Compositional Transformers for Scene Generation
Supplementary Material

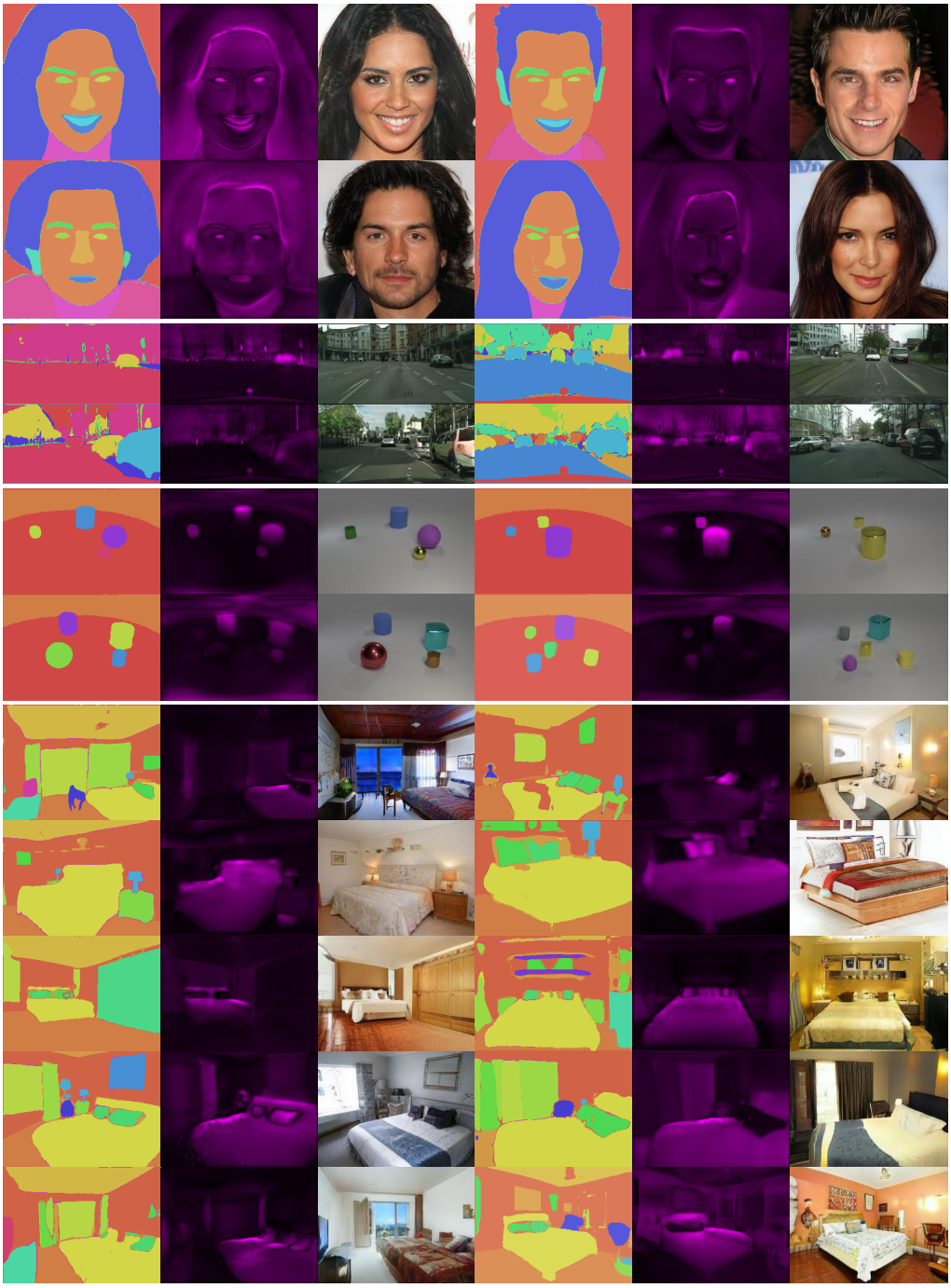


Figure 1: A visualization of the **layouts and (unsupervised) depth maps** produced by GANformer2's (*planning stage*) while synthesizing varied images, making the generative process more structured and interpretable.

