Supplementary Materials: GLIGEN: Open-Set Grounded Text-to-Image Generation

Yuheng Li[1§], Haotian Liu[1§], Qingyang Wu[2], Fangzhou Mu[1], Jianwei Yang[3], Jianfeng Gao[3], Chunyuan Li[3¶], Yong Jae Lee[1¶]

[1]University of Wisconsin-Madison [2]Columbia University [3]Microsoft

https://gligen.github.io/

In this supplemental material, we provide more implementation and training details, and then present more results and discussions.

1. Implementation and training details

We use the Stable Diffusion model [12] as the example to illustrate our implementation details.

Box Grounding Tokens with Text. Each grounded text is first fed into the text encoder to get the text embedding (*e.g.*, 768 dimension of the CLIP text embedding in Stable Diffusion). Since the Stable Diffusion uses features of 77 text tokens outputted from the transformer backbone, thus we choose "`EOS`" token feature at this layer as our grounded text embedding. This is because in the CLIP training, this "`EOS`" token feature is chosen and applied a linear transform (one FC layer) to compare with visual feature, thus this token feature should contain whole information about the input text description. We also tried to directly use CLIP text embedding (after linear projection), however, we notice slow convergence empirically probably due to unaligned space between the grounded text embedding and the caption embeddings. Following NeRF [10], we encode bounding box coordinates with the Fourier embedding with output dimension 64. As stated in the Equation 5 in the main paper, we first concatenate these two features and feed them into a multi-layer perceptron. The MLP consists of three hidden layers with hidden dimension 512, the output grounding token dimension is set to be the same as the text embedding dimension (*e.g.*, 768 in the Stable Diffusion case). We set the maximum number of grounding tokens to be 30 in the bounding box case.

Box Grounding Tokens with Image. We use the similar way to get the grounding token for an image. We use the CLIP image encoder (ViT-L-14 is used for the Stable Diffusion) to get an image embedding. We denote the CLIP training objective as maximizing $(\mathbf{P}_t \boldsymbol{h}_t)^\top (\mathbf{P}_i \boldsymbol{h}_i)$ (we omit normalization), where $\boldsymbol{h}_t$ is "`EOS`" token embedding from the text encoder, $\boldsymbol{h}_i$ is "`CLS`" token embedding from the image encoder, and $\mathbf{P}_t$ and $\mathbf{P}_i$ are linear transformation for text and image embedding, respectively. Since $\boldsymbol{h}_t$ is the text feature space used for grounded text features, to ease our training, we choose to project image features into the text feature space via $\mathbf{P}_t^\top \mathbf{P}_i \boldsymbol{h}_i$, and normalized it to 28.7, which is average norm of $\boldsymbol{h}_t$ we empirically found. We also set the maximum number of grounding tokens to be 30. Thus, 60 tokens in total if one keep both image and text as representations for a grounded entity.

Keypoint Grounding Tokens. The grounding token for keypoint annotations is processed in the same way, except that we also learn N person token embedding vectors $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$ to semantically link keypoints belonging to the same person. This is to deal with the situation in which there are multiple people in the same image that we want to generate, so that the model knows which keypoint corresponds to which person. Each keypoint semantic embedding $f_{\text{text}}(e)$ is processed by using the text encoder, for example, we forward the text: "`left eye`" into the encoder to get its semantic embedding; the dimension of each person token is set the same as text embedding dimension. The grounding token is calculated by:

$$\boldsymbol{h}^e = \text{MLP}(f_{\text{text}}(e) + \boldsymbol{p}_j, \text{Fourier}(\boldsymbol{l})) \tag{1}$$

where $\boldsymbol{l}$ is the x, y location of each keypoint and $\boldsymbol{p}_j$ is the person token for the j'th person. In practice, we set N as 10, which is the maximum number of persons allowed to be generated in each image. Thus, we have 170 tokens in the COCO dataset (*i.e.*, 10*17; 17 keypoint annotations for each person).

Grounding tokens for Spatially Aligned Condition. This type of condition includes edge map, depth map, semantic map, and normal map, etc; they can be represented

§ Part of the work performed at Microsoft; ¶ Co-senior authors

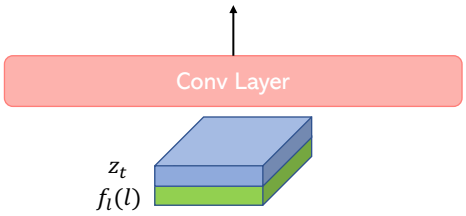


Figure 1. Additional grounding input is fed into the Unet input for spatially aligned conditions.

