Spatial Steerability of GANs via Self-Supervision from Discriminator

Jianyuan Wang, Lalit Bhagat, Ceyuan Yang, Yinghao Xu, Yujun Shen, Hongdong Li, and Bolei Zhou

***Abstract*—Generative models make huge progress to the photorealistic image synthesis in recent years. To enable humans to steer the image generation process and customize the output, many works explore the interpretable dimensions of the latent space in GANs. Existing methods edit the attributes of the output image such as orientation or color scheme by varying the latent code along certain directions. However, these methods usually require additional human annotations for each pretrained model, and they mostly focus on editing global attributes. In this work, we propose a self-supervised approach to improve the spatial steerability of GANs without searching for steerable directions in the latent space or requiring extra annotations. Specifically, we design randomly sampled Gaussian heatmaps to be encoded into the intermediate layers of generative models as spatial inductive bias. Along with training the GAN model from scratch, these heatmaps are aligned with the emerging attention of the GAN's discriminator in a self-supervised learning manner. During inference, users can interact with the spatial heatmaps in an intuitive manner, enabling them to edit the output image by adjusting the scene layout, moving, or removing objects. Moreover, we incorporate DragGAN into our framework, which facilitates fine-grained manipulation within a reasonable time and supports a coarse-to-fine editing process. Extensive experiments show that the proposed method not only enables spatial editing over human faces, animal faces, outdoor scenes, and complicated multi-object indoor scenes but also brings improvement in synthesis quality.**

***Index Terms*—Interpretability, spatial editing, generative models.**

I. Introduction

GENERATIVE Adversarial Network (GAN) has made huge progress to high-quality image synthesis [1], [2], [3], [4], [5]. GAN is formulated as a two-player game between a generator (G) and a discriminator (D) [1], where G maps a random distribution to real-world observation, and D competes with G by distinguishing the generated images from the real ones. It is found that the latent space of G contains disentangled subspaces, which align with various image attributes, e.g., the age of human faces [6], the layout of indoor scenes [7], and the pose of vehicles [8]. Researchers utilize such properties to study the knowledge learned by GANs and facilitate interactive editing over the output image.

However, the existing methods [6], [8], [9], [10], [11], [12] mostly require additional information like extra annotations or human selection. For a target attribute and a given generator, they search for an associated direction in the high-dimensional latent space and then change the image attribute via varying the latent code along the found direction. A typical approach is to first sample numerous images from the latent space, label them regarding the target attribute, and then learn to find the tangent direction, which could be expensive, unstable, and sometimes inapplicable. Some recent works [8], [10], [12] identify the essential directions in the latent space via the techniques like Principal Component Analysis (PCA). Unfortunately, these methods cannot guarantee which attributes will be found, and human still needs to distinguish which attribute each direction corresponds to and select the meaningful ones. Moreover, the spatial steerability of generative models, such as moving an object or changing the local appearance of an object in the output image, is much less explored.

In this work, we propose a novel self-supervision approach called *SpatialGAN* to achieve spatial steerability of GANs without searching for steerable directions in the latent space. It allows human users to perform various spatial manipulations in the image generation, such as moving an object and removing an object in a scene, changing the style of a region, or globally controlling the structure/layout of an image. Some examples are shown in Fig. 1. Previous work shows that the class specific attention maps emerge in image classification networks [13]. We reveal that the discriminator of GAN, as a bi-classifier for adversarial training, also has emerging attention highlighting the informative region of the synthesized image. Therefore, we incorporate a design of spatial heatmaps as inductive bias in the generator, and then learn to align them with the attention maps from the discriminator in a self-supervised learning manner. Specifically, we randomly sample heatmaps and encode them into the intermediate layers of G to guide its spatial focus. To ensure the encoded heatmaps focusing on the meaningful regions of the synthesized image, we regularize the generator's heatmap to be aligned with the discriminator's attention map on the synthesized image. In

Manuscript received 13 January 2024; revised 6 May 2024; accepted 23 June 2024. Date of publication 3 July 2024; date of current version 5 November 2024. This work was supported in part by the Amazon Research Award and NSF under Grant IIS-2339769. Recommended for acceptance by T. Hassner. *(Jianyuan Wang and Lalit Bhagat contributed equally to this work.) (Corresponding author: Bolei Zhou.)*

Jianyuan Wang is with the University of Oxford, OX1 2JD Oxford, U.K (e-mail: jianyuan@robots.ox.ac.uk).

Lalit Bhagat and Bolei Zhou are with the Computer Science Department, University of California, Los Angeles, CA 90095 USA (e-mail: bolei@cs.ucla.edu).

Ceyuan Yang and Yinghao Xu are with the Department of Information Engineering, Chinese University of Hong Kong, Hong Kong.

Yujun Shen is with the Ant Group, Hangzhou 310000, China.

Hongdong Li is with the College of Engineering and Computer Science, The Australian National University, Canberra, ACT 2601, Australia.

Code, models, and demo video are available at https://genforce.github.io/SpatialGAN/

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3422820, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3422820

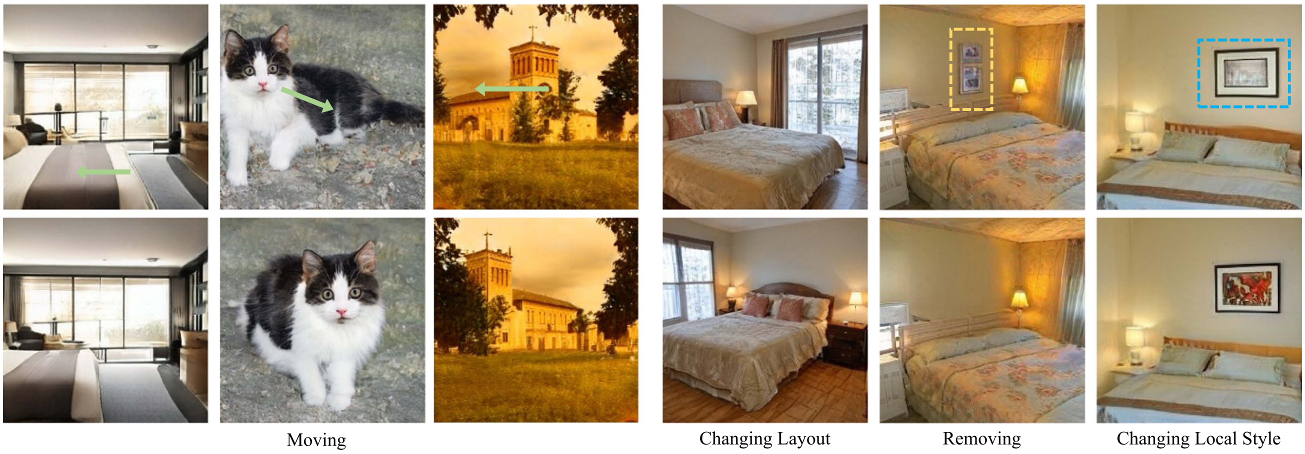


Fig. 1. Illustration of Spatial Manipulations. Our method enables various spatial manipulations for image generation, like moving a bed, a cat, or a building (green arrow), controlling the image layout, removing a drawing (yellow box), and changing the local appearance (blue box).

other words, we utilize the attention map emerging from the discriminator to guide the heatmap in the generator. The whole process follows a self-supervised learning manner and does not involve extra annotation or statistical information. It trains the generator to synthesize the image based on the input heatmaps, and improves the spatial steerability of the model, i.e., we could edit the heatmap to spatially control the output synthesis during inference. The preliminary result of such spatial steerability was shown at our conference version [14], where the main focus was to improve synthesis quality by incorporating heatmaps as an inductive bias, and the steerability is a byproduct. We initially demonstrated this ability in single-object scenes, which sharply focus on one primary element. This could range from artifacts, human faces, animals, to more contextual settings like a lone car on a street or an isolated historical building. Such scenes, due to their focus on a singular primary element, are relatively straightforward to analyze and interpret. Here, moving to indoor scenes, we address the complexities of multi-object indoor scenes. For instance, a living room scene might include a sofa, a coffee table, and artwork on the walls. Each element contributes to the scene's overall composition, requiring careful placement to ensure a realistic representation. These scenes pose a notable challenge due to their multiple points of focus; a single point on the heatmap is insufficient for capturing the scene's full dynamics. To accurately generate such scenes, it is imperative to understand not only each object individually but also how they collectively interact within the space. Compared to the conference version, this journal paper has achieved notable advancements in spatial steerability. Specifically, (i) to enable the spatial steerability in complex indoor scenes with multiple objects, we have developed a new heatmap construction strategy, encoding method, and self-supervision training objective; (ii) our enhanced method facilitates more sophisticated spatial manipulations, such as removing objects and changing the style of a local region, as illustrated in Fig. 1; (iii) we have also significantly improved the synthesis quality of single-object scenes using a refined heatmap processing strategy; (iv) we integrate the recent progress in point-based manipulation (e.g., DragGAN [15]) into our method. This integration combines the strengths of both approaches to achieve high-quality, fine-grained manipulation in a reasonable time. Our study demonstrates not only the unique merits of our framework but also its complementary functionality with DragGAN, highlighting the versatility and effectiveness of our approach; (v) we develop a new user interface to illustrate our manipulation ability; and (vi) we present an expanded set of results and provide a comprehensive analysis that highlights significant advancements in both manipulation capability and synthesis quality.

II. Related Work

A. Generative Adversarial Networks

GANs [1] achieve great success in photorealistic image generation. It aims to learn the target distribution via a minimax two-player game of generator and discriminator. The generator usually takes in a random latent code and produces a synthesis image. Researchers have developed numerous techniques to improve the synthesis quality of GANs, through a Laplacian pyramid framework [16], an all-convolutional deep neural network [2], progressive training [17], spectral normalization [18], [19], and large-sacle GAN training [3], [20]. Some methods also incorporate additional information into discriminator or generator, such as pixel-wise representation [21], 3D pose [22], or neighboring instances [23]. In recent years, the style-based architecture StyleGAN [4] and StyleGAN2 [5] have become the state-of-the-art method for image synthesis by separating high-level attributes. Diffusion models [24], [25], [26], [27] advance image synthesis rapid in recent two years. They are a class of probabilistic generative models that iteratively apply noise to the data and then learn to reverse this process to generate new samples that resemble the original data. Prior works attempt to manipulate the images generated by diffusion models. However, most of these studies have concentrated on text-based editing [28], [29], [30]. Despite the high image quality achieved by diffusion models, it is difficult to have real-time spatial editing with those models. We hope our GAN-based methods can inspire new works on diffusion-based interactive editing.

