GENERATING UNSEEN COMPLEX SCENES: ARE WE THERE YET?

Anonymous authors
Paper under double-blind review

ABSTRACT

Although recent complex scene conditional generation models generate increasingly appealing scenes, it is very hard to assess which models perform better and why. This is often due to models being trained to fit different data splits, and defining their own experimental setups. In this paper, we propose a methodology to compare complex scene conditional generation models, and provide an in-depth analysis that assesses the ability of each model to (1) fit the training distribution and hence perform well on *seen* conditionings, (2) to generalize to *unseen* conditionings composed of *seen* object combinations, and (3) generalize to *unseen* conditionings composed of *unseen* object combinations. As a result, we observe that recent methods are able to generate recognizable scenes given seen conditionings, and exploit compositionality to generalize to unseen conditionings with seen object combinations. However, all methods suffer from noticeable image quality degradation when asked to generate images from conditionings composed of unseen object combinations. Moreover, through our analysis, we identify the advantages of different pipeline components, and find that (1) instance-wise spatial conditioning normalizations increase robustness to both types of unseen conditionings, (2) using a scene-graph perceptual similarity helps improve some dimensions of the generation process, and (3) enhancing the quality of generated masks is a crucial step in improving robustness to both types of unseen conditionings.

1 INTRODUCTION

The recent years have witnessed significant advances in generative models (Goodfellow et al., 2014; Kingma & Welling, 2014; van den Oord et al., 2016a; Miyato & Koyama, 2018; Miyato et al., 2018; Brock et al., 2019), enabling their increasingly widespread use in many application domains (van den Oord et al., 2016b; Vondrick et al., 2016; Zhang et al., 2018a; Hong et al., 2018; Sun & Wu, 2020). Among the most promising approaches, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have achieved remarkable results, generating high quality, high resolution samples in the context of *single class conditional* image generation (Brock et al., 2019). This outstanding progress has paved the road towards tackling more challenging tasks such as the one of *complex scene conditional* generation, where the goal is to generate high quality images with multiple objects and their interactions from a given conditioning (e.g. bounding box layout, segmentation mask, or scene graph). Given the exploding number of possible object combinations and their layouts, the requested conditionings oftentimes require zero-shot generalization. Therefore, successfully generating high quality, high resolution, diverse samples from complex scene datasets such as COCO-Stuff (Caesar et al., 2018) remains a stretch goal.

Despite recent efforts producing increasingly appealing complex scene samples (Hong et al., 2018; Hinz et al., 2019; Park et al., 2019; Ashual & Wolf, 2019; Sun & Wu, 2020; Sylvain et al., 2020), and as previously noted in the *unconditional* GAN literature (Lucic et al., 2018; Kurach et al., 2018), it is unfortunately very hard to assess which models perform better, and perhaps more importantly why. In the case of conditional complex scene generation, this is often due to models being trained to fit different data splits, using different conditioning modalities and levels of supervision – bounding box layouts, segmentation masks, scene graphs –, and reporting inconsistent quantitative metrics (e.g. repeatedly computing previous methods' results using different reference distributions, and/or using different image compression algorithms to store generated images), among other uncontrolled sources of variation. Moreover, these methods disregard the challenges that emerge from their ex-

pected generalization to unseen conditionings. This lack of rigour leads to conclusion replication failure, and hinders the identification of the most promising directions to advance the field.

Therefore, in this paper we aim to provide an in-depth analysis and propose a methodology to compare current conditional complex scene generation methods. We argue that such analysis has the potential to deepen the understanding of such approaches and contribute to their progress. The proposed methodology addresses the following questions: (1) How well does each method perform on *seen* conditionings (training conditionings)?; (2) How well does each method generalize to *unseen* conditionings composed of *seen* object combinations?; and (3) How well does each method generalize to *unseen* conditionings composed of *unseen* object combinations? We investigate the answers to the previous questions both from a scene-wise and an object-wise standpoints. As a result, we observe that: (1) recent methods are capable of generating identifiable scenes from seen conditionings; (2) they exhibit some generalization capabilities when using unseen conditionings composed of seen object combinations, exploiting compositionality to generate scenes with different object arrangements; (3) they suffer from poorer generated image quality when asked for unseen object combinations, especially the method presented in (Sylvain et al., 2020). Finally, through an extensive ablation, we are able to identify the strengths and weaknesses of different pipeline components. In particular, we note that endowing the generator with an instance-wise normalization module (Sun & Wu, 2019) results in increased individual object quality and better robustness to both types of unseen conditionings, whereas exploiting the normalization module of Sylvain et al. (2020) results in improved scene quality. Moreover, we find that including a scene-graph similarity (Sylvain et al., 2020) while training complex scene conditional generation models leads to better conditional consistency, especially for unseen object combinations. Finally, we identify the improvement of generated segmentation masks as one promising avenue to promote generalization to unseen conditionings.

2 RELATED WORK

Evaluation metrics for GANs. The most widely used metrics are the Inception Score (IS) (Salimans et al. (2016)) and the Fréchet Inception Distance (FID) (Heusel et al. (2017)), which aim to capture visual sample quality and sample diversity. On the one hand, IS was designed to evaluate single-object image generation on problems where the expected marginal distribution over classes is uniform, an unrealistic expectation in many real-world datasets with multi-object images. On the other hand, FID was introduced for unconditional image generation and yields a single distribution-specific score, which hinders the analysis of individual failure cases. To overcome the former, researchers have attempted to extend FID to the conditional generation case (DeVries et al., 2019; Benny et al., 2020). To overcome the latter, researchers have proposed to disentangle visual sample quality and diversity into two different metrics (Shmelkov et al., 2018; Sajjadi et al., 2018; Kynkäänniemi et al., 2019; Ravuri & Vinyals, 2019) by modifying the definition of precision-recall in different ways. Finally, the diversity of generated samples has also been quantified by measuring the perceptual similarity between generated image patches, as proposed by Zhang et al. (2018b).

Complex scene conditional generation. The conditional generation literature encompasses a variety of input conditioning modalities. Several existing works focus on generating photo-realistic scenes from detailed semantic segmentation masks (Chen & Koltun, 2017; Qi et al., 2018; Park et al., 2019; Wang et al., 2018; Tang et al., 2020). Although semantic segmentation masks are information rich, they may be difficult to obtain with enough fine-grained details. This is especially relevant in applications where a user is expected to specify the input conditioning. As such, other commonly used input conditioning modalities include text descriptions (Reed et al., 2016; Hong et al., 2018), scene graphs (Johnson et al., 2018; Ashual & Wolf, 2019), and bounding box layouts specifying the position and scale of the objects in the scene (Zhao et al., 2019; Sun & Wu, 2019; 2020; Sylvain et al., 2020). In addition to those, recent works have focused on generating bounding box layouts from a set of classes (Jyothi et al., 2019) and designing remarkably flexible pipelines to enable user interactions (Ashual & Wolf, 2019). From a model perspective, complex scene conditional generation has been improved through the development of tailored class-aware (Park et al., 2019), instance-aware (Sun & Wu, 2019; 2020), and class-and-instance-aware normalizations (Sylvain et al., 2020) incorporated in the GAN generator. The task has also benefited from the introduction of refinement pipelines (Sun & Wu, 2020), as well as the design of a multi-modal perceptual loss to drive scene graph embeddings and generated image embeddings close to each other (Sylvain et al., 2020). All aforementioned models are often trained on a different number of

training samples, use different backbone architectures, and report metrics on different splits and/or using different pre-trained models. This results in repeated reevaluations of baselines to fit each methods' setup, making it hard to pinpoint the most promising research directions.