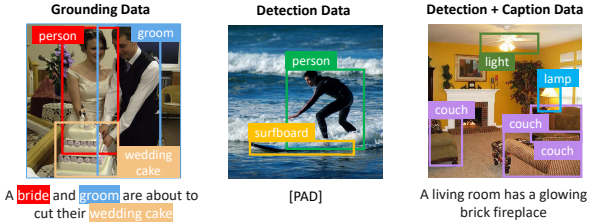


Figure 2. Three different types of grounding data for box.

as $C \times H \times W$ tensor. We resize spatial size into 256×256 and use the convnext-tiny [9] as the backbone to output a feature with spatial size as 8×8, which then is flattened into 64 grounding tokens. We notice that it can help training faster if we also provide the grounding condition l into the Unet input. As shown in the Figure 1, in this case, the input is $\text{CONCAT}(f_l(\boldsymbol{l}), \boldsymbol{z}_t)$ where f_l is a simple downsampling network to reduce $\boldsymbol{l}$ into the same spatial dimension as $\boldsymbol{z}_t$, which is the noisy latent code at the time step t. In this case, the first conv layer of Unet needs to be trainable.

Gated Self-Attention Layers. Our inserted self-attention layer is the same as the original diffusion model self-attention layer at each Transformer block, except that we add one linear projection layer which converts the grounding token into the same dimension as the visual token. For example, in the first layer of the down branch of the UNet [13], the projection layer converts grounding token of dimension 768 into 320 (which is the image feature dimension at this layer), and visual tokens are concatenated with the grounding tokens as the input to the gated attention layer.

Training Details. For all COCO related experiments (Sec. 5.1 in the main paper), we train LDM with batch size 64 using 16 V100 GPUs for 100k iterations. In the scaling up training data experiment (in Sec. 5.2 of the main paper), we train for 400k iterations for LDM, but 500K iterations with batch size of 32 for the Stable diffusion modeL For all training, we use learning rate of 5e-5 with Adam [6], and use warm-up for the first 10k iterations. We randomly drop caption and grounding tokens with 10% probability for classifier-free guidance [4].

Data Details. In the main paper Sec.5.1, we study three different types of data for box grounding. The training data requires both text $\boldsymbol{c}$ and grounding entity $\boldsymbol{e}$ as the full condition. In practice, we can relax the data requirement by considering a more flexible input, *i.e.* the three types of data shown in Figure 2(a). (i) *Grounding data*. Each image is associated with a caption describing the whole image; noun entities are extracted from the caption, and are labeled with bounding boxes. Since the noun entities are taken directly from the natural language caption, they can cover a much richer vocabulary which will be beneficial for open-world vocabulary grounded generation. (ii) *Detection data*. Noun-entities are pre-defined closed-set categories (*e.g.*, 80 object classes in COCO [8]). In this case, we choose to use a null caption token as introduced in classifier-free guidance [4] for the caption. The detection data is of larger quantity (millions) than the grounding data (thousands), and can therefore greatly increase overall training data. (iii) *Detection and caption data*. Noun entities are same as those in the detection data, and the image is described separately with a text caption. In this case, the noun entities may not exactly match those in the caption. For example, in Figure 2(a), the caption only gives a high-level description of the living room without mentioning the objects in the scene, whereas the detection annotation provides more fine-grained object-level details.

2. Ablation Study

Ablation on gated self-attention. As shown in the main paper Figure 3 and Equation 8, our approach uses gated self-attention to absorb the grounding instruction. We can also consider gated cross-attention [1], where the query is the visual feature, and the keys and values are produced using the grounding condition. We ablate this design on COCO2014CD data using LDM. Compare with the table 1 in the main paper, we can find that it leads to similar FID: 5.8, but worse YOLO AP: 16.6 (compared to 21.7 for self-attention in the Table). This shows the necessity of information sharing among the visual tokens, which exists in self-attention but not in cross-attention.

Ablation on null caption. We choose to use the null caption when we only have detection annotations (COCO2014D). An alternative scheme is to simply combine all noun entities into a sentence; *e.g.*, if there are two cats and a dog in an image, then the pseudo caption can be: "`cat, cat, dog`". In this case, the FID becomes worse and increases to 7.40 from 5.61 (null caption, refer to main paper table 1). This is likely due to the pretrained text encoder never having encountered this type of unnatural caption during LDM training. A solution would be to finetune the text encoder or design a better prompt, but this is not the focus of our work.

