution from the ground truth and p_{fake} is the distribution of generated fake images or objects. The input to the discriminator is defined by q.

In order to improve the image quality of the images generated by SG2Im and tackle a few-shot learning scenario, we adapt Reptile [27] algorithm for GANs, similarly to how it

is done in FIGR [5]. However, FIGR is focused on the problem of unconditional image generation and experiments with images that have a single object drawn on them. We, on the other hand, have a setting conditioned on a scene graph with multiple objects, which is a more challenging problem compared to FIGR. We meta-train all components of the SG2Im pipeline on a diverse set of scenes and prove that such procedure enables high-quality image generation from the scene graph.

3.3   MIGS

In this section, we introduce our method of Meta image generation from scene graphs (MIGS) and its components. Our primary goal is to be able to quickly and effectively adapt a model trained on a variety of images to a specific task with just a few training shots. To tackle this problem we refer to the meta-learning models [8, 27] and their application in a few-shot learning setting. To the best of our knowledge, we introduce a few-shot learning scenario for the scene graph to image generation for the first time.

Task Definition   Meta-learning models assume access to a set of tasks T, that consist of different learning problems τ. Each of these tasks τ represent an image generation from scene graph problem and has its own set of image-graph pairs $\mathcal{D}_\tau = \{\mathcal{I}, \mathcal{G}\}_\tau$, that have been grouped together in one task based on certain criteria. Such criteria include the type of objects on the scene, specific background surroundings or any other attributes of either graphs, or images that unite them together. However, the splitting bases should be homogeneous across the whole set of tasks T. The task splitting criteria is dependent on the dataset characteristics. It can be defined based on the scene attributes such as time of the day, the context or by simply clustering the images into a set of clusters based on their visual attributes using an unsupervised clustering method such as [54].

Meta-learning   A loss function on a task τ is denoted as $\mathcal{L}_\tau$. For simplicity we define $\mathcal{L}_\tau$ as a combination of all generator $\mathcal{L}_{\tau,G}$ and discriminator $\mathcal{L}_{\tau,D}$ losses in SG2Im model. Then our meta-learning goal is to find such initial model parameters θ that for a randomly selected task τ the loss $\mathcal{L}_\tau$ will be low after k iterations with only a few data points available. In short, such objective is defined as

$$\min_{\theta}[\mathcal{L}_\tau\left(U_\tau^k(\theta)\right)], \tag{3}$$

where $U_\tau^k(\theta)$ denotes an operator that updates weights θ k times using image-graph pairs from $\mathcal{D}_\tau$.

In order to find such parameters θ, we train the models with the Reptile algorithm [27]. It comprises of inner and outer loops. In the inner loop k iterations of operator U are performed on the locally copied weights θ_l for a randomly sampled training task l. In an outer loop the weights vector of the meta-model θ are updated leveraging the difference between θ and θ_l computed in the inner loop. This updated can be summarized as:

$$\theta \leftarrow \theta + \beta \frac{1}{L} \sum_{l=1}^{L} (\theta_l - \theta), \tag{4}$$

where L is number of tasks and β is the meta learning rate. We perform such updates separately for the image generator model and two discriminators.
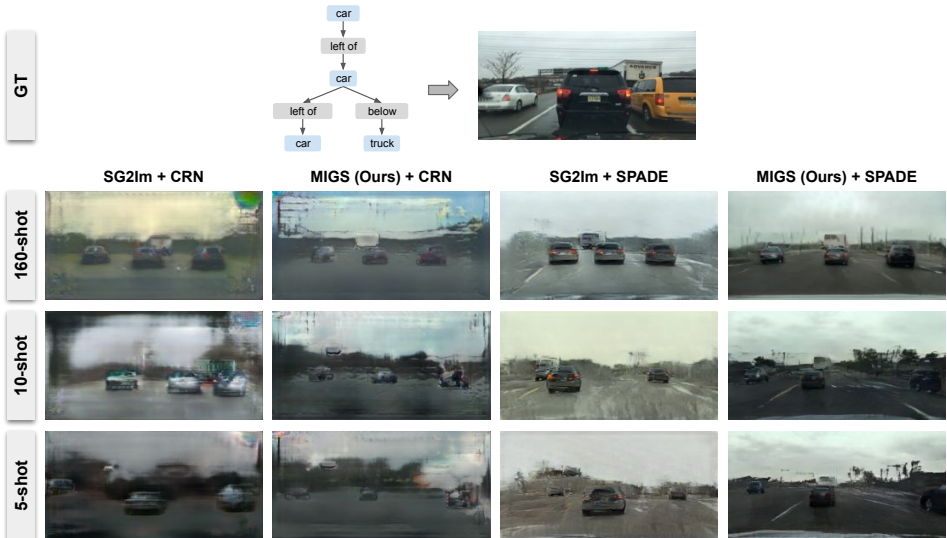
Figure 3: Examples of images generated from BDD dataset for a task with the following attributes: day time: daytime; weather: rainy; driving scenario: highway. All images were generated from the scene graph defined on top. MIGS not only produces more realistic images but also has a higher accuracy when cross-referenced with the provided scene graph, i.e. MIGS + SPADE is the only model for which the truck is clearly seen in all three scenarios.