3 METHODOLOGY

We start by defining three sets of data, composed of triplets of images and their respective *finegrained* and *coarse* conditionings. The finegrained conditionings represent detailed scene layouts by means of scene graphs, bounding boxes or segmentation masks. The coarse conditionings are an abstraction of the finegrained ones into a set of object labels present in the scene. For example, given a scene layout depicting two glasses on a table, the corresponding set of labels is $\{$*glass*, *table*$\}$.

Let $\mathcal{D}_s$ be a set containing triplets of real images and their corresponding conditionings. We use $\mathcal{D}_s$ to train the complex scene conditional generation models, and hereinafter refer to the conditionings used to train the models as *seen* conditionings. Let $\mathcal{D}_u$ be a set containing triplets of real images and their corresponding conditionings, composed of *seen* coarse conditionings but *unseen* finegrained conditionings. Note that we use the term *unseen* to refer to conditionings that are not present in $\mathcal{D}_s$. Similarly, let $\mathcal{D}_{u^2}$ be a set containing triplets of real images and their corresponding conditionings, composed of *unseen* coarse conditionings and *unseen* finegrained conditionings. We use $\mathcal{D}_u$ and $\mathcal{D}_{u^2}$ only for evaluation purposes. After training the models, we use the finegrained conditionings in $\mathcal{D}_s$, $\mathcal{D}_u$ and $\mathcal{D}_{u^2}$ to generate three sets of model samples: $\mathcal{S}_s$, $\mathcal{S}_u$ and $\mathcal{S}_{u^2}$.

The goal of our analysis is three-fold: (1) to assess how well each model fits the training data, (2) to assess each models' generalization to unseen finegrained conditionings, and (3) to assess each models' generalization to unseen coarse conditionings. To do so, we measure discrepancies between the distributions defined by the images in $\mathcal{D}_s$, $\mathcal{D}_u$ and $\mathcal{D}_{u^2}$ and by the images in $\mathcal{S}_s$, $\mathcal{S}_u$ and $\mathcal{S}_{u^2}$, respectively. In particular, we evaluate the quality of the generation processes from both *scene-wise* and *object-wise* standpoints. Note that the scene-wise evaluation considers full real images and full generated images, whereas the object-wise evaluation considers crops of individual objects from real and generated images. Throughout our analysis, we consider the following metrics to assess scene-wise and object-wise generation processes: precision, recall, consistency, FID, LPIPS-based diversity score (DS), and object classification accuracy or image(scene)-to-set prediction F1-score. In the remainder of this section, we will use *image* to refer to either full images or cropped objects.

We follow the definition of precision and recall introduced by Kynkäänniemi et al. (2019). Intuitively, precision captures the sample quality, whereas recall captures the coverage of the generated distribution. In particular, precision verifies whether generated images lie within a reference manifold of real images, and recall verifies whether real images lie within a reference manifold of generated images. The real and generated manifolds are estimated in the feature space, which in our case is defined by a ResNext101 (Xie et al., 2017), and are obtained surrounding each data point with a hyper-sphere that reaches its k-th ($k = 5$) nearest neighbor. Therefore, precision and recall are sensitive to the number of samples used to estimate the reference manifolds, resulting in overestimated manifold volumes when the sample size is small. Hence, when $|\mathcal{D}_u|$ and $|\mathcal{D}_{u^2}|$ are small, we estimate the manifold's hyper-sphere radii using the images in $\mathcal{D}_s \cup \mathcal{D}_u \cup \mathcal{D}_u^2$. Analogously, we estimate the manifold of generated images from $\mathcal{S}_s \cup \mathcal{S}_u \cup \mathcal{S}_u^2$. Although successful conditional generation models should respect their conditionings and generate high quality samples which are consistent with those, the adopted definition of precision does not guarantee that the reference real sample and the generated sample found in its hyper-sphere will have the same coarse conditionings. Therefore, we complement precision with a metric to measure the consistency of the generated samples that fall within the estimated real manifold. More precisely, we compute the intersection over union (IoU) between the classes present in the reference sample and those present in the conditioning of the generated sample. The consistency is averaged over all generated images, assigning consistency 0 to samples lying outside the estimated manifold.

4 ANALYSIS

We perform our analysis using five recently introduced complex scene generation pipelines: (1) **G2im** (Ashual & Wolf, 2019); (2) **G2im-GT** (Ashual & Wolf, 2019) which is a variant of G2im, where the instance segmentation masks and instance bounding box layouts are not generated, but

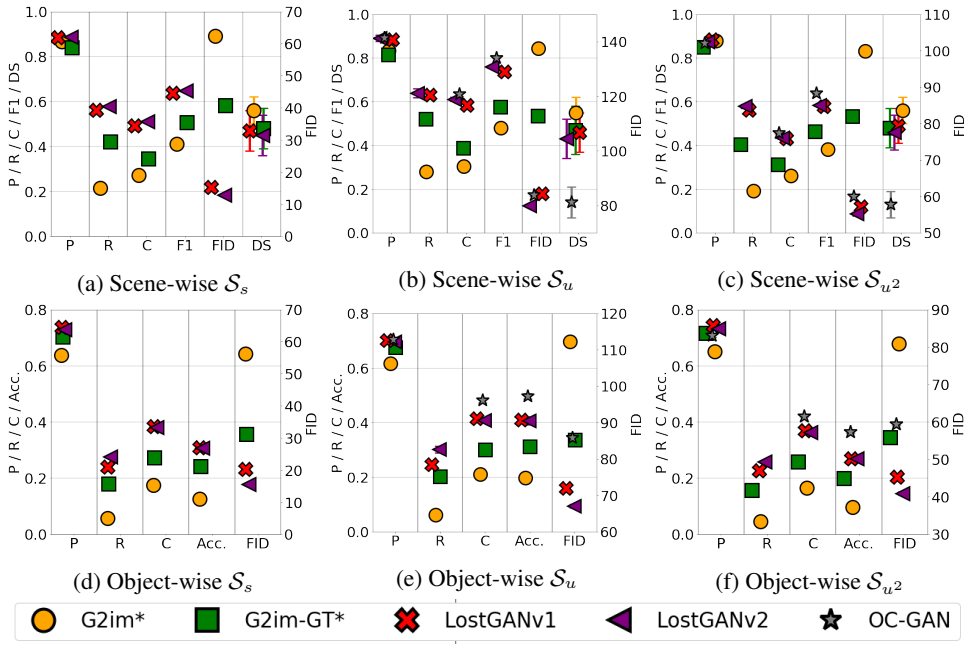


Figure 1: Comparison of state-of-the-art methods in terms of precision (P), recall (R), consistency (C), F1-score (F1), object accuracy (Acc.), FID and DS for (a-d) seen conditionings ($\mathcal{S}_s$), (b-e) unseen finegrained conditionings ($\mathcal{S}_u$), (c-f) unseen coarse conditionings ($\mathcal{S}_{u^2}$). Plots show mean and std over 5 generation processes. *Trained using the open-sourced code. Best viewed in color.

directly taken from the ground-truth data; (3) **LostGANv1** (Sun & Wu, 2019); (4) **LostGANv2** (Sun & Wu, 2020) which is an improved version of LostGANv1; and (5) **OC-GAN** (Sylvain et al., 2020). Detailed description of these approaches can be found in the Appendix C.