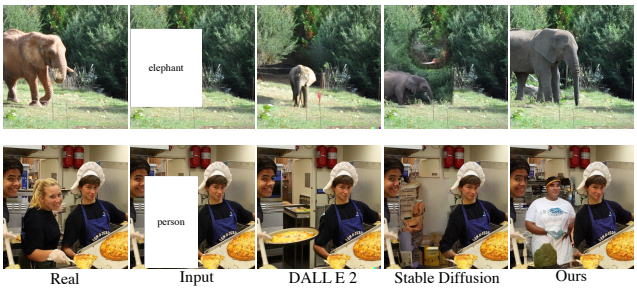


Figure 3. **Inpainting results.** Existing text2img diffusion models may generate objects that do not tightly fit the masked box or miss an object if the same object already exists in the image.

	1%-3%	5%-10%	30%-50%
LDM [12]	25.9	23.4	14.6
GLIGEN-LDM	**29.7**	**30.9**	**25.6**
Upper-bound	41.7	43.4	45.0

Table 1. Inpainting results (YOLO AP) for different size of objects.

3. Grounded inpainting

3.1. Text Grounded Inpainting

Like other diffusion models, GLIGEN can also work for the inpainting task by replacing the known region with a sample from $q(z_t|z_0)$ after each sampling step, where z_0 is the latent representation of an image [12]. One can ground text descriptions to missing regions, as shown in Figure 3. In this setting, however, one may wonder, can we simply use a vanilla text-to-image diffusion model such Stable Diffusion or DALLE2 to fill the missing region by providing the object name as the caption? What are the benefits of having extra grounding inputs in such cases? To answer this, we conduct the following experiment on the COCO dataset: for each image, we randomly mask one object. We then let the model inpaint the missing region. We choose the missing object with three different size ratios with respect to the image: small (1%-3%), median (5%-10%), and large (30%-50%). 5000 images are used for each case.

Table 1 demonstrates that our inpainted objects more tightly occupy the missing region (box) compared to the baselines. Fig. 3 provides examples to visually compare the inpainting results (we use Stable Diffusion for better quality). The first row shows that baselines' generated objects do not follow the provided box. The second row shows that when the missing category is already present in the image, they may ignore the caption. This is understandable as baselines are trained to generate a *whole* image following the caption. Our method may be more favorable for editing applications, where a user might want to generate an object that fully fits the missing region or add an instance of a class that already exists in the image.

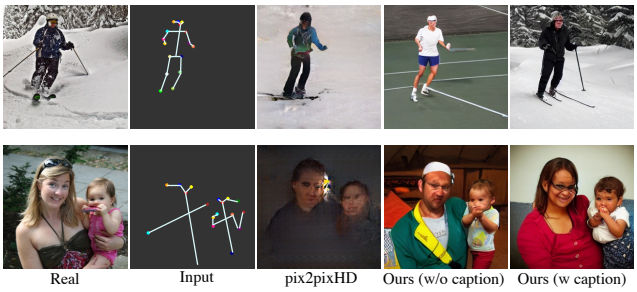


Figure 4. **Keypoint results.** Our model generates higher quality images conditioned on keypoints, and it allows to use caption to specify details such as scene or gender.

Model	FID	AP	AP_{50}	AP_{75}
pix2pixHD [16]	142.4	15.8	33.7	13.0
GLIGEN (w/o caption)	31.02	**31.8**	**53.5**	**31.0**
GLIGEN (w caption)	**27.34**	31.5	52.9	**31.0**
Upper-bound	-	62.4	75.0	65.9

Table 2. Conditioning with Human Keypoints evaluated on COCO2017 validation set. Upper-bound is calculated on real images scaled to 256 $\times$ 256.

3.2. Image Grounded Inpainting

As we previously demonstrated, one can ground text to missing region for inpainting, one can also ground reference images to missing regions. Figure 5 shows inpainting results grounded on reference images. To remove boundary artifacts, we follow GLIDE [11], and modify the first conv layer by adding 5 extra channels (4 for z_0 and 1 for inpainting mask) and make them trainable with the new added layers.

