Appendix

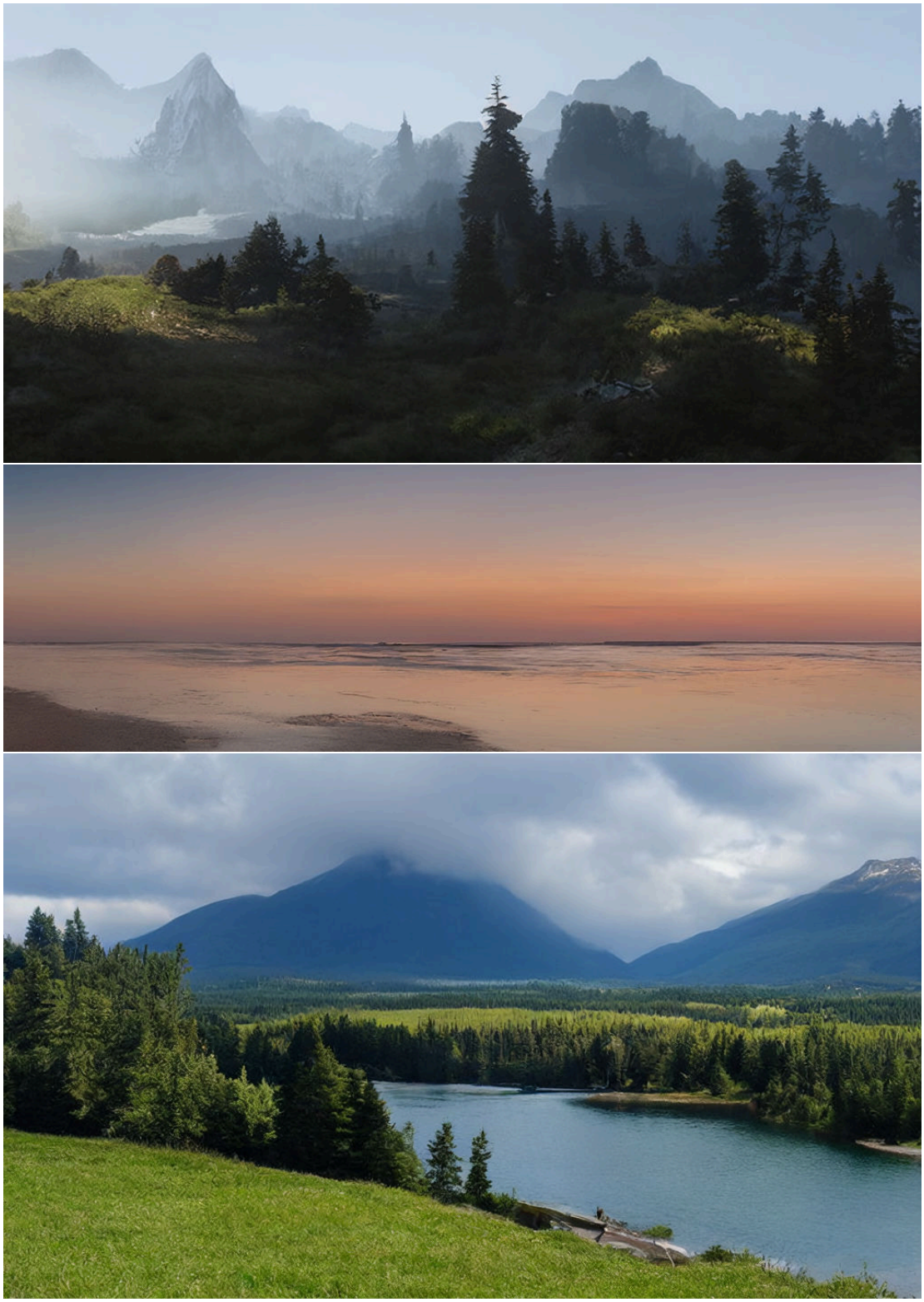

Figure 11. Convolutional samples from the semantic landscapes model as in Sec. 4.3.2, finetuned on 512^2 images.

A. Detailed Information on Denoising Diffusion Models

Diffusion models can be specified in terms of a signal-to-noise ratio $\text{SNR}(t) = \frac{\alpha_t^2}{\sigma_t^2}$ consisting of sequences $(\alpha_t)_{t=1}^T$ and $(\sigma_t)_{t=1}^T$ which, starting from a data sample x_0, define a forward diffusion process q as

$$q(x_t|x_0) = \mathcal{N}(x_t|\alpha_t x_0, \sigma_t^2 \mathbb{I}) \tag{4}$$

with the Markov structure for $s < t$:

$$q(x_t|x_s) = \mathcal{N}(x_t|\alpha_{t|s} x_s, \sigma_{t|s}^2 \mathbb{I}) \tag{5}$$

$$\alpha_{t|s} = \frac{\alpha_t}{\alpha_s} \tag{6}$$

$$\sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}^2 \sigma_s^2 \tag{7}$$

Denoising diffusion models are generative models $p(x_0)$ which revert this process with a similar Markov structure running backward in time, *i.e.* they are specified as

$$p(x_0) = \int_z p(x_T) \prod_{t=1}^T p(x_{t-1}|x_t) \tag{8}$$

The evidence lower bound (ELBO) associated with this model then decomposes over the discrete time steps as

$$-\log p(x_0) \leq \mathbb{KL}(q(x_T|x_0)|p(x_T)) + \sum_{t=1}^T \mathbb{E}_{q(x_t|x_0)} \mathbb{KL}(q(x_{t-1}|x_t, x_0)|p(x_{t-1}|x_t)) \tag{9}$$

The prior $p(x_T)$ is typically choosen as a standard normal distribution and the first term of the ELBO then depends only on the final signal-to-noise ratio $\text{SNR}(T)$. To minimize the remaining terms, a common choice to parameterize $p(x_{t-1}|x_t)$ is to specify it in terms of the true posterior $q(x_{t-1}|x_t, x_0)$ but with the unknown x_0 replaced by an estimate $x_\theta(x_t, t)$ based on the current step x_t. This gives [44]

$$p(x_{t-1}|x_t) \coloneqq q(x_{t-1}|x_t, x_\theta(x_t, t)) \tag{10}$$

$$= \mathcal{N}(x_{t-1}|\mu_\theta(x_t, t), \sigma_{t|t-1}^2 \frac{\sigma_{t-1}^2}{\sigma_t^2} \mathbb{I}), \tag{11}$$

where the mean can be expressed as

$$\mu_\theta(x_t, t) = \frac{\alpha_{t|t-1}\sigma_{t-1}^2}{\sigma_t^2} x_t + \frac{\alpha_{t-1}\sigma_{t|t-1}^2}{\sigma_t^2} x_\theta(x_t, t). \tag{12}$$

In this case, the sum of the ELBO simplify to

$$\sum_{t=1}^T \mathbb{E}_{q(x_t|x_0)} \mathbb{KL}(q(x_{t-1}|x_t, x_0)|p(x_{t-1}) = \sum_{t=1}^T \mathbb{E}_{\mathcal{N}(\epsilon|0,\mathbb{I})} \frac{1}{2} (\text{SNR}(t-1) - \text{SNR}(t)) \|x_0 - x_\theta(\alpha_t x_0 + \sigma_t \epsilon, t)\|^2 \tag{13}$$

Following [29], we use the reparameterization

$$\epsilon_\theta(x_t, t) = (x_t - \alpha_t x_\theta(x_t, t))/\sigma_t \tag{14}$$

to express the reconstruction term as a denoising objective,

$$\|x_0 - x_\theta(\alpha_t x_0 + \sigma_t \epsilon, t)\|^2 = \frac{\sigma_t^2}{\alpha_t^2} \|\epsilon - \epsilon_\theta(\alpha_t x_0 + \sigma_t \epsilon, t)\|^2 \tag{15}$$

and the reweighting, which assigns each of the terms the same weight and results in Eq. (1).

B. Image Guiding Mechanisms

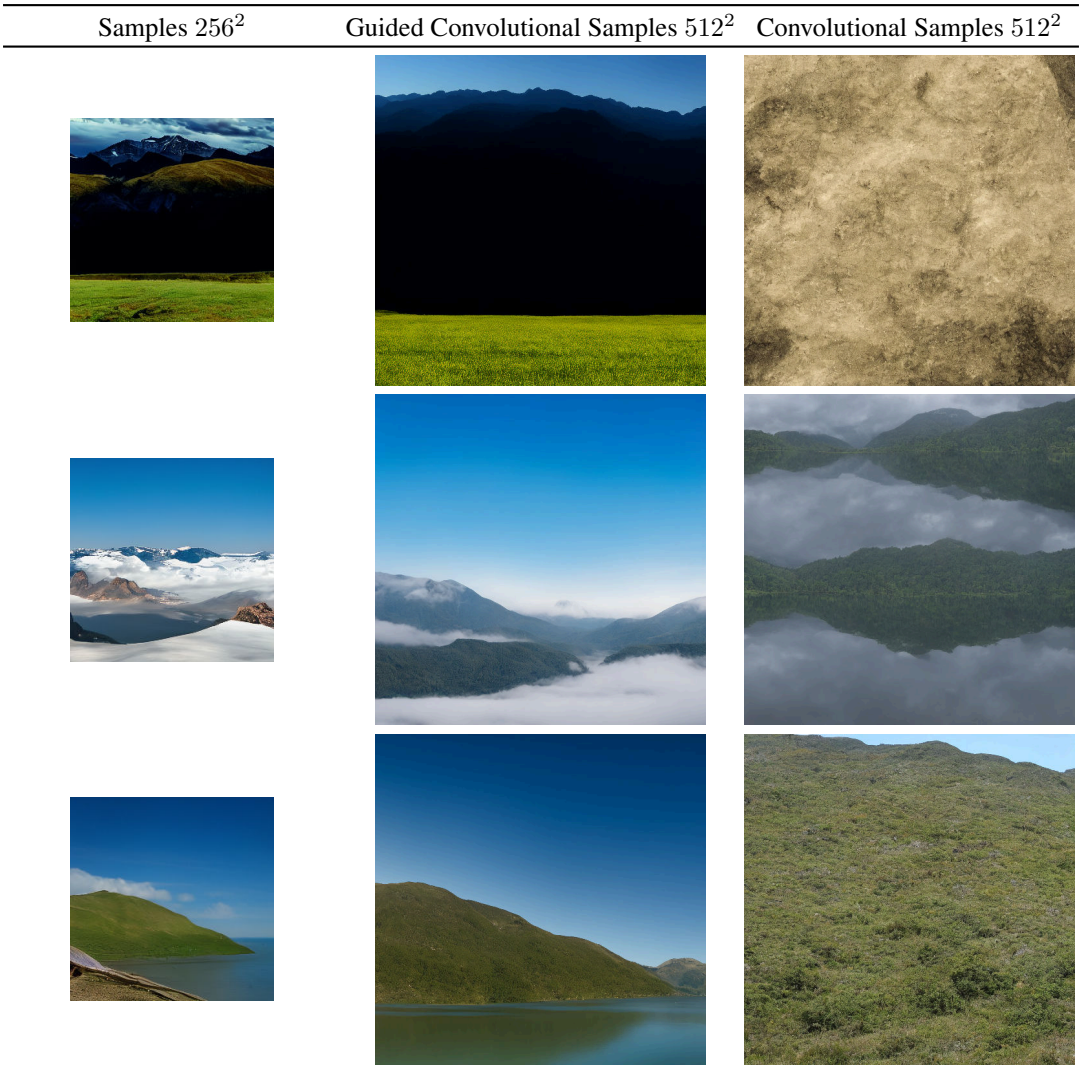

Figure 12. On landscapes, convolutional sampling with unconditional models can lead to homogeneous and incoherent global structures (see column 2). L_2-guiding with a low resolution image can help to reestablish coherent global structures.

An intriguing feature of diffusion models is that unconditional models can be conditioned at test-time [15, 79, 82]. In particular, [15] presented an algorithm to guide both unconditional and conditional models trained on the ImageNet dataset with a classifier $\log p_\Phi(y|x_t)$, trained on each x_t of the diffusion process. We directly build on this formulation and introduce post-hoc *image-guiding*:

For an epsilon-parameterized model with fixed variance, the guiding algorithm as introduced in [15] reads:

$$\hat{\epsilon} \leftarrow \epsilon_\theta(z_t, t) + \sqrt{1-\alpha_t^2}\, \nabla_{z_t} \log p_\Phi(y|z_t)\ . \tag{16}$$

This can be interpreted as an update correcting the "score" ϵ_θ with a conditional distribution $\log p_\Phi(y|z_t)$.