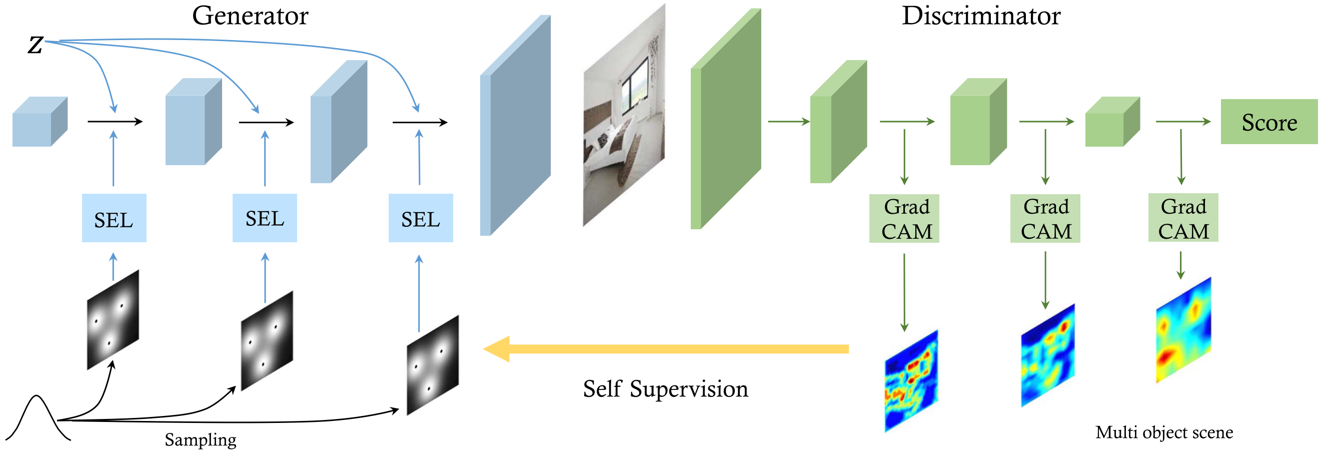


Fig. 2. Illustration of SpatialGAN. We conduct spatial encoding in G and align its spatial awareness with D attention maps. Specifically, we randomly sample spatial heatmaps and encode them into G via the spatial encoding layer (SEL). To implement the alignment during training, we calculate D attention maps over the generated samples via GradCAM.

B. GAN Manipulation

To understand the generation process of GANs and support human customization of the output image, researchers have been trying to control the output synthesis. A popular way is to leverage the rich semantic information in the latent space of GAN. They identify steerable properties as some directions in the latent space and vary the latent code accordingly. For a certain attribute, they search for a certain direction in the latent space, and then alter the target attribute via moving the latent code z along the searched direction [6], [9], [10], [11], [12]. However, for each pre-trained GAN model (i.e., pre-trained latent space), these methods require to annotate a collection of the generated samples to train linear classifiers in the latent space [6], [7], [9] or utilize sample statistics [31], [32]. These requirements are expensive and it can only accommodate limited number of attributes. Some recent works [8], [10], [12], [33] search for steerable directions using techniques like Principal Component Analysis (PCA) in an unsupervised manner. Unfortunately, it does not guarantee the attributes of found directions. For example, in order to achieve spatial control of the image generation, the user has to manually check the effect of found directions, while the one corresponding to the desired spatial manipulation may not exist. Instead, our proposed method brings spatial steerability of GANs without searching for steerable directions in the latent space, which avoids the requirements of extra annotation or human selection. A concurrent work [34] achieves similar capabilities by learning a category-specific middle-level representation. However, this method necessitates a significantly larger amount of training data and computational resources compared to our approach. A recent innovation, DragGAN [15], introduces a point-based manipulation technique that allows for fine editing by performing optimization on the latent code during inference. While this method does provide granular control, it is constrained by the time-intensive nature of the optimization process, particularly when broader, more coarse adjustments are required. In contrast, our method facilitates the movement of objects seamlessly and intuitively without resorting to any form of optimization, thereby offering a more efficient alternative for real-time spatial editing in generative models

III. Method

In this section, we first analyze the spatial attention of the GAN's discriminator in Section III-A, which serves as the guidance for implementing the pseudo attention mechanism in the generator. We then introduce the hierarchical heatmap sampling strategy and the heatmap encoding methods in Section III-B. Compared to our conference version, we find that a heatmap with too fine-grained scale may not benefit the image synthesis, and hence proposes a coarse processing of heatmap to improve the synthesis quality. Furthermore, to address the complexities of indoor scenes with multiple objects, we update the method of heatmap sampling and heatmap encoding. In Section III-C, we discuss utilizing the emerging attention map from the discriminator as a self-supervision signal for image synthesis which paves the way for spatial editing. The overall framework is illustrated in Fig. 2, primarily involving two steps: the explicit encoding of spatial inductive bias into G and using the emerging attention map from D to supervise G. Different from the conference version, we also introduce a new self-supervision objective tailored for intricate indoor scenes, enabling advanced spatial manipulations. Lastly, in Section III-D, we extend our discussion to the integration of our SpatialGAN with DragGAN [15], a point-based image manipulation technique. This synergy leverages the strengths of both approaches, facilitating more efficient and flexible manipulations in generative models.

A. Spatial Attention of Discriminator

We first investigate the behavior of D in the spatial domain, because D is designed to differentiate between the real distribution and the distribution generated by G, i.e., D acts as both an adversary and a teacher for G in this two-player game.

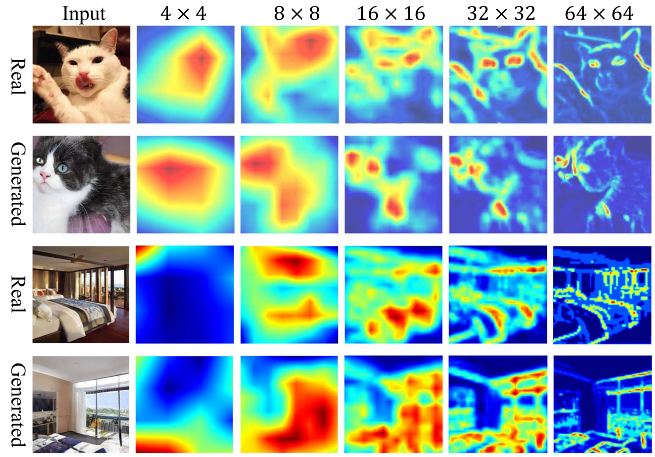


Fig. 3. Spatial visual attention at the intermediate layers of the discriminator, visualized by GradCAM. A bright color indicates a strong contribution to the final score. ‘64 × 64’ indicates being upsampled from a 64 × 64 feature map. The samples are the real images and the images generated by StyleGAN2 [5].

Prior work on network interpretability, like CAM [13], has found that a classification network tends to focus on some discriminative spatial regions to categorize a given image. However, the discriminator in GANs is trained with the relatively weak supervision, i.e., only having real or fake labels. Whether it can learn the attentive property from such a bi-classification task remains unknown. To look under the hood, we apply GradCAM [35] as an interpretability tool on the well-trained discriminator of StyleGAN2 [5].

Specifically, for a certain layer and a certain class, GradCAM calculates the importance weight of each neuron by average-pooling the gradients back-propagated from the final classification score, over the width and height. It then computes the attention map as a weighted combination of the importance weight and the forward activation maps, followed by a ReLU [36] activation. The attention map has the same spatial shape as the corresponding feature map. In this work, we report the GradCAM attention maps all using gradients computed via maximizing the output of D. It reflects the spatial preference of D in making a ‘real’ decision. In practice we find the attention maps are almost the same if instead minimizing the output of D, which indicates the areas that largely contribute to the decision are the same for a discriminator, no matter positively or negatively. The region with higher response within the attention map contributes more to the decision.

Fig. 3 visualizes some GradCAM results under multiple feature resolutions. Our examination of discriminators trained on datasets such as LSUN Cat and LSUN Bedroom reveals insightful aspects of their spatial behavior. We have following observations: (1) D learns its own visual attention on both real and generated images. It suggests that D makes the real/fake decision by paying more attention to some particular regions. (2) The visual attention emerging from D shows a hierarchical property. In the shallow layers (like 64 × 64 and 32 × 32 resolutions), D is attentive to local structures such as edge lines in the image. As the layer goes deeper, D progressively concentrates on the overall location of discriminative contents, e.g., the face of a cat. (3) The hierarchical attention maps have fewer ‘local peaks’ at more abstract feature layers with a lower resolution. For example, there is only one peak in the 4×4 attention maps.

Building on this understanding, the discriminator of GANs has its own visual attention when determining a real or fake image. However, when learning to transform a latent vector into a realistic image, the generator receives no explicit clue about which regions to focus on. Specifically, for a particular synthesis, G has to decode all the needed information from the input latent code. Furthermore, G has no idea about the spatial preference of D on making the real/fake decisions. In [14], we introduced the concept of deploying pseudo attention in G to enhance its synthesis capability and spatial steerability. In subsequent sections, we further refine our methods for both indoor and non-indoor scenes, leading to improvements in synthesis quality and spatial steerability.

