Enhancing Complex Image Synthesis with Conditional Generative Models and Rule Extraction

Chafic Abou Akar*†, Andre Luckow*, Ahmad Obeid*, Christian Beddawi*, Marc Kamradt*, Abdallah Makhoul†

*BMW Group, Munich, Germany
†Univ. Franche-Comté, FEMTO-ST institute, CNRS, Montbéliard, France

***Abstract*—Generative Adversarial Networks (GANs) have shown potential for generating images, but have limitations when applied to complex datasets. To address these limitations, class-conditional training is employed, as it performs better and maintains a high level of semantic diversity. In this work, we propose a new method for training generative models on complex images by extracting rules defining the relationships between objects in the image, cropping significant sub-regions based on these rules, and training the models in a conditional setting using the extracted rules as labels. The proposed approach is evaluated, and the results demonstrate its effectiveness by increasing the training dataset size, and then feeding it to conditional training. As a result, synthesized samples maintain asset fine-grained details and the visibility of small instances.**

***Index Terms*—Conditional Image Generation, Data Partitioning, Generative Models, Image Analysis, Image Synthesis**

I. Introduction

Generative models, particularly Generative Adversarial Networks (GANs) [15], have made significant progress recently. However, training complex datasets with unconditional training can result in missing details and a gap in quality and diversity between the training and generated datasets [6], [36]. State-of-the-art (SOTA) approaches attempted to address the complex dataset generation issue, particularly for single or few assets image datasets such as MNIST, MS-COCO, and ImageNet [14], [30]. Nevertheless, class-conditional training has been shown to be less prone to issues such as mode collapse and connecting, and therefore they perform better, and maintain a high level of semantic diversity since each image class group contains semantically similar image distributions [2], [28]. As a result, some authors partitioned datasets into clusters, with each cluster being linked to a class, and then they used class-conditional architecture to address the complex generation and to consider more training dataset modalities and classes [5], [22], [27], [28].

On the one hand, there is an inverse relationship between the quantity of data and the complexity of the scene, which poses a challenge for current SOTA generative models, as noted in [3]. On another hand, the level of complexity in a scene depends on the objective of the generative model being used. For example, some researchers consider ImageNet to be a complex dataset for fine-grained images [22], whereas others use COCO-Stuff, Cityscape, or ADE20K for generating multi-object scenarios, as discussed in recent studies [9], [10]. As a result, fine-grained features may be omitted or distorted in the generation of complex scenes [6], [14], [20]. To address this, some authors used scene graphs, layouts, semantics, and segmentations as conditioning information in the training for complex scene generation, in order to preserve physical and logical rules [4], [11], [13], [18], [21], [30], [36], [38].

Moreover, many complex image backgrounds (such as the sky, wall, ground, or roads) occupy a significant portion of the image compared to the main labeled objects [12], [13], [18], [30] which significantly affects the context and style of the image [37]. In parallel, domain randomization includes style variations of the surrounding environment to reduce the reality gap and optimize main-object detection training giving it a higher priority over a background context [17], [24].

All of the approaches that we studied, focused on treating the whole image as a single instance, resulting in the generative model attempting to imitate the distribution of the dataset by generating similar images with the same probability of class occurrence as in the real dataset. However, real images often have a multi-modal distribution, with different modes within the same class subset that could potentially be treated as distinct classes [22], [25], [31], [32]. This suggests that there may be value in considering these sub-modes separately when training a generative model.

In this paper, we focus on the single image complexity. We take into consideration the spatial factor of depicted objects, as we observe that often closely situated objects tend to be highly correlated, connected, or associated with specific behaviors, tasks, or relationships. A complex image can thus be thought of as including multiple sub-images with distinct similarities and belonging to different domains. To tackle this challenge, we propose a new approach that involves (1) extracting rules that capture the behavior of all objects in an image, (2) cropping significant sub-regions from complex images based on these rules, and (3) conditional training using these extracted rules. This approach is tested on in-house rendered complex industrial images, Cityscapes [12] and DOTA [35]. By dissecting single complex images, the single complex dataset domain is divided into smaller specific domains. Hence, it provides better diversity, mitigates dataset-based bias [23], and improves the quality of the generation.

The paper is structured as follows: In Section II, we review the main existing data partitioning approaches. In Section III, we propose a new taxonomy for distinguishing between different levels of image complexity and demonstrate how GANs can

imitate the distribution of the number of objects rather than just the class distribution of the dataset. In Section IV, we present our contribution, then we evaluate it and analyze the results in Sections V and VI respectively. Finally, we provide our conclusions in Section VII.

II. Related Work

Training GAN models on a single mixed multi-modal data is not straightforward and leads to failure modes as mode collapse [28]. Hence, data partitioning has been a familiar concept to improve the generation quality and avoid training failure. However, the SOTA considers different techniques to implement clustering algorithms within a generative model training process.

Direct clustering: Authors focused on switching from unconditional GAN training to conditional training by producing synthetic labels. For instance, Noroozi incorporated a clustering network into a conditional GAN (cGAN). It automatically associates pseudo-labels with real and fake images [27]. Plus, Sage *et al.* proposed clustering in the latent space of AutoEncoder or CNN features space of ResNet Classifier to generate "synthetic" data labels[1]. The approach stabilized the disentangled conditional training of DCGAN and iWGAN networks and, therefore, generated a better variety of images. Additionally, they noticed that Gaussian blurring images presented to the discriminator help the GAN network remain stable as well [28]. In contrast, Liu *et al.* trained in [22] a cGAN where class labels are automatically retrieved by the discriminator while clustering its feature space. Therefore, the generator is forced to cover the exploited classes. Although, they show that the cGAN performs better when the number of classes is higher than the number of modes of the dataset. Hence, more generation diversity is manifested per class compared to unconditional training. However, due to the high variance and the complexity of real images, it is hard to define the perfect modes' cardinality, and thus the number of clusters.