So far, this scenario has only been applied to single-class classification models. We re-interpret the guiding distribution $p_\Phi(y|T(\mathcal{D}(z_0(z_t))))$ as a general purpose image-to-image translation task given a target image y, where T can be any differentiable transformation adopted to the image-to-image translation task at hand, such as the identity, a downsampling operation or similar.

As an example, we can assume a Gaussian guider with fixed variance $\sigma^2 = 1$, such that

$$\log p_\Phi(y|z_t) = -\frac{1}{2}\|y - T(\mathcal{D}(z_0(z_t)))\|_2^2 \tag{17}$$

becomes a L_2 regression objective.

Fig. 12 demonstrates how this formulation can serve as an upsampling mechanism of an unconditional model trained on 256^2 images, where unconditional samples of size 256^2 guide the convolutional synthesis of 512^2 images and T is a $2\times$ bicubic downsampling. Following this motivation, we also experiment with a perceptual similarity guiding and replace the L_2 objective with the LPIPS [102] metric, see Sec. 4.4.

C. Additional Results

C.1. Choosing the Signal-to-Noise Ratio for High-Resolution Synthesis

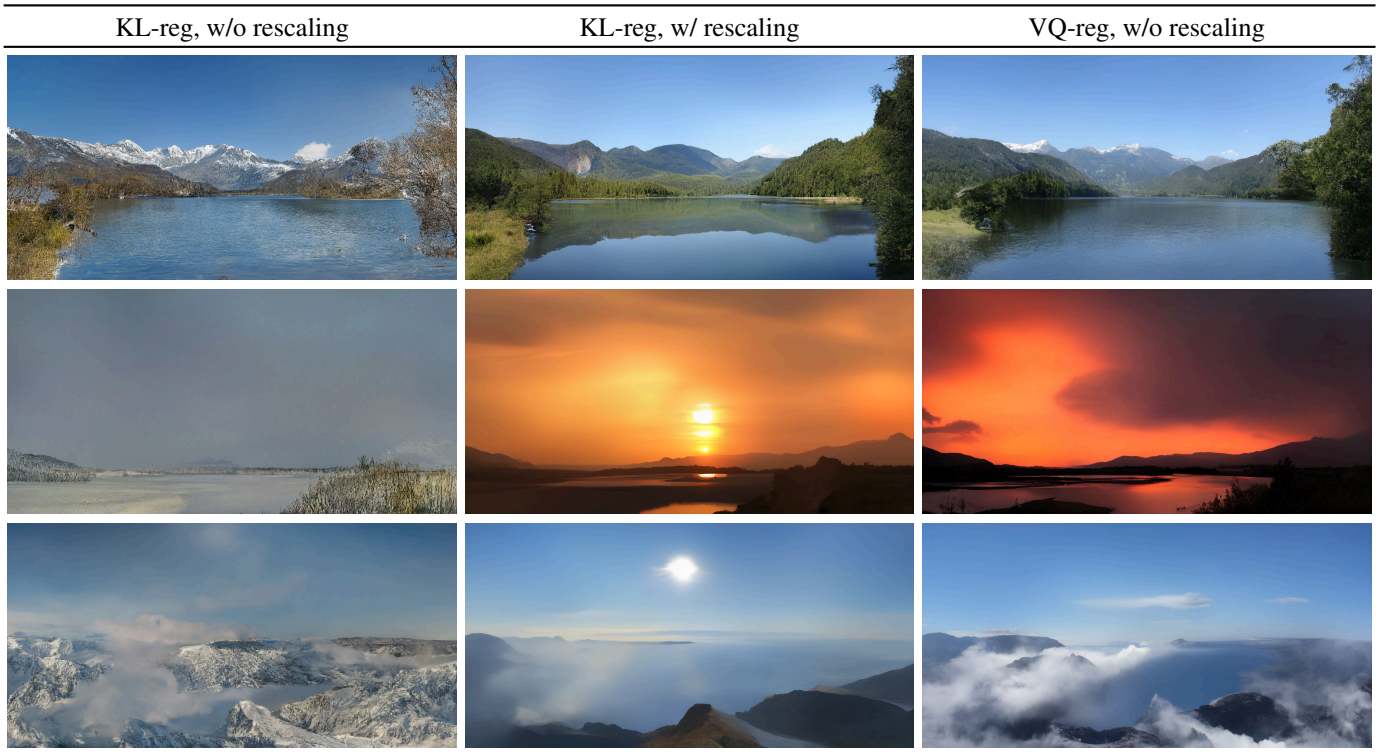

Figure 13. Illustrating the effect of latent space rescaling on convolutional sampling, here for semantic image synthesis on landscapes. See Sec. 4.3.2 and Sec. C.1.

As discussed in Sec. 4.3.2, the signal-to-noise ratio induced by the variance of the latent space (*i.e.* $\mathrm{Var}(z)/\sigma_t^2$) significantly affects the results for convolutional sampling. For example, when training a LDM directly in the latent space of a KL-regularized model (see Tab. 8), this ratio is very high, such that the model allocates a lot of semantic detail early on in the reverse denoising process. In contrast, when rescaling the latent space by the component-wise standard deviation of the latents as described in Sec. F, the SNR is descreased. We illustrate the effect on convolutional sampling for semantic image synthesis in Fig. 13. Note that the VQ-regularized space has a variance close to 1, such that it does not have to be rescaled.

C.2. Full List of all First Stage Models

We provide a complete list of various autoenconding models trained on the OpenImages dataset in Tab. 8.

C.3. Text-to-Image Synthesis

In Fig. 14 we show additional samples from our best text-to-image model for user defined text prompts. For a detailed description of the conditioning mechanism via cross-attention, *cf*. Sec D.2.1.

C.4. Layout-to-Image Synthesis

Here we provide the quantitative evaluation and additional samples for our layout-to-image models from Sec. 4.3.1. We train a model on the COCO [4] and one on the OpenImages [48] dataset, which we subsequently additionally finetune on COCO. Tab 9 shows the result. Our COCO model reaches the performance of recent state-of-the art models in layout-to-image synthesis, when following their training and evaluation protocol [86]. When finetuning from the OpenImages model, we surpass these works. Our OpenImages model surpasses the results of Jahn et al [36] by a margin of nearly 11 in terms of FID. In Fig. 15 we show additional samples of the model finetuned on COCO.

f	$\lvert\mathcal{Z}\rvert$	c	**R-FID** ↓	**R-IS** ↑	**PSNR** ↑	**PSIM** ↓	**SSIM** ↑
16 *VQGAN* [23]	16384	256	4.98	–	19.9 ±3.4	1.83 ±0.42	0.51 ±0.18
16 *VQGAN* [23]	1024	256	7.94	–	19.4 ±3.3	1.98 ±0.43	0.50 ±0.18
8 *DALL-E* [64]	8192	-	32.01	–	22.8 ±2.1	1.95 ±0.51	0.73 ±0.13
32	16384	16	31.83	40.40 ±1.07	17.45 ±2.90	2.58 ±0.48	0.41 ±0.18
16	16384	8	5.15	144.55 ±3.74	20.83 ±3.61	1.73 ±0.43	0.54 ±0.18
8	16384	4	1.14	201.92 ±3.97	23.07 ±3.99	1.17 ±0.36	0.65 ±0.16
8	256	4	1.49	194.20 ±3.87	22.35 ±3.81	1.26 ±0.37	0.62 ±0.16
4	8192	3	0.58	224.78 ±5.35	27.43 ±4.26	0.53 ±0.21	0.82 ±0.10
4†	8192	3	1.06	221.94 ±4.58	25.21 ±4.17	0.72 ±0.26	0.76 ±0.12
4	256	3	0.47	223.81 ±4.58	26.43 ±4.22	0.62 ±0.24	0.80 ±0.11
2	2048	2	0.16	232.75 ±5.09	30.85 ±4.12	0.27 ±0.12	0.91 ±0.05
2	64	2	0.40	226.62 ±4.83	29.13 ±3.46	0.38 ±0.13	0.90 ±0.05
32	KL	64	2.04	189.53 ±3.68	22.27 ±3.93	1.41 ±0.40	0.61 ±0.17
32	KL	16	7.3	132.75 ±2.71	20.38 ±3.56	1.88 ±0.45	0.53 ±0.18
16	KL	16	0.87	210.31 ±3.97	24.08 ±4.22	1.07 ±0.36	0.68 ±0.15
16	KL	8	2.63	178.68 ±4.08	21.94 ±3.92	1.49 ±0.42	0.59 ±0.17
8	KL	4	0.90	209.90 ±4.92	24.19 ±4.19	1.02 ±0.35	0.69 ±0.15
4	KL	3	0.27	227.57 ±4.89	27.53 ±4.54	0.55 ±0.24	0.82 ±0.11
2	KL	2	0.086	232.66 ±5.16	32.47 ±4.19	0.20 ±0.09	0.93 ±0.04

Table 8. Complete autoencoder zoo trained on OpenImages, evaluated on ImageNet-Val. † denotes an attention-free autoencoder.

	COCO256 × 256	OpenImages 256 × 256	OpenImages 512 × 512
Method	FID↓	FID↓	FID↓
LostGAN-V2 [84]	42.55	-	-
OC-GAN [86]	41.65	-	-
SPADE [60]	41.11	-	-
VQGAN+T [36]	56.58	45.33	48.11
LDM-8 (100 steps, ours)	42.06†	-	-
LDM-4 (200 steps, ours)	**40.91***	**32.02**	**35.80**

Table 9. Quantitative comparison of our layout-to-image models on the COCO [4] and OpenImages [48] datasets. †: Training from scratch on COCO; *: Finetuning from OpenImages.

C.5. Class-Conditional Image Synthesis on ImageNet

Tab. 10 contains the results for our class-conditional LDM measured in FID and Inception score (IS). LDM-8 requires significantly fewer parameters and compute requirements (see Tab. 18) to achieve very competitive performance. Similar to previous work, we can further boost the performance by training a classifier on each noise scale and guiding with it, see Sec. B. Unlike the pixel-based methods, this classifier is trained very cheaply in latent space. For additional qualitative results, see Fig. 25 and Fig. 26.

C.6. Sample Quality vs. V100 Days (Continued from Sec. 4.1)

For the assessment of sample quality over the training progress in Sec. 4.1, we reported FID and IS scores as a function of train steps. Another possibility is to report these metrics over the used resources in V100 days. Such an analysis is additionally provided in Fig. 16, showing qualitatively similar results.

C.7. Super-Resolution

For better comparability between LDMs and diffusion models in pixel space, we extend our analysis from Tab. 4 by comparing a diffusion model trained for the same number of steps and with a comparable number [1] of parameters to our LDM. The results of this comparison are shown in the last two rows of Tab. 11 and demonstrate that LDM achieves better performance while allowing for significantly faster sampling. A qualitative comparison is given in Fig. 19 which shows random samples from both LDM and the diffusion model in pixel space.