4. Study for Keypoints Grounding

Although we have thus far demonstrated results with bounding boxes, our approach has flexibility in the grounding condition that it can use for generation. To demonstrate this, we next evaluate our model with another type of grounding condition: human keypoints. We use the COCO2017 dataset; details of the tokenization process for keypoints can be found in the supp. We compare with pix2pixHD [16], a classic image-to-image translation model. Since pix2pixHD does not take captions as input, we train two variants of our model: one uses COCO captions, the other does not. In the latter case, null caption is used as input to the cross-attention layer for a fair comparison.

Fig. 4 shows the qualitative comparison. Clearly, our method generates much better image quality. For our model trained with captions, we can also specify other details such as the scene ("`A person is skiing down a snowy hill`") or person's gender ("`A woman is holding a baby`"). These two inputs complement each other and can enrich a user's controllability for image creation. We measure keypoint correspondence (similar to the YOLO score for boxes) by running a MaskRCNN [3] keypoint detector on the generated images. Both of our model

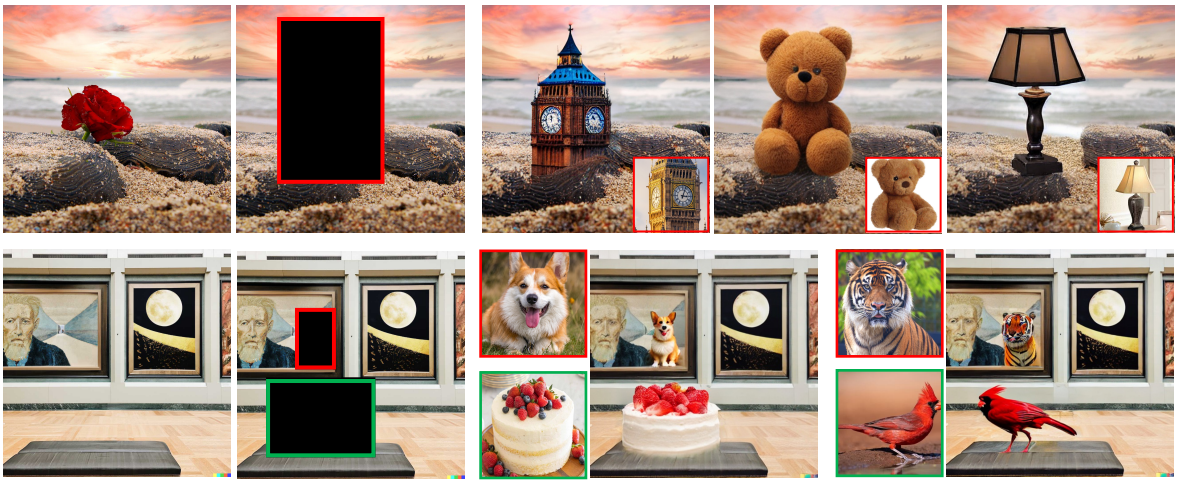


Figure 5. **Image grounded Inpainting.** One can use reference images to ground holes they want to fill in.

variants produce similar results; see Table 2.

5. Additional quantitative results

In this section, we show more studies with our pretrained model using our largest data (GoldG, O365, CC3M, SBU). We had reported this model's zero-shot performance on LVIS [2] in the main paper Table 3. Here we finetune this model on LVIS, and report its GLIP-score in Table 3. Clearly, after finetuning, we show much more accurate generation results, surpassing the supervised baseline LAMA [7] by a large margin.

Similarly, we also test this model's zero-shot performance on the COCO2017 val-set, and its finetuning results are in Table 4. The results show the benefits of pretraining which can largely improve layout correspondence performance.

6. More qualitative results and discussion

We show qualitative comparisons with layout2img baselines in Figure 6, which complements the results in Sec. 5.1 of the main paper. The results show that our model has comparable image quality when built upon LDM, but has more visual appeal and details when built upon the Stable Diffusion model.

