3LG-LDM: A 3D-layout guided Latent Diffusion Model

Pu Ching, Hsiao-Wei Chen, Yen-Pin Cheng, Jie-Ying Li, Pin-Xuan Liu

1. Introduction

Image synthesis is one of the computer vision fields with the most spectacular recent development. A learned latent code or sampled noise (e.g., Gaussian) can be utilized to generate a novel image by learning the data distribution. To address the problem, especially for the high-resolution synthesis of complex, natural scenes, likelihood-based methods[1, 2] are applied to model the training data distribution; GAN-based methods[3, 4, 5] implicitly learn the distribution via an extra discriminator. Among the approaches mentioned above, diffusion models (DMs)[6] achieve an impressive synthesis result. Through the denoising autoencoder, a pure noise map can be gradually transformed into a realistic image. Recently, incredible applications built upon DMs like image synthesis[7, 8, 9, 10], super-resolution[11], colorization[8], or stroke-based synthesis[12] outperform other types of generative models. However, the diffusion process upon pixel space is computationally demanding and takes hundreds of GPU days. Aiming to reduce the calculation complexity while maintaining the high sampling quality, latent diffusion[13] is proposed to operate the diffusion process on the pretrained latent space. Except for the dimension reduction on the denoising autoencoder, proposed cross-attention layers provide a flexible solution to fuse multi-modality conditions. An interesting application built upon this structure is that the image synthesis can be conditioned on a 2D-layout given labels. Nevertheless, based on our observations, a 2D-layout lacks information about the orientation and depth of the generated object images, e.g. the horse faces in different directions in Fig. 1. In this work , we' re going to propose a 3D-layout guided latent diffusion model making use of the 3D structures to synthesize 2D images. We believe such improvement may provide an interface for applications like a virtual tour or synthetic data generation.

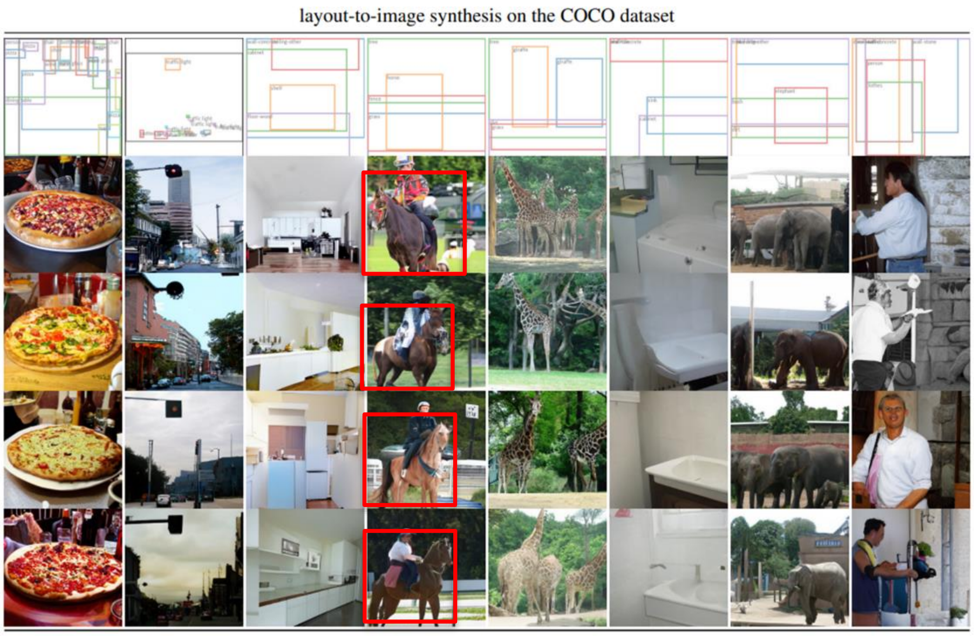


Fig.1 Layout-to-image synthesis results from LDM

2. Technical part

In practical terms, we' ll separate the whole framework into 2 stages: 1) 3D object detection and 2) Latent diffusion.

For the initial experiments on 3D object detection, we' re going to apply the ground truth 3D bounding box annotated from the Objectron dataset[14]. To explore the effects of noisy bounding boxes, we' ll also apply the on-the-shelf detector[15] to estimate the scales, rotation, and translation of detected objects on the COCO dataset[16]. Given the 3D bounding box, we can obtain a projected 2D layout map (See Fig. 2) that implies the 3D geometry of each object. Following the preprocessing mentioned above, we can automatically prepare the paired data between the ground truth images and the layout maps. Note that we discovered that providing the whole 3D representation (e.g., voxel) may be costly on memory space, so we assume that using the projected map can provide sufficient information but is computationally efficient.