B. Encoding Spatial Heatmap in Generator

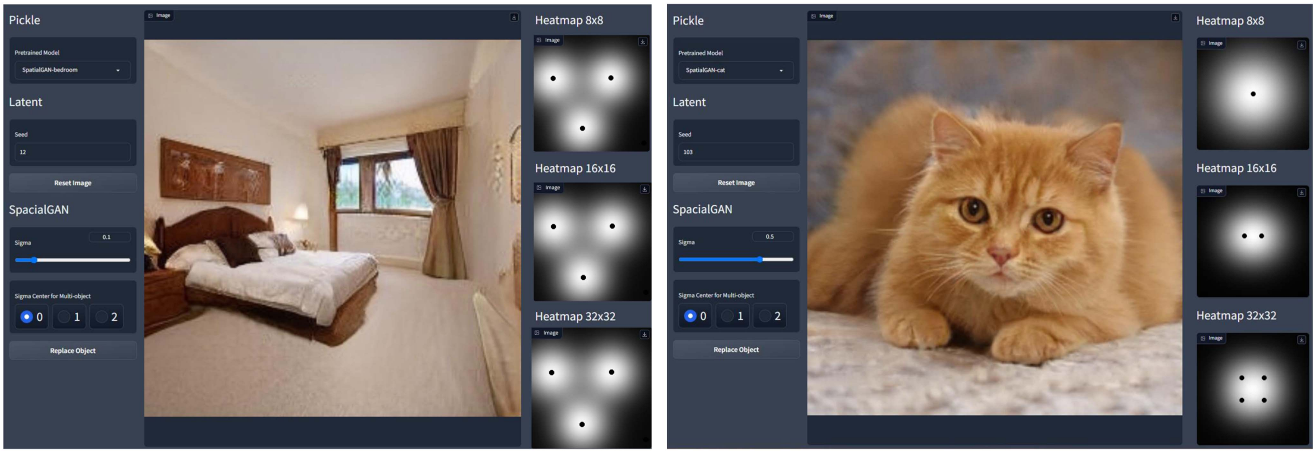
Hierarchical Heatmap Sampling: Inspired by the observation in Section III-A, we propose a hierarchical heatmap sampling algorithm for single-object scenes. The heatmap is responsible for teaching G which regions to pay more attention to. Each heatmap is abstracted as a combination of several sub-regions and a background. We formulate each sub-region as a 2D map, H_i, which is sampled subject to a Gaussian distribution,

$$H_i \sim \mathcal{N}(\mathbf{c_i}, \mathbf{cov}), \tag{1}$$

where $\mathbf{c_i}$ and $\mathbf{cov}$ denote the mean and the covariance. According to the definition of 2D Gaussian distribution, $\mathbf{c_i}$ just represents the coordinates of the region center. The final heatmap can be written as the sum of all sub-maps, $H = \sum_{i=1}^{n} H_i$, where n denotes the total number of local regions for G to focus on.

As pointed out in the prior works [4], [7], the generator in GANs learns image synthesis in a coarse-to-fine manner, where the early layers provide a rough template and the latter layers refine the details. To match such a mechanism and be consistent with the hierarchical spatial attention of discriminators, we design a hierarchical heatmap sampling algorithm. Concretely, we first sample a spatial heatmap with (1) for the most abstract level (i.e., with the lowest resolution), and derive the heatmaps for other resolutions based on the initial one. The number of centers, n, and the covariance, $\mathbf{cov}$, adapt according to the feature resolution.

Multi-Object Heatmap Sampling: Given that indoor scenes often have several independent objects, a straightforward idea is to model each object by a hierarchical set of Gaussian heatmaps as discussed above and encode such a complicated inductive bias into each layer of G. However, with hierarchical sampling, modeling multiple objects in an image would lead to too many sub-regions at high-resolution, which would interact and possibly conflict with each other. In practice, we find such a conflict would let the model confused and hinder the optimization during the training process. Especially, a small modification of a heatmap sub-region would spread to other sub-regions and result in a dramatic change over the output synthesis, which troubles the



Pretrained Model: SpatialGAN-bedroom
Seed: 12
Sigma: 0.1
Sigma Center for Multi-object: ● 0 ○ 1 ○ 2

Pretrained Model: SpatialGAN-cat
Seed: 103
Sigma: 0.5
Sigma Center for Multi-object: ● 0 ○ 1 ○ 2

Fig. 4. User interface for interactive editing. Users can drag the Gaussian centers to alter heatmaps and synthesize an image corresponding to the new heatmap. In addition to that users can also change the sigma value of the heatmap center and the local style corresponding to that center. The Bedroom model (left) uses the same heatmap for all the resolutions while the Cat model (right) uses hierarchical heatmaps. *Readers are suggested to view the demo video of interactive editing on our project page.*

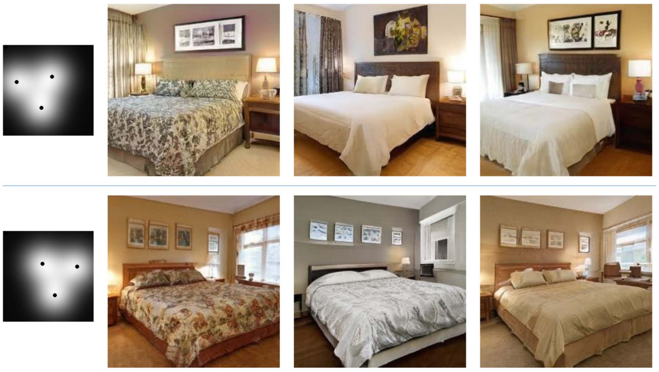


Fig. 5. Controlling the room layout by using the same spatial heatmap. For each row, we adopt different latent codes but use the same spatial heatmap. The images in the same row show a similar spatial layout, while their appearances are different. The appearances include colour, texture, lighting, and so on.

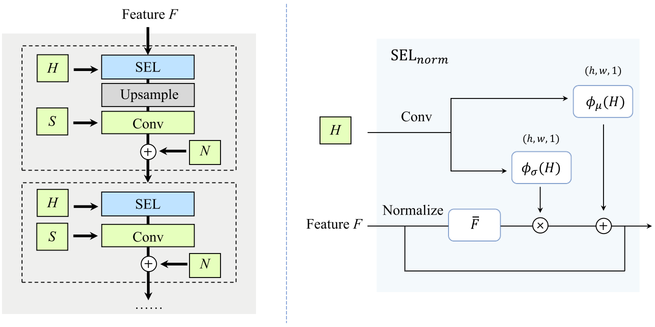


Fig. 6. Spatial Encoding, where the left shows how the spatial encoding layer (SEL) works over StyleGAN2 at each resolution, and the right describes the internal of the SEL$_{norm}$. The symbol ‘S’ represents the style in StyleGAN2, ‘N’ is the noise, and ‘H’ indicates the spatial heatmap. Derived from [38], we incorporate the spatial heatmaps into G via normalization and denormalization.

spatial editing. Therefore, we abandon the hierarchical sampling for the multi-object setting. Instead, we model multiple objects by different Gaussian distributions at the coarsest resolution, and use this heatmap as the inductive bias for *all* the synthesis layers of G, which can be noticed in our illustration for the user interface Fig. 4. The heatmap in fact points to the rough location of various objects, which can be viewed as an abstracted scene layout. In this way, we can control the input heatmap to keep the scene layout, as shown in Fig. 5.

Heatmap Encoding: We incorporate the spatial heatmaps into the intermediate features of G to raise its spatial controllability (no matter for single-object or multi-object scenes). It usually can be conducted in two ways, via feature concatenation or feature normalization [37], [38]. We use a spatial encoding layer (SEL), whose two variants are denoted as SEL$_{concat}$ and SEL$_{norm}$. Specifically, the variant SEL$_{concat}$ processes the concatenation of heatmap and feature via a convolution layer, and outputs new feature for the next layer. Inspired by SPADE [38], the variant SEL$_{norm}$ integrates the hierarchical heatmaps into the per-layer feature maps of G with normalization and denormalization operations, as

$$SEL_{norm}(F, H) = \phi_\sigma(H)\,\frac{F - \mu(F)}{\sigma(F)} + \phi_\mu(H), \qquad (2)$$

where F denotes an intermediate feature map produced by G, which is with the same resolution as H. $\mu(\cdot)$ and $\sigma(\cdot)$ respectively stand for the functions of computing channel-wise mean and standard deviation. $\phi_\mu(\cdot)$ and $\phi_\sigma(\cdot)$ are two learnable functions, whose outputs are point-wise and with a shape of $(h, w, 1)$. Besides, as shown in Fig. 6, we use a residual connection to stabilize the intermediate features. If not particularly specified, we adopt the variant SEL$_{norm}$ since it shows a slightly better performance.

It is worth noting, although we learn the SEL$_{norm}$ architecture from SPADE [38], these two methods are clearly different since SPADE targets at synthesizing images based on a given semantic segmentation mask, whose training requires paired ground-truth data, while our model is trained with completely unlabeled data.

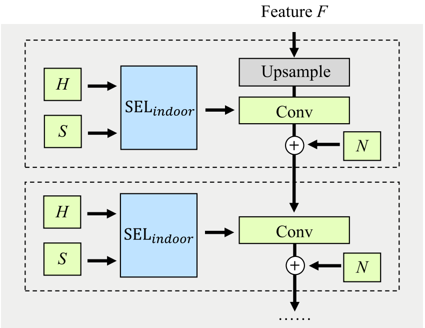


Fig. 7. Illustration of SEL$_{indoor}$, where we integrate the heatmap H and style S to avoid modifying the intermediate features twice like in Fig. 6.

Meanwhile, SEL$_{norm}$ is just a replaceable component of our approach.

Coarse Heatmap Processing: To better understand the role of heatmap encoding, we study the mapping from the input heatmap to the synthesized image. In our conference version [14], the heatmap H is sampled as the same size as the target image, processed by function $\phi_\mu(\cdot)$ and $\phi_\sigma(\cdot)$, and then down-sampled to match the resolution of intermediate features. It was designed to keep the heatmap continuous and consistent with the final output. However, here we find this is not the optimal choice. Instead, if we first down-sample the fine-grained heatmap to the intermediate resolution and then process it via a convolution layer, the synthesis quality would be improved and the computation cost is decreased. In other words, we process a 'coarse' heatmap. We speculate two reasons for this observation: (1) A spatially continuous inductive bias may not be necessary. The heatmap (sampled by a 2D Gaussian distribution) is continuous while the real attention could be discrete, because the real-world distribution of objects may be discrete. The network possibly only needs a rough spatial structure instead of a fine-grained guidance. (2) For heatmap processing, a global understanding matters. With the same convolutional layer, down-sampling the input in fact enlarges the receptive field by times, providing a global view. We verify these two hypotheses and provide detailed analysis in Section IV-C.

Multi-Object Heatmap Encoding: To further improve the spatial steerability of the model, we review the heatmap encoding process. It is worth noting that the intermediate features of G are successively modified by heatmap H and style S, as illustrated by the left of Fig. 6. This architecture was designed following the philosophy that for an image, the spatial structure should be determined first and then comes to other components like appearance. However, the feature change brought by styles may weaken the effect of spatial heatmaps. We empirically find this phenomenon is obvious in indoor scenes, where we suppose the multi-object setting increases the difficulty of utilizing the encoded spatial heatmaps. Therefore, we propose a variant SEL$_{indoor}$, as shown in Fig. 7. It integrates the spatial heatmaps H and styles S together and encodes their combination to G, which only modifies the generator features once. Since H is a heatmap within the range (0,1), the integration can be effectively conducted via an element-wise product, that SEL$_{indoor}(S, H) = S * H$, from $(1, 1, c)$ and $(h, w, 1)$ to a shape of (h, w, c).