[1] It is not possible to exactly match both architectures since the diffusion model operates in the pixel space

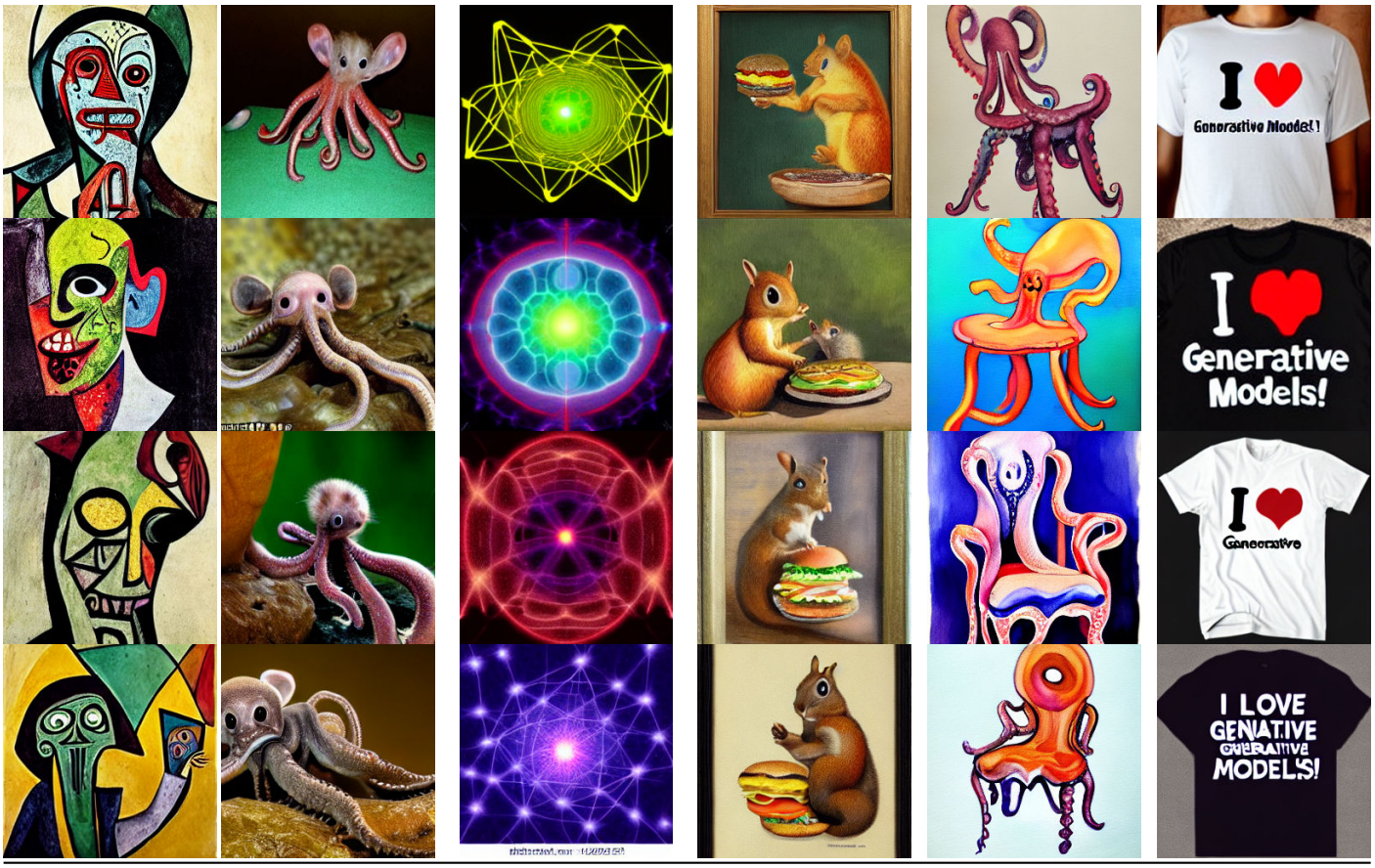

Figure 14. More samples for user-defined text prompts from our big model for text-to-image synthesis, *LDM-8 (KL)*, which was trained on the LAION database. Samples generated with 200 DDIM steps and $\eta = 1.0$. We use unconditional guidance [31] with $s = 10.0$.

Method	FID↓	IS↑	Precision↑	Recall↑	Nparams	
SR3 [70]	11.30	-	-	-	625M	-
ImageBART [21]	21.19	-	-	-	3.5B	-
ImageBART [21]	7.44	-	-	-	3.5B	0.05 acc. rate*
VQGAN+T [23]	17.04	70.6$\pm$1.8	-	-	1.3B	-
VQGAN+T [23]	5.88	**304.8**$\pm$3.6	-	-	1.3B	0.05 acc. rate*
BigGan-deep [3]	6.95	203.6$\pm$2.6	**0.87**	0.28	340M	-
ADM [15]	10.94	100.98	0.69	**0.63**	554M	250 DDIM steps
ADM-G [15]	4.59	186.7	0.82	0.52	608M	250 DDIM steps
ADM-G,ADM-U [15]	3.85	221.72	0.84	0.53	n/a	2 $\times$ 250 DDIM steps
CDM [30]	4.88	158.71$\pm$2.26	-	-	n/a	2 $\times$ 100 DDIM steps
LDM-8 (ours)	17.41	72.92$\pm$2.6	0.65	0.62	395M	200 DDIM steps, 2.9M train steps, batch size 64
LDM-8-G (ours)	8.11	190.43$\pm$2.60	0.83	0.36	506M	200 DDIM steps, classifier scale 10, 2.9M train steps, batch size 64
LDM-8 (ours)	15.51	79.03$\pm$1.03	0.65	**0.63**	395M	200 DDIM steps, 4.8M train steps, batch size 64
LDM-8-G (ours)	7.76	209.52$\pm$4.24	0.84	0.35	506M	200 DDIM steps, classifier scale 10, 4.8M train steps, batch size 64
LDM-4 (ours)	10.56	103.49$\pm$1.24	0.71	0.62	400M	250 DDIM steps, 178K train steps, batch size 1200
LDM-4-G (ours)	**3.95**	178.22$\pm$2.43	0.81	0.55	400M	250 DDIM steps, unconditional guidance [31] scale 1.25, 178K train steps, batch size 1200
LDM-4-G (ours)	**3.60**	247.67$\pm$5.59	**0.87**	0.48	400M	250 DDIM steps, unconditional guidance [31] scale 1.5, 178K train steps, batch size 1200

Table 10. Comparison of a class-conditional ImageNet *LDM* with recent state-of-the-art methods for class-conditional image generation on the ImageNet [12] dataset.*: Classifier rejection sampling with the given rejection rate as proposed in [65].

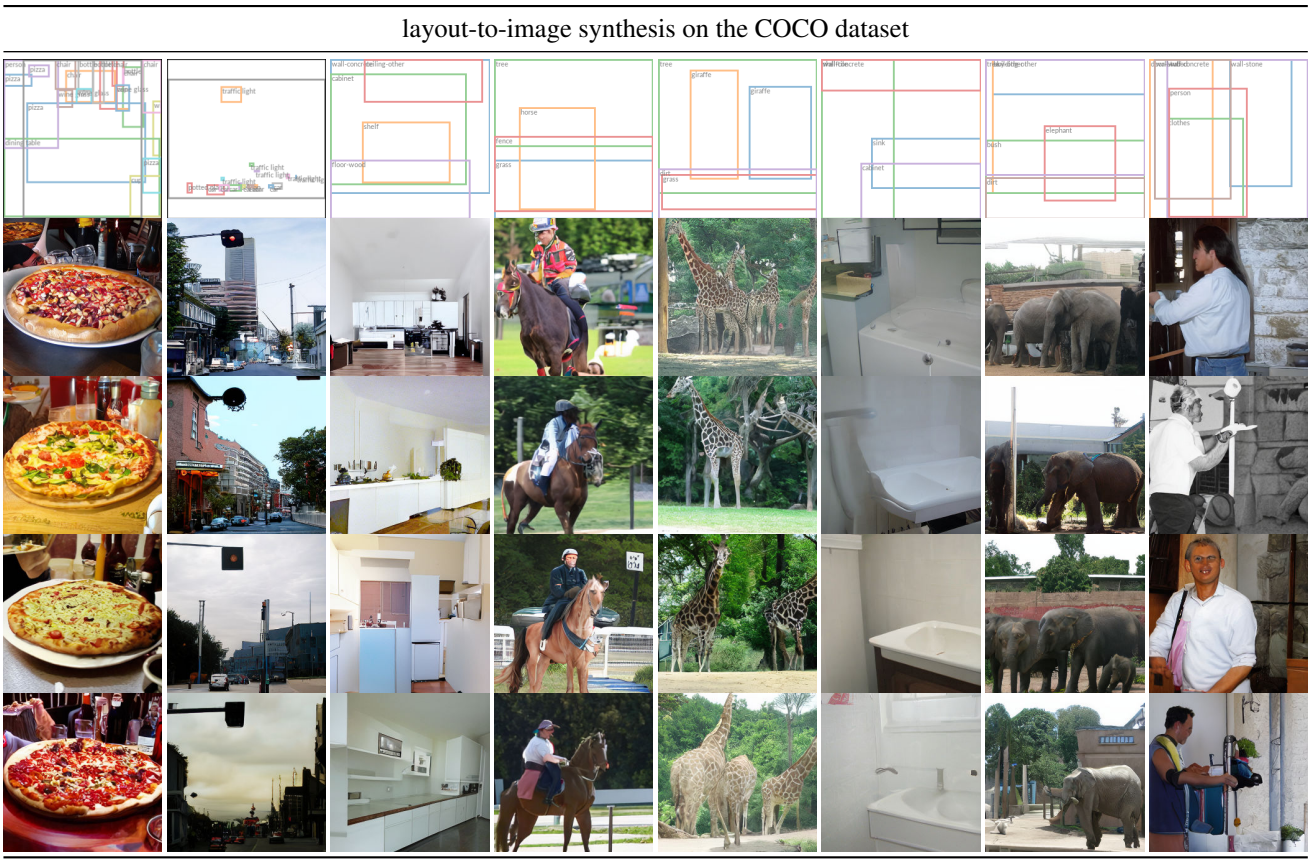

Figure 15. More samples from our best model for layout-to-image synthesis, *LDM-4*, which was trained on the OpenImages dataset and finetuned on the COCO dataset. Samples generated with 100 DDIM steps and $\eta = 0$. Layouts are from the COCO validation set.

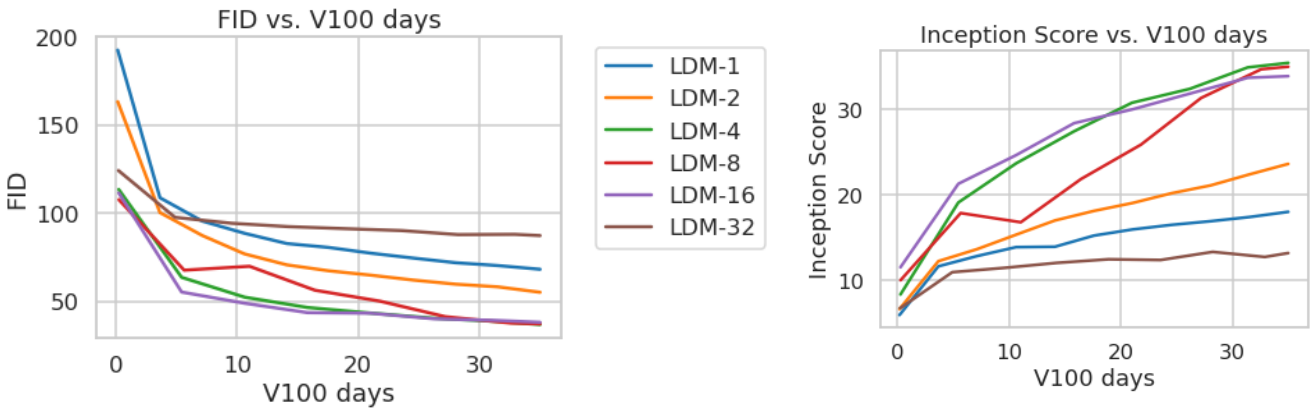

Figure 16. For completeness we also report the training progress of class-conditional *LDMs* on the ImageNet dataset for a fixed number of 35 V100 days. Results obtained with 100 DDIM steps [81] and $\kappa = 0$. FIDs computed on 5000 samples for efficiency reasons.

C.7.1 LDM-BSR: General Purpose SR Model via Diverse Image Degradation

To evaluate generalization of our LDM-SR, we apply it both on synthetic LDM samples from a class-conditional ImageNet model (Sec. 4.1) and images crawled from the internet. Interestingly, we observe that LDM-SR, trained only with a bicubicly downsampled conditioning as in [70], does not generalize well to images which do not follow this pre-processing. Hence, to obtain a superresolution model for a wide range of real world images, which can contain complex superpositions of camera

Method	FID ↓	IS ↑	PSNR ↑	SSIM ↑
Image Regression [70]	15.2	121.1	**27.9**	**0.801**
SR3 [70]	5.2	**180.1**	26.4	0.762
LDM-4 (ours, 100 steps)	**2.8**†/**4.8**‡	166.3	24.4$\pm$3.8	0.69$\pm$0.14
LDM-4 (ours, 50 steps, guiding)	4.4†/6.4‡	153.7	25.8$\pm$3.7	0.74$\pm$0.12
LDM-4 (ours, 100 steps, guiding)	4.4†/6.4‡	154.1	25.7$\pm$3.7	0.73$\pm$0.12
LDM-4 (ours, 100 steps, +15 ep.)	**2.6**† / **4.6**‡	169.76$\pm$5.03	24.4$\pm$3.8	0.69$\pm$0.14
Pixel-DM (100 steps, +15 ep.)	5.1† / 7.1‡	163.06$\pm$4.67	24.1$\pm$3.3	0.59$\pm$0.12

Table 11. $\times 4$ upscaling results on ImageNet-Val. (256^2); †: FID features computed on validation split, ‡: FID features computed on train split. We also include a pixel-space baseline that receives the same amount of compute as *LDM-4*. The last two rows received 15 epochs of additional training compared to the former results.