Multi-stage training: Authors considered that it is more relevant to cluster samples of generated synthetic data to produce the labels. Afterward, the real dataset is accordingly partitioned and fed to a cGAN training: Considering a pre-trained (unconditional) generator, Liu *et al.* inverted all training set images into the generator's latent space, calculated their latent distances [26] which serve as descriptive features, and therefore, as data labels [23]. Afterward, the generator and discriminator are converted to conditional variants and fine-tuned with the new self-labeled samples. In this case, the generator is forced to reconsider rare semantic attributes and produce more realistic images. However, the first training is still unconditional and may suffer from collapse, which we are trying to avoid as previously mentioned. On the contrary, Watanabe *et al.* trained a classifier using generated images and their corresponding labels to generate new artificial labels and refined them using self-attention. These labels reflect semantic similarities, which is easier for the GAN to map with latent vectors instead of separate class labels [33]. Despite the efficiency of rare feature exploration, a complete GAN training - consuming and vulnerable to failure modes - must precede the labels' extraction of another cGAN training. In addition to specifying the cluster numbers, another connecting data problem remains, especially when a single instance could belong to more than a single cluster.

Multi-model training: Other authors preferred not only to split the training data but to split the generative model as well. Armandpour *et al.* solved the complex high-dimensionality distribution learning problem - usually leading to mode collapse and connecting - by breaking/partitioning the data space into smaller regions with simpler distribution (including "connected" manifold). Then, they trained a separate generator for each "disconnected" data manifold to avoid missing modes [2]. Still, some of the smaller regions could have little data, so training their corresponding separate generators may collapse.

Nearest neighbor guidance: Casanova *et al.* introduced instance-conditioned GAN (IC-GAN) to present the neighbors of an instance image as real samples for the discriminator instead of partitioning data into clusters. Hence, the generator produces images similar to the instance's neighborhood [9]. Similarly to the previously mentioned approaches, all authors consider the whole image at once for clustering, which is highly efficient for a single asset or simple images. Conversely, complex scene images contain many combinations and regions (sub-images) where each one belongs to a distinct cluster. In addition, image details may not occur equally in all the images. Consequently, fine-grained features are omitted or presented with high artifacts. As a solution, we propose a single GAN conditional training, where first, we self-extract image regions based on the provided labels and object distances and then the conditional training labels.

III. Image Content Understanding

In this section, we argue a new image taxonomy based on image complexity and asset behaviors, and the ability of GANs to imitate these connections.

A. Image Content & Taxonomy

An image is a visual representation of one or many items (objects) that may occur more than once (instance). Therefore, we propose the following taxonomy dividing images into 4 clusters based on objects' occurrences: (1) Single-object single-instance image: as in classification datasets, (2) Single-object multi-instance image, (3) Multi-object single-instance image, and (4) Multi-object multi-instance image.

However, it is important to differentiate between a single object image and a single object dataset. In a single object dataset, all images highlight the same single object class, while in opposition, the dataset may contain multiple single object images for different objects.

Moreover, instances I_1, I_2, ..., I_i are related by a behavior $\mathcal{B}(I_1, I_2, ..., I_i)$ defined by a rule $\mathcal{R}(c_1, c_2, ..., c_n)$ of n object classes (c), where $n \leq i$. This behavior is executed to fulfill the

[1]Synthetic data labels refer to automatically generated labels instead of manually being defined.

requirement or need of a predefined task. E.g., in an industrial plant, small load carrier (KLT) boxes are perfectly stacked and arranged on a pallet placed on a wheeled dolly for ease of movement. For this reason, it is important to detect assets in their dynamic active functional behavior as much as detecting them in a standalone or idle state. Therefore, it could be beneficial to generate additional images of such scenarios for training deep learning models.

B. Image Distribution

As previously mentioned, most GAN research applications directly focused on a single object image data augmentation. Nevertheless, an industrial image visualizes one or many behaviors of multiple assets. In this section, we will check if a GAN can learn a behavioral content distribution:

1) We rendered using NVIDIA Omniverse 19,039 images for a random number of KLT box stacks.
2) We used the same KLT dataset to train a StyleGAN3 model. Afterward, we generated 10,000 images as well.
3) Using the same KLT detection model - trained on a Faster-RCNN (FRCNN) Resnet-101 architecture - we counted the instance occurrences per image in the two previous datasets, and visualized the results in Fig. 1.

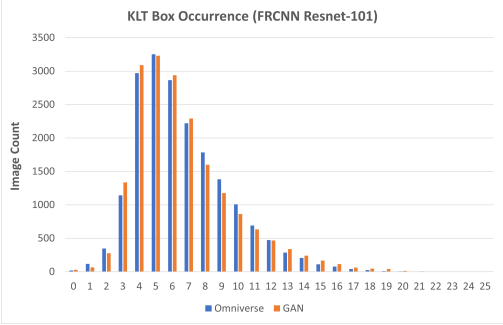

Fig. 1. KLT boxes occurrence in Omniverse and GAN-based datasets of 19039 images

In Fig. 1, we notice that instance occurrence distributions in Omniverse-rendered and GAN-synthesized images are highly similar, which shows that a GAN is capable of reproducing behavioral relationships based on asset occurrences and boundings. However, a complex image encounters multiple behaviors simultaneously, which is challenging to traditional GAN approaches. In further sections, we propose a complex data processing pipeline enhancing the output quality of generative models entitled conditional rule-based GAN (CRGAN).