Testing   For testing the trained meta-model, we first fine-tune the trained weights θ on the training split of each specific test task l to obtain the final weights θ_l for this task. Then, we generate images from the (unseen) test set scene graphs of each task.

     To fairly compare our meta-models to baseline methods, we use transfer learning on each of the non-meta models' weights Φ and fine-tune them on each task to obtain the corresponding Φ_l. Then for each task l we evaluate the images generated by θ_l and Φ_l.

4   Experiments

We evaluate our method on Berkeley Deep Drive [41] (BDD), Action Genome [15] (AG) and Visual Genome [23] (VG) datasets on different baselines. We show that our proposed method is independent of the generator architecture as the performance gain happens in two different generator architectures, namely CRN [4] and SPADE [28]. To measure the quality and realism of the images generated by our proposed method quantitatively, Fréchet Inception Distance [13] (FID) and Kernel Inception Distance [2] (KID) are reported. Moreover, generated image samples are compared to related work in diverse scenarios. We refer the reader to the supplement for the results with more metrics such as precision and recall [30] and the architecture details of the CRN, SPADE and GCN networks.

4.1   Datasets

As there are no established datasets for conditional image generation with meta-learning, we investigated and selected datasets that can be naturally categorized in a set of tasks.
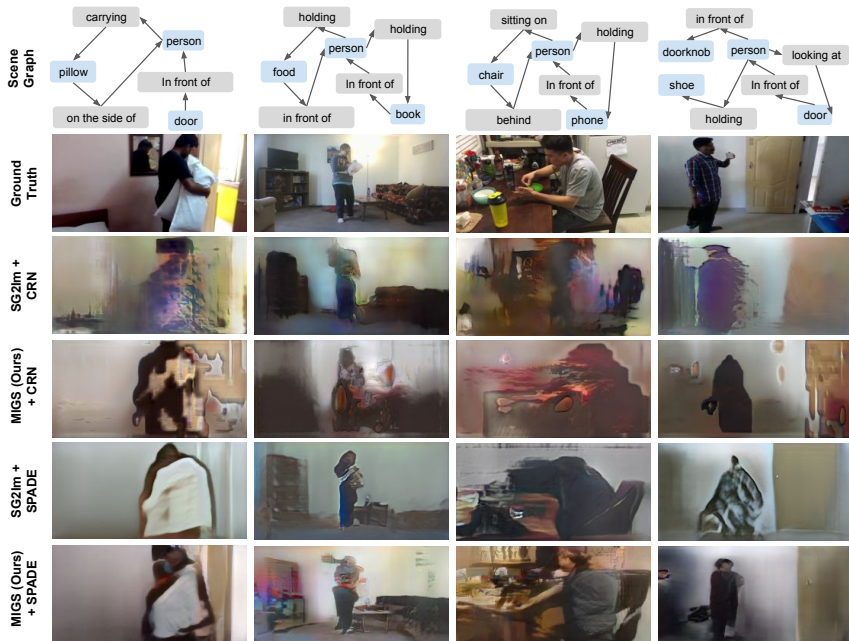
Figure 4: Examples of images generated from the Action Genome dataset. Each column correspond to one task (i.e. a video sequence) that each model was fine-tuned on. The MIGS results on each video notably contain more details as opposed to the baseline counterpart.