Figure 17. *LDM-BSR* generalizes to arbitrary inputs and can be used as a general-purpose upsampler, upscaling samples from a class-conditional *LDM* (image *cf*. Fig. 4) to 1024^2 resolution. In contrast, using a fixed degradation process (see Sec. 4.4) hinders generalization.

noise, compression artifacts, blurr and interpolations, we replace the bicubic downsampling operation in LDM-SR with the degration pipeline from [101]. The BSR-degradation process is a degradation pipline which applies JPEG compressions noise, camera sensor noise, different image interpolations for downsampling, Gaussian blur kernels and Gaussian noise in a random order to an image. We found that using the bsr-degredation process with the original parameters as in [101] leads to a very strong degradation process. Since a more moderate degradation process seemed apppropiate for our application, we adapted the parameters of the bsr-degradation (our adapted degradation process can be found in our code base at `https://github.com/CompVis/latent-diffusion`). Fig. 17 illustrates the effectiveness of this approach by directly comparing *LDM-SR* with *LDM-BSR*. The latter produces images much sharper than the models confined to a fixed pre-processing, making it suitable for real-world applications. Further results of *LDM-BSR* are shown on LSUN-cows in Fig. 18.

D. Implementation Details and Hyperparameters

D.1. Hyperparameters

We provide an overview of the hyperparameters of all trained *LDM* models in Tab. 12, Tab. 13, Tab. 14 and Tab. 15.

	CelebA-HQ 256×256	FFHQ 256×256	LSUN-Churches 256×256	LSUN-Bedrooms 256×256
f	4	4	8	4
z-shape	$64 \times 64 \times 3$	$64 \times 64 \times 3$	-	$64 \times 64 \times 3$
$\lvert\mathcal{Z}\rvert$	8192	8192	-	8192
Diffusion steps	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear
N_{params}	274M	274M	294M	274M
Channels	224	224	192	224
Depth	2	2	2	2
Channel Multiplier	1,2,3,4	1,2,3,4	1,2,2,4,4	1,2,3,4
Attention resolutions	32, 16, 8	32, 16, 8	32, 16, 8, 4	32, 16, 8
Head Channels	32	32	24	32
Batch Size	48	42	96	48
Iterations*	410k	635k	500k	1.9M
Learning Rate	9.6e-5	8.4e-5	5.e-5	9.6e-5

Table 12. Hyperparameters for the unconditional *LDMs* producing the numbers shown in Tab. 1. All models trained on a single NVIDIA A100.

	LDM-1	*LDM-2*	*LDM-4*	*LDM-8*	*LDM-16*	*LDM-32*
z-shape	$256 \times 256 \times 3$	$128 \times 128 \times 2$	$64 \times 64 \times 3$	$32 \times 32 \times 4$	$16 \times 16 \times 8$	$88 \times 8 \times 32$
$\lvert\mathcal{Z}\rvert$	-	2048	8192	16384	16384	16384
Diffusion steps	1000	1000	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear	linear	linear
Model Size	396M	391M	391M	395M	395M	395M
Channels	192	192	192	256	256	256
Depth	2	2	2	2	2	2
Channel Multiplier	1,1,2,2,4,4	1,2,2,4,4	1,2,3,5	1,2,4	1,2,4	1,2,4
Number of Heads	1	1	1	1	1	1
Batch Size	7	9	40	64	112	112
Iterations	2M	2M	2M	2M	2M	2M
Learning Rate	4.9e-5	6.3e-5	8e-5	6.4e-5	4.5e-5	4.5e-5
Conditioning	CA	CA	CA	CA	CA	CA
CA-resolutions	32, 16, 8	32, 16, 8	32, 16, 8	32, 16, 8	16, 8, 4	8, 4, 2
Embedding Dimension	512	512	512	512	512	512
Transformers Depth	1	1	1	1	1	1

Table 13. Hyperparameters for the conditional *LDMs* trained on the ImageNet dataset for the analysis in Sec. 4.1. All models trained on a single NVIDIA A100.

D.2. Implementation Details

D.2.1 Implementations of τ_θ for conditional *LDMs*

For the experiments on text-to-image and layout-to-image (Sec. 4.3.1) synthesis, we implement the conditioner τ_θ as an unmasked transformer which processes a tokenized version of the input y and produces an output $\zeta := \tau_\theta(y)$, where $\zeta \in \mathbb{R}^{M \times d_\tau}$. More specifically, the transformer is implemented from N transformer blocks consisting of global self-attention layers, layer-normalization and position-wise MLPs as follows[2]:

[2]adapted from https://github.com/lucidrains/x-transformers

	LDM-1	*LDM-2*	*LDM-4*	*LDM-8*	*LDM-16*	*LDM-32*
z-shape	$256 \times 256 \times 3$	$128 \times 128 \times 2$	$64 \times 64 \times 3$	$32 \times 32 \times 4$	$16 \times 16 \times 8$	$88 \times 8 \times 32$
$\lvert\mathcal{Z}\rvert$	-	2048	8192	16384	16384	16384
Diffusion steps	1000	1000	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear	linear	linear
Model Size	270M	265M	274M	258M	260M	258M
Channels	192	192	224	256	256	256
Depth	2	2	2	2	2	2
Channel Multiplier	1,1,2,2,4,4	1,2,2,4,4	1,2,3,4	1,2,4	1,2,4	1,2,4
Attention resolutions	32, 16, 8	32, 16, 8	32, 16, 8	32, 16, 8	16, 8, 4	8, 4, 2
Head Channels	32	32	32	32	32	32
Batch Size	9	11	48	96	128	128
Iterations*	500k	500k	500k	500k	500k	500k
Learning Rate	9e-5	1.1e-4	9.6e-5	9.6e-5	1.3e-4	1.3e-4

Table 14. Hyperparameters for the unconditional *LDMs* trained on the CelebA dataset for the analysis in Fig. 6. All models trained on a single NVIDIA A100. *: All models are trained for 500k iterations. If converging earlier, we used the best checkpoint for assessing the provided FID scores.

Task	Text-to-Image	Layout-to-Image		Class-Label-to-Image	Super Resolution	Inpainting	Semantic-Map-to-Image
Dataset	LAION	OpenImages	COCO	ImageNet	ImageNet	Places	Landscapes
f	8	4	8	4	4	4	8
z-shape	$32 \times 32 \times 4$	$64 \times 64 \times 3$	$32 \times 32 \times 4$	$64 \times 64 \times 3$	$64 \times 64 \times 3$	$64 \times 64 \times 3$	$32 \times 32 \times 4$
$\lvert\mathcal{Z}\rvert$	-	8192	16384	8192	8192	8192	16384
Diffusion steps	1000	1000	1000	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear	linear	linear	linear
Model Size	1.45B	306M	345M	395M	169M	215M	215M
Channels	320	128	192	192	160	128	128
Depth	2	2	2	2	2	2	2
Channel Multiplier	1,2,4,4	1,2,3,4	1,2,4	1,2,3,5	1,2,2,4	1,4,8	1,4,8
Number of Heads	8	1	1	1	1	1	1
Dropout	-	-	0.1	-	-	-	-
Batch Size	680	24	48	1200	64	128	48
Iterations	390K	4.4M	170K	178K	860K	360K	360K
Learning Rate	1.0e-4	4.8e-5	4.8e-5	1.0e-4	6.4e-5	1.0e-6	4.8e-5
Conditioning	CA	CA	CA	CA	concat	concat	concat
(C)A-resolutions	32, 16, 8	32, 16, 8	32, 16, 8	32, 16, 8	-	-	-
Embedding Dimension	1280	512	512	512	-	-	-
Transformer Depth	1	3	2	1	-	-	-

Table 15. Hyperparameters for the conditional *LDMs* from Sec. 4. All models trained on a single NVIDIA A100 except for the inpainting model which was trained on eight V100.

$$\zeta \leftarrow \text{TokEmb}(y) + \text{PosEmb(y)} \tag{18}$$

$$\text{for } i = 1, \dots, N:$$

$$\zeta_1 \leftarrow \text{LayerNorm}(\zeta) \tag{19}$$

$$\zeta_2 \leftarrow \text{MultiHeadSelfAttention}(\zeta_1) + \zeta \tag{20}$$

$$\zeta_3 \leftarrow \text{LayerNorm}(\zeta_2) \tag{21}$$

$$\zeta \leftarrow \text{MLP}(\zeta_3) + \zeta_2 \tag{22}$$

$$\zeta \leftarrow \text{LayerNorm}(\zeta) \tag{23}$$

$$\tag{24}$$

With ζ available, the conditioning is mapped into the UNet via the cross-attention mechanism as depicted in Fig. 3. We modify the "ablated UNet" [15] architecture and replace the self-attention layer with a shallow (unmasked) transformer consisting of T blocks with alternating layers of (i) self-attention, (ii) a position-wise MLP and (iii) a cross-attention layer;

see Tab. 16. Note that without (ii) and (iii), this architecture is equivalent to the "ablated UNet".

While it would be possible to increase the representational power of τ_θ by additionally conditioning on the time step t, we do not pursue this choice as it reduces the speed of inference. We leave a more detailed analysis of this modification to future work.

For the text-to-image model, we rely on a publicly available[3] tokenizer [95]. The layout-to-image model discretizes the spatial locations of the bounding boxes and encodes each box as a (l, b, c)-tuple, where l denotes the (discrete) top-left and b the bottom-right position. Class information is contained in c.
See Tab. 17 for the hyperparameters of τ_θ and Tab. 13 for those of the UNet for both of the above tasks.

Note that the class-conditional model as described in Sec. 4.1 is also implemented via cross-attention, where τ_θ is a single learnable embedding layer with a dimensionality of 512, mapping classes y to $\zeta \in \mathbb{R}^{1\times 512}$.

input	$\mathbb{R}^{h\times w\times c}$
LayerNorm	$\mathbb{R}^{h\times w\times c}$
Conv1x1	$\mathbb{R}^{h\times w\times d\cdot n_h}$
Reshape	$\mathbb{R}^{h\cdot w\times d\cdot n_h}$
$\times T$ { SelfAttention	$\mathbb{R}^{h\cdot w\times d\cdot n_h}$
$\times T$ { MLP	$\mathbb{R}^{h\cdot w\times d\cdot n_h}$
$\times T$ { CrossAttention	$\mathbb{R}^{h\cdot w\times d\cdot n_h}$
Reshape	$\mathbb{R}^{h\times w\times d\cdot n_h}$
Conv1x1	$\mathbb{R}^{h\times w\times c}$

Table 16. Architecture of a transformer block as described in Sec. D.2.1, replacing the self-attention layer of the standard "ablated UNet" architecture [15]. Here, n_h denotes the number of attention heads and d the dimensionality per head.

	Text-to-Image	**Layout-to-Image**
seq-length	77	92
depth N	32	16
dim	1280	512

Table 17. Hyperparameters for the experiments with transformer encoders in Sec. 4.3.

D.2.2 Inpainting

For our experiments on image-inpainting in Sec. 4.5, we used the code of [85] to generate synthetic masks. We use a fixed set of 2k validation and 30k testing samples from Places [104]. During training, we use random crops of size 256×256 and evaluate on crops of size 512×512. This follows the training and testing protocol in [85] and reproduces their reported metrics (see † in Tab. 7). We include additional qualitative results of *LDM-4, w/ attn* in Fig. 20 and of *LDM-4, w/o attn, big, w/ ft* in Fig. 21.

D.3. Evaluation Details

This section provides additional details on evaluation for the experiments shown in Sec. 4.

D.3.1 Quantitative Results in Unconditional and Class-Conditional Image Synthesis

We follow common practice and estimate the statistics for calculating the FID-, Precision- and Recall-scores [28,49] shown in Tab. 1 and 10 based on 50k samples from our models and the entire training set of each of the shown datasets. For calculating FID scores we use the `torch-fidelity` package [58]. However, since different data processing pipelines might lead to different results [62], we also evaluate our models with the script provided by Dhariwal and Nichol [15]. We find that results

[3] https://huggingface.co/transformers/model_doc/bert.html#berttokenizerfast

mainly coincide, except for the ImageNet and LSUN-Bedrooms datasets, where we notice slightly varying scores of 7.76 (`torch-fidelity`) vs. 7.77 (Nichol and Dhariwal) and 2.95 vs 3.0. For the future we emphasize the importance of a unified procedure for sample quality assessment. Precision and Recall are also computed by using the script provided by Nichol and Dhariwal.