IV. Conditional Rule-based GAN

In this section, we propose in detail our conditional rule-based GAN (CRGAN) as illustrated in Fig. 2.

A. Rule Extraction

The first phase of our proposed framework is to understand the provided datasets and extract all significant rules applied based on the instances' distance from each other.

Point calculation: An image visualizes i instances. Each instance I_j of an image is determined by its bounding box coordinates B_j (Left, Top, Right, Bottom), its bbox 2D midpoint p_j (x_{p_j}, y_{p_j}) and its object class c_j.
k-d tree creation: Behavioral bindings are defined by assets that are frequently found close to each other. To get the instance's neighbors, we created a "K-dimensional (K-d)" tree structure containing all calculated midpoints. Theoretically, a K-d tree is a data structure organizing points in a k-dimensional space - in our case a 2D space - and applies binary search in a quadratic time complexity $\mathcal{O}(n^2)$ with imposed constraints for nearest neighbor searches [8]. As a result, we have i lists $\{\ p_j$: sort(p_1, p_2, p_3, ..., p_i, p_j) $\}$, where the jth list contains all midpoints sorted by ascending distance to p_j.
Combination framing: A combination-frame is a set of k instances which their larger bounding box B_{j+k} covers the main instance I_j and the next k nearest instances. B_{j+k} is limited to a maximum width and height less than the image's minimum width or height. We considered all possible combinations for every instance I_j with its 0 to (maximum) $i-1$ neighbor-instances.
Combination-frame cleaning: To preserve only significant combination-frames highlighting only the bound assets without any intruders, we ignore each combination-frame containing an additional foreigner-instance not belonging to the combination's instances e.g. a background instance, or a partially appearing instance at the frame edge, etc.
Empirical rule extraction: We formulate empirical rules "o_1 c_1, o_2 c_2, ..., and o_n c_n " by exactly counting the occurrence o_m of every object class c_m, so-called condition, within the same combination-frames.
Rule fuzzification: Occurrences are fuzzified for generalization purposes, and especially when it comes to small portable assets. In such cases, the asset occurrence reflects the asset's current functional state. For instance, a single KLT box could indicate that it is used for picking up or storing parts. In the case of two boxes, we can sort, split, or assemble parts. However, for a large number, boxes are stacked for transportation. Yet, for a significantly larger number, the boxes are stored in a warehouse.
Frequent rule selection: The above-mentioned process is applied to all the datasets' images. Afterward, we merge and count all rules. Then, we select every rule with a normalized coefficient higher than a predefined threshold, i.e. support σ. As a result, the possible conditions are limited to "equal", "greater than or equal", and "inclusive between" constraints. Each rule $\mathcal{R}$ is composed of one or many conditions.

B. Region Exploration and Image Cropping

Condition exploitation: A single condition can be developed into multiple "=" meta-conditions. For instance:

$$\text{“}|c_m| \geq o_m\text{”} = \begin{cases} \text{“}|c_m| = o_m\text{”} \\ \text{“}|c_m| = (o_m + 1)\text{”} \\ \vdots \\ \text{“}|c_m| = max(|c_m|)\text{”} \end{cases} \quad (1)$$