Berkeley Deep Drive [41] (BDD) This dataset consists of images from city streets, residential areas, and highways in different scene conditions. As BDD does not contain scene graphs associated with images, we construct spatial scene graphs automatically, leveraging the ground truth bounding boxes, leading to six mutually exclusive spatial relationships: `left of`, `right of`, `above`, `below`, `inside`, and `surrounding`. As we are mostly interested in the objects and their relationships, we pre-process images and crop out the area of interest, which contains all the objects and as little background as possible.

For our meta-learning purpose, we split BDD into tasks using provided attributes for all images, such as time of day, weather conditions, and driving scenarios. We then filter out tasks that have less than 500 images available, which results in a total of 23 different meta-learning tasks. We use 20 tasks for training and validation and 3 for testing.

Action Genome [15] (AG) The Action Genome dataset was originally designed for the action recognition problem. It consists of video frames of humans interacting with objects in a scene. We use human-object relationships labels from Action Genome directly to construct semantically meaningful scene graphs.

As the AG dataset consists of a large number of videos with different actions, we use those videos as task splitting criteria. We remove the frames of all videos that do not have persons or only persons and no other objects, as it is impossible to construct meaningful scene graphs from them. Furthermore, we extract as tasks only the horizontal videos with at least 30 frames annotated by a human. This procedure provides us with 735 tasks which we split into 662 train and validation and 73 test tasks accordingly.

Visual Genome [23] (VG)   The Visual Genome dataset provides images with their corresponding scene graphs and bounding box annotations which makes it suitable for the image generation from scene graphs problem. Since the VG dataset does not have specific annotations for scene attributes, we construct the tasks by splitting all the images into 100 classes using SCAN [54] pre-trained on the imagenet dataset in an unsupervised manner. We use the first 60 clusters for the pre-training step and the last 40 clusters for evaluation.

4.2   Experimental setup

We train our models to generate images of size 128×256 for BDD and AG, and 64×64 for VG. For all meta-learning models, we use an inner learning rate of 0.0001 and train for 10 inner iterations with Adam optimizer. The outer loop has a learning rate of 1 and uses SGD. The number of training iterations is dependant on the model and dataset, e.g. AG is more diverse than BDD and takes longer to converge. Meta-learning models on AG are trained for 40000 outer loop iterations and models on BDD are trained for 30000.

The baseline model, SG2Im [18] is trained on the same data as MIGS (all training tasks) in the pre-training step. During the testing phase, the pre-trained model is used as an initialization and is fine-tuned and tested in the few-shot setting similarly to MIGS. In all datasets, both models are trained until convergence. The SG2Im model is trained for 250k iterations on AG, 200k iterations on BDD, and 40k iterations on VG. The MIGS model is trained for 40k and 30k iterations on AG and BDD respectively, and 8k iterations on VG.

4.3   Results

The quantitative results of our experiments are shown in Table 1, Table 2. We compare our method to two baselines which are SG2Im with the original CRN decoder and another version with the SPADE network as the decoder to improve the generation quality. More qualitative results are provided in the supplementary material.

BDD Results   We show the performance of the mentioned models on BDD with different shot values ranging from 5 to 160 both quantitatively in Table 1 and qualitatively in Figure 3. Among all three experiments, we observe that FID and KID improve almost twice compared to the corresponding model with no meta-learning. We also perform a user study on BDD images, asking users to rank the quality of images from the same scene graph generated by different methods, and determine whether the scene represents the specified attribute. The results of the user study show that MIGS + SPADE is generally chosen as the most realistic method, MIGS + CRN as second and SG2Im + SPADE, SG2Im + CRN as third and fourth rank respectively. The exact percentages of rankings are shown in Table 3. The values show the percentages of users choosing the method as the specified rank based on the image quality. For the attribute representation, MIGS + SPADE and MIGS + CRN stand as 1st and 2nd rank, while SG2Im + SPADE and SG2Im + CRN are ranked 3rd and 4th.

Figure 3 shows example scene graph and images generated with baselines and our method on a single testing task from BDD. It is clear from these images, that our method outperforms all baselines and is able to generate realistic-looking images with a high level of detail even in an extremely challenging scenario where only 5 frames are available for training. Additionally, example images generated for a diverse set of training tasks may be seen on Figure 1. Our method successfully captures the differences in the scenes associated with daytime and driving scenario change.

