Unlocking Pre-trained Image Backbones for Semantic Image Synthesis

Supplementary Material

In Appendix A, we present more details on our experimental setup to ease reproduction of our work. In Appendix B, we provide additional experimental results to evaluate our model and compare to prior work.

A. More details on the experimental setup

In Table S1 we provide the links to the datasets and models used in our work and their licensing.

A.1. Architecture details

Generator. As can be seen in Figure 3 of the main paper, our generator consists of a UNet-like architecture with two pyramidal paths. The *label map encoding* takes the input segmentation map, and progressively downsamples it to produce label-conditioned multi-scale features. These features are then used in the *image generation path*, which progressively upsamples the signal to eventually produce an RGB image. The stochasticity of the images generated is based on conditioning on the noise vector $\mathbf{z}$. We provide a schematic overview of the noise injection operation in Figure S1. Notably, we follow [27] and normalize every noise vector to the unit sphere before feeding it to the generator $\tilde{\mathbf{z}} = \frac{\mathbf{z}}{\|\mathbf{z}\|_2}$. In Table S2, we provide additional information on the label map encoding and the image generation paths.

In the label map encoding branch, each block is made of the elements listed in Table S2. Cross-attention and self-attention are only applied at lower resolutions (64×64, and lower) where we use an embedding dimension that is half of the original feature dimension. We downscale the feature maps by using convolution layers with a stride of 2.

In the image synthesis branch, we follow the same architecture as OASIS [52] with the only difference being the SPADE conditioning maps which are given by the label map encoding path instead of a resized version of the label maps. We also remove the hyperbolic tangent at the output of the network as we found it leads to a more stable generator.

For the contrastive learning branch, features obtained from VGG19 go through three convolutional blocks and two linear layers for projection. We sample 128 different patches to obtain negative samples from the image.

Discriminator. We provide additional details of our discriminator architecture in Table S3. The residual blocks are made of one convolution with kernel size 3 followed by leaky ReLU, then a pointwise convolution with leaky ReLU. For the full resolution channel, we set the dimensionality to 128. For the lower resolution channels, we stick to the same dimensionality as the corresponding encoder feature. The dimensionality of the final convolution before predicting the segmentations is set to 256.

We use spectral norm on all convolutional and linear layers in both the generator and the discriminator.

Feature conditioning. In [51] the authors observe that when using a fixed feature encoder in the GAN discriminator, only a subset of features is covered by the projector. They therefore propose propose to dilute prominent features, encouraging the discriminator to utilize all available information equally across the different scales. We believe that the reason behind this is that feature encoders trained for a discriminative task will have different structures than those trained on generative tasks. For the former, models tend to capture a subset of key features useful for discrimination, while disregarding other less relevant features. On the latter however, the model needs an extensive representation of the different objects it should generate. In practice, this translates to feature encoders having poor conditioning. The range of activations differs greatly from one feature to the other, which leads to bias towards a minority features that have a high amplitude of activations. A simple way to resolve this issue is by applying normalization these features to have a distribution with zero mean and a unit standard deviation across the batch.

In some situations, linear scaling of the features might not be enough to have proper conditioning of the features. Accordingly, we reduce the dynamic range of the feature maps before the normalization by using a sigmoid activation at the feature outputs of the pretrained encoder.

A.2. Computation of the mIoU evaluation metrics

To compute the mIoU metric, we infer segmentation maps for generated images. We infer segmentation maps for the generated images using the same networks as in OASIS [52]: UperNet101 [61] for ADE-20K, multi-scale DRN-D-105 [64] for Cityscapes, and DeepLabV2 [8] for COCO-Stuff. We also measure mIoU using Mask2Former [10] with Swin-L backbone [35] (mIoU_{MF}), which yields more accurate segmentations, and thus a more accurate comparison to the ground-truth masks.

In Table S4 we compare the segmentation accuracy on the three datasets we used in our experiments. The results confirm that Mask2Former is more accurate for all three datasets, in particular on COCO-Stuff, where it boosts mIoU by more than 19 points w.r.t. DeepLab-v2.