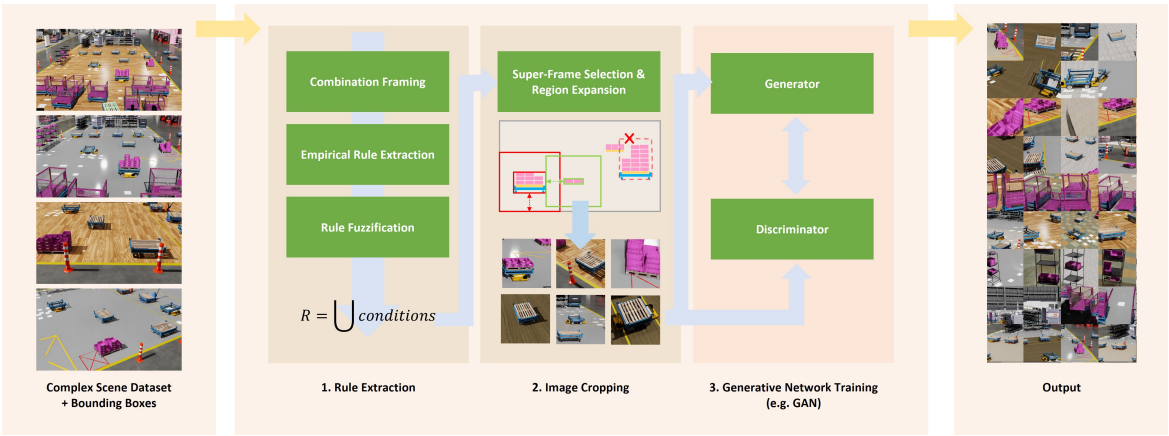

Fig. 2. Conditional rule-based GAN (CRGAN) overview

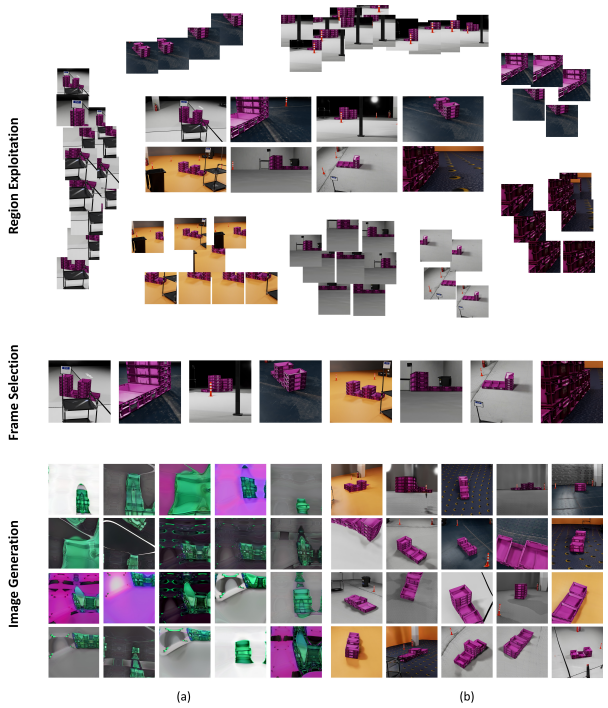

Fig. 3. Image generation models based on (a) region exploration only (b) frame selection training datasets at 5000 steps

where $max(|c_m|)$ is the maximal possible occurence of the object class c_m in the image. Or,

$$\text{“}o_m \leq |c_m| \leq o_m^{'}\text{”} = \begin{cases} \text{“}|c_m| = o_m\text{”} \\ \text{“}|c_m| = (o_m + 1)\text{”} \\ \vdots \\ \text{“}|c_m| = o_m^{'}\text{”} \end{cases} \tag{2}$$

where o_m and $o_m^{'}$ are the minimum and maximum occurrences of the object class c_m in a “inclusive between” constraint.
For each meta-condition, we extract all bounding boxes related to the condition asset c_m. Then, we calculate their midpoints and apply K-d tree as previously mentioned in Section IV-A. We apply KNN to search for the closest $o_m + k$ instances of the asset c_m.
This step is repeated for every condition in $\mathcal{R}$. As a result, multiple frames[2] F are provided per condition.
Condition combination: We apply Cartesian product between different $\mathcal{R}$'s conditions frames F to create all possible combinations.
Region exploitation: For each combination, we examine its bounding box dimension as explained previously in IV-A combination framing, and combination-frame cleaning, to exploit regions that exactly satisfy the rule.
Region maximum expansion: We expand the frame bounding box in the four directions while keeping into consideration the alignment of the asset to the center and without including any foreign assets in the frame. The expansion stops when it collides with the initial image edges or any foreign asset bounding boxes.
Square region fine-tuning: We equally crop from all dimensions to find the inscribed square of the bounding box area while preserving the alignment condition as in the above step.
Frame selection: We ensure a set of these instances to guarantee frame distinction: For the same image, we consider the superset of instances satisfying the maximum valid occurrence. Thus, only one frame containing this superset is considered to avoid semantic duplicates, image translation, or image scaling. Otherwise, the new cropped dataset will mainly include similar frames that are only different by insignificant translations and incomplete assets. For e.g., and for the same dataset, we extracted two cropped datasets: one preserving frame distinction while the other does not. As seen in Fig. 3 for the second dataset, the model is having a hard time asserting the shape and color of the assets. Thus, distinct frames have a significant impact on the generation quality.
Image cropping: Finally, we crop these regions and resize them accordingly to satisfy the GAN training dataset condi-

[2]A frame F is a sub-image of the initial image that we are exploiting.

tions and the image resolution ρ. However, since the dataset is exploited and images are cropped according to well-defined rules, these subsets belong to well-defined classes as well. Hence, they are subject to cGAN training.

V. Experimental Setup

In this section, we present our experimentation setup, including our dataset acquisition, training materials, and the evaluation metrics that we propose to assess our approach.

A. Dataset Acquisition

We conducted our supervised training using an in-house dataset consisting of 20,728 rendered images in 1080p using NVIDIA Omniverse [1], as sampled in Fig. 4. Moreover, we utilized the following public datasets: Cityscapes [12], and DOTA [35].

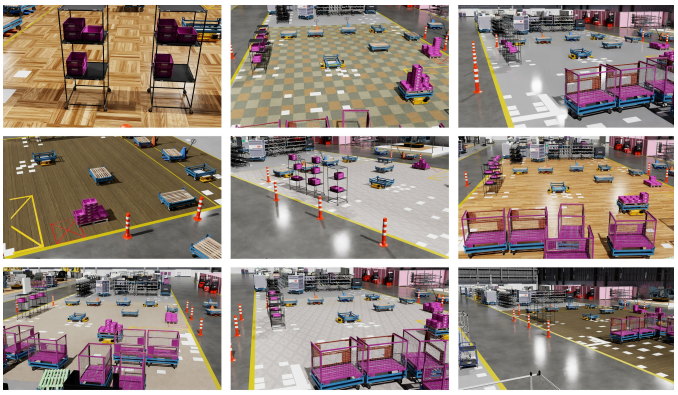

Fig. 4. Initial Omniverse-rendered complex dataset sample

We used NVIDIA Omniverse to build an industrial area filled with various single and group of assets and we rendered it using NVIDIA A6000 GPUs with 48 GB. Then, we used Isaac Sim to generate all bounding box annotations for the following industrial assets: small load carrier (KLT) box, stillage, jack, rack, smart transport robot (STR), dolly, and pallet. However, the 3D scene is inspired by a real factory setup and presents different realistic asset combinations. Additionally, we applied texture, position, and appearance domain randomizations to leverage all scenarios. Finally, we removed the annotations of occluded assets by a rack or a stillage, since they are hollow which negatively affects their appearance in extracted rules. The rendered dataset used in this study is available on request from the corresponding author.