D.3.2 Text-to-Image Synthesis

Following the evaluation protocol of [64] we compute FID and Inception Score for the Text-to-Image models from Tab. 2 by comparing generated samples with 30000 samples from the validation set of the MS-COCO dataset [50]. FID and Inception Scores are computed with `torch-fidelity`.

D.3.3 Layout-to-Image Synthesis

For assessing the sample quality of our Layout-to-Image models from Tab. 9 on the COCO dataset, we follow common practice [36, 84, 86] and compute FID scores the 2048 unaugmented examples of the COCO Segmentation Challenge split. To obtain better comparability, we use the exact same samples as in [36]. For the OpenImages dataset we similarly follow their protocol and use 2048 center-cropped test images from the validation set.

D.3.4 Super Resolution

We evaluate the super-resolution models on ImageNet following the pipeline suggested in [70], *i.e*. images with a shorter size less than 256 px are removed (both for training and evaluation). On ImageNet, the low-resolution images are produced using bicubic interpolation with anti-aliasing. FIDs are evaluated using `torch-fidelity` [58], and we produce samples on the validation split. For FID scores, we additionally compare to reference features computed on the train split, see Tab. 4 and Tab. 11.

D.3.5 Efficiency Analysis

For efficiency reasons we compute the sample quality metrics plotted in Fig. 5, 16 and 6 based on 5k samples. Therefore, the results might vary from those shown in Tab. 1 and 10. All models have a comparable number of parameters as provided in Tab. 13 and 14. We maximize the learning rates of the individual models such that they still train stably. Therefore, the learning rates slightly vary between different runs *cf*. Tab. 13 and 14.

D.3.6 User Study

For the results of the user study presented in Tab. 5 we followed the protocoll of [70] and and use the 2-alternative force-choice paradigm to assess human preference scores for two distinct tasks. In Task-1 subjects were shown a low resolution/masked image between the corresponding ground truth high resolution/unmasked version and a synthesized image, which was generated by using the middle image as conditioning. For SuperResolution subjects were asked: *'Which of the two images is a better high quality version of the low resolution image in the middle?'*. For Inpainting we asked *'Which of the two images contains more realistic inpainted regions of the image in the middle?'*. In Task-2, humans were similarly shown the low-res/masked version and asked for preference between two corresponding images generated by the two competing methods. As in [70] humans viewed the images for 3 seconds before responding.

E. Computational Requirements

Method	Generator Compute	Classifier Compute	Overall Compute	Inference Throughput*	N_{params}	FID↓	IS↑	Precision↑	Recall↑
LSUN Churches 256^2									
StyleGAN2 [41]†	64	-	64	-	59M	3.86	-	-	-
LDM-8 (ours, 100 steps, 410K)	18	-	18	6.80	256M	4.02	-	0.64	0.52
LSUN Bedrooms 256^2									
ADM [15]† (1000 steps)	232	-	232	0.03	552M	1.9	-	0.66	0.51
LDM-4 (ours, 200 steps, 1.9M)	60	-	55	1.07	274M	2.95	-	0.66	0.48
CelebA-HQ 256^2									
LDM-4 (ours, 500 steps, 410K)	14.4	-	14.4	0.43	274M	5.11	-	0.72	0.49
FFHQ 256^2									
StyleGAN2 [41]	32.13‡	-	32.13†	-	59M	3.8	-	-	-
LDM-4 (ours, 200 steps, 635K)	26	-	26	1.07	274M	4.98	-	0.73	0.50
ImageNet 256^2									
VQGAN-f-4 (ours, first stage)	29	-	29	-	55M	0.58††	-	-	-
VQGAN-f-8 (ours, first stage)	66	-	66	-	68M	1.14††	-	-	-
BigGAN-deep [3]†	128-256		128-256	-	340M	6.95	$203.6_{\pm 2.6}$	0.87	0.28
ADM [15] (250 steps) †	916	-	916	0.12	554M	10.94	100.98	0.69	0.63
ADM-G [15] (25 steps) †	916	46	962	0.7	608M	5.58	-	0.81	0.49
ADM-G [15] (250 steps)†	916	46	962	0.07	608M	4.59	186.7	0.82	0.52
ADM-G,ADM-U [15] (250 steps)†	329	30	349	n/a	n/a	3.85	221.72	0.84	0.53
LDM-8-G (ours, 100, 2.9M)	79	12	91	1.93	506M	8.11	$190.4_{\pm 2.6}$	0.83	0.36
LDM-8 (ours, 200 ddim steps 2.9M, batch size 64)	79	-	79	1.9	395M	17.41	72.92	0.65	0.62
LDM-4 (ours, 250 ddim steps 178K, batch size 1200)	271	-	271	0.7	400M	10.56	$103.49_{\pm 1.24}$	0.71	0.62
LDM-4-G (ours, 250 ddim steps 178K, batch size 1200, classifier-free guidance [31] scale 1.25)	271	-	271	0.4	400M	3.95	$178.22_{\pm 2.43}$	0.81	0.55
LDM-4-G (ours, 250 ddim steps 178K, batch size 1200, classifier-free guidance [31] scale 1.5)	271	-	271	0.4	400M	3.60	$247.67_{\pm 5.59}$	0.87	0.48

Table 18. Comparing compute requirements during training and inference throughput with state-of-the-art generative models. Compute during training in V100-days, numbers of competing methods taken from [15] unless stated differently;*: Throughput measured in samples/sec on a single NVIDIA A100;†: Numbers taken from [15] ;‡: Assumed to be trained on 25M train examples; ††: R-FID vs. ImageNet validation set

In Tab 18 we provide a more detailed analysis on our used compute ressources and compare our best performing models on the CelebA-HQ, FFHQ, LSUN and ImageNet datasets with the recent state of the art models by using their provided numbers, *cf*. [15]. As they report their used compute in V100 days and we train all our models on a single NVIDIA A100 GPU, we convert the A100 days to V100 days by assuming a $\times 2.2$ speedup of A100 vs V100 [72][4]. To assess sample quality, we additionally report FID scores on the reported datasets. We closely reach the performance of state of the art methods as StyleGAN2 [41] and ADM [15] while significantly reducing the required compute resources.

[4]This factor corresponds to the speedup of the A100 over the V100 for a U-Net, as defined in Fig. 1 in [72]

F. Details on Autoencoder Models

We train all our autoencoder models in an adversarial manner following [23], such that a patch-based discriminator D_ψ is optimized to differentiate original images from reconstructions $\mathcal{D}(\mathcal{E}(x))$. To avoid arbitrarily scaled latent spaces, we regularize the latent z to be zero centered and obtain small variance by introducing an regularizing loss term L_{reg}.
We investigate two different regularization methods: (i) a low-weighted Kullback-Leibler-term between $q_\mathcal{E}(z|x) = \mathcal{N}(z; \mathcal{E}_\mu, \mathcal{E}_{\sigma^2})$ and a standard normal distribution $\mathcal{N}(z; 0, 1)$ as in a standard variational autoencoder [45, 67], and, (ii) regularizing the latent space with a vector quantization layer by learning a codebook of $|\mathcal{Z}|$ different exemplars [93].
To obtain high-fidelity reconstructions we only use a very small regularization for both scenarios, *i.e.* we either weight the $\mathbb{KL}$ term by a factor $\sim 10^{-6}$ or choose a high codebook dimensionality $|\mathcal{Z}|$.

The full objective to train the autoencoding model $(\mathcal{E}, \mathcal{D})$ reads:

$$L_{\text{Autoencoder}} = \min_{\mathcal{E},\mathcal{D}} \max_{\psi} \Big(L_{rec}(x, \mathcal{D}(\mathcal{E}(x))) - L_{adv}(\mathcal{D}(\mathcal{E}(x))) + \log D_\psi(x) + L_{reg}(x; \mathcal{E}, \mathcal{D}) \Big) \tag{25}$$

DM Training in Latent Space Note that for training diffusion models on the learned latent space, we again distinguish two cases when learning $p(z)$ or $p(z|y)$ (Sec. 4.3): (i) For a KL-regularized latent space, we sample $z = \mathcal{E}_\mu(x) + \mathcal{E}_\sigma(x) \cdot \varepsilon =: \mathcal{E}(x)$, where $\varepsilon \sim \mathcal{N}(0, 1)$. When rescaling the latent, we estimate the component-wise variance

$$\hat{\sigma}^2 = \frac{1}{bchw} \sum_{b,c,h,w} (z^{b,c,h,w} - \hat{\mu})^2$$

from the first batch in the data, where $\hat{\mu} = \frac{1}{bchw} \sum_{b,c,h,w} z^{b,c,h,w}$. The output of $\mathcal{E}$ is scaled such that the rescaled latent has unit standard deviation, *i.e.* $z \leftarrow \frac{z}{\hat{\sigma}} = \frac{\mathcal{E}(x)}{\hat{\sigma}}$. (ii) For a VQ-regularized latent space, we extract z *before* the quantization layer and absorb the quantization operation into the decoder, *i.e.* it can be interpreted as the first layer of $\mathcal{D}$.

G. Limitations & Societal Impact

Limitations While LDMs significantly reduce computational requirements compared to pixel-based approaches, their sequential sampling process is still slower than that of GANs. Moreover, the use of LDMs can be questionable when high precision is required: although the loss of image quality is very small in our $f = 4$ autoencoding models (see Fig. 1), their reconstruction capability can become a bottleneck for tasks that require fine-grained accuracy in pixel space. We assume that our superresolution models (Sec. 4.4) are already somewhat limited in this respect.

Societal Impact Generative models for media like imagery are a double-edged sword: On the one hand, they enable various creative applications, and in particular approaches like ours that reduce the cost of training and inference have the potential to facilitate access to this technology and democratize its exploration. On the other hand, it also means that it becomes easier to create and disseminate manipulated data or spread misinformation and spam. In particular, the deliberate manipulation of images ("deep fakes") is a common problem in this context, and women in particular are disproportionately affected by it [13, 24].

Generative models can also reveal their training data [5, 87], which is of great concern when the data contain sensitive or personal information and were collected without explicit consent. However, the extent to which this also applies to DMs of images is not yet fully understood.

Finally, deep learning modules tend to reproduce or exacerbate biases that are already present in the data [22, 37, 88]. While diffusion models achieve better coverage of the data distribution than *e.g.* GAN-based approaches, the extent to which our two-stage approach that combines adversarial training and a likelihood-based objective misrepresents the data remains an important research question.

For a more detailed discussion of the ethical considerations of deep generative models, see *e.g.* [13].

H. Additional Qualitative Results

Finally, we provide additional qualitative results for our landscapes model (Fig. 11, 22, 23 and 24), our class-conditional ImageNet model (Fig. 25 - 26) and our unconditional models for the CelebA-HQ, FFHQ and LSUN datasets (Fig. 27 - 30). Similar as for the inpainting model in Sec. 4.5 we also fine-tuned the semantic landscapes model from Sec. 4.3.2 directly on 512^2 images and depict qualitative results in Fig. 11 and Fig. 22. For our those models trained on comparably small datasets, we additionally show nearest neighbors in VGG [77] feature space for samples from our models in Fig. 31 - 33.