4.1 Dataset

We perform our analysis on the challenging COCO-Stuff (Caesar et al. (2018)), given its ubiquitous use in conditional image generation. The training and evaluation splits consist of 75k and 3k images respectively, following Sun & Wu (2019). We use the whole training set as $\mathcal{D}_s$, and divide the validation set into three sets, namely $\mathcal{D}_u$, $\mathcal{D}_{u^2}$ and a set used for hyper-parameter search and validation $\mathcal{D}_v$[1]. Note that $\mathcal{D}_v$ is defined and used as in (Ashual & Wolf, 2019). We use 128x128 resolution images, as all studied methods support this resolution. Finally, it is worth noting that the proportions of long-tail classes[2] in the previously defined $\mathcal{D}_s$, $\mathcal{D}_u$ and $\mathcal{D}_{u^2}$ are 47%, 37% and 50%, respectively, resulting in more well represented classes in $\mathcal{D}_u$ than $\mathcal{D}_{u^2}$.

4.2 Fitting seen conditionings

We start by analyzing the quality of the generation processes of the methods from a scene-wise perspective[3]. All methods have been trained for 125 epochs to give them equal opportunity to fit the training data. Figure 1a depicts the precision (P), recall (R), consistency (C), F1-score (F1), FID and DS achieved by the methods under study. We observe that all methods have rather high precision scores ($\sim$ 90%). Despite its remarkable precision, G2Im and G2Im-GT suffer from poor recall and consistency. At the same time, LostGANv1 and LostGANv2 exhibit at least $\sim$ 10% higher recall and consistency than G2Im and/or G2Im-GT, with LostGANv2 being slightly better than LostGANv1,

[1] $\mathcal{D}_v$ is included when computing the reference manifold, and its corresponding generated images $\mathcal{S}_v$ are included when computing the manifold of generated images.

[2] We consider long-tail classes the non top-25 most frequent classes in the COCO-Stuff dataset.

[3] OC-GAN's code is not publicly available, nor are its pretrained models. Their authors kindly shared their evaluation images $\mathcal{S}_u$ and $\mathcal{S}_{u^2}$ through personal communitation. Those were obtained after a 170 epochs training process. Unfortunately, we could not get any generated images from the training conditionings, and so we do not consider this method to evaluate $\mathcal{S}_s$ nor do we compute its recall in the next sections.

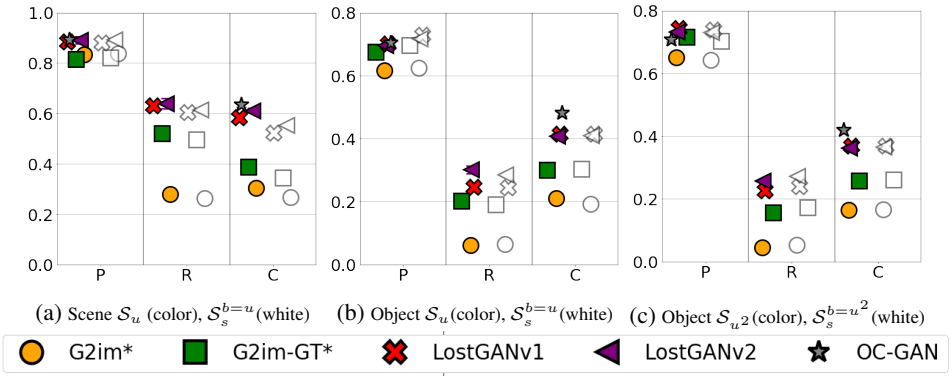


Figure 2: Comparison of state-of-the-art methods in terms of precision (P), recall (R), and consistency (C), and matching class distributions for (a-b) seen and unseen finegrained conditionings, (c) seen and unseen coarse conditionings. *Trained using the open-sourced code. Best viewed in color.

hence suggesting better generated distribution coverage and conditional consistency, which could be attributed to its mask refinement process and/or its larger number of parameters. Despite the superiority of LostGANs, their achieved consistency is only around 50%, indicating that image conditionings are often violated. Additionally, their recall is slightly below 60%, which implies that about 40% of the real images fall outside of the manifold of generated images. Moreover, when compared to Grid2Im-GT, LostGANs' superiority cannot be attributed to having access to improved layouts (Grid2Im-GT uses ground truth layouts), but rather to their modeling choices. It is worth noting that although G2Im has higher precision than G2Im-GT, G2Im-GT obtains significantly better recall. This could suggest that Grid2Im is only able to model some modes of the layout distribution. If we consider F1-score and FID, we see that the trend is preserved, with LostGANs being the top performers. When it comes to DS, all methods behave similarly. Although recall can also be thought of as a diversity metric, it is concerned with diversity in the whole image space, whereas DS tries to capture the intra-conditioning diversity. The absence of a clear leading method in terms of DS raises the question whether there could be any major improvement given the nature of the dataset, which contains a single instantiation of each finegrained conditioning.

Figure 1d shows a similar trend for object-wise metrics, with LostGANv2 scoring higher recall but similar precision and consistency than LostGANv1, suggesting that refined masks improve more scene than object quality. It is worth noting that object-wise metrics exhibit lower scores than scene-wise metrics, suggesting that the generation processes might lead to appealing generated scenes, which however suffer from high frequency details.

Finally, Figure 3 depicts two examples of images resulting from conditional generation processes from seen conditionings, showing in most cases recognizable objects in the scene, which could however be significantly improved, and emphasizing the quantitative superiority of LostGANv2.

4.3 Generalization to unseen finegrained conditionings

Here we compare the quality of the generation processes of all methods, when generating images from unseen finegrained conditionings ($\mathcal{S}_u$). We start by assessing the methods' performances from both scene-wise and object-wise standpoints, see Figures 1b and 1e. In this case, we consistently see that OC-GAN and LostGANv1/v2 show the most competitive results across most of the metrics. In particular, OC-GAN performs on par with the best version of LostGAN for both scene and object-wise precision, i.e. it matches the top scene precision performer and the top object precision performer at the same time, suggesting that this method does not trade scene precision for object precision. Moreover, OC-GAN consistently exhibits improved performance in terms of scene/object consistency, as well as scene F1-score and object accuracy, which highlights that its high scene/object precision also respects the (coarse) conditionings. When it comes to FID, LostGANv2 leads both the scene-wise and object-wise rankings, reiterating their ability to generate high quality images from unseen finegrained conditionings. However, OC-GAN's object-wise FID ranks the model as $3rd$, together with G2Im-GT, whereas G2Im-GT achieves worse FID than OC-GAN scene-wise. This suggests that OC-GAN may suffer from poorer object diversity, resulting in lower

(a) Generated images for seen conditionings

(b) Generated images for unseen finegrained conditionings

(c) Generated images for unseen coarse conditionings

Figure 3: Qualitative results. Generated images with resolution 128×128 for several methods.