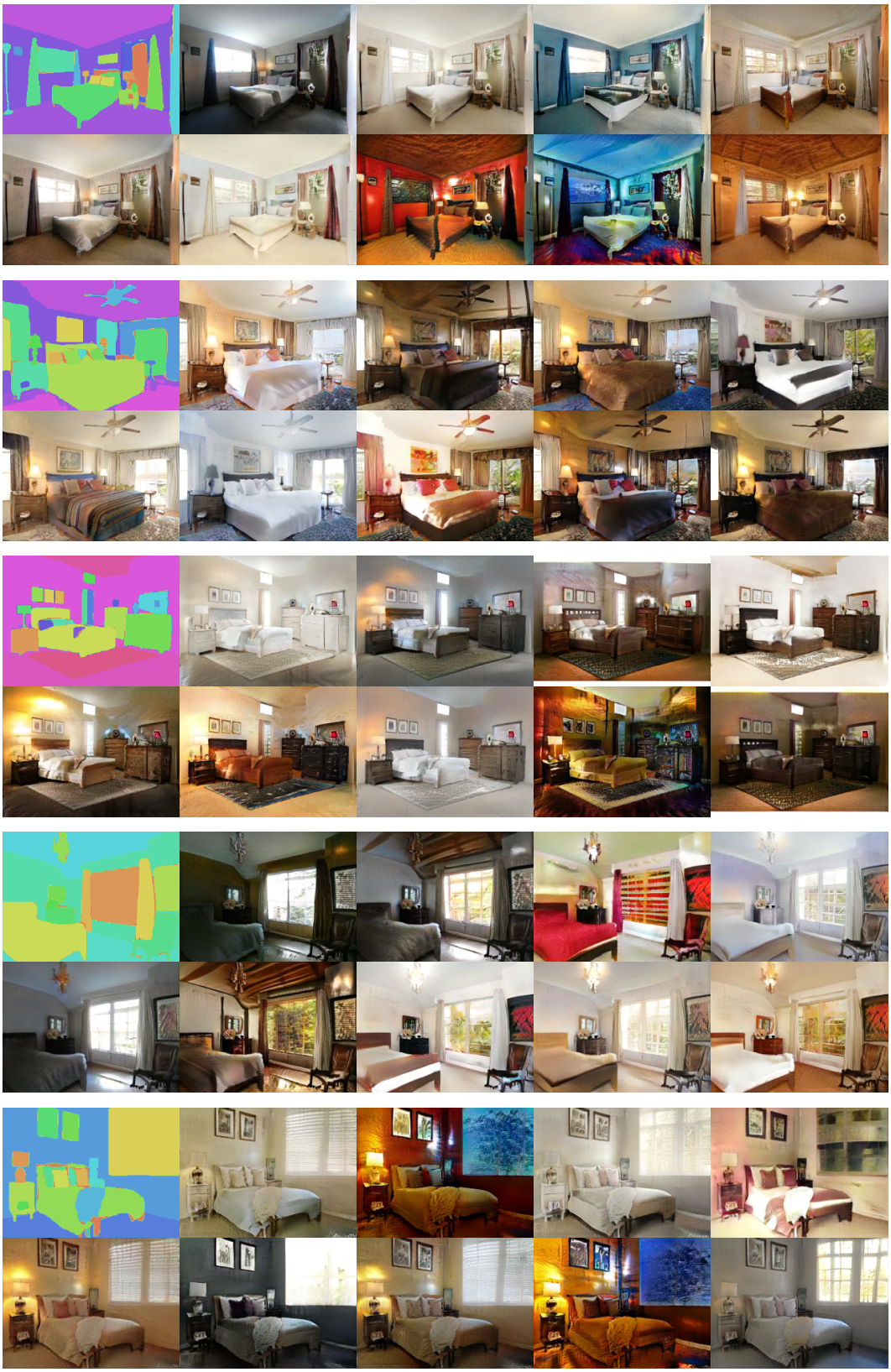


Figure 2: **GANformer2 style variations** conditioned on the same layouts for **LSUN-Bedrooms**.