B. Rule Extraction & Image Cropping

In Tables I, II, and III, we list the extracted rules related to our dataset, Cityscapes, and DOTA respectively.

C. Training Backbones

For our approach, we tested it with StyleGAN3 [19] as a backbone. For each dataset, we trained 2 models: the first is unconditional, trained on a single random, yet largest square crop per image. The second one is conditionally trained with

TABLE I
Our Dataset's extracted rules: $\sigma = 0.005$ and $\rho = 256 \times 256$ px

Rules	Image #
1. $1 \leq \lvert\text{dolly}\rvert \leq 8$	1,226
2. $1 \leq \lvert\text{dolly}\rvert \leq 8$ **AND** $1 \leq \lvert\text{pallet}\rvert \leq 2$	14,741
3. $\lvert\text{dolly}\rvert = 1$ **AND** $\lvert\text{STR}\rvert = 1$	8,041
4. $\lvert\text{pallet}\rvert = 1$	2,780
5. $1 \leq \lvert\text{dolly}\rvert \leq 4$ **AND** $1 \leq \lvert\text{jack}\rvert \leq 2$ **AND** $1 \leq \lvert\text{pallet}\rvert \leq 4$	16,857
6. $1 \leq \lvert\text{dolly}\rvert \leq 8$ **AND** $1 \leq \lvert\text{stillage}\rvert \leq 4$	8,276
7. $\lvert\text{stillage}\rvert = 1$	349
8. $1 \leq \lvert\text{dolly}\rvert \leq 4$ **AND** $1 \leq \lvert\text{pallet}\rvert \leq 2$ **AND** $\lvert\text{STR}\rvert = 1$	19,374
9. $1 \leq \lvert\text{rack}\rvert \leq 2$	5,453
10. $3 \leq \lvert\text{dolly}\rvert \leq 9$ **AND** $1 \leq \lvert\text{pallet}\rvert \leq 8$ **AND** $1 \leq \lvert\text{stillage}\rvert \leq 4$	3,832
11. $2 \leq \lvert\text{dolly}\rvert \leq 8$ **AND** $1 \leq \lvert\text{jack}\rvert \leq 2$ **AND** $1 \leq \lvert\text{pallet}\rvert \leq 4$ **AND** $1 \leq \lvert\text{STR}\rvert \leq 2$	37,187
Total	**118,116**

TABLE II
DOTA's extracted rules: $\sigma = 0.001$ and $\rho = 256 \times 256$ px

Rules	Image #
1. $1 \leq \lvert\text{small-vehicle}\rvert \leq 9$	1,329
2. $\lvert\text{plane}\rvert \geq 1$	992
3. $\lvert\text{harbor}\rvert \geq 1$	1,095
4. $1 \leq \lvert\text{large-vehicle}\rvert \leq 9$	1,319
5. $1 \leq \lvert\text{storage-tank}\rvert \leq 9$	655
6. $\lvert\text{large-vehicle}\rvert \geq 1$ **AND** $\lvert\text{small-vehicle}\rvert \geq 1$	809
7. $\lvert\text{bridge}\rvert \geq 1$	615
8. $\lvert\text{swimming-pool}\rvert \geq 1$	264
9. $1 \leq \lvert\text{ship}\rvert \leq 9$	729
10. $1 \leq \lvert\text{tennis-court}\rvert \leq 8$	727
11. $3 \leq \lvert\text{large-vehicle}\rvert \leq 4$ **AND** $\lvert\text{plane}\rvert \geq 9$	55
12. $\lvert\text{bridge}\rvert \geq 3$ **AND** $\lvert\text{ship}\rvert \geq 1$	112
13. $1 \leq \lvert\text{harbor}\rvert \leq 9$ **AND** $1 \leq \lvert\text{ship}\rvert \leq 9$	1,749
14. $2 \leq \lvert\text{large-vehicle}\rvert \leq 9$ **AND** $\lvert\text{storage-tank}\rvert \geq 9$	230
15. $\lvert\text{harbor}\rvert \geq 9$ **AND** $1 \leq \lvert\text{swimming-pool}\rvert \leq 8$	147
16. $\lvert\text{helicopter}\rvert = 1$ **AND** $\lvert\text{plane}\rvert \geq 9$	101
17. $\lvert\text{baseball-diamond}\rvert = 1$	299
18. $\lvert\text{bridge}\rvert \geq 9$ **AND** $\lvert\text{ground-track-field}\rvert \geq 9$ **AND** $\lvert\text{soccer-ball-field}\rvert \geq 9$	10
19. $\lvert\text{roundabout}\rvert = 1$	299
20. $\lvert\text{bridge}\rvert \geq 9$ **AND** $\lvert\text{roundabout}\rvert = 1$ **AND** $\lvert\text{ship}\rvert \geq 9$	15
21. $\lvert\text{basketball-court}\rvert = 1$	69
Total	**11,620**

TABLE III
Cityscapes' extracted rules: $\sigma = 0.0025$ and $\rho = 512 \times 512$ px