Name	Link
ImageNet	https://www.image-net.org
COCO-Stuff	https://cocodataset.org
Cityscapes	https://www.cityscapes-dataset.com
ADE-20K	https://groups.csail.mit.edu/vision/datasets/ADE20K/
Detectron2	https://github.com/facebookresearch/detectron2
ConvNext	https://github.com/facebookresearch/ConvNeXt
Swin	https://github.com/microsoft/Swin-Transformer
EfficientNet	https://github.com/lukemelas/EfficientNet-PyTorch
VGG19	https://github.com/pytorch/vision/blob/main/torchvision/models/vgg.py
Deeplab-v2	https://github.com/kazuto1011/deeplab-pytorch/
UperNet101	https://github.com/CSAILVision/sceneparsing
MS DRN-D-105	https://github.com/fyu/drn
Mask2Former	https://github.com/facebookresearch/Mask2Former
Self-supervised FID [39]	https://github.com/stanis-morozov/self-supervised-gan-eval

Name	License
ImageNet	Terms of access: https://www.image-net.org/download.php
COCO-Stuff	https://www.flickr.com/creativecommons
Cityscapes	https://www.cityscapes-dataset.com/license
ADE-20K	https://groups.csail.mit.edu/vision/datasets/ADE20K/terms/
Detectron2	Apache-2.0 license
R50	BSD
ConvNext	MIT License
Swin	MIT License
EfficientNet	Apache-2.0 license
VGG19	BSD-3-Clause license
UperNet101	BSD-3-Clause license
MS DRN-D-105	BSD-3-Clause license
Deeplab-v2	MIT License
Mask2Former	MIT License

Table S1. Links to the assets used in our work and the corresponding licensing information.

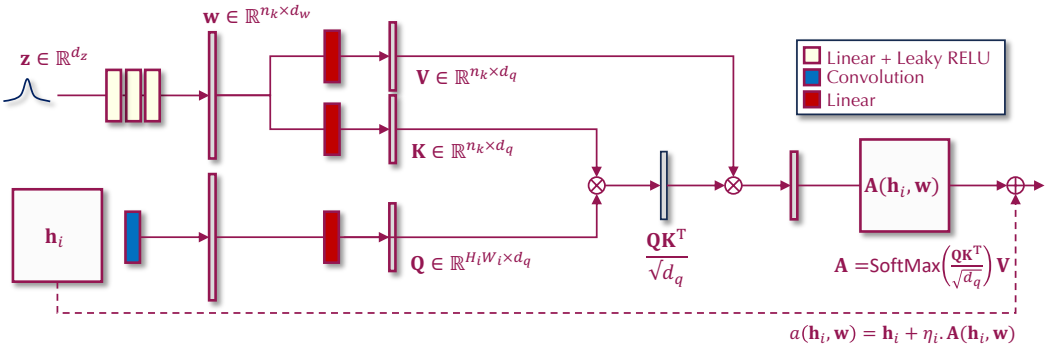

Figure S1. Schematic overview of our noise injection mechanism using cross-attention.

A.3. Influence of face blurring

For Cityscapes we use the release of the dataset with blurred faces and licence plates, which is available publicly on the website listed in Table S1. We blurred faces in ADE-20K and COCO-Stuff.

To assess the impact of blurring, we train OASIS on blurred images using the original source code from the authors and compare to their reported results on the non-blurred data. We report our results in Table S5. Here, and elsewhere in the paper, we also use the blurred data to compute FID w.r.t. the generated images. We see that blurring has a negative impact on FID, most notably for COCO-Stuff (+1.8), and to a lesser extent for ADE-20K (+0.8) and Cityscapes (+0.3). The mIoU_{MF} scores also degrade on all the datasets when using blurred data: on COCO-Stuff, ADE-20K and Cityscapes by 5.0, 3.9, and 0.4 points respectively. Note that in all comparisons to the state of the art, we report metrics obtained using models trained on blurred data for our approach, and models trained on non-blurred data for other approaches. Therefore, the real gains of our method over OASIS (and probably other methods as well) are even larger than what is shown in our comparisons in Table 1 in the main paper.