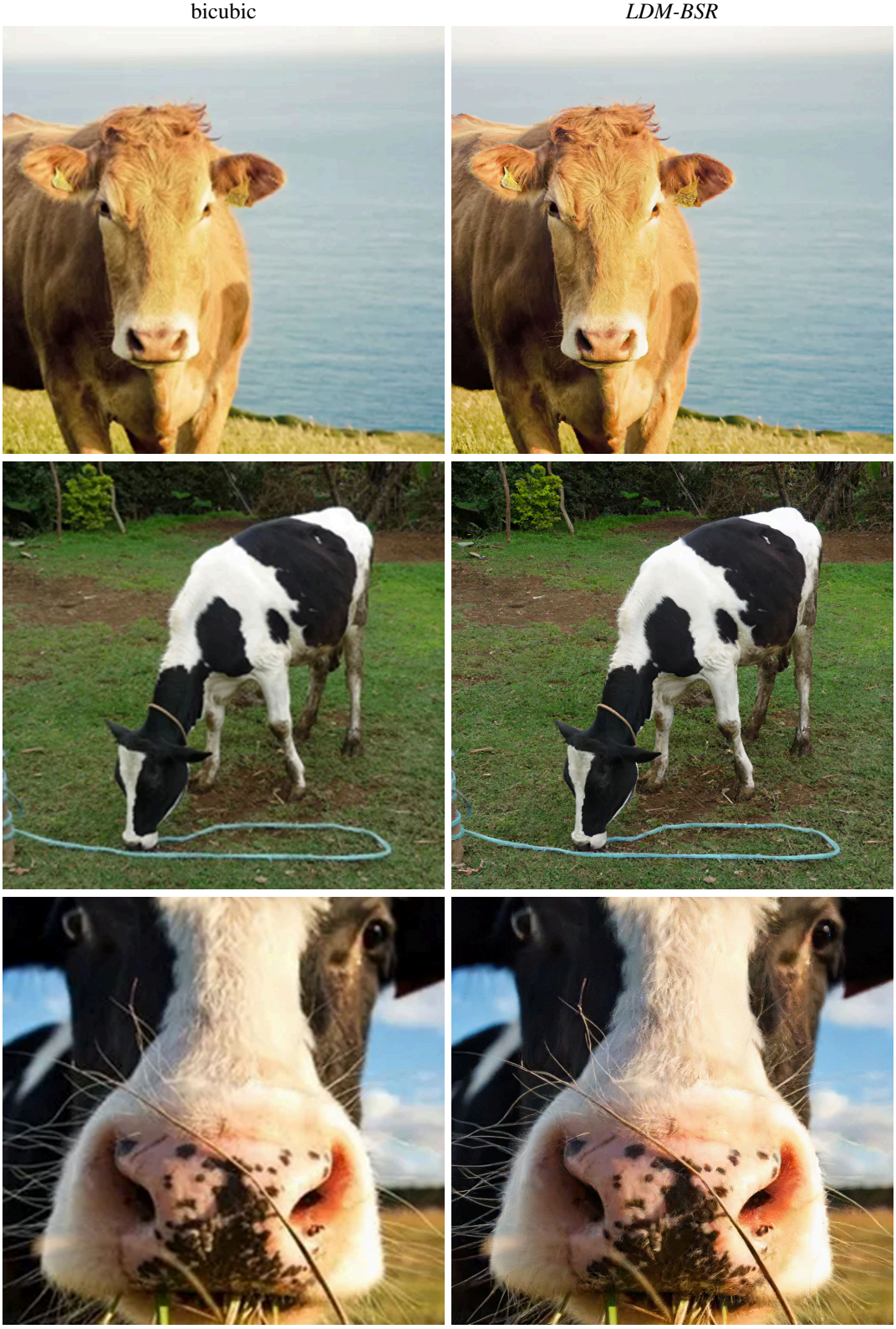

Figure 18. *LDM-BSR* generalizes to arbitrary inputs and can be used as a general-purpose upsampler, upscaling samples from the LSUN-Cows dataset to 1024^2 resolution.

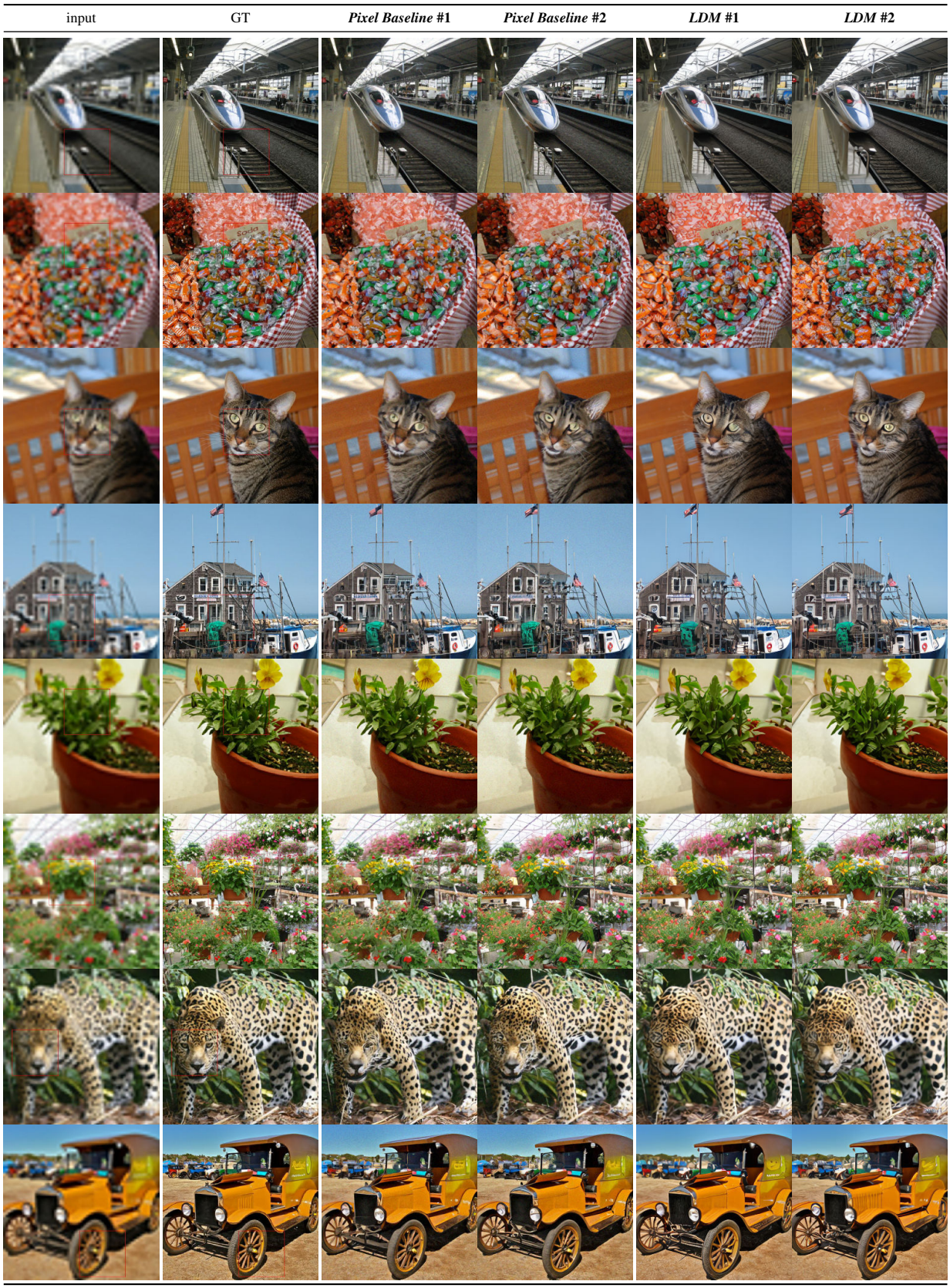

Figure 19. Qualitative superresolution comparison of two random samples between LDM-SR and baseline-diffusionmodel in Pixelspace. Evaluated on imagenet validation-set after same amount of training steps.

Figure 20. Qualitative results on image inpainting. In contrast to [85], our generative approach enables generation of multiple diverse samples for a given input.

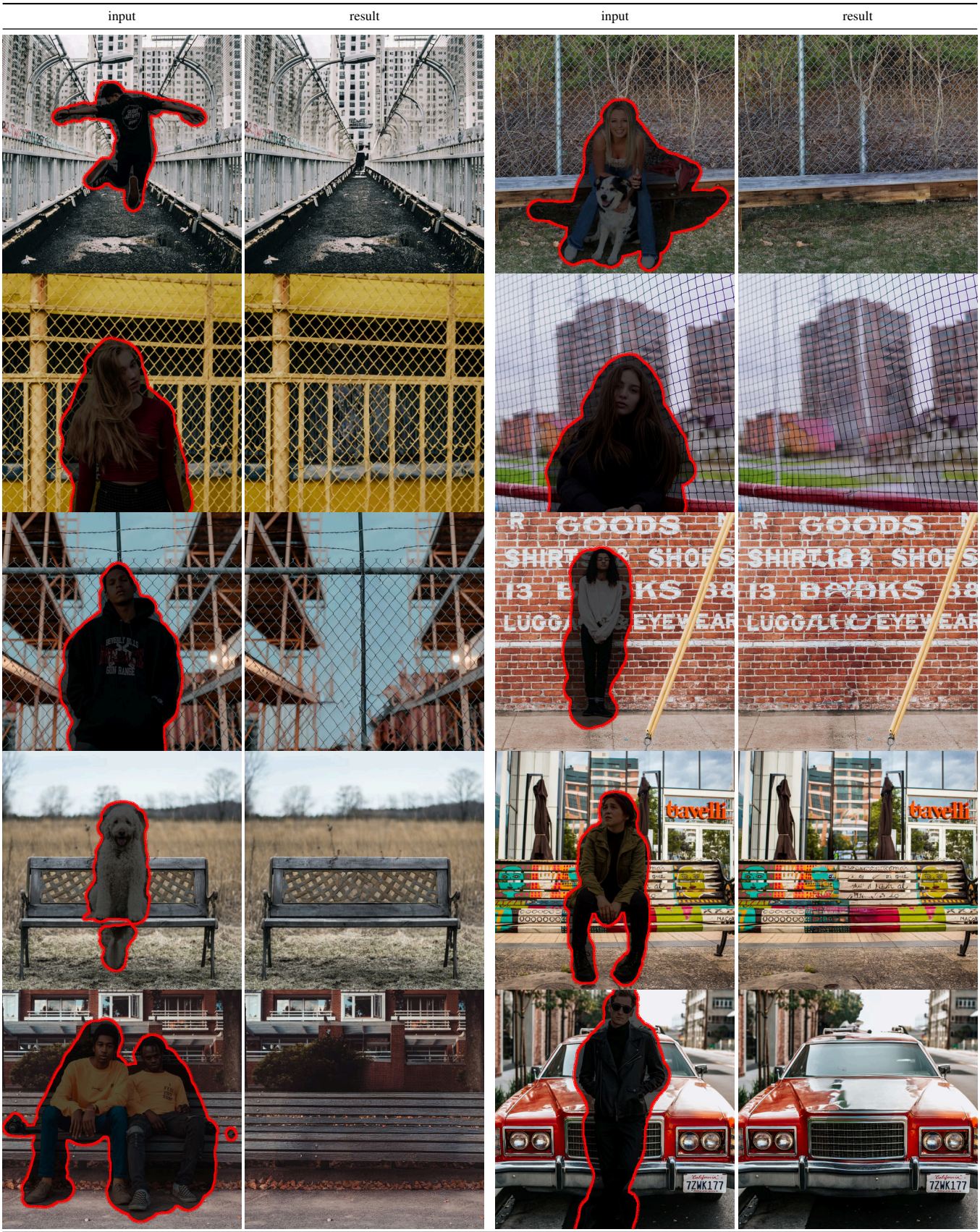

Figure 21. More qualitative results on object removal as in Fig. 10.