Rules	Image #
1. $1 \leq \lvert\text{pole}\rvert \leq 9$	5,363
2. $1 \leq \lvert\text{traffic sign}\rvert \leq 4$	395
3. $1 \leq \lvert\text{car}\rvert \leq 2$	4,195
4. $1 \leq \lvert\text{pole}\rvert \leq 4$ **AND** $1 \leq \lvert\text{traffic sign}\rvert \leq 4$	4,209
5. $\lvert\text{traffic light}\rvert = 1$	9
6. $\lvert\text{car group}\rvert = 1$	5,083
7. $1 \leq \lvert\text{person}\rvert \leq 2$	734
8. $\lvert\text{fence}\rvert = 1$	1,301
9. $\lvert\text{terrain}\rvert = 1$	1,351
10. $\lvert\text{car}\rvert = 1$ **AND** $1 \leq \lvert\text{pole}\rvert \leq 2$	3,579
11. $\lvert\text{static}\rvert = 1$	23
12. $\lvert\text{pole}\rvert = 1$ **AND** $\lvert\text{traffic light}\rvert = 1$	499
13. $\lvert\text{cargroup}\rvert = 1$ **AND** $\lvert\text{pole}\rvert = 1$	2,940
14. $\lvert\text{wall}\rvert = 1$	724
15. $\lvert\text{bicycle}\rvert = 1$	323
16. $\lvert\text{person}\rvert = 1$ **AND** $\lvert\text{pole}\rvert = 1$	683
17. $\lvert\text{pole}\rvert = 1$ **AND** $\lvert\text{terrain}\rvert = 1$	960
18. $\lvert\text{fence}\rvert = 1$ **AND** $\lvert\text{pole}\rvert = 1$	975
19. $\lvert\text{traffic light}\rvert = 1$ **AND** $\lvert\text{traffic sign}\rvert = 1$	18
20. $\lvert\text{person group}\rvert = 1$	244
21. $\lvert\text{pole}\rvert = 1$ **AND** $\lvert\text{traffic light}\rvert = 1$ **AND** $\lvert\text{traffic sign}\rvert = 1$	280
22. $\lvert\text{car}\rvert = 1$ **AND** $\lvert\text{pole}\rvert = 1$ **AND** $\lvert\text{traffic sign}\rvert = 1$	1,385
23. $\lvert\text{car}\rvert = 1$ **AND** $\lvert\text{car group}\rvert = 1$	1,941
24. $\lvert\text{rider}\rvert = 1$	18
25. $\lvert\text{cargroup}\rvert = 1$ **AND** $\lvert\text{pole}\rvert = 1$ **AND** $\lvert\text{traffic sign}\rvert = 1$	1,517
Total	**38,749**

our proposed approach in which we associated a class ID with each of the extracted rules. All experiments were executed on NVIDIA A100-SXM4-40GB GPUs.

VI. Results Analysis

A. Qualitative assessment

In our approach, we cropped sub-images from the initial higher-resolution images, so fine-grained details are maintained in a better way than downsampling high-dimension images into lower-resolution images as done in the SOTA. Therefore, by visually assessing the image generation quality, our proposed approach conserves details and the assets are sharper and more visible, while in the traditional StyleGAN3 training, far and background objects are subject to incomplete formation and artifacts.

Ours: Comparing our proposed approach to the traditional results, the stillage cage, the pallet's wood, and the KLT box shapes are better conserved as shown in Fig. 5, and 6. Still, and as previously mentioned, we assume that the artifact in some specific classes results from the small size of the extracted training dataset belonging to its corresponding rule.

Cityscapes: It is an extreme case dataset due to the perspective effect and the capture viewpoint: far assets - even if they are distant from each other - are closer (and smaller) when projecting in a 2D space, affecting random rules. Therefore, a data cleaning process or adding depth information to our pipeline is essential. Still, our approach extracted 25 rules, and expanded the training dataset from 22,973 to 38,749 distinct images (168.67%). As a result, it has achieved better results, in a lower number of training steps (6200 steps), like car and street marking shapes.

DOTA: Satellite images are perfect 2D images (no depth perspective) that highlight the efficiency of our proposed approach. We extracted 21 rules and the training dataset was augmented from 1,869 images to 11,620 distinct images (621.72%). We executed it at a resolution of 256x256 px to preserve the appearance of small instances that are sometimes omitted in traditional approaches.

B. Quantitative assessment

For each of the 6 models, we generated, respecting the same class distributions, approx. 20,000 images for assessment. We conducted our experiment assessment using the Frechet Inception Score (FID) as it is one of the most widely accepted scores for assessing image generation based on its quality and diversity - the lower the value, the better it is [7], [16]. We calculated the FID to compare synthesized datasets at different resolution scales (512, 256, 128, 64, and 32 px) to their corresponding original-scale training dataset using the following available repository: [29], [34]. We assume that the more we downsample an image, and due to the compression algorithm, the more fine-grained features are lost, like small and far assets, surface details, etc. Therefore, by calculating multi-scale FID, we prove that our approach is capable of maintaining better fine-grained features compared to the traditional GAN approaches that take the training image as a single instance.

Fig. 5. Our proposed approach to maintaining complex image details - Comparing Ground Truth (GT) images' subpart to the traditional generation vs our approach. (Zoom in for better comparison)

TABLE IV
Multi-Scale FID Calculation and Comparison

Dataset	$FID_{512}\downarrow$	$FID_{256}\downarrow$	$FID_{128}\downarrow$	$FID_{64}\downarrow$	$FID_{32}\downarrow$
Ours	-	13.42	62.07	162.10	218.44
Ours (CRGAN)	-	**11.22**	**30.25**	**113.57**	**177.95**
Cityscapes	15.45	21.15	61.35	199.03	317.61
Cityscapes (CRGAN)	**14.86**	**18.00**	**37.58**	**141.60**	**253.92**
DOTA	-	31.04	49.55	139.28	217.31
DOTA (CRGAN)	-	**21.78**	**35.31**	**108.16**	**178.75**

In Table IV, we notice that our approach, the so-called CRGAN, has achieved better FID scores (equivalent to lower scores) at different resolution scales. This proves our previously mentioned objective of maintaining fine-grained details from complex image datasets.