Parameter	Description
Hyperparameters	
$\mathbf{z}$ dimension	64
$\mathbf{w}$ dimension	256
Batch size	64
Learning rate	10^{-3}
β_1 for Adam	0
β_2 for Adam	0.99
EMA beta	0.9999
Label map encoding	
Pyramid block	Conv2d(kernel_size=3), BN, GELU, CrossAttention, BN, GELU, SelfAttention, GELU, BN, Conv2d(kernel_size=1)
Self Attention channel divider	2
Cross Attention channel divider	2
Conv block	Conv2d(kernel_size=3), BN, GELU, Conv2d(kernel_size=1)
Block type	[Conv, Conv, Conv, Linear, Linear]
Image synthesis branch	
Channel base	64
Number of residual blocks	6
Channel depths	[1024, 1024, 1024, 512, 256, 128, 64]
Residual block	SPADE, Leaky RELU, Conv2d(3)
Pyramid dimensionality	64
Hyperbolic tangent on output	No
Contrastive learning branch	
Perceptual network	VGG19
Contrastive encoding channels	[64, 128, 256, 512, 512]
Contrastive embedding dimension	256
Number of patches	128

Table S2. Architecture details of the generator.

Parameter	Description
Hyperparameters	
Number of multiscale backbone features	4
Full resolution embedding dimension	128
Number of residual blocks	5
Decoder	
Residual block	Conv2d(kernel_size=3), Leaky RELU, Conv2d(kernel_size=1), Leaky RELU
Leaky RELU slope	0.2
Penultimate channel dimension	256
Feature conditioning	
Conditioning normalization	Batch Norm w.o learned affine
Conditioning non-linearity	Hyperbolic tangent

Table S3. Architecture details of the discriminator.

	ADE-20K	Cityscapes	COCO-Stuff
UperNet101	42.7	—	—
MS DRN-D-105	—	61.3	—
DeepLab-v2	—	—	35.3
Mask2Former	45.3	69.9	54.5

Table S4. Segmentation performance in terms of mIoU on real images using different segmentation models. To match the setting used in our semantic image synthesis experiments, evaluation images are downsampled to 256×256 for ADE-20K and COCO, and to 256×512 for Cityscapes.

Dataset	Model	Blurring	FID (↓)	mIoU_{MF}(↑)
COCO-Stuff	OASIS	✗	17.0	52.1
	OASIS	✓	18.8	47.1
	DP-SIMS (ours)	✓	**13.6**	**65.2**
ADE-20K	OASIS	✗	28.3	53.5
	OASIS	✓	29.1	49.6
	DP-SIMS (ours)	✓	**22.7**	**67.8**
Cityscapes	OASIS	✗	47.7	72.0
	OASIS	✓	48.0	71.6
	DP-SIMS (ours)	✓	**38.2**	**78.5**

Table S5. Influence of face blurring on the performance of OASIS.

A.4. Carbon footprint estimation

On COCO-Stuff, it takes approximately 10 days to train our model using 8 GPUs. On ADE-20K and Cityscapes the training times are about 6 and 4 days respectively. Given a thermal design power (TDP) of the V100-32G GPU equal to 250W, a power usage effectiveness (PUE) of 1.1, a carbon intensity factor of 0.385 kg CO_2 per KWh, a time of 240 hours $\times$ 8 GPUs = 1920 GPU hours. The $250 \times 1.1 \times 1920$ = 528 kWh used to train the model is approximately equivalent to a CO_2 footprint of 528 $\times$ 0.385 = 208 kg of CO_2 for COCO-Stuff. For ADE-20K this amounts to 124 kg of CO_2, and 83 kg of CO_2 for Cityscapes.