Figure 3: **GANformer2 style variations** conditioned on the same layouts for the **CelebA dataset**.

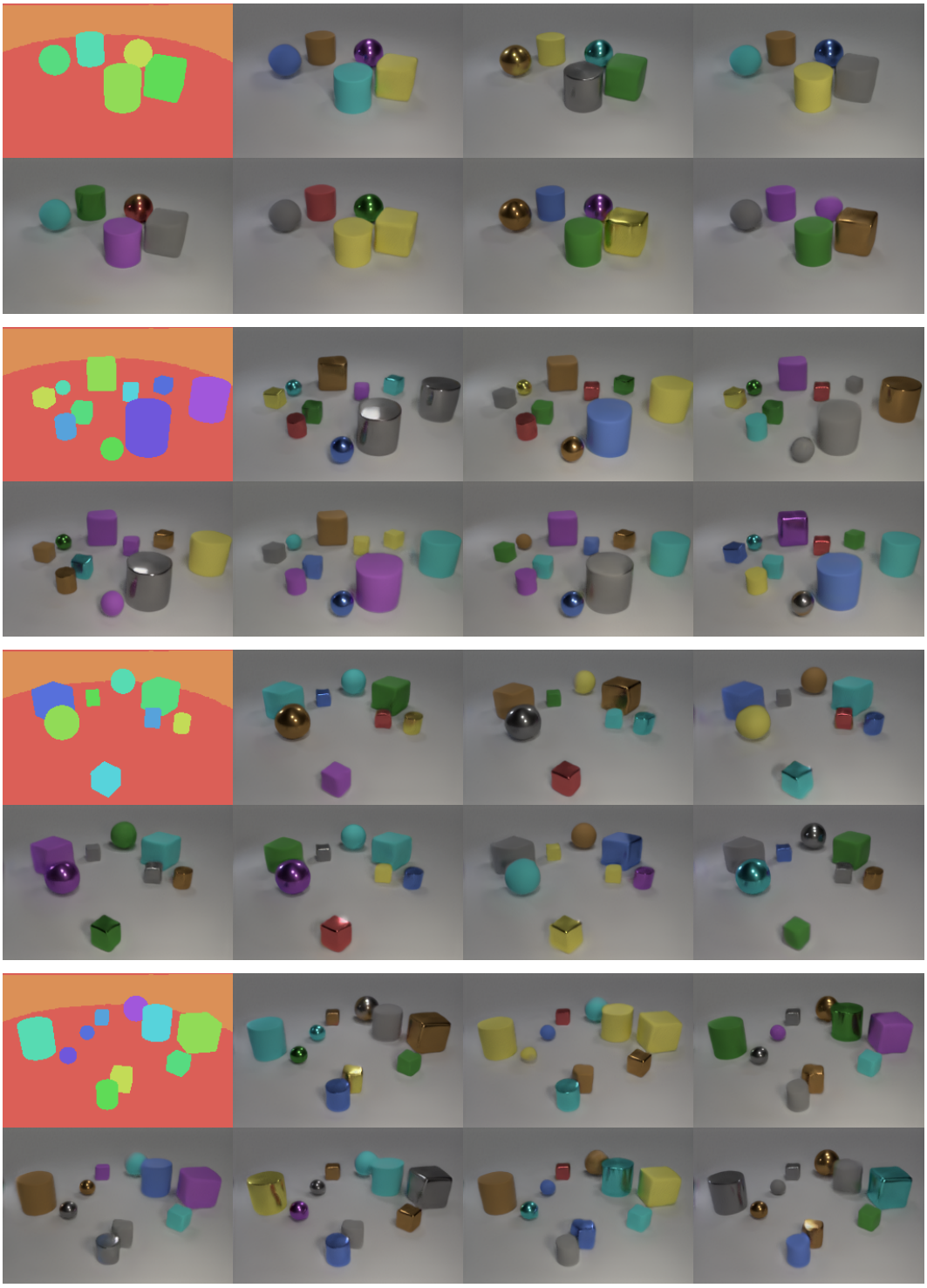


Figure 4: **GANformer2 style variations** conditioned on the same layouts for the **CLEVR dataset**.