VII. Conclusion & Future Work

In this paper, we define a complex scene image as a composition of various behaviors between the image objects. Hence, it is possible to decompose an image into multiple sub-images based on objects' behaviors and relationships. In parallel, it is more important to detect assets in their dynamic active states, especially in the industrial field. Therefore, we proposed our framework that consumes a labeled complex dataset, extracts objects' behavioral rules, and, consequently, splits the images (and their sub-images) into classes satisfying the previously extracted rules. As a result, the training dataset is increased in size, labeled, and more focused on specific assets highlighting their details. After feeding them into a conditional training of StyleGAN3, the model takes into consideration more details from a larger dataset which enhanced our image generation with better shape formations and completions.

However, on one side, our framework is modular, so it can be used in association with other generative models, e.g., new GANs or diffusion models. On the other hand, it makes the

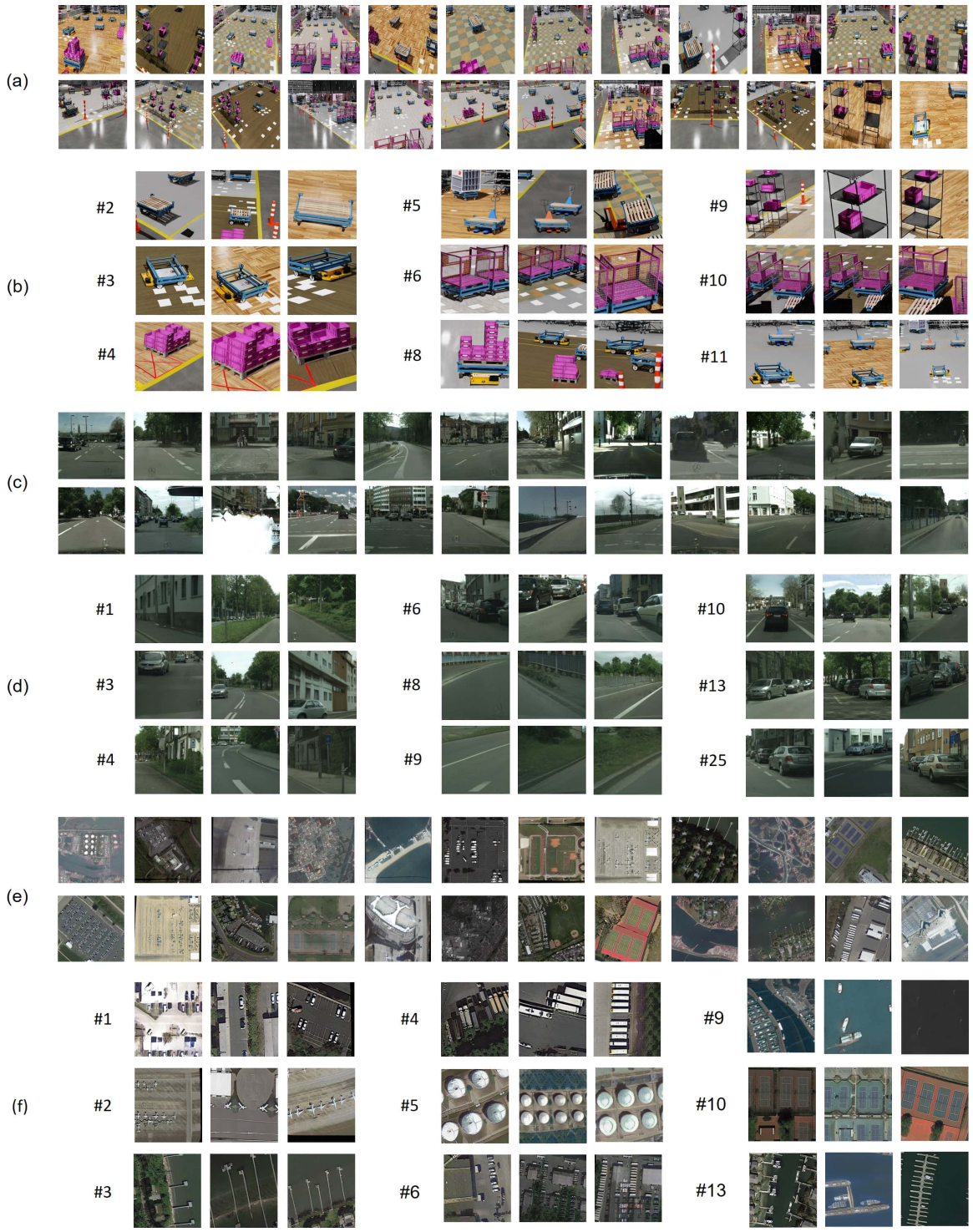

Fig. 6. Samples of generated images for our dataset (a) without and (b) with our approach at 25,000 steps, Cityscapes (c) without and (d) with our approach at 6,200 steps, and DOTA (e) without and (f) with our approach at 7,500 steps, and using StyleGAN3 as backbone

generation quality highly dependent on that generative model, and it requires a labeled dataset for rule extraction. Thus, the ability to discover and cluster image areas and sub-regions in an unsupervised manner is the subject of future work. In addition, including depth information could improve the quality of the extracted rules. Moreover, it is worth balancing the conditional class samples by automatically picking distinct and the most representative samples of that class distribution to avoid long-tail class distributions [9].