FID, and that having access to high quality (GT) layouts may lead to better object quality. When it comes to DS, OC-GAN has by far the lowest result, suggesting the method trades image quality for intra-conditioning diversity. G2Im and G2Im-GT are the weakest performers, with G2Im-GT achieving more competitive results.

When analyzing the generalization gap, we see that scene-wise precision achieves the same score for both the seen conditionings ($\mathcal{S}_s$) and the unseen finegrained ones ($\mathcal{S}_u$), whereas object-wise precision is slightly better for $\mathcal{S}_s$ than for $\mathcal{S}_u$. However, recall and consistency show consistently better scores for $\mathcal{S}_u$ than for $\mathcal{S}_s$ (see Figures 1a, 1b, 1d, 1e). We hypothesize that the lower presence of long-tail classes in $\mathcal{S}_u$ results in better metrics for this split. To confirm the hypothesis, we subsample $\mathcal{D}_s$ to match the same class distribution as in $\mathcal{D}_u$, creating the split $\mathcal{D}_s^{b=u}$, and generate the corresponding set of model samples $\mathcal{S}_s^{b=u}$. Results are presented in Figure 2, and show that for $\mathcal{S}_s^{b=u}$ the recall performance gap with $\mathcal{S}_u$ is roughly closed, whereas the consistency performance gap becomes smaller. Overall, the results observed for $\mathcal{S}_u$ indicate good generalization to unseen finegrained conditionings for all methods. This is further supported by the visualizations in Figure 3, where the quality of generated images for unseen finegrained conditionings equates the one of generated images for seen conditionings, suggesting that state-of-the-art models properly exploit the compositionality of the scenes, and as such are able to generate images with recognizable objects from unseen finegrained conditionings.

4.4 Generalization to unseen coarse conditionings

Here we compare the quality of the generation processes for all methods, when generating images from unseen coarse conditionings ($\mathcal{S}_{u^2}$). We start by assessing the methods' performances from both scene-wise (see Figure 1c) and object-wise (see Figure 1f) standpoints. Similarly to previous experiments, we observe that all methods achieve high precision values, especially scene-wise, and

	↑SP	↑SR	↑SC	↑OP	↑OR	↑OC	↑F1	↑ Acc.	↑DS	↓SFID	↓OFID
unseen finegrained conditionings $\mathcal{S}_u$											
ISLA	89.7	**65.2**	63.6	73.1	28.4	45.9	76.7	43.1	**0.41**	78.5	66.1
SPADE(oc)	90.9	57.0	65.7	70.2	26.6	41.5	75.9	39.3	0.17	78.7	67.6
ISLA + M	91.6	57.7	68.1	**73.8**	31.6	49.3	77.0	45.4	0.12	75.2	61.7
ISLA + SG	90.5	64.9	66.1	70.5	**32.2**	45.3	78.9	42.1	**0.41**	78.4	62.6
ISLA + M + SG	**92.1**	58.8	**72.5**	72.6	28.7	**50.6**	**80.8**	**49.3**	0.18	**72.6**	**59.2**
unseen coarse conditionings $\mathcal{S}_{u^2}$											
ISLA	88.4	58.5	45.9	74.7	24.9	40.5	59.6	30.7	**0.44**	53.5	38.2
SPADE(oc)	90.1	53.2	47.6	73.6	24.5	35.7	59.2	26.4	0.20	55.6	40.2
ISLA + M	91.7	55.3	53.1	**77.7**	28.1	43.4	61.1	29.0	0.11	52.3	36.0
ISLA + SG	88.4	**59.1**	47.2	74.2	**28.3**	42.7	62.2	31.4	**0.44**	52.1	34.2
ISLA + M + SG	**91.8**	54.8	**55.9**	74.3	25.0	**43.7**	**64.3**	**34.5**	0.19	**50.2**	**33.2**

Table 1: Ablation results. We report average precision, recall and consistency for scenes (SP, SR, SC) and objects (OP, OR, OC), F1-score, Accuracy (Acc.), DS, scene and object FIDs (SFID and OFID). Results show the mean over 5 random seeds at test time. Extended table in Appendix E.

relatively low recall and consistency values. G2Im exhibits competitive results in terms of scene average precision but lags behind in the vast majority of the metrics. Interestingly, the high scene precision of G2Im does not translate to its object precision, suggesting that the method does not capture enough high frequency detail. The overall quality of the scene generation process of OC-GAN is mostly on par with the one of LostGANv1/v2, achieving similar precision and consistency and slightly lower scene FID score. However, OC-GAN results in lower FID scores object-wise, suggesting that OC-GAN suffers from object diversity, as in $\mathcal{S}_u$. Once again, OC-GAN shines when it comes to preserving conditional consistency (see consistency, F1, object accuracy metrics) and but appears to be the weakest in terms of DS.

Interestingly, when analyzing the generalization gap, we see that scene-wise and object-wise precision and recall achieve roughly the same scores for the seen conditionings ($\mathcal{S}_s$) and the unseen coarse ($\mathcal{S}_{u^2}$) ones. However, scene and object consistencies are more accurate for $\mathcal{S}_s$ than for $\mathcal{S}_{u^2}$. If we compare the generalization to $\mathcal{S}_u$ against the generalization to $\mathcal{S}_{u^2}$, we observe that for scenes and objects, $\mathcal{S}_{u^2}$ presents lower recall and consistency for most of the methods. Further, object-wise precision is also affected for $\mathcal{S}_{u^2}$, altering the methods' ranking. We note that OC-GAN loses its $1st$ ranked position (shared with the LostGANs) in $\mathcal{S}_u$ to become $3rd$ (together with G2Im-GT) in $\mathcal{S}_{u^2}$, and suffers an important loss in terms of object FID, suggesting that OC-GAN may suffer particularly from unseen coarse conditionings. For completeness, and to assess whether the marginally lower recall and consistency of $\mathcal{S}_{u^2}$ come from the heavier long tail of $\mathcal{S}_{u^2}$, we subsample $\mathcal{D}_s$ to match the same class distribution as in $\mathcal{D}_{u^2}$. In this case, this experiment can only be performed object-wise, as none of the class combinations in $\mathcal{D}_{u^2}$ appears in $\mathcal{D}_s$. The results presented in Figure 2 show that precision, recall and consistency of both splits are very similar.

Finally, looking at Figure 3, we observe that images generated from unseen coarse conditionings present lower quality than images generated from seen or unseen finegrained conditionings for all methods. For instance, if we consider the conditioning in Figure 3c, which depicts a person skiing with a dog, we observe that the overall quality of the generated images is rather low, with oversmoothed or implausible looking persons and, in some cases, invisible dogs. Similarly, if we consider the conditioning of an airplane on a paper, most of the resulting generated images also exhibit rather low quality. The most compelling image is the one generated by G2Im-GT, which directly leverages a high quality (GT) layout and can focus on filling in the object textures. Additional qualitative results can be found in Appendix F.