Semantic Synthesis on Flickr-Landscapes [23] (512^2 finetuning)

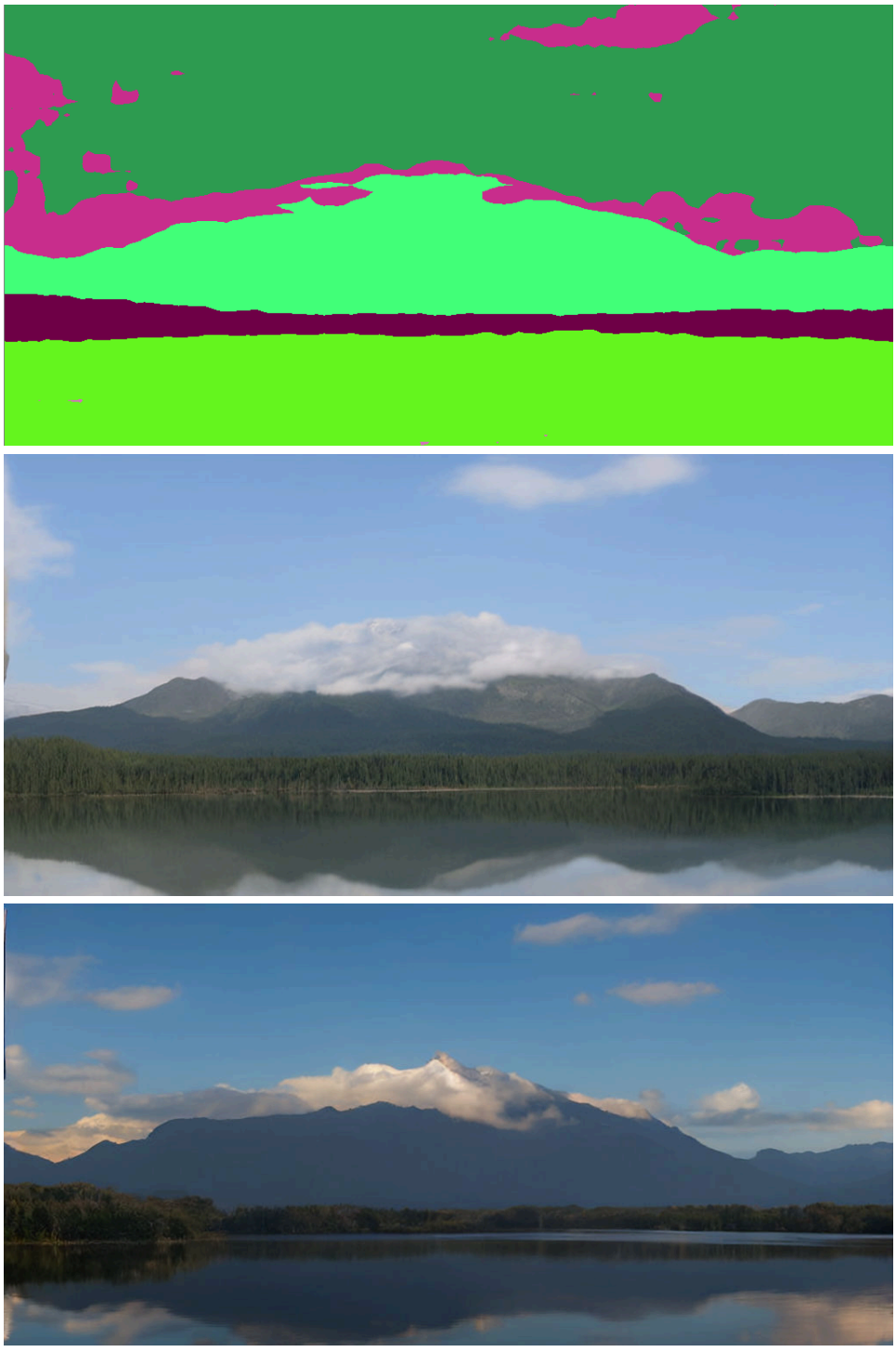

Figure 22. Convolutional samples from the semantic landscapes model as in Sec. 4.3.2, finetuned on 512^2 images.

Figure 23. A *LDM* trained on 256^2 resolution can generalize to larger resolution for spatially conditioned tasks such as semantic synthesis of landscape images. See Sec. 4.3.2.

Semantic Synthesis on Flickr-Landscapes [23]

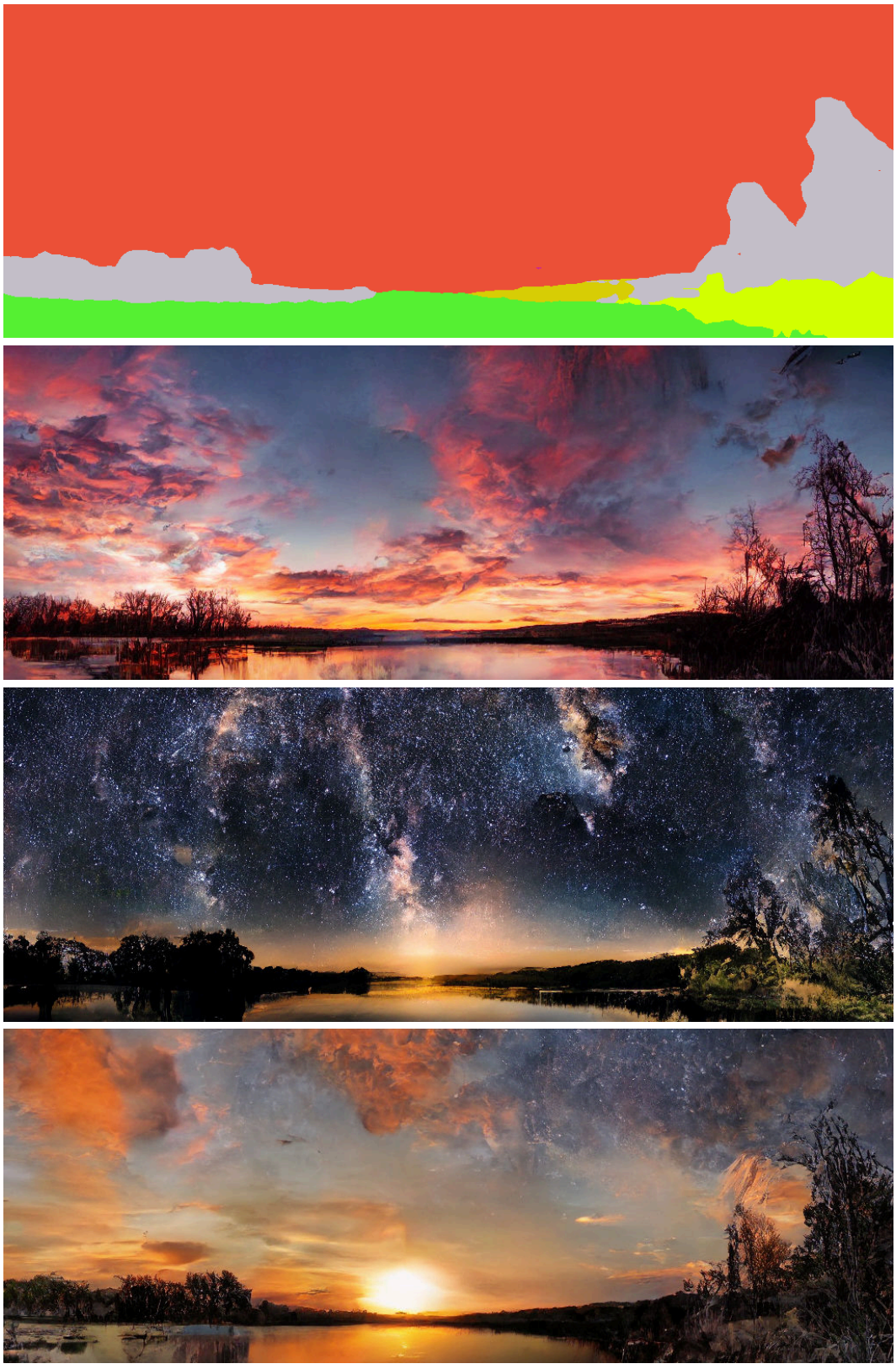

Figure 24. When provided a semantic map as conditioning, our *LDMs* generalize to substantially larger resolutions than those seen during training. Although this model was trained on inputs of size 256^2 it can be used to create high-resolution samples as the ones shown here, which are of resolution 1024×384.

Random class conditional samples on the ImageNet dataset

Figure 25. Random samples from *LDM-4* trained on the ImageNet dataset. Sampled with classifier-free guidance [31] scale $s = 5.0$ and 200 DDIM steps with $\eta = 1$..

Random class conditional samples on the ImageNet dataset

Figure 26. Random samples from *LDM-4* trained on the ImageNet dataset. Sampled with classifier-free guidance [31] scale $s = 3.0$ and 200 DDIM steps with $\eta = 1$..

Random samples on the CelebA-HQ dataset

Figure 27. Random samples of our best performing model *LDM-4* on the CelebA-HQ dataset. Sampled with 500 DDIM steps and $\eta = 0$ (FID = 5.15).

Random samples on the FFHQ dataset

Figure 28. Random samples of our best performing model *LDM-4* on the FFHQ dataset. Sampled with 200 DDIM steps and $\eta = 1$ (FID = 4.98).

Random samples on the LSUN-Churches dataset

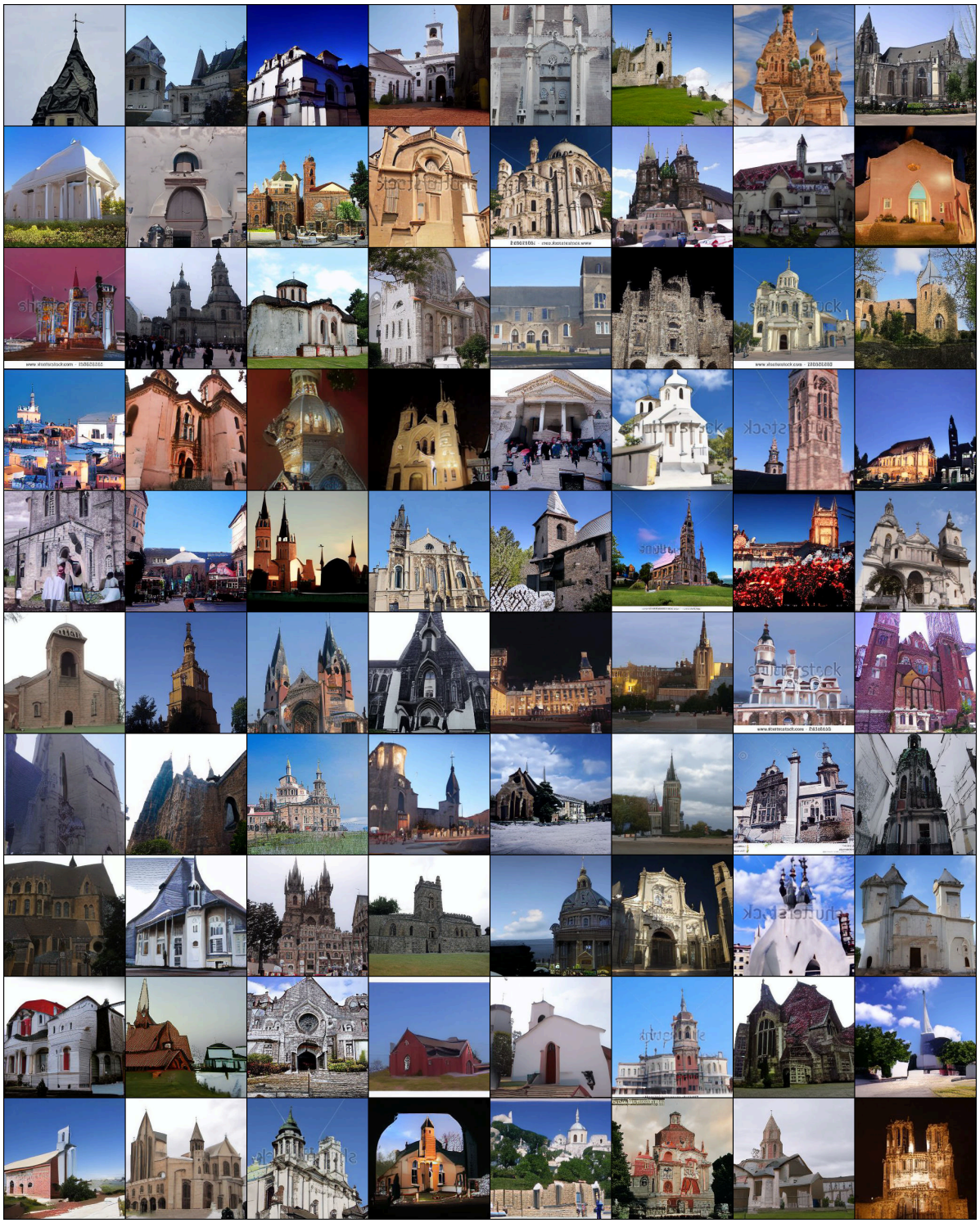

Figure 29. Random samples of our best performing model *LDM-8* on the LSUN-Churches dataset. Sampled with 200 DDIM steps and $\eta = 0$ (FID = 4.48).