Moreover, since now one 2D Gaussian (sub-heatmap H_i) represents a separate object, we could employ a unique appearance code for each sub-heatmap, to reduce their interaction. We utilize n style codes S_i to depict the indoor objects and one S_{bg} to control the background appearance:

$$SEL_{indoor}(S, H) = \sum_{i=1}^{n} S_i * H_i + S_{bg}. \quad (3)$$

The value of H_i would decay to zero in its edge areas and hence the corresponding appearance code S_i would gradually lose the effect. All the style codes S_i and S_{bg} share the same feature dimension. For simplicity, we encode SEL$_{indoor}(S, H)$ into G as same as encoding S, just except being spatial-aware. This new heatmap encoding method enhances the spatial steerability of the model, supporting the editing operations like removing an object or changing the style of a region, as shown in Fig. 1.

C. Spatial Alignment via Self Supervision

Encoding heatmaps into G can explicitly raise its spatial steerability, but the heatmaps fed into G are completely arbitrary. Without further guidance, how G is supposed to utilize the heatmaps is ambiguous, which influences the spatial steerability of the model. For example, G has no idea about 'whether to pay *more* or *less* attention to the highlighted regions in the heatmap'. Instead, D learns its own visual attention based on the semantically meaningful image contents. To make the best usage of the introduced spatial inductive bias, we propose to involve the spatial attention of D as a self-supervision signal, which does not require any extra annotation.

Specifically, at each optimization step of G, we use D to process the synthesized image and generate a corresponding visual attention map with the help of GradCAM. Besides competing with D, G is further trained to minimize the distance between the attention map induced from D and the input heatmap H. The loss function can be written as

$$\mathcal{L}_{align} = ||\ \text{GradCAM}_D[G(H, \mathbf{z})] -\ H\ ||_1. \quad (4)$$

We truncate the $\mathcal{L}_{align}$ values if smaller than a constant τ, since the sampled heatmaps are not expected to perfectly match the real attention maps shaped by semantics. The threshold τ is set as 0.25 for all the experiments. Note that D is not updated in the process above and only used as a self supervision signal to train G. Such a regularization loss aligns the spatial awareness of G with the spatial attention of D, narrowing the information gap between them. It employs D to tell G how to leverage the encoded inductive bias and hence raises the spatial controllability of G. As an adversary, D can also be a good teacher.

Multi-Object Self-Supervision Objective: For conciseness, the sub-heatmaps for different indoor objects are treated equally in the sampling stage. However, some objects may take over much attention in real-world situations, e.g., a very large bed. Consequently, we relax the training objective for spatial awareness alignment. Still leveraging discriminator attention as the self-supervision signal, we compare it to the heatmap H and each sub-heatmap H_i, with the minimum distance as the optimization term:

$$\mathcal{L}_{align\_indoor} = \min_{n+1} || \text{GradCAM}_D[G(H, \mathbf{z})] - \widehat{H} ||_1, \quad (5)$$

where $\widehat{H} \in \{H_1, H_2, \ldots, H_n, H\}$. The value truncation and discriminator freezing in $\mathcal{L}_{align}$ are still used here.

D. Synergy Between DragGAN and SpatialGAN

In this section, we detail the integration of our SpatialGAN method with the latest point-based image manipulation technique, DragGAN [15]. DragGAN demonstrates proficiency in relocating specific parts of an image from one point to another. Specifically, DragGAN assumes the feature space of a GAN model is discriminative enough to support precise point tracking and motion supervision, which is consistent with the observation in our paper. Therefore, the method optimizes the GAN latent code $\mathbf{w}$ to encourage some given handle points (i.e., starting points) to move towards their target destinations. To ensure stable movement and precise manipulation, this process is typically repeated $30 - 200$ times.

Direct application of DragGAN's techniques to SpatialGAN yielded inferior manipulation results, mainly because of the architectural differences between SpatialGAN and StyleGAN2, especially the inclusion of heatmaps in our design. We discovered that effective manipulation in SpatialGAN requires point-based optimization of both the latent space and the heatmaps. We have explored two distinct approaches for heatmap optimization: one by directly optimizing the pixels of heatmaps H, and the other by focusing on the optimization of the heatmap centers $\mathbf{c}$. Preliminary results indicate that optimizing the centers of heatmaps yields more effective manipulation. By simultaneously optimizing heatmap centers and latent codes, we not only augment DragGAN's capabilities to support SpatialGAN but also drastically reduces the optimization steps for image manipulation, requiring just 15–70 steps

However, although DragGAN shows an impressive ability for granular manipulation, its iterative optimization during inference is time-consuming. Unlike DragGAN, our SpatialGAN does not require any optimization process during inference. To leverage the strengths of both methodologies, we propose a two-step manipulation process: initially, our method is employed for coarse movement, followed by the application of DragGAN to refine the movement. Given the handle points and the target destinations, SpatialGAN will first manipulate the image by altering the heatmaps to enable the corresponding movement, which can skip a lot of iterative optimization steps required by DragGAN. Then, we adopt DragGAN over the manipulated image to ensure the handle points precisely match the target destinations. Specifically, the coarse movement is guided by the condition: if the starting points are within a specified radius r of the heatmap centers c_i, the centers are moved to new positions c_i' according to the formula:

$$c_i' = c_i + \alpha \cdot (t - p), \quad (6)$$

where p is the start point, t is the target point, and α is a factor determining the extent of movement towards the target. In cases where the starting points are outside the radius, the heatmap centers remain unchanged. Following this, point-based optimization takes over for fine manipulation, where the heatmap centers and latent code are jointly optimized. The process, described here for a single handle point, can be easily extended to multiple points. This hybrid methodology significantly reduces the optimization steps required for precise alignment, harnessing the strengths of both SpatialGAN's efficient coarse adjustment and DragGAN's fine manipulation. As a result, the optimization steps are further condensed to approximately 10–20 iterations, greatly diminishing the time required for manipulation.

The successful integration of SpatialGAN with DragGAN demonstrates the adaptability of our method, illustrating its compatibility with various manipulation techniques. This integration highlights the potential of SpatialGAN as an advanced toolkit for the community engaged in spatial editing within generative models. The comprehensive methodology of our approach, is systematically illustrated in Fig. 8.

IV. Experiments

We evaluate the proposed SpatialGAN on multiple benchmarks, covering faces, indoor scenes, and outdoor scenes. Section IV-A provides the implementation details. The main comparison and experimental results are presented in Section IV-B, that our SpatialGAN can support multiple types of editing, enhance the spatial controllability of G, and improve the synthesis quality. Section IV-C includes comprehensive ablation studies.

A. Implementation Details

Datasets: We conduct the experiments on the FFHQ [4], LSUN Cat [39], Church [39], and Bedroom [39] datasets. The FFHQ dataset consists of 70 K high-resolution (1024×1024) images of human faces, under Creative Commons BY-NC-SA 4.0 license [40]. Usually, the images are horizontally flipped to double the size of training samples. The LSUN Cat dataset contains 1600 K real-world images of different cats, the LSUN Church dataset includes 126 K images with church scenes, and the LSUN Bedroom dataset provides around 300 K complex indoor bedroom images from different views. Following the setting of [41], we respectively take 200 K LSUN Cat samples and 200 K LSUN Bedroom images for training. We use all the FFHQ and LSUN Church images for training. It is worth noting that all images are resized to 256×256 resolution.

Spatial Heatmap Sampling and Encoding: In practice, we find the GradCAM maps on the fine resolutions are too sensitive to semantic cues. Therefore, we only conduct encoding on the level 0,1,2 of G, i.e., resolution 4×4, 8×8, and 16×16. For non-indoor scenes, we heuristically generate 1,2,4 centers (in other words, sub-heatmaps) on these three levels. The center for the level 0 heatmap, denoted as $\mathbf{c_0^0}$, is sampled using a Gaussian distribution with a mean positioned at half the height and width $(\frac{h}{2}, \frac{w}{2})$, and a standard deviation of a third of the height and width $(\frac{h}{3}, \frac{w}{3})$. To maintain heatmap consistency across different levels, the centers for levels 1 and 2 ($\mathbf{c_k^1}$ and $\mathbf{c_k^2}$) are sampled

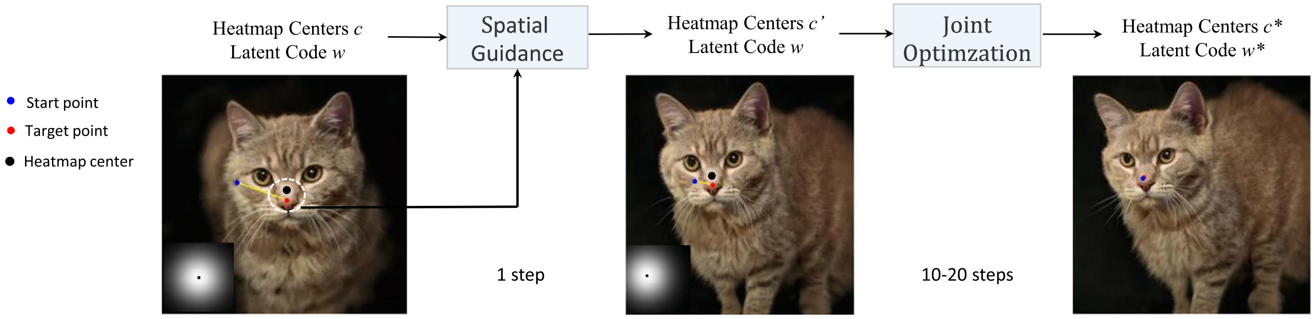


Fig. 8. Overview of SpatialGAN + DragGAN pipeline. In this example, the cat's face is initially marked for repositioning. Subsequently, a coarse spatial adjustment is performed by shifting heatmap centers. Then we jointly optimize the heatmap centers and latent codes, to culminate in the precise alignment of features with predetermined targets, highlighting the model's fine-tuning abilities for detailed editing.

based on the position of the level 0 center. In this context, $\mathbf{c}_\mathbf{k}^\mathbf{1}$ and $\mathbf{c}_\mathbf{k}^\mathbf{2}$ represent multiple potential centers at levels 1 and 2, respectively, with 'k' indicating the specific center number. The standard deviations for these are set at $(\frac{h}{6}, \frac{w}{6})$. It's important to note that if the center at level 0 shifts, the heatmaps at other levels will adjust accordingly. Following the coarse-to-fine manner, we decrease each center's influence area level by level. Besides, we drop the sampling if the level 0 center is outside the image. In our observation, the results of the proposed method are robust to these hyperparameters for heatmap sampling. Therefore, we use the same hyperparameters for heatmap sampling on the FFHQ, LSUN Cat, and LSUN Church datasets. For the indoor setting, we take $n = 3$ sub-heatmaps to construct a complete heatmap, whose centers are sampled via a Gaussian distribution with a mean of $(\frac{h}{2}, \frac{w}{2})$ and a standard deviation of $(\frac{h}{2}, \frac{w}{2})$. We encode this heatmap to all the synthesis layers of G. More implementation details are provided in the Supplementary Material.