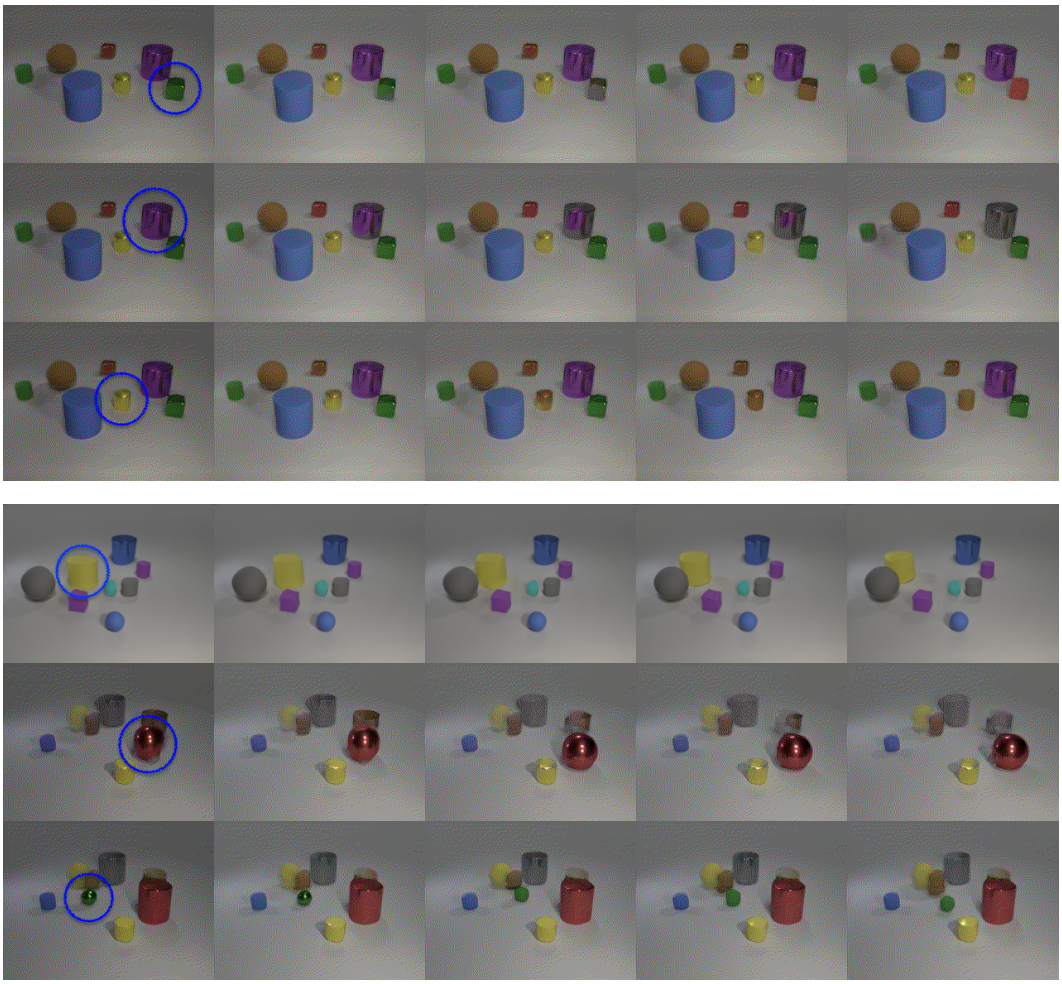
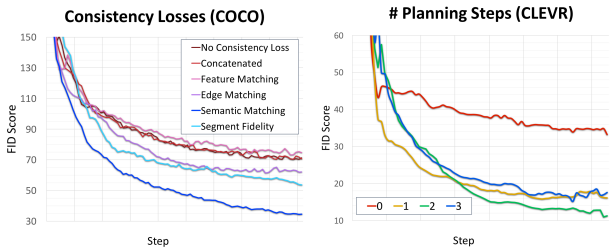