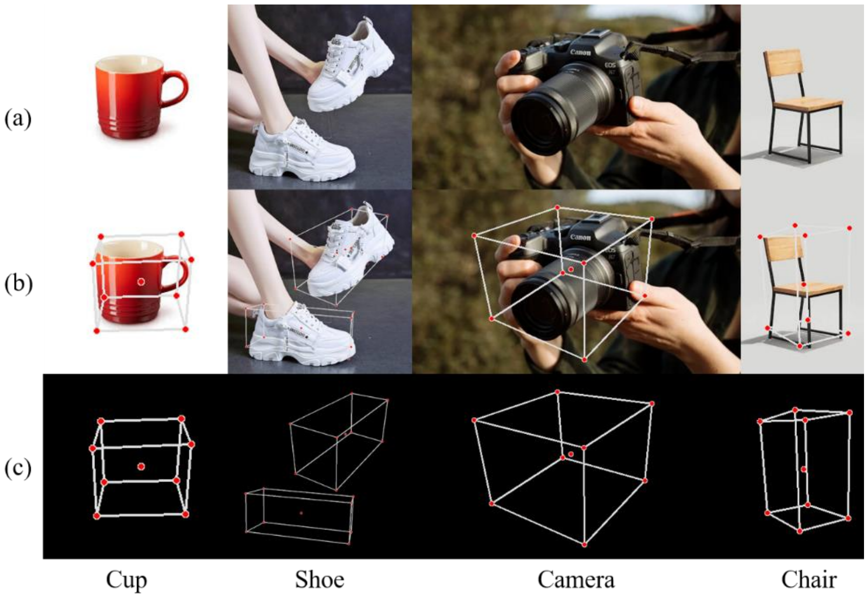


Fig.2 projected 3D layout map, (a) original image. (b) annotated image. (c) 3D layout map.

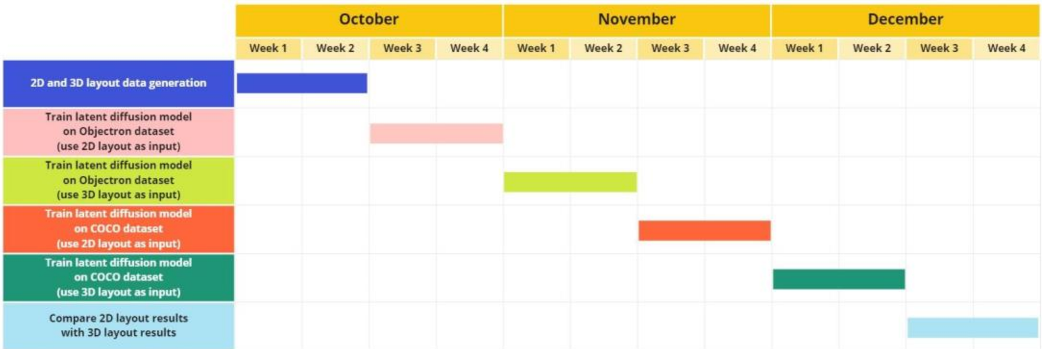
Conditioned on the processed data, we can project the layout map into an embedding vector using τ_θ and then pass it to the cross-attention layers implementing Attention$(Q, K, V) =$ softmax $(\frac{QK^T}{\sqrt{d}}) \cdot$ V, with $Q = W_Q^{(i)} \cdot \rho_i(z_t), K = W_K^{(i)} \cdot \tau_\theta(y), V = W_V^{(i)} \cdot \tau_\theta(y)$. The detailed loss function is listed in equation 1, where ϵ is the sampled noise, ϵ_θ is our noise estimator, t is the timestep, z_t is the latent map at timestep t, and y is our layout map. To implement the complete LDM structure, here we choose LDM-8 as our pretrained image autoencoder.

$$L_{LDM} := \mathbb{E}_{\varepsilon(x), y, \epsilon \sim N(0,1), t}[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2]. \quad (1)$$

3. Milestones

First, we get the 2D and 3D bounding box ground-truth data from the Objectron dataset to generate 2D and 3D layout data. Since the COCO dataset has no ground-truth data, we will use 2D and 3D object detectors to create 2D and 3D layout data. Next, we will train latent diffusion model s on the 2D and 3D layout data datasets. There will be four models that need to be trained. Finally, we will compare the results of 2D and 3D layout models to check if the 3D layout data help genera te images with objects' orientation and depth. So our milestones will be set as follows :

1. 2D and 3D layout data generation
2. Train latent diffusion model on Objectron dataset(use the 2D layout as input)
3. Train latent diffusion model on Objectron dataset(use the 3D layout as input)
4. Train latent diffusion model on COCO dataset(use the 2D layout as input)
5. Train latent diffusion model on COCO dataset(use the 3D layout as input)
6. Compare 2D layout results with 3D layout results



4. References

[1]Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark C hen, and Ilya Sutskever. Zero-shot text-to-image generation. CoRR, abs/2102.12092, 2021.

[2]Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images wit h VQ-VAE-2. In NeurIPS, pages 14837 – 14847, 2019.

[3]Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In Int. Conf. Learn. Represent., 2019.

[4]Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Oz air, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. CoRR, 2014.

[5]Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative a dversarial networks. In IEEE Conf. Comput. Vis. Pattern Recog., pages 4401 – 4410, 2019.

[6]Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupe rvised learning using nonequilibrium thermodynamics. CoRR, abs/1503.03585, 2015.

[7]Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In NeurIP S, 2020.

[8]Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Scorebased generative modeling through stochastic differential equations. CoRR, abs/2 011.13456, 2020.

[9]Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. CoRR, abs/2105.05233, 2021.
[10]Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. CoRR, abs/2106.15282, 2021.
[11]Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. CoRR, abs/2104.07636, 2021.
[12]Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. CoRR, abs/2108.01073, 2021.
[13]Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-¨ resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684 – 10695, 2022.
[14]Adel Ahmadyan, Liangkai Zhang, Jianing Wei, Artsiom Ablavatski, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. arXiv preprint arXiv:2012.09988, 2020.
[15] HOU, Tingbo, et al. Mobilepose: Real-time pose estimation for unseen objects with weak shape supervision. arXiv preprint arXiv:2003.03522, 2020.
[16] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." European conference on computer vision. Springer, Cham, 2014.