Training and Evaluation: We implement our SpatialGAN on the official implementation of StyleGAN2, such that the state-of-the-art image generation method StyleGAN2 [5] serves as our baseline. We follow the default training configuration of [41] for the convenience of reproducibility, and keep the hyperparameters unchanged to validate the effectiveness of our proposed framework. For example, we train all the models with a batch size of 64 on 8 GPUs and continue the training until 25 M images have been shown to the discriminator. Our method increases the training time by around 30% compared to the baseline. We use FrÃ©chet Inception Distance (FID) [42] between 50 K generated samples and all the available real samples as the image generation quality indicator.

B. Main Results

Enhanced Spatial Controllability: Building on our previous discussion, we introduced spatial awareness into the generator (G), aiming to enhance its spatial steerability. In this section, we present qualitative results from various datasets to demonstrate that G effectively concentrates on areas indicated by the input heatmaps, thereby facilitating a range of spatial editing applications. For the indoor scenes, by keeping the spatial heatmaps unchanged, we could control the overall layout (Fig. 5). The bedrooms generated with the same heatmap (each row) will arrange objects in a similar manner, even though the object semantics vary. Furthermore, for the indoor scenes with multi-object heatmap sampling, we can move objects as shown in Fig. 9(a). For example, in the right most column of Fig. 9(a), we drag the sub-heatmap center to the left, and the bed correspondingly moves to the left, with the object trajectory denoted by a yellow arrow. It is also worth noting that G could adaptively modify the nearby texture and structure to produce a reasonable image.

Moreover, since we model the objects with independent sub-heatmaps and style codes for the indoor setting, it allows removing objects or changing the style of a partial region. As shown in Fig. 9(b), we can remove various objects in the indoor scenes. For example, in the left two columns, a window is gradually removed as we decrease the area of the associated sub-heatmap. In other columns, the objects like a painting or a bed are removed. Although the model slightly adjusts the nearby region to keep the synthesis reasonable, the overall image is unchanged. The partial editing samples are shown in Fig. 9(c). As mentioned in Section III-B, we can edit the style code of a specific sub-heatmap and hence change the appearance of the corresponding region. For instance, we can edit the region specified by the blue boxes to different types of paintings and windows.

Turning our attention to single object scenes, we find that by keeping the spatial heatmaps unchanged, we can control the pose of human faces and the viewpoint of churches as shown in Fig. 10(a) and (b). For instance, as we move the level 0 heatmap of the sample in row 1 in Fig. 10(c), the cat bodies move under the guidance of heatmap movement, indicated by red arrows. In the second row of Fig. 10(c), the change in level 1 heatmap leads to a movement in cat eyes. As we slightly push the top two centers of the level 2 heatmap to the right, the cat ears subtly turn right while other parts, even the cat whiskers, remain unchanged. Overall, these qualitative samples, spanning both indoor and non-indoor scenes, demonstrate the spatial steerability introduced by our SpatialGAN.

Quantitative Analysis of Spatial Steerability: In order to assess our method's spatial steerability in a quantitative manner, we examine how the movement of objects is influenced by alterations in the heatmap. Ideally, if the center of a sub-heatmap H_i moves by certain pixels, the corresponding object should also move by the same pixels in the same direction. With

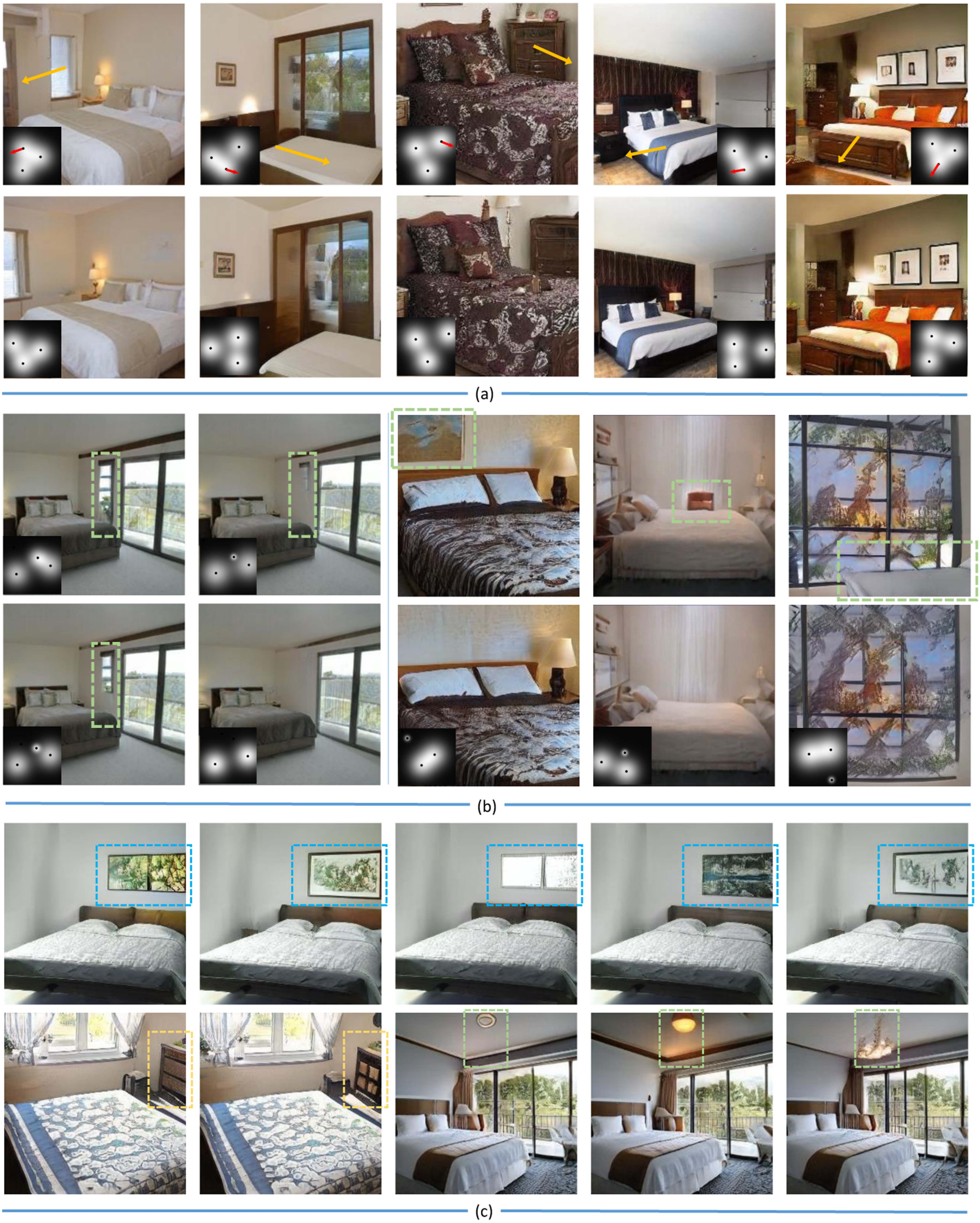


Fig. 9. Manipulating Multi-Object Indoor Scenes. (a) Rearranging Objects: By moving one sub-heatmap center (yellow arrows), corresponding objects like windows and beds are disentangled and moved, with the generator adjusting nearby regions for coherence. (b) Removing Objects: Objects can be removed by eliminating their associated sub-heatmaps, as shown by the gradual removal of elements like windows and beds, leaving the background and other objects mostly unchanged. (c) Replacing Objects: The appearance of local regions is altered using unique style codes for each sub-heatmap, enabling variations in paintings, windows, and light types, as indicated by the blue and green boxes.

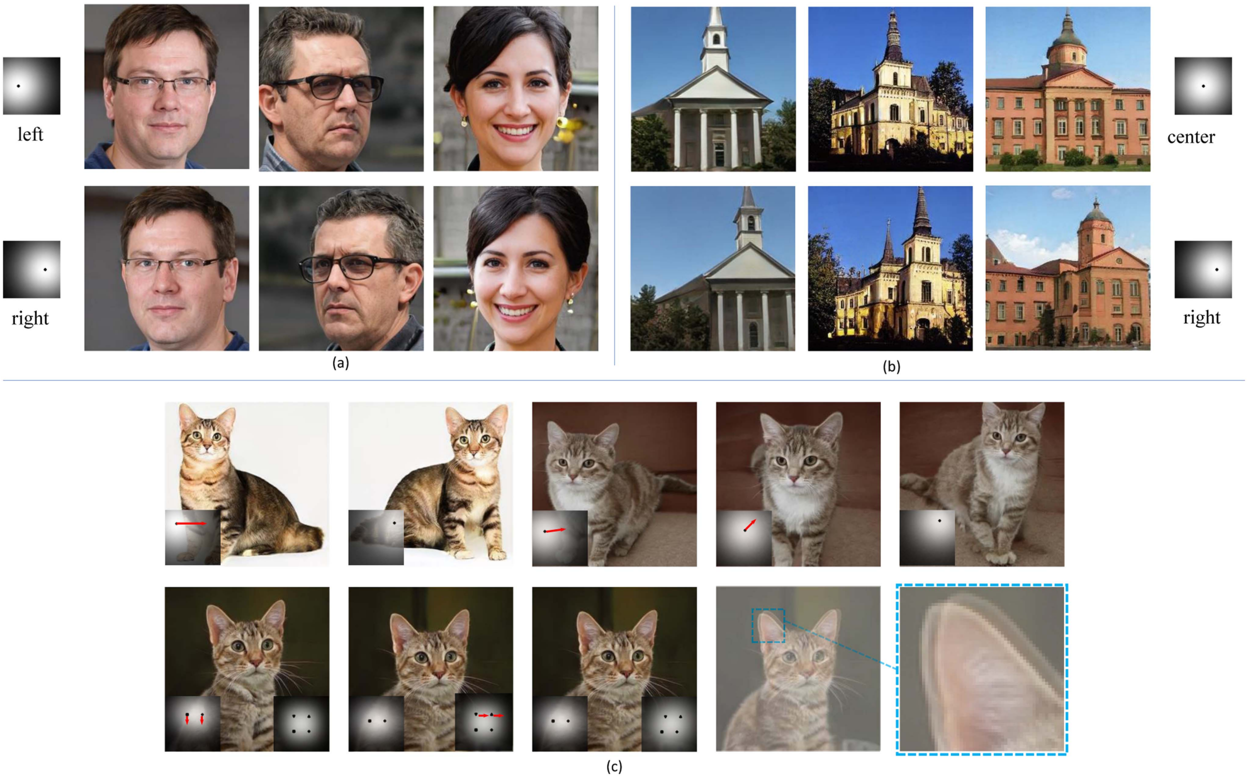


Fig. 10. Qualitative results on the FFHQ (a), the LSUN Church (b), and LSUN Cat dataset (c). Each row uses the same spatial heatmap but different latent codes, and each column uses the same latent code. We can see that the spatial heatmap roughly controls the pose of the face and the viewpoint of the church building, which facilitates the interactive spatial editing of the output image. To further show the hierarchical structure, we move the heatmap to the fine-grained level in the last row. Different from the body movement, the change in 8×8 heatmap (two centers) mainly moves the cat eyes, and the change in 16×16 heatmap (four centers) leads to subtle movement of the cat ears. It is worth noting that, as the content is being manipulated, our G knows to adjust the nearby regions to make everything coherent.