Method	Decoder	FID ↓	KID ·10^3 ↓	FID ↓	KID ·10^3 ↓	FID ↓	KID ·10^3 ↓
		160-shot		10-shot		5-shot	
SG2Im [18]	CRN	194	210	176	186.5	196.8	224.2
MIGS(Ours)	CRN	158.5	156.4	157	158.4	183.5	187.6
SG2Im	SPADE	66.1	42.2	70.6	48.3	95.2	73.1
MIGS(Ours)	SPADE	**49.5**	**26.7**	**46.1**	**24**	**53.5**	**30.7**

Table 1: Quantitative results on BDD100k fine-tuned on 5,10 and 160 shots.

AG Results We train the Action Genome model on all training images for each testing task (approximately 30 frames) and evaluate the model on the frames extracted from the full videos that were not used for training. The testing set has approximately 65000 images.

AG dataset is extremely challenging for the scene graph to image generation, as it contains labeling only for a few chosen objects on the image and most commonly those objects are quite small, e.g. phone or book. Figure 4 shows the example images generated by our model as well as baselines on AG, while the quantitative performance on AG is shown in Table 2. Due to the difficulty of the used dataset, it is quite expected that the results look different from the ones on BDD. However, even in this scenario, the number of details increases with our method compared to a corresponding baseline. This improvement can be also verified quantitatively from the substantial decrease in both FID and KID.

Method	Decoder	FID ↓	KID ·10^3 ↓
SG2Im [18]	CRN	198	163.4
MIGS(Ours)	CRN	174.5	137.8
SG2Im	SPADE	141.3	76.3
MIGS(Ours)	SPADE	**98.1**	**47.4**

Table 2: Quantitative results on Action Genome dataset compared to related work.

Method	Decoder	Rank			
		1 (%)	2 (%)	3 (%)	4 (%)
SG2Im [18]	CRN	15.79	23.34	26.55	**35.58**
MIGS(Ours)	CRN	24.46	**25.57**	25.16	24.81
SG2Im	SPADE	25.36	24.60	**28.29**	21.74
MIGS(Ours)	SPADE	**34.72**	26.48	20.0	17.79

Table 3: User study ranking results on randomly sampled images from BDD dataset.

VG Results The performance of MIGS compared to SG2Im [18] on the VG dataset is presented in Table 4 and Figure 5. As shown in the results, MIGS outperforms the baseline in all metrics for different shot values, even with less number of training epochs. Despite the higher diversity of the images in the VG dataset and their wild nature, MIGS is able to generate images which look more realistic compared to the baseline.

4.4 Discussion

Our experiments show that using meta-learning for the task of image generation from scene graphs, outperforms the respective baselines almost twice, in terms of the employed metrics. The proposed method is shown to be advantageous in all scenarios even when using only 5 training samples. The results obtained on BDD qualitatively differ, which can be attributed

Method	Decoder	FID ↓	KID ·10^3 ↓	FID ↓	KID ·10^3 ↓	FID ↓	KID ·10^3 ↓
		160-shot		10-shot		5-shot	
SG2Im [18] (All epochs)	SPADE	55.20	35.54	81.42	59.39	91.79	68.52
MIGS(Ours, 1/3 epochs)	SPADE	54.83	34.21	76.56	52.02	84.87	59.38
MIGS(Ours, All epochs)	SPADE	**54.24**	**29.00**	**75.96**	**50.69**	**83.54**	**55.28**

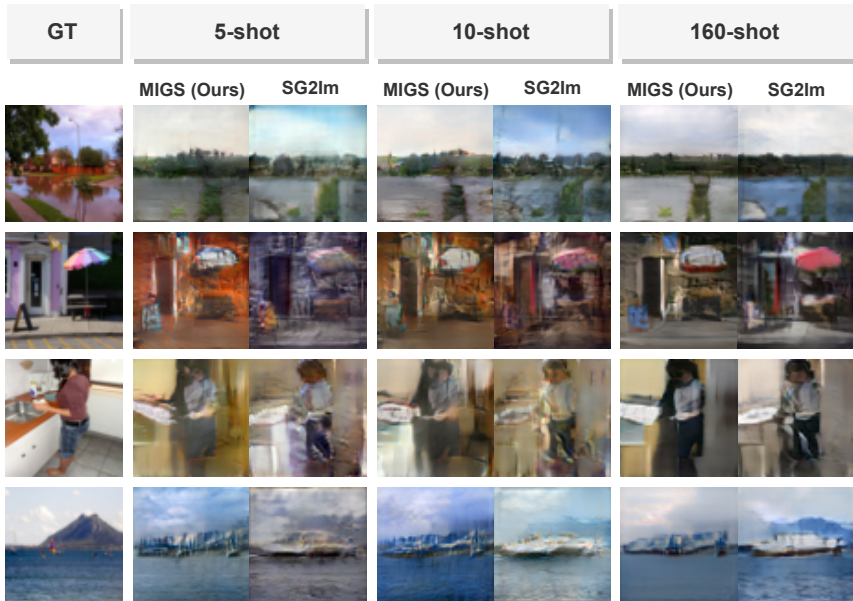Table 4: Quantitative results on VG fine-tuned on 5, 10 and 160 shots.