Figure 5: **Object controllability.** GANformer2 supports entity-level manipulation, and separates style from structure, enabling independent control over objects and attributes. For each segment s_i, we can modify its **structural** (rows 1-3) or **textural** (rows 4-6) properties by varying the latents u_i or w_i respectively, the former used during **planning** while the latter through **execution**.



Steps	0	1	2	3
CLEVR	11.95	7.54	**4.70**	5.25
FFHQ	8.21	**7.77**	8.84	9.92
Cityscapes	9.83	8.51	7.26	**6.21**
Bedrooms	11.45	9.37	**6.05**	7.21
COCO	31.15	28.45	24.30	**21.58**

Figure 6: **Learning curves and variation experiments** when varying the loss function for the execution stage **(left)**, and the number of recurrent planning steps for scene layout synthesis **(center, right)**. In each step, a random number of segments is generated, sampled from a trainable normal distribution, to flexibly model highly-structured scenes and capture conditional object dependencies within them while still maintaining good computational efficiency. 0 denotes a non-compositional model that generates the layouts in a single pass rather than as a collection of segments.

A Overview

In the following, we provide additional qualitative and quantitative experiments for the GANformer2 model. **Figure 9** shows visualizations of the model's generative process as it produces layouts that consist of depth maps and segmentations, that in turn guide and facilitate the synthesis of output photo-realistic images. The model's compositional structure does not only enhance its transparency, but also increases its controllability, allowing GANformer2 to manipulate properties of individual objects without negatively altering their surroundings, and even develop capacity for amodal completion of occluded objects, as illustrated in **figure 13**. The obtained decoupling between structure and style is further illustrated through **figures 10-12**, respectively showing variability along images style while maintaining consistency over the structure.

We believe the diversity achieved by GANformer2 results from the new structural losses we introduce (section 3.3, main paper), which we compare to several baselines and alternatives in **section C**. In **section D**, we proceed through additional model ablation and variation studies, to asses the contribution of its different architectural components, and the number of recurrent planning steps in particular. **Section E** focuses on the training configuration, comparing and contrasting between paired vs. unpaired settings. We conclude the supplementary by providing implementation details (**section F**), description of baselines and competing methods (**section G**), and information about data preparation procedures (**section H**).

B Style & Structure Disentanglment

GANformer2 decomposes the synthesis into two stages of planning and execution, the former produces the scene layout and structure while the latter controls its texture, colors and style. Figures **figures 10-12** demonstrate the high content variability achieved over different datasets, while still complying with shared source layouts. Indeed, the styles vary largely between one sample to another, and variation is notably observed in local structures and elements, while still conforming to the layout at the global scale. This is most noticeable in the CelebA case (figure 11), where some of the images produced for given layouts depict young people while other old ones, and likewise some feature straight hair while other curly hair, even when originating from the same layout.

The high diversity and consistency demonstrated here, which also quantitatively surpass the competing approaches (section 4.2, main paper), may be attributed to the new structural loss functions we employ (section 3.3, main paper). These purely generative losses liberate us from resorting to perceptual and feature-matching objectives common to prior work, which impose unnecessary and unjustified pixel-wise similarity conditions on the generated images, inhibiting their diversity. By sidestepping this reliance, we can unlock wider variation among the synthesized scenes, both in terms of structure and style. See a comparison between synthesized samples of different approaches at figure .