the help of an off-the-shelf instance segmentation model like Mask RCNN [43], we can roughly quantify the movement of a specific object. In detail, for one sub-heatmap H_i, we view its corresponding object as the one where H_i center lies in. Its object center is the average position of the pixels belonging to this object, segmented by Mask RCNN. We move the H_i center by $\mathbf{p}$ and generate a new image, where $\mathbf{p}$ is a random 2D vector. Assuming the overall appearance remains unchanged, we traverse the objects in the new image and look for the object with the smallest feature distance to the one in the original image. We view it as the moved object and compute the vector of its center movement as $\mathbf{q}$. Project $\mathbf{q}$ into the $\mathbf{p}$ to get its movement in the desired direction, $\frac{\mathbf{p}\cdot\mathbf{q}}{\|\mathbf{p}\|}$ as a scalar. We pick its ratio over the desired movement scalar, $\frac{\mathbf{p}\cdot\mathbf{q}}{\|\mathbf{p}\|^2}$, as the indicator of how effectively the heatmap controls the synthesis, named as Co-move Ratio. The closer this indicator is to 1, the better. We evaluate a model's Co-move Ratio by averaging the results of 50 k synthesis samples. The ablation study was conducted on the indoor dataset LSUN Bedroom because it contains multiple independent objects and there are publicly available Mask RCNN models trained on similar scenes.

The results are provided in Table I. The baseline StyleGAN2 is denoted as 'N/A' because it does not support such a moving manipulation. It is noticed that our conference version can move objects but just roughly, with a Co-move Ratio of 0.32, because this version was not designed for scenes with multiple independent objects. By adopting multi-object heatmap sampling, we mitigate the sub-heatmap conflict mentioned in Section III-B and increase the ratio to 0.41. The largest improvement, from 0.41 to 0.62, is brought by SEL$_{indoor}$, where we (1) integrate the spatial heatmaps H and styles S to ensure only modifying the intermediate features once, and (2) employ a unique appearance code for each sub-heatmap. The indoor self-supervision objective further improves the ratio to 0.69. These results verify the effectiveness of our designs for multi-object scenes, which improves the model's ability to move objects as desired.

TABLE I
QUANTITATIVE ANALYSIS ON THE IMPROVEMENT OF SPATIAL CONTROLLABILITY

Method	Co-move Ratio
Baseline	N/A
Ours Conf [14]	0.32
+ Multi-obj Sampling	0.41
+ SEL$_{indoor}$	0.62
+ $\mathcal{L}_{align\_indoor}$	0.69

The experiments were conducted on the LSUN Bedroom dataset. Co-move Ratio indicates the ability to spatially move objects, the closer to 1 the better.

Comparison of Manipulation Ability: As discussed before, we design a two-stage manipulation strategy to leverage the benefits

TABLE II
QUANTITATIVE ANALYSIS OF SPEED PERFORMANCE

Method	Move 30 pix	Move 60 pix	Optimization steps
DragGAN [15]	15 sec	31 sec	30-200
Heatmap & Latent Code	18 sec	35 sec	15-70
SpatialGAN + DragGAN	**6 sec**	**11 sec**	**10-20**

This table presents the results of speed comparison experiments conducted using the LSUN Cat dataset. It showcases the average time required to shift the cat's face by a specified pixel distance.

of both SpatialGAN and DragGAN, which largely reduces the time required for point-based manipulation. Here we provide a quantitative analysis of the speed in Table II, averaging the results from 50 manipulation samples. For example, for the LSUN Cat dataset, moving the cat's face by 60 pixels takes DragGAN approximately 31 seconds and $30-200$ optimization steps. Instead, the combination of SpatialGAN and DragGAN not only reduces the manipulation time to 11 seconds but also significantly decreases the optimization steps to $10-20$. This hybrid two-stage approach offers both coarse and fine manipulation capabilities, providing a more efficient, dynamic, and versatile tool for image manipulation tasks. Please also notice that, without point-based optimization, our SpatialGAN can conduct manipulation within just one second, although it does not support granular manipulation.

In addition to speed comparisons, we evaluated the effectiveness of our SpatialGAN-DragGAN hybrid approach in terms of the relative distance between the start and target points during image manipulation tasks. With DragGAN alone, the final average distance relative to the image size was approximately 0.015. When employing our SpatialGAN combined with DragGAN, this relative distance averaged around 0.027. This result demonstrates that, while our method is three times quicker than DragGAN alone, both methods exhibit comparable accuracy in spatial manipulation.

Interactive Interface: To further enhance user engagement and control, we have developed an interactive interface (UI), as illustrated in Fig. 4. Compared to the conference version, this interface is versatile, supporting models trained under both multi-object indoor and single-object settings. Moreover, utilizing a well-trained SpatialGAN model, users can initiate the process by selecting a random seed to generate an initial image. The interface automatically generates heatmaps with preset centers and sigma values. Users can interactively modify these heatmaps by clicking and dragging the centers, triggering real-time image synthesis reflective of these adjustments. The heatmap alterations are displayed in sequence, corresponding to feature resolutions of 4×4, 8×8, and 16×16, or levels 0, 1, and 2, visible on the interface's right side. For the indoor model, a consistent multi-object heatmap is applied across all layers, while for other models, a hierarchical heatmap structure is employed. Another addition to our interface is the ability to adjust the heatmap areas. In the context of the multi-object setting, this feature offers the ability to manipulate the size or even facilitate the removal of objects associated with specific heatmap centers. Furthermore, for nuanced control in the multi-object setting, users can select specific centers to alter their corresponding heatmap areas.

TABLE III
QUANTITATIVE RESULTS ON THE LSUN CAT, FFHQ, LSUN CHURCH, AND LSUN BEDROOM DATASETS, ALL TRAINED WITH **25M** IMAGES SHOWN TO DISCRIMINATOR

Method	Bedroom FID ↓	Cat FID ↓	FFHQ FID ↓	Church FID ↓
Baseline	4.27	8.36	3.66	3.73
EqGAN-SA [14]	2.95	6.81	2.96	3.11
SpatialGAN	**2.72**	**6.57**	**2.91**	**2.86**

The baseline uses the architecture of StyleGAN2 [4]. We use FID as the metric for image generation quality. ↓ denotes smaller is better.

TABLE IV
ABLATION STUDY OF HEATMAP SAMPLING ON THE SINGLE-OBJECT SCENES

FID ↓	Baseline	Gau. Noise	Non-Hie	Hie
Cat	8.36	8.33	7.03	**6.57**
FFHQ	3.66	3.67	3.29	**2.91**
Church	3.73	3.69	3.20	**2.86**

We separately throw random Gaussian noise, spatial heatmap without hierarchical sampling, and our hierarchical heatmap as the input to the spatial encoding layer. The hierarchical sampling consistently shows a better performance.

Additionally, our user interface now integrates with the DragGAN optimization pipeline, enhancing the granularity of user control. As discussed in Section III-D, the combined optimization process allows users to precisely adjust object placements with DragGAN, as well as to perform coarse manipulations with SpatialGAN. Consequently, the UI provides an intuitive platform for engaging with the advanced features of the system, streamlining the user experience in creative image manipulation.

Enhanced Synthesis Quality: SpatialGAN, our proposed model, notably elevates the quality of synthesis. We present quantitative evidence of this improvement in Table III. Relative to the baseline StyleGAN2 and our conference version EqGAN-SA, SpatialGAN demonstrates consistent enhancements across a variety of datasets. For the indoor dataset LSUN Bedroom, SEL_{indoor} and $\mathcal{L}_{align\_indoor}$ show an impressive performance, improving the baseline FID from 4.27 to 2.72. It is also better than our conference version, whose FID is 2.95, which shows the advantage of the specific design for multi-object scenes. Transitioning to non-indoor datasets, the LSUN Cat dataset saw an FID improvement from 8.36 to 6.57. For context, our conference version had an FID of 6.81 for this dataset. Similarly, the Church dataset's FID was reduced from 3.73 to 2.86, surpassing the conference version's FID of 3.11. The results on the LSUN Cat, FFHQ, and LSUN Church datasets are slightly better than our conference version [14] because we adopt the coarse heatmap processing introduced in Section III-B.

C. Analysis and Discussion

Hierarchical Heatmap Sampling for Single Object Scenes: We conduct an ablation study to validate the effect of hierarchical heatmap sampling on non-indoor scenes, as provided in Table IV. Specifically, 2D Gaussian noise is first considered

TABLE V
ABLATION STUDY OF SPATIAL ENCODING ON SINGLE-OBJECT SCENES

Method	Cat FID ↓	FFHQ FID ↓	Church FID ↓
Baseline	8.36	3.66	3.73
Flatten	8.52	3.75	3.79
SEL$_{concat}$	6.73	3.02	2.93
SEL$_{norm}$	**6.57**	**2.91**	**2.86**

We flatten the spatial heatmap and incorporate the vectorized one into latent code, denoted as ‘Flatten’. It destroys the $2D$ space structure, and hence cannot improve over the baseline. Instead, encoding heatmaps in the spatial domain is beneficial. The two variants of SEL show a similar result, where SEL*norm* is slightly better.

as a straightforward baseline experiment since it provides non-structured spatial information. Accordingly, 2D Gaussian noise introduces no performance gains. It indicates, merely feeding a spatial heatmap but without any region to be emphasized is insufficient to raise spatial awareness.