REFERENCES

[1] Chafic Abou Akar, Jimmy Tekli, Daniel Jess, Mario Khoury, Marc Kamradt, and Michael Guthe. Synthetic object recognition dataset for industries. In *2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, volume 1, pages 150–155. IEEE, 2022.

[2] Mohammadreza Armandpour, Ali Sadeghian, Chunyuan Li, and Mingyuan Zhou. Partition-guided gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5099–5109, 2021.

[3] Samaneh Azadi, Michael Tschannen, Eric Tzeng, Sylvain Gelly, Trevor Darrell, and Mario Lucic. Semantic bottleneck scene generation. *arXiv preprint arXiv:1911.11357*, 2019.

[4] Samaneh Azadi, Michael Tschannen, Eric Tzeng, Sylvain Gelly, Trevor Darrell, and Mario Lucic. Unconditional synthesis of complex scenes using a semantic bottleneck. 2020.

[5] Fan Bao, Chongxuan Li, Jiacheng Sun, and Jun Zhu. Why are conditional generative models better than unconditional ones? *arXiv preprint arXiv:2212.00362*, 2022.

[6] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4502–4511, 2019.

[7] Yaniv Benny, Tomer Galanti, Sagie Benaim, and Lior Wolf. Evaluation metrics for conditional image generation. *International Journal of Computer Vision*, 129:1712–1731, 2021.

[8] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[9] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. *Advances in Neural Information Processing Systems*, 34:27517–27529, 2021.

[10] Arantxa Casanova, Michal Drozdzal, and Adriana Romero-Soriano. Generating unseen complex scenes: are we there yet? *arXiv preprint arXiv:2012.04027*, 2020.

[11] Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alexander G Hauptmann. A comprehensive survey of scene graphs: Generation and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[13] Stanislav Frolov, Avneesh Sharma, Jörn Hees, Tushar Karayil, Federico Raue, and Andreas Dengel. Attrlostgan: attribute controlled image synthesis from reconfigurable layout and style. In *DAGM German Conference on Pattern Recognition*, pages 361–375. Springer, 2021.

[14] Raghudeep Gadde, Qianli Feng, and Aleix M Martinez. Detail me more: Improving gan's photo-realism of complex scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13950–13959, 2021.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[16] 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. *Advances in neural information processing systems*, 30, 2017.

[17] Dániel Horváth, Gábor Erdős, Zoltán Istenes, Tomáš Horváth, and Sándor Földi. Object detection using sim2real domain randomization for robotic applications. *IEEE Transactions on Robotics*, 2022.

[18] Tianyu Hua, Hongdong Zheng, Yalong Bai, Wei Zhang, Xiao-Ping Zhang, and Tao Mei. Exploiting relationship for complex-scene image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1584–1592, 2021.

[19] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

[20] Avisek Lahiri, Arnav Kumar Jain, Sanskar Agrawal, Pabitra Mitra, and Prabir Kumar Biswas. Prior guided gan based semantic inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13696–13705, 2020.

[21] Wentong Liao, Kai Hu, Michael Ying Yang, and Bodo Rosenhahn. Text to image generation with semantic-spatial aware gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18187–18196, 2022.

[22] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14286–14295, 2020.

[23] Yunzhe Liu, Rinon Gal, Amit H. Bermano, Baoquan Chen, and Daniel Cohen-Or. Self-conditioned gans for image editing. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022.

[24] Nathan Morrical, Jonathan Tremblay, Yunzhi Lin, Stephen Tyree, Stan Birchfield, Valerio Pascucci, and Ingo Wald. Nvisii: A scriptable tool for photorealistic image generation. *arXiv preprint arXiv:2105.13962*, 2021.

[25] Hariharan Narayanan and Sanjoy Mitter. Sample complexity of testing the manifold hypothesis. *Advances in neural information processing systems*, 23, 2010.

[26] Yotam Nitzan, Rinon Gal, Ofir Brenner, and Daniel Cohen-Or. Large: Latent-based regression through gan semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19239–19249, 2022.

[27] Mehdi Noroozi. Self-labeled conditional gans. *arXiv preprint arXiv:2012.02162*, 2020.

[28] Alexander Sage, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Logo synthesis and manipulation with clustered generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5879–5888, 2018.

[29] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.3.0.

[30] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2647–2655, 2021.

[31] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[32] Huan Wan, Hui Wang, Bryan Scotney, Jun Liu, and Wing WY Ng. Within-class multimodal classification. *Multimedia Tools and Applications*, 79:29327–29352, 2020.

[33] Tomoki Watanabe and Paolo Favaro. A unified generative adversarial network training via self-labeling and self-attention. *arXiv preprint arXiv:2106.09914*, 2021.

[34] Dai-Jie Wu. GitHub - jaywu109/faster-pytorch-fid — github.com. https://github.com/jaywu109/faster-pytorch-fid, 2023.

[35] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3974–3983, 2018.

[36] Zuopeng Yang, Daqing Liu, Chaoyue Wang, Jie Yang, and Dacheng Tao. Modeling image composition for complex scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7764–7773, 2022.

[37] Suiyun Zhang, Zhizhong Han, Yu-Kun Lai, Matthias Zwicker, and Hui Zhang. Stylistic scene enhancement gan: mixed stylistic enhancement generation for 3d indoor scenes. *The Visual Computer*, 35(6):1157–1169, 2019.

[38] Guangming Zhu, Liang Zhang, Youliang Jiang, Yixuan Dang, Haoran Hou, Peiyi Shen, Mingtao Feng, Xia Zhao, Qiguang Miao, Syed Afaq Ali Shah, et al. Scene graph generation: A comprehensive survey. *arXiv preprint arXiv:2201.00443*, 2022.