Figure 5: Example images generated with MIGS + SPADE and SG2Im + SPADE on the Visual Genome dataset.

not only to a more challenging setting of the AG but also to the unsuitability of this dataset for the image generation task. As it is intended for action recognition on video, only the objects that the person interacts with are annotated. It is thus common for some instances of the objects to be ignored in certain frames, until the action takes place, or be unannotated through the whole video if not used by the main actor. Such discrepancies may confuse the model and result in poor quality. Additionally, this dataset contains a lot of very small objects compared to the frame size. We believe that with a better dataset for semantically meaningful scene graphs the model should demonstrate results similarly to BDD. The results on the VG dataset show that using a simple yet effective task construction scheme, such as clustering, which could be applied to any other dataset, combined with the meta-learning approach can improve the performance of the image generation. The effect of task construction on the performance of meta-learning is an interesting topic and leaves room for research for future.

5   Conclusion

We propose MIGS, a meta-learning approach for the generation of images from scene graphs. To our knowledge, this is the first work for few-shot image generation of scenes in the wild. The proposed method could be applied to a different range of generator architectures and different datasets. We plan to replace the current SG2Im framework with [12] as part of the future work. The results of the evaluation on three datasets show that our proposed meta-learning-based image generation scheme proves to improve the quality of generated images significantly in all scenarios. We show that it is possible to generate high-quality images only given 5 shots of data. The performance gain compared to previous work is shown both quantitatively and qualitatively in the results.

References

[1] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *IEEE International Conference on Computer Vision*, 2019.

[2] Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR*. OpenReview.net, 2019.

[4] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE international conference on computer vision*, 2017.

[5] Louis Clouâtre and Marc Demers. Figr: Few-shot image generation with reptile. *arXiv preprint arXiv:1901.02199*, 2019.

[6] Helisa Dhamo, Azade Farshad, Iro Laina, Nassir Navab, Gregory D Hager, Federico Tombari, and Christian Rupprecht. Semantic image manipulation using scene graphs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[7] Helisa Dhamo, Fabian Manhardt, Nassir Navab, and Federico Tombari. Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

[9] Sarthak Garg, Helisa Dhamo, Azade Farshad, Sabrina Musatian, Nassir Navab, and Federico Tombari. Unconditional scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[11] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping Images to Scene Graphs with Permutation-Invariant Structured Prediction. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[12] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.

[15] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[16] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 2016.

[18] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR*. OpenReview.net, 2018.

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[21] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalanditis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision (IJCV)*, 2017.

[24] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. Storygan: A sequential conditional gan for story visualization. *arXiv preprint arXiv:1812.02784*, 2018.

[25] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10551–10560, 2019.

[26] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[27] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[28] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[29] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[30] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lučić, Olivier Bousquet, and Sylvain Gelly. Assessing Generative Models via Precision and Recall. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[31] Daniel Seita. Bdd100k: A large-scale diverse driving video database. *The Berkeley Artificial Intelligence Research Blog. Version*, 511:41, 2018.

[32] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *ICCV*, October 2019.

[33] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R. Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *AAAI*, 2021.

[34] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pages 268–285. Springer, 2020.

[35] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[36] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, pages 8798–8807, 2018.

[37] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene Graph Generation by Iterative Message Passing. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, pages 5907–5915, 2017.

[39] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. *NeurIPS*, 2:8, 2018.

[40] Zizhao Zhang, Yuanpu Xie, and Lin Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In *CVPR*, 2018.

[41] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *CVPR*, 2019.

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.