Table 1: **Paired vs. Unpaired Training**

	Paired	Unpaired	Parallel
CLEVR	**4.70**	6.58	6.72
FFHQ	**7.77**	8.12	8.35
Cityscapes	**6.21**	7.32	7.81
Bedrooms	**6.05**	8.24	8.55
COCO	**21.58**	25.02	27.41

Table 2: **Hyperparameters**

	Max #Latents	Latents Overall Dim	R1 reg weight (γ)
FFHQ	20	128	10
CLEVR	16	512	40
Cityscapes	64	512	20
Bedroom	64	512	100
COCO	64	512	100

Table 3: **Dataset Configurations**

Dataset	Size	Resolution	Augment
FFHQ	70,000	256×256	Flip
CLEVR	100,015	256×256	None
Cityscapes	24,998	256×256	Flip
Bedrooms	3,033,042	256×256	None
COCO	287,330	256×256	Crop + Flip

Table 4: **COCO Subset Statistics**

Category	Size	Category	Size
People	**50293**	**Sports**	**37304**
Children	13840	Skiing	5840
Eating	5096	Baseball	10067
Playing	5798	Tennis	11721
Rural	12084	Skating	6412
Others	13475	Surfing	3264
Animals	**64066**	**Indoors**	**30946**
Dogs	5245	Bathrooms	17121
Cats	5543	Kitchens	10427
Birds	8153	Bedrooms	3398
Sheep	8134		
Bears	8138	**Outdoors**	**20713**
Elephants	16204	Beaches	8396
Giraffes	6352	Cities	8578
Zebras	6297	Streets	3739
Vehicles	**53714**	**Misc**	**30294**
Airplanes	21232	Desserts	11370
Buses	7436	Food	4176
Trains	6661	Toys	7242
Bikes	18385	Electronics	7506

C Structural Losses Comparison

As discussed in section 3.3 (main paper), we introduce two new losses to the model's execution stage, where we transform input layouts into output photo-realistic images. The new losses of **Semantic Matching** and **Segment Fidelity** respectively encourage structural consistency between the layouts and the images, and fidelity at the level of the individual segment. In figure 14, we compare their performance in terms of FID score with several alternative objectives over the COCO dataset.

Specifically, we explore the following baselines: (1) Using *no consistency* loss at all (training with the standard fidelity loss only $\mathcal{L}(D(\cdot))$), in hopes that the model's layout-conditioned feature modulation will serve as an architectural bias to promote structural alignment; (2) A simple *concatenation* of the layout S and image X as they are fed into the discriminator D; (3) A *Feature-Marching loss*, as widely used in prior work, $\mathcal{L}_{FM}(f(X), f(X'))$, that compares VGG features of the generated image with those of the source natural image X' that underlies the input layout S; and (4) an *Edge-Matching* loss, $\mathcal{L}_{EM}(e(S), e(S'))$, that compares the binary segmentation edges between the input layout S and a layout S' induced by the generated image X.

As figure 14 shows, our newly proposed losses, and the Semantic-Matching loss especially, surpass the discussed baselines and effectively encourage GANformer2 to generate high-quality images. For the Semantic-Matching loss, $\mathcal{L}_{SM}(S, S')$, we believe that providing semantic pixel-wise guidance to the generator in terms of the classes each region in the image should depict, while not limiting its content to match an arbitrary natural image, as is the case in feature-matching losses, accelerates the generator's learning without inhibiting its output diversity, which in turn yields better FID scores. Likewise, for the Segment-Fidelity loss, $\frac{1}{n}\sum \mathcal{L}_{SF}(D(s_i, x_i))$, promoting fidelity of individual segments x_i, rather than just of the whole picture X, naturally enhances learning, especially for rich and highly-structured scenes, as we observe for the COCO dataset.