Besides, we also use multiple-resolution spatial heatmaps but discard the hierarchical constraint, referred to as Non-Hie in Table IV. Namely, spatial heatmaps at different resolutions are independently sampled. The baseline is obviously improved by this non-hierarchical spatial heatmap, demonstrating the effectiveness of the spatial awareness of G. Moreover, when our hierarchical sampling is adopted, we observe further improvements over the synthesis quality.

Spatial Encoding for Single Object Scenes: In order to raise the spatial controllability of G, there exist several alternatives to implement spatial encoding. Therefore, on non-indoor scenes, we conduct an ablation study to test various methods. For example, the first way of feeding the spatial heatmap is to flatten the 2D heatmap as a vector, and then concatenate it with the original latent code. This setting aims at validating whether maintaining 2D structure of spatial heatmap is necessary. Besides, we also use two different SEL modules (i.e., SEL$_{concat}$ and SEL$_{norm}$) mentioned in Section III-B. Table V presents the results. Apparently, simply feeding the spatial heatmap but without the explicit $2D$ structure leads to no gains compared to the baseline. It might imply that it is challenging to use a vector (like the original latent code) to raise the spatial awareness of the generator. Instead, the proposed SEL module could introduce substantial improvements, demonstrating the effectiveness of the encoding implementation. For a fair comparison, all the ablation studies use $\mathcal{L}_{align}$.

Exploring Heatmap Dynamics in Point-Based Optimization: As previously discussed in the Section III-D, the integration of DragGAN into our method involves the simultaneous optimization of heatmap and latent code for point-based manipulations. In the beginning, we observed that optimizing only the latent code – while keeping heatmap points static for single-object scenarios – imposes limitations. The unchanged heatmap restricts image transformation, while the evolving latent code attempts to induce change. This discrepancy hinders effective manipulation, leading to distorted outcomes without meaningful alteration.

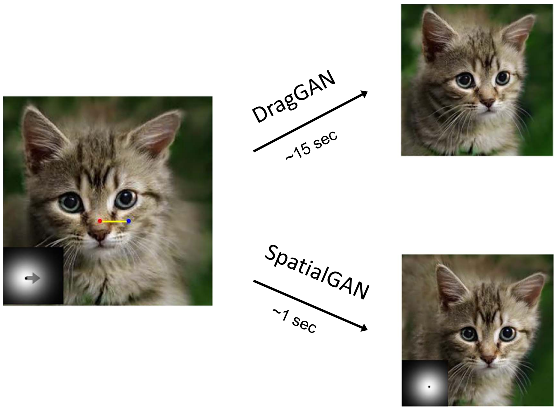


Fig. 11. Speed Comparison between DragGAN and SpatialGAN. The images illustrate shifting a cat’s face a few pixels to the right. While DragGAN requires approximately 15 seconds to complete this task, our SpatialGAN accomplishes the same in just 1 s.

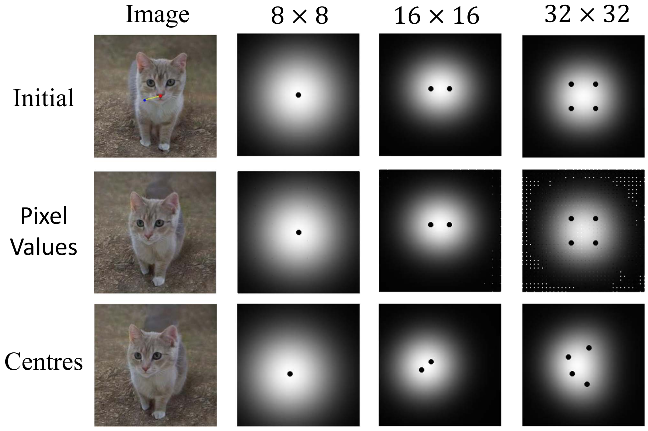


Fig. 12. Illustrative comparison of optimization strategies in DragGAN. **Initial Optimization (Top):** Shows the Initial image. **Heatmap Adjustment (Middle):** Demonstrates the effect of modifying the heatmap values, which leads to selective adversarial heatmap adjustments. **Center Optimization (Bottom):** Depicts the outcome of optimizing heatmap centers, which results in efficient and harmonious image manipulation. The columns represent the progression of heatmaps.

Consequently, we experimented with modifying the pixel values of heatmaps H together with the latent code. This approach resulted in selective pixel adjustments. As depicted in Fig. 12, we can observe that only certain pixel values of the heatmaps would be changed, and such changes mostly happen at the 32 × 32 level. Although this strategy achieved the desired image manipulation, its lack of explainability and the challenge it poses for further user manipulation are significant drawbacks.

To synergize with our spatial GAN method, we then optimize the heatmap centers, guiding them towards the intended directional movement of images. This strategy enables the heatmap to not only accommodate but also enhances the DragGAN optimization process. We employed an alternating optimization strategy, where in one iteration, we focused on modifying the latent code to initiate directional changes, followed by adjusting the heatmap centers in the subsequent iteration. This collaborative

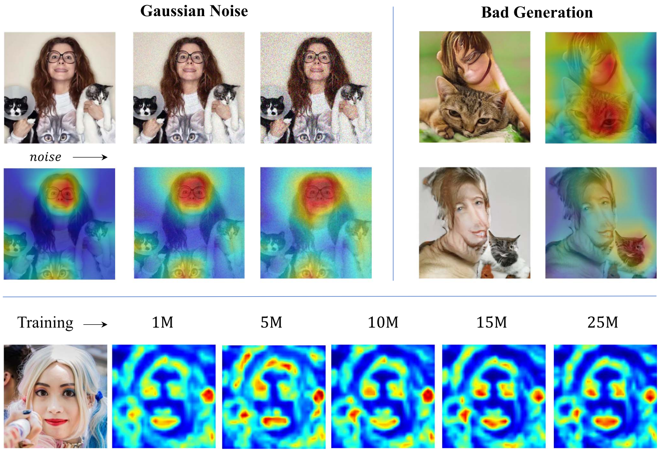


Fig. 13. Robustness and Consistency. We test the response of D to noisy images and bad generation samples in the top. The bottom visualizes that the D's attention is consistent over the training.

adjustment of both elements demonstrated a more efficient and harmonious manipulation process.

Whether Visual Attention of D is Robust and Consistent? As discussed in Section III-C, the self-supervised alignment loss uses the D attention maps to guide G. It assumes the attention map from D is stable enough to serve as a self-supervision signal and valid over the whole training. To validate the design, we first explore the *robustness* of D. As shown in the top left of Fig. 13, we add random Gaussian noise to a real image from the LSUN Cat dataset, destroying its texture. As the noise amplitude increasing, we can visually see the noise pattern and the local appearance has been over smoothed. D is still attentive to the original important regions, e.g., the human and cat faces. We then test its response to terrible samples generated by a poorly-trained G, illustrated in the right top of Fig. 13. The samples contain distorted human, cat, and background. That is, the visual attention of D is sufficiently robust to the noise perturbation and the generated artifacts. Furthermore, as indicated at the bottom of Fig. 13, we validate whether the visual attention is *consistent* throughout the entire training process. At a very early stage of training, D has already localized the discriminative regions. The focus of such visual attention is consistently maintained till the end of the training. The robustness and consistency property of D attention could successfully provide a support for our self-supervision objective.

Visualization of Generator Intermediate Features: The spatial attention of generator is worth investigating. However, CAM or GradCAM is not a suitable visualization tool because they both require a classification score, which is not applicable for a generator. Introducing another classifier may be a solution but it would introduce the bias of the classifier. As an alternative solution, we could directly average the intermediate features of the generator along the feature dimension. Such a visualization can be viewed as *the contribution of a layer towards certain pixels*. As shown in Fig. 14, our generator shows a much clearer spatial awareness than the baseline which presents random spatial focus, particularly at the 32 × 32 resolution.

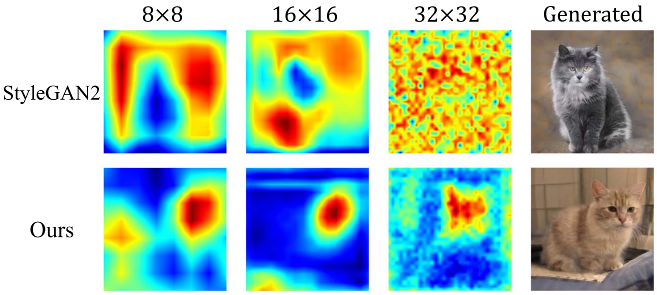


Fig. 14. Visualization of intermediate features of the generator. Our method shows a much clearer spatial awareness than the baseline, whose spatial focus is close to random.

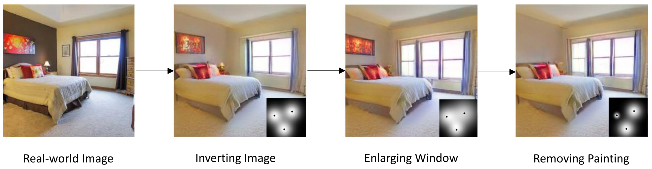


Fig. 15. Real image manipulation. We apply GAN inversion on a real image to invert it to the latent space of our model, and then enlarge its window and remove its painting.

Is a Heatmap Necessary to be Continuous? As discussed in Section III-B, coarse heatmap processing could improve the synthesis quality (Table III) even though it reduces the heatmap continuity, which is somehow counter-intuitive. We study the importance of heatmap continuity by randomly inserting extreme values (0 or 1) into heatmaps, i.e., adding impulse noise. It would lead to some local jumps but keep the overall spatial structure in heatmaps. The synthesis quality stays stable as we gradually increase the impulse noise percentage. For example, on the LSUN Cat dataset, the model keeps an FID of 6.93 even if using impulse noise over 5% heatmap pixels. The synthesis quality is not sensitive to the continuity of the spatial inductive bias, possibly because the real-world semantics distribution is not necessary to be continuous. Besides, we empirically find that heatmap continuity would also not affect spatial controllability.