4.5 Ablation of pipeline components

In this subsection, we switch the gears of the analysis and design an ablation study to better understand how elements of complex scene generation pipelines affect their performance. In particular, we inspect the following components: (1) the spatial conditioning normalization module; (2) the effect of imperfect instance segmentation mask prediction on the image quality; and (3) the influ-

ence of scene graph loss introduced in OC-GAN. We carry our analysis on models derived from LostGANv2. We use hyperbands (Li et al., 2017) to find the best hyperparameters for each model.

Spatial conditioning normalization modules. We compare LostGANv2's ISLA-norm to the variant of SPADE introduced in OC-GAN, hereinafter called SPADE(oc). The results of these comparisons are presented in Table 1 both for unseen finegrained conditionings and for unseen coarse conditionings. When comparing ISLA to SPADE(oc) in the unseen finegrained conditionings scenario, we observe that ISLA improves upon diversity metrics such as scene recall, object recall and DS on average. The high scene variation achieved by ISLA could possibly be explained given it design, which leverages latent vectors to compute instance-wise normalizing parameters. Interestingly, ISLA also favors object-wise metrics, by improving object consistency and accuracy. The same trend can be observed in the case of unseen coarse conditionings. This might be due to the fact that ISLA considers individual objects information through its input. However, SPADE(oc) dominates the scene-wise metrics, leading to better scene precision and scene consistency on average. This might be due to the module's design, which directly exploits bounding box semantic segmentation masks. Additional ablations with variants of SPADE can be found in Appendix E.

Ground-truth masks (M). As expected, using GT masks in ISLA+M greatly improves over ISLA. In particular, for $\mathcal{S}_u$, using ground truth masks improves upon most of the ISLA results on average, at the expense of scene recall and DS decrease. We observe the same trend for $\mathcal{S}_{u^2}$, where improvements are mainly observed in terms of object and scene precision and consistency metrics. Again, using GT masks reduces the scene layout variability and translates into lower diversity-related scores. Moreover, using unseen GT masks forces the generator to hallucinate objects with unseen shapes. This may in some cases lead to undesirable outcomes, with worse generated image quality than when using predicted masks.

Scene-graph similarity module (SG). When endowing ISLA with SG (ISLA+SG), we observe notable boosts in scene-wise conditional consistency metrics in $\mathcal{S}_u$ (see scene consistency and F1). This is not surprising given that the module's goal is to match information from scene graphs and generated images to encode the location of specific objects in the scene. It is worth noting that adding the SG module does not come at the expense of intra-conditioning diversity but harms scene recall on average. In $\mathcal{S}_{u^2}$, we observe a marked increase in object recall, object consistency, F1, and both FIDs, while noticing a slight decrease in terms of average precision, emphasizing the general benefits of the SG module for unseen coarse conditionings. When using both GT masks and the SG module (ISLA+M+SG), we observe further improvements. For both $\mathcal{S}_u$ and $\mathcal{S}_{u^2}$, ISLA+M+SG achieves the highest results in terms of average scene precision, consistency, F1, as well as FID, and improves upon object consistency, accuracy and FID, while harming diversity metrics. Note that ISLA+M+SG slightly mitigates the decrease in DS/recall experienced by ISLA+M. The compelling improvements of ISLA+M+SG suggest that the quality of the generated masks plays a crucial role in improving generalization to unseen conditionings. Figure 3 shows generated images for both variants of the model with the SG module. We can see that the generated images display high frequency details for the scenarios of seen and unseen finegrained conditionings. For unseen coarse conditionings, we can appreciate better results for ISLA+SG that could be combining both instance-wise compositionality from ISLA and the consistency boost of SG module. Moreover, ISLA+M+SG leverages GT masks, showing further quality improvement.

5 CONCLUSIONS

We proposed a methodology to analyze current complex scene conditional generation methods. Using this methodology, we evaluated recent complex scene generation pipelines from the points of view of training data fitting as well as their ability to generalize unseen scenes. Through an extensive evaluation, we observed that current methods: (1) fit the training distribution with a moderate success, generating recognizable scenes, but suffering from the loss of high frequency details at individual object level; (2) display decent generalization to unseen finegrained conditionings; and (3) have significant space for improvement when it comes to generating images from unseen coarse conditionings. Thus, complex scene generation is still a stretch goal, especially for unseen conditionings, where the methods' performances are still in their infancy. To provide some pointers on how to move towards high quality complex scenes generation for unseen scenarios, we performed an ablation study that highlights the importance of model components when it comes to generalization.

REFERENCES

Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4561–4569, 2019.

Yaniv Benny, Tomer Galanti, Sagie Benaim, and Lior Wolf. Evaluation metrics for conditional image generation. *arXiv preprint arXiv:2004.12361*, 2020.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1xsqj09Fm`.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1511–1520, 2017.

Terrance DeVries, Adriana Romero, Luis Pineda, Graham W Taylor, and Michal Drozdzal. On the evaluation of conditional gans. *arXiv preprint arXiv:1907.08175*, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

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. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Generating multiple objects at spatially distinct locations. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=H1edIiA9KQ`.

Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

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*, pp. 1219–1228, 2018.

Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9895–9904, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. The gan landscape: Losses, architectures, regularization, and normalization. *arXiv preprint arXiv:1807.04720*, 2018.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, pp. 3929–3938, 2019.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pp. 700–709, 2018.

Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=ByS1VpgRZ`.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Luis Pineda, Amaia Salvador, Michal Drozdzal, and Adriana Romero. Elucidating image-to-set prediction: An analysis of models, losses and datasets. *CoRR*, abs/1904.05709, 2019. URL http://arxiv.org/abs/1904.05709.

Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8808–8816, 2018.

Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 12268–12279. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9393-classification-accuracy-score-for-conditional-generative-models.pdf.

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.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, pp. 5228–5237, 2018.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

K. Shmelkov, C. Schmid, and K. Alahari. How good is my GAN? In *Proceedings of European Conference on Computer Vision*, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 10531–10540, 2019.

Wei Sun and Tianfu Wu. Learning layout and style reconfigurable gans for controllable image synthesis. *arXiv preprint arXiv:2003.11571*, 2020.

Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. *arXiv preprint arXiv:2003.07449*, 2020.

Hao Tang, Dan Xu, Yan Yan, Philip HS Torr, and Nicu Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7870–7879, 2020.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1747–1756. JMLR.org, 2016a. URL http://dl.acm.org/citation.cfm?id=3045390.3045575.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016b. URL https://arxiv.org/abs/1609.03499.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 613–621. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6194-generating-videos-with-scene-dynamics.pdf.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018.

S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, 2017.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 2018a.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018b.

Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

A EXTENDED EXPERIMENTAL SETUP

Dataset details. We use COCO-Stuff (Caesar et al. (2018)), with training and evaluation splits consisting of 75k and 3k images respectively, following Sun & Wu (2019). We use the whole training set as $\mathcal{D}_s$, and divide the validation set into three sets, namely $\mathcal{D}_u$, $\mathcal{D}_{u^2}$ and a set used for hyper-parameter search and validation $\mathcal{D}_v$. This results in $|\mathcal{D}_s| = 75$k, $|\mathcal{D}_u| = 675$, $|\mathcal{D}_{u^2}| = 1375$ and $|\mathcal{D}_v| = 1024$ triplets to evaluate the quality of the generation processes. We use the above described splits to compute the scene-wise metrics, and crop individual objects from the scenes to compute object-wise metrics. In particular, we consider objects from *things* categories exclusively, given that crops of *stuff* categories most often also contain foreground objects from *things* categories. This results in 145k objects crops extracted from $\mathcal{D}_s$, 1163 from $\mathcal{D}_u$, 2706 from $\mathcal{D}_{u^2}$ and 2062 from $\mathcal{D}_v$. In both cases and only for evaluation purposes, we preprocessed the data sets by merging redundant and prone-to-confusion classes (see Appendix B for a detailed explanation of the dataset preprocessing).

Evaluation networks. Our object classifier has a ResNext101 backbone and is trained on individual objects cropped from the images in $\mathcal{D}_s$. Similarly, the image(scene)-to-set predictor has a ResNext101 backbone and is trained on $\mathcal{D}_s$. In both cases, the architecture and optimization hyper-parameters are chosen independently on $\mathcal{D}_v$ and by means of the Hyperband algorithm (Li et al., 2017), following (Pineda et al., 2019). Both evaluation networks are trained on the merged COCO-Stuff categories.

Evaluation details. We report all results with 128x128 images, as all studied methods support this resolution size. The image features to compute precision, recall and consistency metrics are extracted from the evaluation networks (image-to-set or object classifier). In both cases, we use the features from the last fully connected layer before the classifier. For each finegrained conditioning, we generate images with each method from 5 random seeds, which change the input noise z fed to the generators, and so allow for some image variation. The metrics are computed and averaged across the 5 seeds for $\mathcal{S}_s$, $\mathcal{S}_u$ and $\mathcal{S}_{u^2}$.

B DATASET DETAILS AND CATEGORIZATION CLEANING

In COCO-Stuff the distinctions between *stuff* categories and some thing categories is difficult to grasp even for humans. Analyzing a confusion matrix extracted for feature extractor on object crops and some designed rules, we cleaned the COCO-Stuff category list to minimize mistakes in

evaluation due to poor dataset categorization. Note that is is only done for evaluation, since all studied generative models have been trained with the original COCO-Stuff categories.

We extracted VGG16 Simonyan & Zisserman (2014) features for all cropped training objects. We computed pair-wise Euclidean distances between all feature crops, and assigned as the predicted class the one corresponding to the closest crop. Using these class assignations, we obtain a confusion matrix, that illustrates which are the most confused class and by what other classes they are confused. To decide which classes to merge, we applied the following steps:

- For each target class class, we took the 5-top most confused classes that have, at least, the same probability as the target class.
- We discarded the class 'person' from any possible confusion, as this miss-classification can come from the bounding box crop.
- We also do not consider some confused classes where the confusion can come from the bounding box: for instance, confusing 'building' crops for 'tree' or 'sky' class.
- We did not mix *stuff* classes that contain the suffix '-other', as they usually contain a high variety of classes by themselves that are difficult to categorize. Exception: when the entire super-class (grouping of COCO-Stuff classes) can be mixed into one, including classes with suffix '-other'.
- Only merge classes that by individual inspection, are semantically close.

C METHODS UNDER STUDY

G2im (Ashual & Wolf, 2019) uses a scene graph as input conditioning, where nodes represent objects in the scene with their class annotation, and edges represent the relative positions among those objects. The scene graph is used to generate instance segmentation masks and instance bounding box layouts, which are fused to obtain a semantic segmentation mask of the scene. The resulting mask is then combined with object features, obtained separately by feeding object crops from training images to an auto-encoder, to create the final generated image.

G2im-GT (Ashual & Wolf, 2019) is a variant of G2im, where the instance segmentation masks and instance bounding box layouts are not generated, but directly taken from the ground-truth data.

LostGANv1 (Sun & Wu, 2019) uses instance bounding box layouts with their corresponding class annotations as input conditioning. This input is used to compute object embeddings, which are subsequently used to generate instance segmentation masks. The instance segmentation masks are then fed to a scene generator which uses a spatial conditioning module, called ISLA-norm, that learns independent normalization parameters per object instance. These normalization parameters are applied through all stages of the scene generation process.

LostGANv2 (Sun & Wu, 2020) takes the same input conditioning, and computes the same object embeddings as LostGANv1. In this case, both the instance segmentation mask generator and the scene generator are extended to follow a multi-stage generation process. In particular, the instance segmentation masks are refined using the mask from the previous stage and some intermediate image features from the scene generator. In turn, each time the mask is refined, it is used to update the normalization parameters used in the scene generator, therefore using a stage-wise ISLA-norm.

OC-GAN (Sylvain et al., 2020) takes the same input conditioning as the LostGANs. In contrast to LostGANv1's ISLA-norm, OC-GAN uses a variant of SPADE's normalization module (Park et al., 2019), which spatially conditions the scene generation on instance segmentation masks, per instance bounding box boundaries, as well as a semantic bounding box layout. Moreover, OC-GAN uses a scene-graph similarity module, called SGSM, that attends scene visual features with graph features, and matches global and local information from both the scene graph and the generated image to encode the location of specific objects in the scene. This scene graph similarity module is used at training time but discarded at test time.

	↑ **Mean F1-score**			↑ **Mean accuracy**		
	$\mathcal{S}_s$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$	$\mathcal{S}_s$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$
GT	75.2 ± 0.0	80.9 ± 0.0	64.7 ± 0.0	61.0 ± 0.0	70.1 ± 0.0	54.8 ± 0.0
G2im*	41.2 ± 0.1	48.1 ± 0.5	38.1 ± 0.3	12.6 ± 0.1	19.7 ± 0.4	9.7 ± 0.3
G2im-GT*	50.6 ± 0.1	57.5 ± 0.7	46.3 ± 0.5	24.2 ± 0.1	31.2 ± 0.2	19.8 ± 0.2
LostGANv1	63.7 ± 0.1	73.8 ± 0.6	57.7 ± 0.4	$\mathbf{30.9} \pm 0.1$	41.0 ± 0.6	26.8 ± 0.3
LostGANv2	$\mathbf{64.8} \pm 0.1$	75.9 ± 0.1	58.6 ± 0.3	$\mathbf{30.8} \pm 0.1$	40.6 ± 1.1	26.3 ± 0.5
OC-GAN	-	$\mathbf{80.0} \pm 0.1$	$\mathbf{63.8} \pm 0.3$	-	$\mathbf{49.6} \pm 0.9$	$\mathbf{36.7} \pm 0.5$

Table 2: F1-score and accuracy averaged across examples in splits $\mathcal{S}_s$, $\mathcal{S}_u$ and $\mathcal{S}_{u^2}$. This table presents the mean and standard deviation of 5 different random seeds for the generative models. * Trained using the open-sourced code.