Lastly, we show more grounded text2img results with bounding boxes in Figure 7 and other modality grounding results in Figure 8 9 10 11 12 13. Note that our keypoint model only uses keypoint annotations from COCO [8] which is not linked with person identity, but it can successfully utilize and combine the knowledge learned in the text2img training stage to control keypoints of a specific person. Out of curiosity, we also tested whether the keypoint grounding information learned on humans can be transferred to other non-humanoid categories such as cat or lamp for keypoint grounded generation, but we find that our model struggles in such cases even with scheduled sampling. Compared to

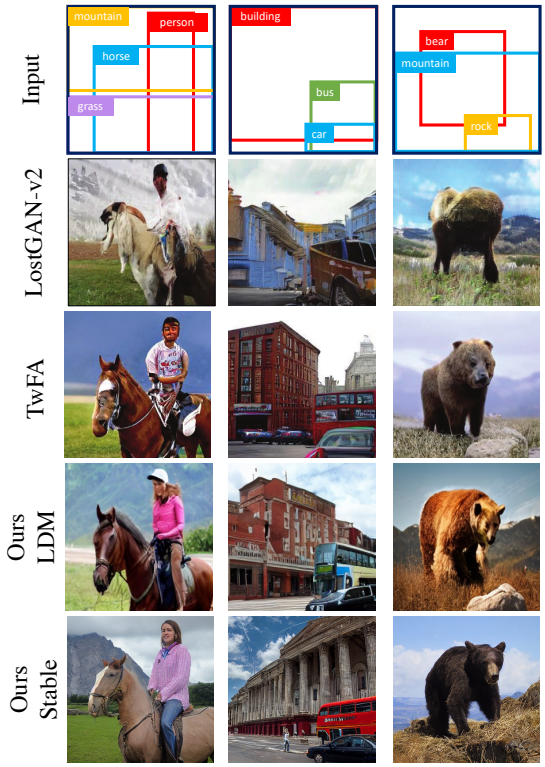


Figure 6. Layout2img comparison. Our model generates better quality images, especially when using stable diffusion. Baseline images are all copied from TwFA [17]

bounding boxes, which only specify a coarse location and size of an object in the image and thus can be shared across all object categories, keypoints (i.e., object parts) are not always shareable across different categories. Thus, while keypoints enable more fine-grained control than boxes, they are less generalizable.

Model	Pre-training data	Traing data	FID	AP	AP_r	AP_c	AP_f
LAMA [7]	–	LVIS	151.96	2.0	0.9	1.3	3.2
GLIGEN-LDM	COCO2014CD	–	22.17	6.4	5.8	5.8	7.4
GLIGEN-LDM	COCO2014D	–	31.31	4.4	2.3	3.3	6.5
GLIGEN-LDM	COCO2014G	–	13.48	6.0	4.4	6.1	6.6
GLIGEN-LDM	GoldG,O365	–	8.45	10.6	5.8	9.6	13.8
GLIGEN-LDM	GoldG,O365,SBU,CC3M	–	10.28	11.1	9.0	9.8	13.4
GLIGEN-LDM	GoldG,O365,SBU,CC3M	LVIS	**6.25**	**14.9**	**10.1**	**12.8**	**19.3**
Upper-bound	–	–	–	25.2	19.0	22.2	31.2

Table 3. GLIP-score on LVIS validation set. Upper-bound is provided by running GLIP on real images scaled to 256 × 256.

Model	FID	YOLO score AP	AP_{50}	AP_{75}
LostGAN-V2 [14]	42.55	9.1	15.3	9.8
OCGAN [15]	41.65		–	
HCSS [5]	33.68		–	
LAMA [7]	31.12	13.40	19.70	14.90
TwFA [17]	22.15	–	28.20	20.12
GLIGEN-LDM	**21.04**	22.4	36.5	24.1
After pretrain on GoldG,O365,SBU,CC3M				
GLIGEN-LDM (*zero-shot*)	27.03	19.1	30.5	20.8
GLIGEN-LDM (*finetuned*)	21.58	**30.8**	**42.3**	**35.3**

Table 4. Image quality and correspondence to layout are compared with baselines on COCO2017 val-set.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022. 2

[2] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. Lvis: A dataset for large vocabulary instance segmentation. *CVPR*, pages 5351–5359, 2019. 4

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 3

[4] Jonathan Ho. Classifier-free diffusion guidance. *ArXiv*, abs/2207.12598, 2022. 2

[5] Manuel Jahn, Robin Rombach, and Björn Ommer. High-resolution complex scene synthesis with transformers. *ArXiv*, abs/2105.06458, 2021. 5

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 2

[7] Z. Li, Jingyu Wu, Immanuel Koh, Yongchuan Tang, and Lingyun Sun. Image synthesis from layout with locality-aware mask adaption. *ICCV*, pages 13799–13808, 2021. 4, 5

[8] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 4

[9] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[10] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[11] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022. 3

[12] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CVPR*, pages 10674–10685, 2022. 1, 3

