SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis

Objective

Improve Stable Diffusion!

Architecture & Scale

Model	SDXL	SD 1.4/1.5	SD 2.0/2.1
# of UNet Params	2.6B	860M	865M
Transformer blocks	[0, 2, 10]	[1, 1, 1, 1]	[1, 1, 1, 1]
Channel mult.	[1, 2, 4]	[1, 2, 4, 4]	[1, 2, 4, 4]
Text encoder	CLIP ViT-L & OpenCLIP ViT-bigG	CLIP ViT-L	OpenCLIP ViT-H
Context dim.	2048	768	1024
Pooled text emb.	OpenCLIP ViT-bigG	N/A	N/A

Table 1: Architecture comparision

Micro-Conditioning (Image Size)

Latent diffusion requires *minimal image size*. To cope with this one typically do one of the followings:

1. Discard images that do not meet the requirement: e.g. Stable Diffusion ($\sim$512 pixels)
2. Upscale images that are too small

However, each has the following shortcoming respectively

1. May discard large portion of the data
2. Introduce unwanted artifacts

Micro-Conditioning (Image Size)

Instead authors propose to *condition* the UNet with the original image size $\mathbf{c}_{\text{size}} = (h_{\text{orig}}, w_{\text{orig}})$. Then at the inference one can choose the *apparent resolution* of the generating image the size conditioning.

Micro-Conditioning (Image Size)

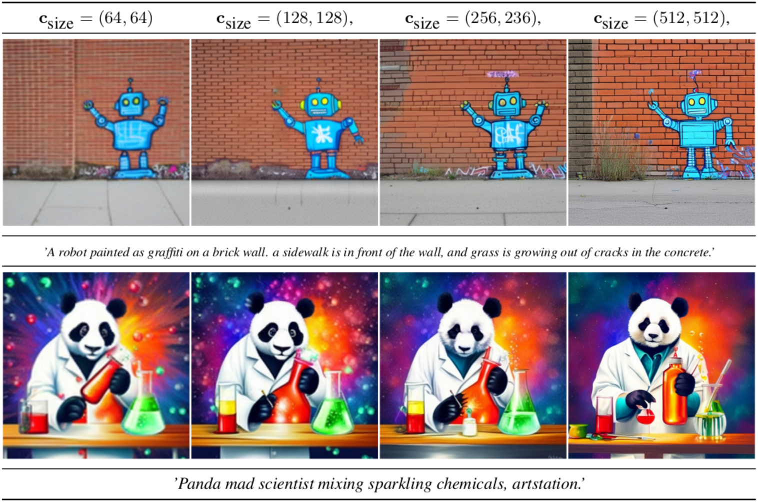

Figure 1: The effect of size conditioning

Micro-Conditioning (Image Size)

model	FID-5k↓	IS-5k↑
CIN-512-only	43.84	110.64
CIN-nocond	39.76	211.50
CIN-size-cond	36.53	215.34

Table 2: Class conditional ImageNet

Micro-Conditioning (Cropping Parameters)

Typical collating of the batch in deep learning includes following procedure:

1. resize an image so that the shortest size match the desired target size
2. randomly crop the image along the longer axis

However, such cropping could have unwanted effect on the actual image generation.

Micro-Conditioning (Cropping Parameters)

Figure 2: Cropping affecting the generated images

Micro-Conditioning (Cropping Parameters)

To cope with this authors propose to *condition* the UNet with $\mathbf{c}_{\text{crop}} = (c_{\text{top}}, c_{\text{left}})$ where $c_{\text{top}}, c_{\text{left}}$ indicate the numbers of pixels cropped from the top-left corner along height and width axes respectively. At inference, one can set $\mathbf{c}_{\text{crop}} = (0, 0)$.

Micro-Conditioning (Cropping Parameters)

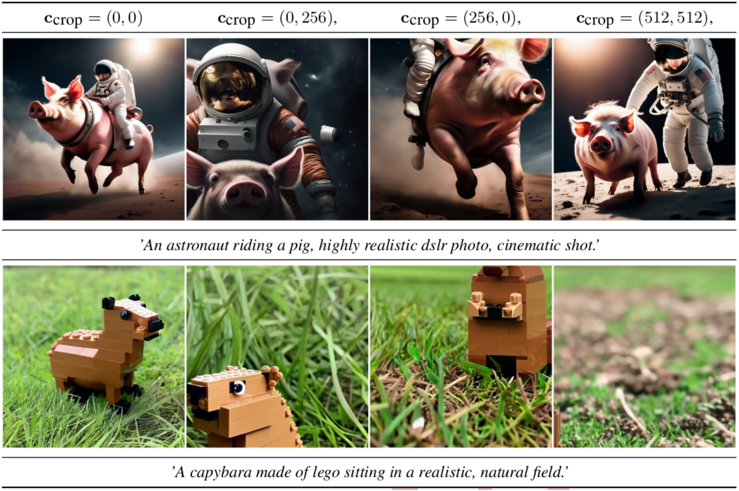

Figure 3: The effect of crop conditioning

Multi-Aspect Training

The real world dataset contains images of various aspect-ratio. However, the typical diffusion models only generate square images. To cope with this the author propose to finetune the model with images with different aspect ratios (with pixel counts as close to 1024^2 as possible:

1. Partition the dataset into buckets of different aspect ratios
2. Randomly choose a bucket, and compose a batch with images from that bucket
3. Condition the model with $\mathbf{c}_{\text{ar}} = (h_{\text{tgt}}, w_{\text{tgt}})$

Improved Autoencoder

The performance of the autoencoder affects the local and high frequency details in generation. To further improve the autoencoder, the authors train the autoencoder with larger batch-size (256 vs 9), and use the exponential moving average.

model	PNSR↑	SSIM↑	LPIPS↓	rFID↓
SDXL-VAE	24.7	0.73	0.88	4.4
SD-VAE 1.x	23.4	0.69	0.96	5.0
SD-VAE 2.x	24.5	0.71	0.92	4.7

Table 3: Autoencoder performance

Multi-Stage Optimization

1. Pretrain the baseline model with the internal dataset at resolution 256x256 with size and crop conditioning
2. Finetune with 512x512 images
3. Multi-aspect training
4. Train refinement model

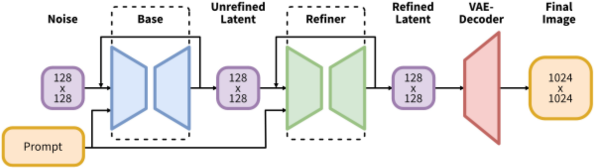

Figure 4: Framework

Refinement

Figure 5: Effect of refinement

Thank You

Q & A