	↑ **DS**		
	$\mathcal{S}_s$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$
GT	-	-	-
G2im*	0.56 ± 0.06	0.55 ± 0.07	0.56 ± 0.06
G2im-GT*	0.48 ± 0.09	0.47 ± 0.11	0.48 ± 0.09
LostGANv1	0.47 ± 0.09	0.46 ± 0.09	0.49 ± 0.08
LostGANv2	0.45 ± 0.09	0.43 ± 0.09	0.46 ± 0.08
OC-GAN	-	0.14 ± 0.07	0.13 ± 0.06

Table 3: Diversity score using LPIPS metric in all evaluation splits $\mathcal{S}_s$, $\mathcal{S}_u$, $\mathcal{S}_{u^2}$. * Trained using the open-sourced code.

D METRIC TABLES

Metrics reported in the analysis are detailed as follows: Table 2 presents the F1-score and accuracy, Table 3 the diversity score, and Table 4 the FID values for both scenes and objects.

E EXTENDED ABLATION METRICS

The variants of SPADE that are ablated are the following: (1) **SPADE** uses a global semantic segmentation mask and object boundaries as in (Park et al., 2019), (2) **SPADE(i)** that uses object embeddings and instance masks, (3) **SPADE(s)** that uses a global semantic segmentation mask obtained from fusing instance masks, and (4) **SPADE(oc)** that uses object embeddings, instance segmentation masks, object boundaries and a bounding box semantic layout, as proposed by Sylvain et al. (2020). The results of these comparisons are presented in detail in Tables 5, 6, 7, 8.

F ADDITIONAL QUALITATIVE RESULTS

Figure 4 shows more qualitative results, for state-of-the-art methods and two of the best ablated pipelines.

	↓ FID			↓ Object FID		
	$\mathcal{S}_s$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$	$\mathcal{S}_s$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$
GT	0.0	0.0	0.0	0.0	0.0	0.0
G2im*	62.3 ± 0.1	137.6 ± 1.6	99.9 ± 1.0	56.2 ± 0.1	112.3 ± 0.5	80.8 ± 0.6
G2im-GT*	40.7 ± 0.1	112.9 ± 1.9	82.1 ± 0.8	31.0 ± 0.5	85.2 ± 0.9	55.8 ± 0.4
LostGANv1	15.2 ± 0.1	84.2 ± 1.0	57.2 ± 0.4	20.3 ± 0.2	71.9 ± 0.5	45.2 ± 0.2
LostGANv2	**12.8** ± 0.1	**80.0** ± 0.4	**55.2** ± 0.5	**15.5** ± 0.6	**67.1** ± 0.7	**40.9** ± 0.3
OC-GAN	-	85.8 ± 0.5	60.1 ± 0.2	-	85.8 ± 0.6	59.4 ± 0.3

Table 4: FID for each data split and model. As a reference, ground-truth images from each of the $\mathcal{D}_s$, $\mathcal{D}_u$ and $\mathcal{D}_{u^2}$ splits are used accordingly.Both splits have different number of points, so FID not comparable across splits. This table presents the mean and standard deviation of 5 different random seeds for the generative models.* Trained using the open-sourced code.

	↑ Scene Precision		↑ Scene Recall		↑ Scene Consistency	
	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$
LostGANv2	89.1 ± 1.4	87.5 ± 0.3	**63.9** ± 2.0	**58.0** ± 0.9	61.1 ± 0.9	43.5 ± 0.4
OC-GAN	89.2 ± 0.9	86.9 ± 0.6	-	-	63.6 ± 1.0	45.8 ± 0.5
ISLA	89.7 ± 0.9	88.4 ± 0.5	**65.2** ± 1.1	**58.5** ± 0.7	63.6 ± 1.2	45.9 ± 0.8
SPADE	89.4 ± 0.4	**91.1** ± 0.3	53.2 ± 0.5	49.2 ± 0.5	61.5 ± 0.6	46.0 ± 0.3
SPADE(i)	85.9 ± 0.7	85.9 ± 0.5	53.0 ± 0.5	46.3 ± 0.8	49.5 ± 0.4	37.7 ± 0.7
SPADE(s)	90.0 ± 1.0	86.3 ± 0.7	47.8 ± 1.2	44.0 ± 0.7	53.1 ± 0.4	38.4 ± 0.6
SPADE(oc)	90.9 ± 0.5	90.1 ± 0.6	57.0 ± 0.5	53.2 ± 0.7	65.7 ± 0.7	47.6 ± 0.3
ISLA+M	**91.6** ± 0.4	**91.7** ± 0.5	57.7 ± 1.4	55.3 ± 0.6	68.1 ± 1.4	53.1 ± 0.5
ISLA+SG	90.5 ± 0.5	88.4 ± 0.7	**64.9** ± 0.7	**59.1** ± 0.4	66.1 ± 1.1	47.2 ± 0.5
ISLA+M+SG	**92.1** ± 0.5	**91.8** ± 0.4	58.8 ± 1.4	54.8 ± 0.6	**72.5** ± 0.7	**55.9** ± 0.6

Table 5: Ablation results for all ablated pipelines, scene-wise precision, recall and coverage for the two splits $\mathcal{S}_u$ and $\mathcal{S}_{u^2}$. This table presents the mean and standard deviation of 5 random seeds that control the input noise.

	↑ Object Precision		↑ Object Recall		↑ Object Consistency	
	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$
LostGANv2	69.6 ± 0.3	73.2 ± 0.5	30.2 ± 1.2	25.8 ± 0.5	40.9 ± 0.7	36.3 ± 0.6
OC-GAN	70.4 ± 0.7	70.9 ± 0.8	-	-	48.2 ± 0.9	42.1 ± 0.2
ISLA	**73.1** ± 0.5	74.7 ± 0.6	28.4 ± 0.8	24.9 ± 0.7	45.9 ± 0.9	40.5 ± 0.2
SPADE	68.1 ± 0.6	70.0 ± 0.5	24.5 ± 0.6	20.4 ± 0.4	38.1 ± 0.9	32.5 ± 0.5
SPADE(i)	69.5 ± 0.8	71.2 ± 1.0	21.1 ± 1.2	19.5 ± 0.6	35.6 ± 0.9	32.4 ± 0.4
SPADE(s)	70.4 ± 1.2	72.0 ± 0.6	21.7 ± 0.9	18.9 ± 0.6	35.0 ± 1.0	31.8 ± 0.6
SPADE(oc)	70.2 ± 0.9	73.6 ± 0.4	26.6 ± 1.1	24.5 ± 0.5	41.5 ± 0.7	35.7 ± 0.7
ISLA+M	**73.8** ± 0.4	**77.7** ± 0.3	**31.6** ± 0.2	**28.1** ± 0.8	**49.3** ± 0.6	**43.4** ± 0.9
ISLA+SG	70.5 ± 1.4	74.2 ± 1.0	**32.2** ± 0.8	**28.3** ± 0.5	45.3 ± 1.3	**42.7** ± 1.1
ISLA+M+SG	**72.6** ± 1.7	74.3 ± 0.7	28.7 ± 1.2	25.0 ± 0.4	**50.6** ± 1.2	**43.7** ± 0.7