[13] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). 2

[14] Wei Sun and Tianfu Wu. Learning layout and style reconfigurable gans for controllable image synthesis. *TPAMI*, 44:5070–5087, 2022. 5

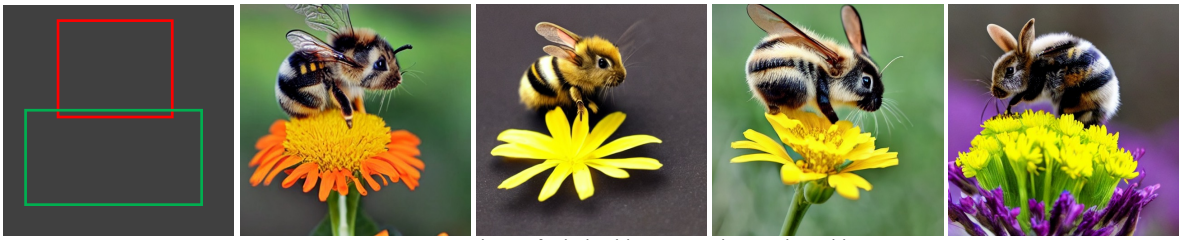
[15] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R. Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. *ArXiv*, abs/2003.07449, 2021. 5

[16] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 3

[17] Zuopeng Yang, Daqing Liu, Chaoyue Wang, J. Yang, and Dacheng Tao. Modeling image composition for complex scene generation. *CVPR*, pages 7754–7763, 2022. 4, 5

Caption: "Space view of a planet and its sun"

Grounded text: planet, sun



Caption: "a a photo of a hybrid between a bee and a rabbit"

Grounded text: hybrid between a bee and a rabbit, flower

Caption: "cartoon sketch of a little girl with a smile and balloons, old style, detailed, elegant, intricate"

Grounded text: girl with a smile, balloon, balloon, balloon

Caption: "Walter White in GTA v"

Grounded text: Walter White, car, bulldog

Caption: "two pirate ships on the ocean in minecraft"

Grounded text: a pirate ship, a pirate ship

Figure 7. Bounding box grounded text2image generation. Our model can ground noun entities in the caption for controllable image generation

Caption: “Steve Jobs is working with his laptop”
Grounded keypoints: plotted dots on the left

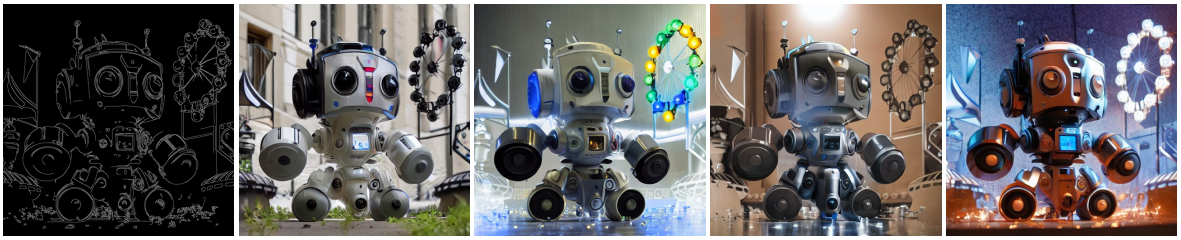
Caption: “Barack Obama is sitting at a desk”
Grounded keypoints: plotted dots on the left

Figure 8. Results for keypoints grounded generation.

Caption: “a small church is sitting in a garden”
Grounded hed map: the left image

Caption: “fox wallpaper, digit art, colorful”
Grounded hed map: the left image

Figure 9. Results for HED map grounded generation.



Caption: “A Humanoid Robot Designed for Companionship”
Grounded canny map: the left image

Caption: “a chair and a table”
Grounded canny map: the left image

Figure 10. Results for canny map grounded generation.

Caption: “a busy street with many people”
Grounded depth map: the left image

Caption: “a butterfly, ultra details”
Grounded depth map: the left image

Figure 11. Results for depth map grounded generation.

Caption: “a long hallway with pipes on the ceiling”
Grounded normal map: the left image

Caption: “the front of a building ”
Grounded normal map: the left image

Figure 12. Results for normal map grounded generation.

Caption: “a man is drawing”
Grounded semantic map: the left image

Caption: “a photo of a bedroom”
Grounded semantic map: the left image

Figure 13. Results for semantic map grounded generation.