How Important is the Receptive Field for Heatmap Processing? Another hypothesis for the effect of coarse heatmap processing is that a global understanding matters. A large receptive field will introduce a global view of the spatial structure. Without coarse heatmap processing, the FID of our model on the LSUN Cat dataset is 6.81, with the convolution kernel size as 3 × 3. Increasing the kernel size to 7 × 7 would improve the result to 6.72. If adopting dilated convolution [44] to further increase the receptive field, it would become 6.61, closer to the result of coarse heatmap processing (6.57). The experiments verify our speculation that a global understanding of spatial structure helps. For simplicity, we do not use dilated convolution in our method.

Real Image Manipulation: Fig. 15 shows an example of real-image manipulation on a bedroom scene. We first invert the real image into the latent space and then alter the heatmap to manipulate the inverted image. It should be noted that the editability of the inverted latent code highly depends on the GAN

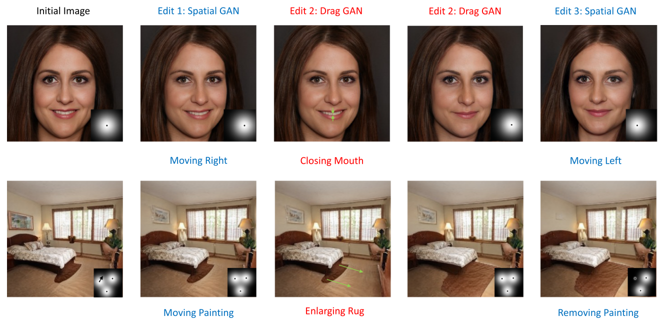


Fig. 16. More qualitative samples of the of SpatialGAN + DragGAN pipeline, where we first perform manipulation using SpatialGAN, followed by DragGAN, and finally another editting by SpatialGAN.

inversion quality, and it remains very challenging to recover the details of multi-object images [45]. Our method can achieve reasonable editing output.

Synergy Between DragGAN and SpatialGAN: Fig. 16 shows more qualitative samples from integrating DragGAN and SpatialGAN.

V. Conclusion and Discussions

In this paper, we propose a method to improve the spatial steerability and synthesis quality of GANs. Specifically, we notice that D spontaneously learns its visual attention, which can serve as a self-supervision signal to raise the spatial awareness of G. Therefore, we encode spatial heatmaps into the intermediate features of G, and align the heatmaps and the attention of D during training. Qualitative results show that our method successfully makes G concentrate on specific regions. This method enables multiple spatial manipulations like moving or removing objects in the synthesis by altering the encoded heatmaps, and consistently improves the synthesis quality on various datasets.

Limitation: Though simple and effective, our SpatialGAN is heuristic and built upon existing techniques. In addition, we notice the spatial encoding operation would sometimes lead to a synthesis blurring at the location of heatmaps boundaries, which may visually affect the manipulation quality. Sometimes, altering one sub-heatmap would change the appearance of some remote areas, which is not desired by our design. We consider SpatialGAN as an empirical study and hope it can inspire more work on improving the image synthesis quality and controllability of GANs.

Ethical Consideration: This paper focuses on improving the spatial controllability of GANs. Although only using public datasets for research and following their licences, the abuse of our method may bring negative impacts through deep fake generation. Such risks would increase as the synthesis results of GANs are becoming more and more realistic. From the perspective of academia, these risks may be mitigated by promoting the research on deep fake detection. It also requires the management on the models trained with sensitive data.

References

[1] I. Goodfellow et al., "Generative adversarial nets," in *Adv. Neural Inform. Process. Syst.*, 2014, pp. 2672–2680.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2016.

[3] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 9256–9290.

[4] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.

[5] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8110–8119.

[6] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9243–9252.

[7] C. Yang, Y. Shen, and B. Zhou, "Semantic hierarchy emerges in deep generative representations for scene synthesis," *Int. J. Comput. Vis.*, vol. 129, pp. 1451–1466, 2021.

[8] Y. Shen and B. Zhou, "Closed-form factorization of latent semantics in GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1532–1540.

[9] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola, "Ganalyze: Toward visual definitions of cognitive image properties," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 5744–5753.

[10] A. Voynov and A. Babenko, "Unsupervised discovery of interpretable directions in the GAN latent space," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9786–9796.

[11] C. Yang, Y. Shen, and B. Zhou, "Semantic hierarchy emerges in deep generative representations for scene synthesis," *Int. J. Comput. Vis.*, vol. 129, pp. 1451–1466, 2021.

[12] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "Ganspace: Discovering interpretable GAN controls," in *Proc. Adv. Neural Inform. Process. Syst.*, 2020, pp. 9841–9850.

[13] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.

[14] J. Wang, C. Yang, Y. Xu, Y. Shen, H. Li, and B. Zhou, "Improving GAN equilibrium by raising spatial awareness," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11285–11293.

[15] X. Pan, A. Tewari, T. Leimkühler, L. Liu, A. Meka, and C. Theobalt, "Drag your GAN: Interactive point-based manipulation on the generative image manifold," in *Proc. ACM SIGGRAPH Conf.*, 2023, pp. 1–11.

[16] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. Adv. Neural Inform. Process. Syst.*, 2015, pp. 1486–1494.

[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 213–238.

[18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 376–401.

[19] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.

[20] J. Donahue and K. Simonyan, "Large scale adversarial representation learning," in *Proc. Adv. Neural Inform. Process. Syst.*, 2019, pp. 10542–10552.

[21] E. Schonfeld, B. Schiele, and A. Khoreva, "A U-Net based discriminator for generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8207–8216.

[22] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11453–11464.

[23] A. Casanova, M. Careil, J. Verbeek, M. Drozdzal, and A. Romero Soriano, "Instance-conditioned GAN," in *Proc. Adv. Neural Inform. Process. Syst.*, 2021, pp. 27517–27529.

[24] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6840–6851.

[25] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2015, pp. 2256–2265.

[26] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 11918–11930.

[27] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” 2020, *arXiv: 2011.13456*.
[28] A. Nichol et al., “Glide: Towards photorealistic image generation and editing with text-guided diffusion models,” 2021, *arXiv:2112.10741*.
[29] B. Kawar et al., “Imagic: Text-based real image editing with diffusion models,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 6007–6017.
[30] Y. Huang et al., “Diffusion model-based image editing: A survey,” 2024, *arXiv:2402.17525*.
[31] A. Plumerault, H. L. Borgne, and C. Hudelot, “Controlling generative models with continuous factors of variations,” in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 4006–4022.
[32] A. Jahanian *, L. Chai *, and P. Isola, “On the ”steerability” of generative adversarial networks,” in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 6879–6909.
[33] J. Zhu, Y. Shen, Y. Xu, D. Zhao, and Q. Chen, “Region-based semantic factorization in GANs,” in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 27612–27632.
[34] D. Epstein, T. Park, R. Zhang, E. Shechtman, and A. A. Efros, “BlobGAN: Spatially disentangled scene representations,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 616–635.
[35] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
[36] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
[37] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 1501–1510.
[38] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2337–2346.
[39] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop,” 2015, *arXiv:1506.03365*.
[40] T. A. Tero Karras, Samuli Laine, Flickr-faces-hq dataset (FFHQ), 2019. [Online]. Available: https://github.com/NVlabs/ffhq-dataset
[41] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training generative adversarial networks with limited data,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 12 104–12 114.
[42] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local nash equilibrium,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6629–6640.
[43] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2961–2969.
[44] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in *Proc. Int. Conf. Learn. Representations*, 2016.
[45] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, “GAN inversion: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3121–3138, Mar. 2023.

Jianyuan Wang received the BE degree with first-class honors from the Australian National University in 2019. He is currently working toward the PhD degree with the Visual Geometry Group (VGG), University of Oxford. He serves as the reviewer for *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ICLR, CVPR, ICCV, and so on. His research interests include visual geometry and generative modelling.

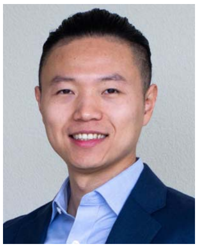
Lalit Bhagat received the BTech degree in computer science and engineering with honors from the JIIT, Noida, India, in 2021. He is currently working toward the MS degree in computer science with the University of California, Los Angeles (UCLA). He is an award recipient of the GCD Fellowship with UCLA. His research interests include computer vision and generative modelling.

Ceyuan Yang received the BEng degree from Honors College, Northwestern Polytechnical University, in 2018, and the PhD degree from The Chinese University of Hong Kong in 2022. He is a young researcher with Shanghai AI Laboratory. His research interests include computer vision and machine learning, particularly in video understanding, generative models and representation learning.

Yinghao Xu received the graduation degree from Zhejiang University in 2019 working closely with Dr. Lechao Cheng. He is currently working toward the third-year PhD degree with Multimedia Lab (MMLab), Department of Information Engineering in The Chinese University of Hong Kong. His supervisor is Prof. Bolei Zhou. His research interests include video understanding, generative models as well as structural representation for vision perception.

Yujun Shen received the BS degree from Tsinghua University and the PhD degree from the Chinese University of Hong Kong. He is a senior research scientist with Ant Research. Before that, he worked as a senior researcher with ByteDance Inc. His research interests include computer vision and deep learning, particularly in 3D vision and generative models. He is an award recipient of Hong Kong PhD Fellowship.

Hongdong Li is a professor with the Australian National University (ANU). He was a visiting professor with the Robotics Institute, Carnegie Mellon University (CMU) in 2017. His research interests include 3D vision reconstruction, structure from motion, multi-view geometry, and visual perception. He is an associate editor for *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and serves as an area chair for recent years’ CVPR, ICCV, and ECCV. He is also a general co-chair for ACCV 2018 and ACCV 2022. He won a number of paper awards in computer vision and pattern recognition, including the CVPR Best Paper Award 2012, the Marr Prize (Honorable Mention) at ICCV 2017, and so on.



Bolei Zhou received the PhD degree from MIT in 2018. He is an assistant professor with the Computer Science Department, the University of California, Los Angeles (UCLA). His research interest lies at the intersection of computer vision and machine autonomy, focusing on enabling interpretable human-AI interaction. He and his colleagues have developed many widely used interpretation methods, such as CAM and Network Dissection, as well as computer vision benchmarks Places and ADE20 K. He is an associate editor for the *International Journal of Computer Vision* and has been area chair for CVPR, ICCV, ECCV, and AAAI. He received the NSF CAREER Award, Intel’s Rising Star Faculty Award, and MIT Tech Review’s Innovators under 35 in Asia-Pacific Award.