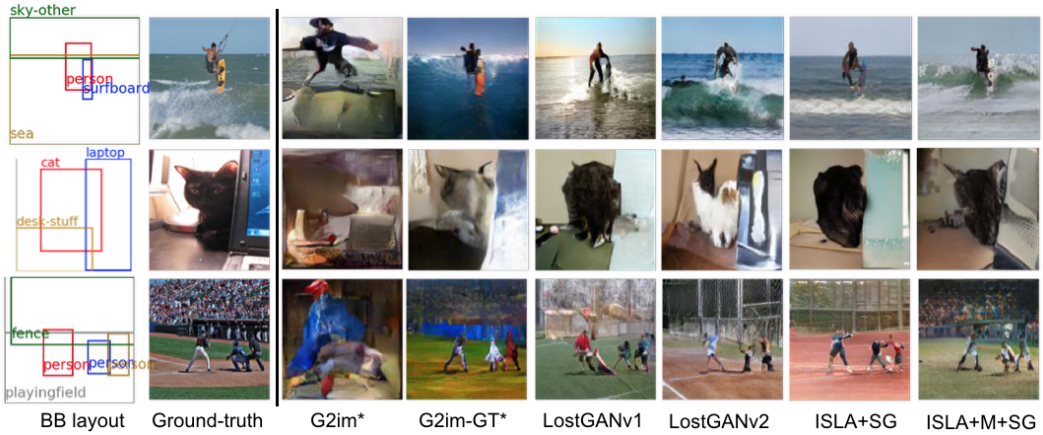
Table 6: Ablation results for all ablated pipelines. Metrics reported are object-wise precision, recall and coverage for the two splits $\mathcal{S}_u$ and $\mathcal{S}_{u^2}$.This table presents the mean and standard deviation of 5 random seeds that control the input noise.

	↑ **Mean F1-score**		↑ **Mean accuracy**		↑ **DS**	
	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$	$\mathcal{S}_u$	$\mathcal{S}_{u^2}$
ISLA	76.7 ± 0.3	59.6 ± 0.2	43.1 ± 0.7	30.7 ± 0.3	**0.41** ± 0.10	**0.44** ± 0.09
SPADE	74.1 ± 0.4	57.6 ± 0.2	39.8 ± 0.6	25.0 ± 0.3	0.01 ± 0.01	0.01 ± 0.01
SPADE(i)	64.9 ± 0.5	52.6 ± 0.3	32.1 ± 0.6	23.8 ± 0.3	**0.26** ± 0.16	**0.26** ± 0.15
SPADE(s)	65.9 ± 0.4	53.2 ± 0.3	29.5 ± 0.7	22.4 ± 0.5	0.15 ± 0.13	0.17 ± 0.14
SPADE(oc)	75.9 ± 0.5	59.2 ± 0.3	39.3 ± 0.9	26.4 ± 1.0	0.17 ± 0.07	0.20 ± 0.07
ISLA+M	77.0 ± 0.3	61.1 ± 0.2	45.4 ± 0.3	29.0 ± 0.3	0.12 ± 0.08	0.11 ± 0.06
ISLA+SG	78.9 ± 0.1	62.2 ± 0.1	42.1 ± 0.6	31.4 ± 0.8	**0.41** ± 0.10	**0.44** ± 0.08
ISLA+M+SG	**80.8** ± 0.4	**64.3** ± 0.4	**49.3** ± 1.0	**34.5** ± 0.3	0.18 ± 0.07	0.19 ± 0.08

Table 7: Ablation results for all ablated pipelines. F1-score, accuracy and DS for the two splits $\mathcal{S}_u$ and $\mathcal{S}_{u^2}$.This table presents the mean and standard deviation of 5 random seeds that control the input noise.

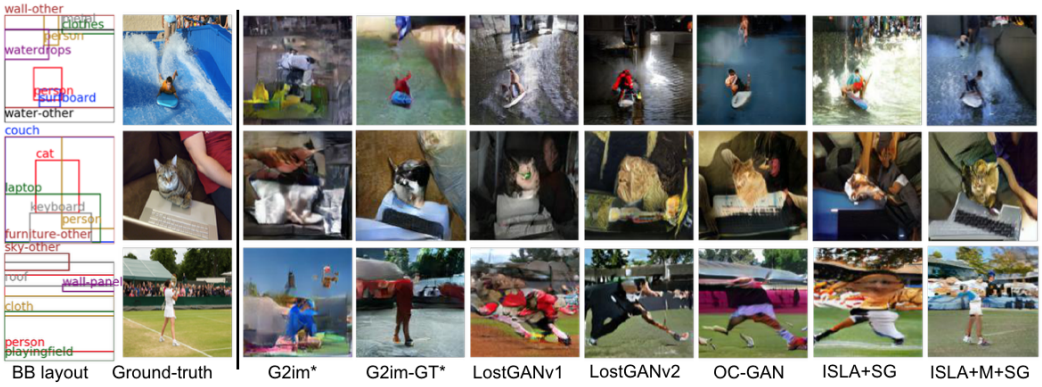
	# Parameters	↓ **FID**		↓ **Object FID**	
	Generator	$\mathcal{D}_u$	$\mathcal{D}_{u^2}$	$\mathcal{D}_u$	$\mathcal{D}_{u^2}$
ISLA	40.5M	78.5 ± 0.3	53.5 ± 0.5	66.1 ± 0.3	38.2 ± 0.3
SPADE	40.8M	83.8 ± 0.3	62.2 ± 0.2	70.8 ± 0.1	44.9 ± 0.9
SPADE(i)	40.8M	98.4 ± 0.4	64.2 ± 0.4	76.3 ± 0.3	44.5 ± 0.2
SPADE(s)	40.8M	98.4 ± 0.7	64.5 ± 0.3	77.8 ± 0.3	44.1 ± 0.2
SPADE(oc)	41.7M	78.7 ± 0.5	55.6 ± 0.2	67.6 ± 0.5	40.2 ± 0.1
ISLA+M	35.3M	75.2 ± 0.3	52.3 ± 0.1	61.7 ± 0.1	36.0 ± 0.2
ISLA+SG	47.5M	78.4 ± 0.6	52.1 ± 0.2	62.6 ± 0.4	34.2 ± 0.1
ISLA+M+SG	42.3M	**72.6** ± 0.5	**50.2** ± 0.2	**59.2** ± 0.2	**33.2** ± 0.1

Table 8: Ablation results for all ablated pipelines. Number of parameters per generator for each of the pipelines, scene FID and object FID for the two splits $\mathcal{S}_u$ and $\mathcal{S}_{u^2}$.This table presents the mean and standard deviation of 5 random seeds that control the input noise.



(a) Generated images for seen conditionings

(b) Generated images for unseen finegrained conditionings



(c) Generated images for unseen coarse conditionings

Figure 4: Extended qualitative results. Generated images with resolution 128×128 for several methods. * Trained using the open-sourced code.