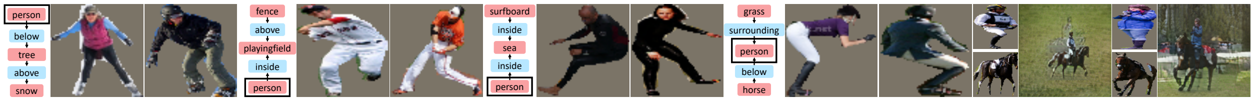

Figure A. Top-2 retrieved **person** crops using our crop selector with different scene graphs during inference on COCO.

The performance of the proposed Crop Selection. (R1) As shown in Fig. A, by encoding the context information with GCN, our selector can retrieve different *person*s with different poses and uniforms for different scene graphs. These scene-compatible crops will simplify the generation process and significantly improve the image quality over the random selection. The ablation study *"w/o crop selection"* in Tab. 2 validates our claim.

What do u_{img} and u^{attn} refer to. (R1) Object-Image Fuser aims at fusing different objects to the virtual *"image"* object via *"in image"* relationship. Objects are merged through attention maps. u^{attn} is the fused object feature after calculating attentions with *"in image"* relationships. u_{img} is the feature map of virtual *"image"* object. Eq. (4) aggregates the objects to the *"image"* via λ_{attn} (we use 1 in our experiments). Additionally, there is a typo in Eq. (3), where D should be N, the number of objects in that image. Will revise in the next version.

Necessity of components. (R1) Our PasteGAN is based on SG2IM [4]. Compared to SG2IM, our method has several novelties: 1) a semi-parametric setting; 2) an object-image fuser; 3) a crop selector; 4) an object^2 refiner. Our ablation study (Tab. 2) shows the necessity of these components. For the other components like D_{img} or D_{obj}, they are not our main contributions, and their effectiveness has been validated in [4]. We got similar conclusions on our PasteGAN.

Reproducibility of PasteGAN. (R1, R2) Most of hyper-parameters are used in [4] and we followed their settings. For the newly-introduced ones, the hyper parameters are tuned heuristically. We will release the code for reproducibility.

Crop's resolution. (R1) Comparing to the size of an image, most objects usually occupy a small region, so we specify the size of an object to half of the image size to reduce the computational cost. Experimental results show that the image-size crops don't bring significant improvements and slower the inference.

Limited novelty. (R2, R3) Our proposed PasteGAN aims at enabling the model to finely control the appearance of the objects in generated image through a semi-parametric setting. As R2 said, the semi-parametric setting has its advantages naturally over generating the images from scratch, and to the best of our knowledge, our PasteGAN is the first semi-parametric method to generate the image from the scene graph. Moreover, this is not a trivial combination of [3] and [4], as we propose: 1) a crop selector to retrieve the most scene-matching crops by encoding the scene graph; 2) an object^2 refiner to translate the object crops to the target appearance based on their connections; 3) an object-image fuser to merge the objects into a latent scene feature map. All these modules make our PasteGAN outperform the baselines, which is proven by our experimental results. Additionally, the high diversity score and the qualitative results both prove that by alternating the input object crops, we can finely control the generated images, especially the objects' appearance, and synthesize diverse images, which is the goal of this work.

Limited resolution. (R2, R3) Most of the SOTA high-resolution generative models either leverage a cascaded generator, like StackGAN++, or generate specific images, such as birds or street views, like [3]. As generating general images from scene graphs is an emerging and challenging topic, how to encode the scene graph is the most critical part of the investigation, so most of the existing methods choose to generate low-resolution images for more efficient model training. For a fair comparison with baselines, SG2IM [4] and Layout2IM [16], we follow the most commonly-used size, 64×64. We also evaluate our method on 128×128 image generation. We get **14.5** and **10.4** (Inception Score) on COCO and VG, respectively, which outperforms 12.4 (COCO) of [Hong et al.] on the same size. For higher resolution, like 256×256, we may need to redesign a multi-stage cascaded generator, which worths investigating in the future.

Missing reference of [Hong et al.]. (R2) The two methods work on different but related problems, image generation from free-form texts v.s. from more structured scene graphs. Besides the semi-parametric setting, the most significant difference of the methodology part is that [Hong et al.] uses the textual information to generate object masks and heuristically aggregates them to a soft semantic map for the image decoding, while ours utilizes a 2D-GCN-based *object-image fuser* to merge the objects in a learnable way. We will cite and discuss it in the next version.

Crop Selection. (R3) If the crop selector is jointly trained with our PasteGAN, the visual code of each object will change at every training step. Correspondingly, we need to extract the visual code for all the training set whenever the model gets updated, which is nearly impossible. Therefore, we alternatively utilize the pre-trained model to extract the visual codes offline and directly use them during the training and inference.

Pair-wise relationships discriminator. (R3) Using an additional relationship discriminator to ensure the consistency of the pair-wise relations is a great idea worth trying. We used a GCN-based RelationNet to distinguish the pair-wise relations in the generated image as the additional discriminator, but cannot observe any improvements on these two benchmark datasets. We will continue investigating the idea in the future.

Mistake in captions of Fig. 2. (R3) Will revise in the next version.