D Model Ablations & Iterative Generative

To validate the efficacy of our approach and better assess the relative contribution of each design choice, we perform ablation and variation studies over the planning and execution stages. We begin by exploring the impact of varying the number of recurrent generation steps for planning the scene layout, and further compare it with a single-pass approach that generates the layout in a non-compositonal fashion – as a one image like standard GANs, rather then as a collection of interacting segments. We note that each generation step creates a random number of segments, sampled from a trainable normal distribution, and therefore reducing the number of steps does not limit the maximum number of segments the model can create overall. Adding more recurrent steps can instead enhance the

model's capability to capture conditional dependencies across segments, while keeping the planning process shorter can naturally increase the computational efficiency.

Compositional Layout Synthesis. As figure 14 shows, compositional layout generation performs substantially better than the standard single-pass approach (denoted by 0) across all datasets. Intuitively, we believe that the efficiency gains arise from the ability of the recurrent approach to decompose the combinatorial space of possible scene layouts into several smaller tasks, such that each step focuses on a few segments only, rather than modeling the whole scene at once. This could be especially useful for highly-structured scenes with multiple objects and dependencies.

Layout Synthesis Length. We can see how the optimal number of synthesis steps vary between different datasets: while CelebA layouts seem simple enough for the model to comprehend and synthesize in just one step, CLEVR and LSUN-Bedrooms benefit from 2 steps, and the richer Cityscapes and COCO see performance improvement in even 3 layout generation steps. The most effective lengths seem to indicate the compositionality degree of each dataset: In CLEVR, objects are mostly independent of each other (holding only weak relations of not mutually occupying the same space), and so their segments can be produced mostly in parallel, over less recurrent steps. Meanwhile, more intricate scenes, as in COCO and Cityscapes, benefit from a longer sequential generation that can explicitly capture conditional dependencies among objects (e.g. generating a cup only after creating the table it is placed on), demonstrating the strength of recurrent synthesis.

Layout Refinement. In the execution stage, which transforms layouts into output images, we explore the contribution of the layout refinement mechanism (section 3.3, main paper). It introduces a sigmoidal gate $\sigma(g(S, X, W))$ to support local layout adjustments, meant to increase the model's flexibility and expressivity during the translation. We study ablations over CLEVR, either not applying this refinement or using limited gating versions, constraining the inputs to be the latents W, layout S, or image X only. We see that compared to the default model's FID score of *4.70*, using weaker refinements lead to deterioration of 0.72, 0.78 and 0.85 when inputting the latents, layout or image respectively, and a larger reduction of 1.45 points, when ablating the gating mechanism completely. These results provide evidence for the benefit of using the gating mechanism to refine the scene's layout during the execution stage.

E Paired vs. Unpaired Training

We train GANformer2 over two sets: of images $\{X_i\}$ and layouts $\{S_i\}$, used during the planning and execution stages respectively. The training sets can either be paired: listing the alignment between images and layouts, namely $\{(S_i, X_i)\}$, or unpaired: $\{X_i\}, \{S_i\}$, and our training scheme accommodates both options: For the paired case, we first train the planning stage over layouts $\{S_i\}$, and the execution stage conditionally over $\{(S_i, X_i)\}$, such that it learns to translate *ground-truth layouts* into output images. Then, to make both stages work in tandem, we fine-tune them together, where this time the execution stage translates *generated layouts* $\{S_i^g\}$ into output images $\{X_i^g\}$.

For the unpaired case, we can either (1) train the planning stage over the layouts $\{S_i\}$, and then fine-tune it together with the execution stage, or (2) jointly train the two stages from scratch (we call this scheme *Parallel*). Since the Segment-Fidelity loss assumes access to paired layout-image samples, both fake and real, we do not use it in the unpaired and parallel cases, and instead use the Edge-Matching loss, described in C. For both the Edge-Matching and Semantic-Matching losses, we adjust them for both the generator and the discriminator to be computed over generated pairs $\{(S_i^g, X_i^g)\}$, which are available even when the training data is unpaired. Table 4 compares the model performance between the paired, unpaired and parallel settings for different datasets. We note that the model manages to perform well even in the unpaired and parallel settings, but achieves strongest results in the paired case.