Random samples on the LSUN-Bedrooms dataset

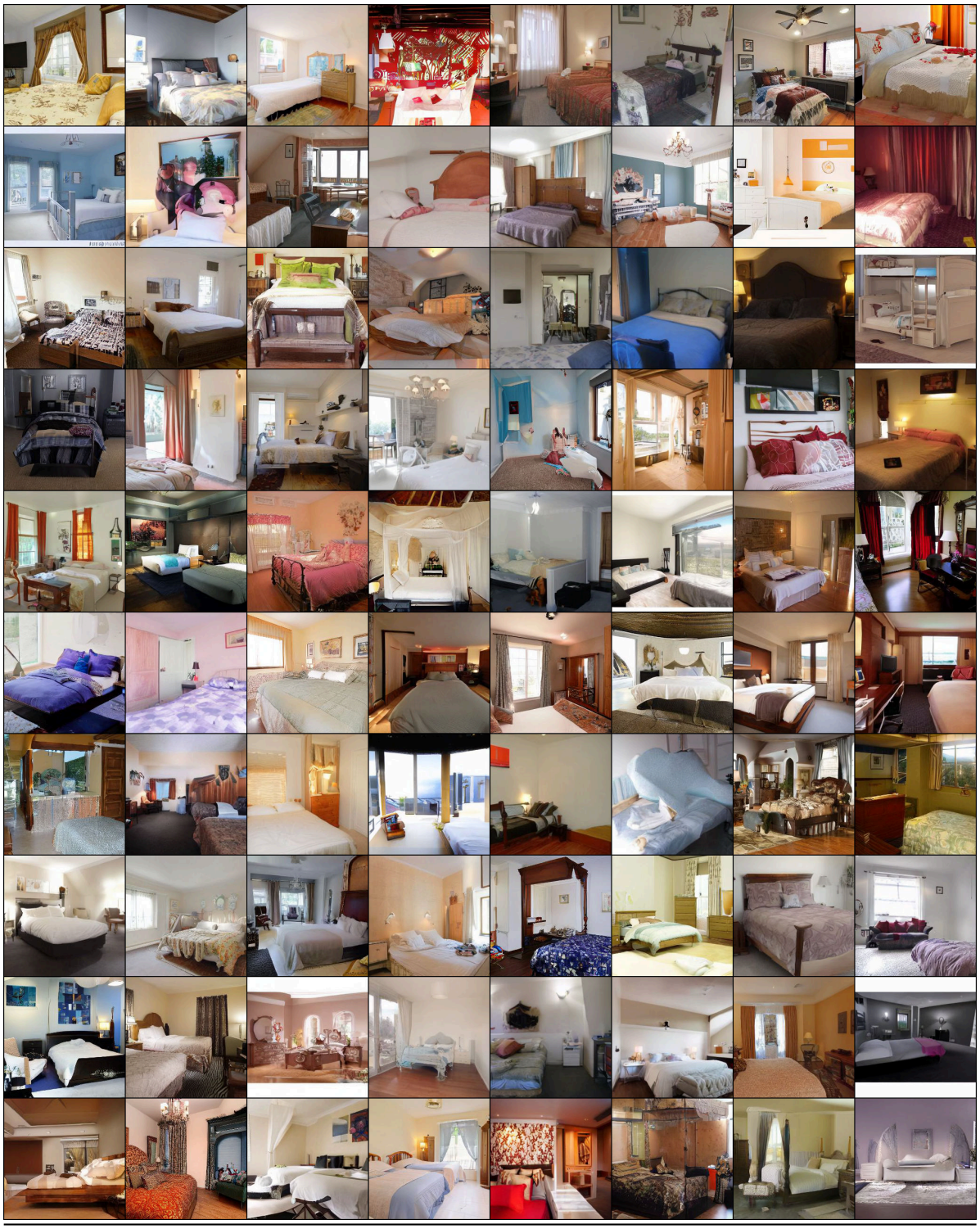

Figure 30. Random samples of our best performing model *LDM-4* on the LSUN-Bedrooms dataset. Sampled with 200 DDIM steps and $\eta = 1$ (FID = 2.95).

Nearest Neighbors on the CelebA-HQ dataset

Figure 31. Nearest neighbors of our best CelebA-HQ model, computed in the feature space of a VGG-16 [77]. The leftmost sample is from our model. The remaining samples in each row are its 10 nearest neighbors.

Nearest Neighbors on the FFHQ dataset

Figure 32. Nearest neighbors of our best FFHQ model, computed in the feature space of a VGG-16 [77]. The leftmost sample is from our model. The remaining samples in each row are its 10 nearest neighbors.

Nearest Neighbors on the LSUN-Churches dataset

Figure 33. Nearest neighbors of our best LSUN-Churches model, computed in the feature space of a VGG-16 [77]. The leftmost sample is from our model. The remaining samples in each row are its 10 nearest neighbors.

References

[1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1122–1131. IEEE Computer Society, 2017. 1

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. 3

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Int. Conf. Learn. Represent.*, 2019. 1, 2, 6, 7, 8, 15, 22

[4] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1209–1218. Computer Vision Foundation / IEEE Computer Society, 2018. 6, 13, 14

[5] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021. 23

[6] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 2020. 3

[7] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *ICLR*. OpenReview.net, 2021. 1

[8] Lu Chi, Borui Jiang, and Yadong Mu. Fast fourier convolution. In *NeurIPS*, 2020. 8

[9] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *CoRR*, abs/2011.10650, 2020. 3

[10] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *CoRR*, abs/1904.10509, 2019. 3

[11] Bin Dai and David P. Wipf. Diagnosing and enhancing VAE models. In *ICLR (Poster)*. OpenReview.net, 2019. 2, 3

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE Computer Society, 2009. 1, 5, 7, 15

[13] Emily Denton. Ethical considerations of generative ai. AI for Content Creation Workshop, CVPR, 2021. 23

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 6

[15] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *CoRR*, abs/2105.05233, 2021. 1, 2, 3, 4, 6, 7, 8, 11, 15, 19, 20, 22

[16] Sander Dieleman. Musings on typicality, 2020. 1, 3

[17] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. *CoRR*, abs/2105.13290, 2021. 6

[18] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015. 3

[19] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1, 3

[20] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Adv. Neural Inform. Process. Syst.*, pages 658–666, 2016. 3

[21] Patrick Esser, Robin Rombach, Andreas Blattmann, and Björn Ommer. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis. *CoRR*, abs/2108.08827, 2021. 6, 15

[22] Patrick Esser, Robin Rombach, and Björn Ommer. A note on data biases in generative models. *arXiv preprint arXiv:2012.02516*, 2020. 23

[23] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *CoRR*, abs/2012.09841, 2020. 2, 3, 4, 6, 7, 14, 15, 23, 28, 30

[24] Mary Anne Franks and Ari Ezra Waldman. Sex, lies, and videotape: Deep fakes and free speech delusions. *Md. L. Rev.*, 78:892, 2018. 23

[25] Kevin Frans, Lisa B. Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *ArXiv*, abs/2106.14843, 2021. 3

[26] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *CoRR*, 2014. 1, 2

[27] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017. 3

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Adv. Neural Inform. Process. Syst.*, pages 6626–6637, 2017. 1, 6, 20

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1, 2, 3, 4, 6, 10

[30] 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. 1, 3, 15

[31] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 6, 7, 15, 22, 31, 32

[32] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976. IEEE Computer Society, 2017. 3, 4

[33] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. 4

[34] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs &outputs. *CoRR*, abs/2107.14795, 2021. 4

[35] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. Perceiver: General perception with iterative attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR, 2021. 4

[36] Manuel Jahn, Robin Rombach, and Björn Ommer. High-resolution complex scene synthesis with transformers. *CoRR*, abs/2105.06458, 2021. 13, 14, 21

[37] Niharika Jain, Alberto Olmo, Sailik Sengupta, Lydia Manikonda, and Subbarao Kambhampati. Imperfect imaganation: Implications of gans exacerbating biases on facial data augmentation and snapchat selfie lenses. *arXiv preprint arXiv:2001.09528*, 2020. 23

[38] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. 5, 6

[39] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4401–4410, 2019. 1

[40] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5, 6

[41] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *CoRR*, abs/1912.04958, 2019. 2, 6, 22

[42] Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Score matching model for unbounded data score. *CoRR*, abs/2106.05527, 2021. 6

[43] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2018. 3

[44] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *CoRR*, abs/2107.00630, 2021. 1, 3, 10

[45] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014. 1, 3, 4, 23

[46] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *CoRR*, abs/2106.00132, 2021. 3

[47] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *ICLR*. OpenReview.net, 2021. 1

[48] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale. *CoRR*, abs/1811.00982, 2018. 6, 13, 14

[49] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *CoRR*, abs/1904.06991, 2019. 6, 20

[50] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 6, 21

[51] Yuqing Ma, Xianglong Liu, Shihao Bai, Le-Yi Wang, Aishan Liu, Dacheng Tao, and Edwin Hancock. Region-wise generative adversarial imageinpainting for large missing areas. *ArXiv*, abs/1909.12507, 2019. 8

[52] 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. 1

[53] Lars M. Mescheder. On the convergence properties of GAN training. *CoRR*, abs/1801.04406, 2018. 3

[54] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 3

[55] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. 4

[56] Gautam Mittal, Jesse H. Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. *CoRR*, abs/2103.16091, 2021. 1

[57] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *ArXiv*, abs/1901.00212, 2019. 8

[58] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. Version: 0.3.0, DOI: 10.5281/zenodo.4957738. 20, 21

[59] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4, 7

[60] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 14

[61] Gaurav Parmar, Dacheng Li, Kwonjoon Lee, and Zhuowen Tu. Dual contradistinctive generative autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 823–832. Computer Vision Foundation / IEEE, 2021. 6

[62] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On buggy resizing libraries and surprising subtleties in fid calculation. *arXiv preprint arXiv:2104.11222*, 2021. 20

[63] David A. Patterson, Joseph Gonzalez, Quoc V. Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *CoRR*, abs/2104.10350, 2021. 2

[64] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. 1, 2, 3, 4, 6, 14, 21

[65] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, pages 14837–14847, 2019. 1, 2, 3, 15

[66] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016. 4

[67] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning, ICML*, 2014. 1, 4, 23

[68] Robin Rombach, Patrick Esser, and Björn Ommer. Network-to-network translation with conditional invertible neural networks. In *NeurIPS*, 2020. 3

[69] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI (3)*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015. 2, 3, 4

[70] 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. 1, 4, 7, 15, 16, 17, 21

[71] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *CoRR*, abs/1701.05517, 2017. 1, 3

[72] Dave Salvator. NVIDIA Developer Blog. https://developer.nvidia.com/blog/getting-immediate-speedups-with-a100-tf32, 2020. 22

[73] Robin San-Roman, Eliya Nachmani, and Lior Wolf. Noise estimation for generative diffusion models. *CoRR*, abs/2104.02600, 2021. 3

[74] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *CoRR*, abs/2111.01007, 2021. 6

[75] Edgar Schönfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8204–8213. Computer Vision Foundation / IEEE, 2020. 6

[76] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs, 2021. 6

[77] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *Int. Conf. Learn. Represent.*, 2015. 23, 37, 38, 39

[78] Charlie Snell. Alien Dreams: An Emerging Art Scene. https://ml.berkeley.edu/blog/posts/clip-art/, 2021. [Online; accessed November-2021]. 2

[79] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *CoRR*, abs/1503.03585, 2015. 1, 3, 4, 11

[80] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 4

[81] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*. OpenReview.net, 2021. 3, 5, 6, 16

[82] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020. 1, 3, 4, 11

[83] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13693–13696. AAAI Press, 2020. 2

[84] Wei Sun and Tianfu Wu. Learning layout and style reconfigurable gans for controllable image synthesis. *CoRR*, abs/2003.11571, 2020. 14, 21

[85] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor S. Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *ArXiv*, abs/2109.07161, 2021. 8, 20, 26

[86] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R. Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 2647–2655. AAAI Press, 2021. 13, 14, 21

[87] Patrick Tinsley, Adam Czajka, and Patrick Flynn. This face does not exist... but it might be yours! identity leakage in generative models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1320–1328, 2021. 23

[88] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. 23

[89] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *NeurIPS*, 2020. 3

[90] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *CoRR*, abs/2106.05931, 2021. 2, 3, 6

[91] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, 2016. 3

[92] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *CoRR*, abs/1601.06759, 2016. 3

[93] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NIPS*, pages 6306–6315, 2017. 2, 4, 23

[94] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 3, 4, 5, 6

[95] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019. 20

[96] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. VAEBM: A symbiosis between variational autoencoders and energy-based models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 6

[97] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using VQ-VAE and transformers. *CoRR*, abs/2104.10157, 2021. 3

[98] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. 5, 6

[99] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan, 2021. 3, 4

[100] Jiahui Yu, Zhe L. Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-form image inpainting with gated convolution. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4470–4479, 2019. 8

[101] K. Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. *ArXiv*, abs/2103.14006, 2021. 17

[102] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 7, 12

[103] Shengyu Zhao, Jianwei Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I-Chao Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *ArXiv*, abs/2103.10428, 2021. 8

[104] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1452–1464, 2018. 8, 20

[105] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. LAFITE: towards language-free training for text-to-image generation. *CoRR*, abs/2111.13792, 2021. 6