F Implementation & Training Details

We implement all the unconditional methods within the shared GANformer codebase [33], to ensures they are tested under comparable conditions in terms of training details, model sizes, and optimization scheme. For the conditional models, we use the authors' official implementations, likewise implemented as extensions to the Pix2PixHD repository for conditional generative modeling. All approaches have been trained with images of 256×256 resolution and data augmentation as detailed

in section H. See table 5 for the particular settings and hyperparameters of each model. The overall latent dimension is chosen based on performance among $\{128, 256, 512\}$. The R1 regularization factor γ is likewise chosen based on performance and training stability among $\{1, 10, 20, 40, 80, 100\}$.

In terms of the loss function, optimization and training configuration, we adopt the settings and techniques used in StyleGAN2 and GANformer [33, 42], including in particular style mixing, Xavier Initialization, stochastic variation, exponential moving average for weights, and a non-saturating logistic loss with lazy R1 regularization. We use Adam optimizer with batch size of 32 (4 times 8 using gradient accumulation), equalized learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_1 = 0.999$ as well as leaky ReLU activations with $\alpha = 0.2$, bilinear filtering in all up/downsampling layers and minibatch standard deviation layer at the end of the discriminator. The mapping layer of the generator consists of 8 layers, and ResNet connections are used throughout the model, for the mapping network, synthesis network and discriminator. All models have been trained for the same number of training steps, roughly spanning 10 days on 1 NVIDIA V100 GPU per model.

G Baselines & Prior Approaches

We compare GANformer2 to both unconditional and conditional generative models. First, we inspect unconditional methods which synthesize images from scratch, including in particular: (1) a baseline GAN [23], (2) the StyleGAN2 [42] model, (3) SAGAN [81] which utilizes self-attention across spatial regions, (4) k-GAN [69] that blends together k generated images through alpha-composition, (5) VQGAN [21], a visual autoregessive autoencoder, (6) SBGAN [5], a non-compositional two-stage approach, and (7) the original GANformer. We then proceed to compare our execution stage to popular conditional semantic generation models, including the aforementioned SBGAN and also: (8) Pix2PixHD [72] which uses a U-Net [62] to translate source to target images, (9) BicycleGAN [86] that promotes cycle consistency among domains, and (10) SPADE [60], which performs spatial modualtion using a fixed set of trainable semantic category vectors. For disentanglement and controllability experiments, we compare the GANformer2 to a baseline GAN, StyleGAN2 and GANformer, as well as: (11) MONet, and (12) Iodine, two sequential variational autoencoders.

H Data Preparations

We train all models on images of 256×256 resolution, padded as necessary. See dataset statistics in table 6. The images in the Cityscapes and FFHQ datasets are mirror-augmented, while the images in the COCO dataset are both mirror-augmented and also randomly cropped, to increase the effective training set size. We assume access to a training data of image and panoptic segmentations [43], indicating the segment unique identity and its semantic class. Contrary to prior conditional works which rely on costly hand-annotated ground-truth segmentations, and to demonstrate the model robustness, we instead intentionally explore training on auto-predicted segmentations (either produced in an unsupervised manner [39] for CLEVR [37] or by pre-trained segmentor [74] otherwise).

For the COCO dataset [48], we note that it introduces challenges along two dimensions, being both (1) **highly-structured**, with each scene populated by many objects that hold intricate relations and dependencies, but also (2) visually and semantically **diverse**, consisting of varied images from a wide range of domains. In order to isolate the former challenge of modeling compositional scenes from the latter important but different challenge of covering a diverse image distribution, we study training on a topical partition of COCO, named COCO_p, that groups images into 7 semantically-related splits, listed in table 7. To partition the dataset, we cluster t-SNE processed ResNet activations of the COCO images into 31 subsets, which are then semantically grouped into the 7 splits. We train the models on each split separately, and report the mean scores. As expected, and also empirically suggested by table 1 (main paper), the partition leads to improved visual quality across all models – both baselines and new ones, likely due to the more uniform resulting training distributions.