B. Additional experimental results

B.1. Frozen *vs*. finetuned backbones

We experimented with training the feature backbone, rather than fixing it as in our default setup, and initializing from scratch or using a pre-trained model. We report the results on COCO-Stuff in Table S6. All tested alternatives provide worse performance than our default setting (fixed pre-trained backbone). When finetuning the backbone it is better to start from the pre-trained model, and using a fixed randomly initialized results in the worst performance.

Backbone	Initialization	FID	mIoU_{MF}
Fixed	Random	43.3	42.9
Finetuned	Random	18.9	52.6
Finetuned	IN-21k pre-trained	17.8	60.1
Fixed	IN-21k pre-trained	**13.6**	**65.2**

Table S6. Performance comparison between fixing and finetuning the disrciminator encoder.

Moreover, we find that using a fixed pre-trained backbone also results in significantly faster convergence compared to the alternatives. In Fig. S2, we report training progress for models trained on COCO-Stuff with both a frozen and trainable encoder. We additionally evaluate the trainable encoder with random *vs*. ImageNet-21k initializations. The fixed encoder model converges much faster than its trainable counterpart. For example, while the frozen model requires approximately 12 hours to achieve an FID below 25, the trainable models require more than a week of training to achieve the same score.

B.2. Quantifying bias towards ImageNet classes

Our discriminator backbones are pre-trained for ImageNet classification, as is the Inceptionv3 model [54] used in the computation of the FID metric. Therefore, the question arises whether our results are influenced by a bias of the features towards the classes in the ImageNet dataset. To analyze this, we report in Tab. S7 a quantitative comparison

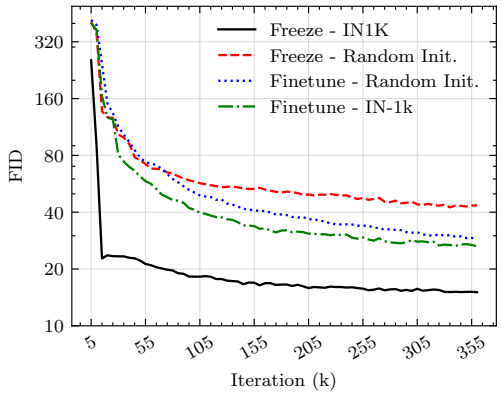

Figure S2. Convergence speed comparison for COCO-Stuff training with learnable *vs*. frozen encoder

following the approach outlined in [39], where we compute the Fréchet distance between two Gaussians fitted to feature representations of the SwAV Resnet50 network that was pre-trained in a self-supervised manner on ImageNet-1k. Our models retain state-of-the-art performance with respect to this metric on all the three datasets studied, further corroborating our results.

Additionally, we further experiment with the influence of the backbone pre-training in Table S8. Differently from the main paper where FID with the Inceptionv3 features is studied, here we find than the IN-21k checkpoint brings about better performance than its IN-1k counterpart. While the fine-tuning at high resolution (384 vs 224) also improves SwAV-FID.

	OASIS	SDM	PITI	DP-SIMS
COCO-Stuff	3.09	2.68	2.52	**2.14**
ADE-20K	4.35	3.85	—	**2.84**
Cityscapes	4.75	3.94	—	**3.71**

Table S7. Evaluation of SwAV Resnet50 FID on ADE-20K for different methods. We use a ConvNext-L backbone for DP-SIMS.

Pre-training	Acc@1	FID_{SwAV} (↓)
IN-1k@224	84.3	3.03
IN-21k@224	86.6	2.97
IN-21k@384	87.5	**2.84**

Table S8. Evaluation of SwAV Resnet50 FID with different pre-trainings evaluated on ADE-20K.

B.3. Influence of diversity loss

Our diversity loss is similar to prior works [38, 63] with a few notable differences. Mainly, the hinge term and the image distance space. In [38] it is shown that this loss formulation is a lower bound for the averaged gradients over

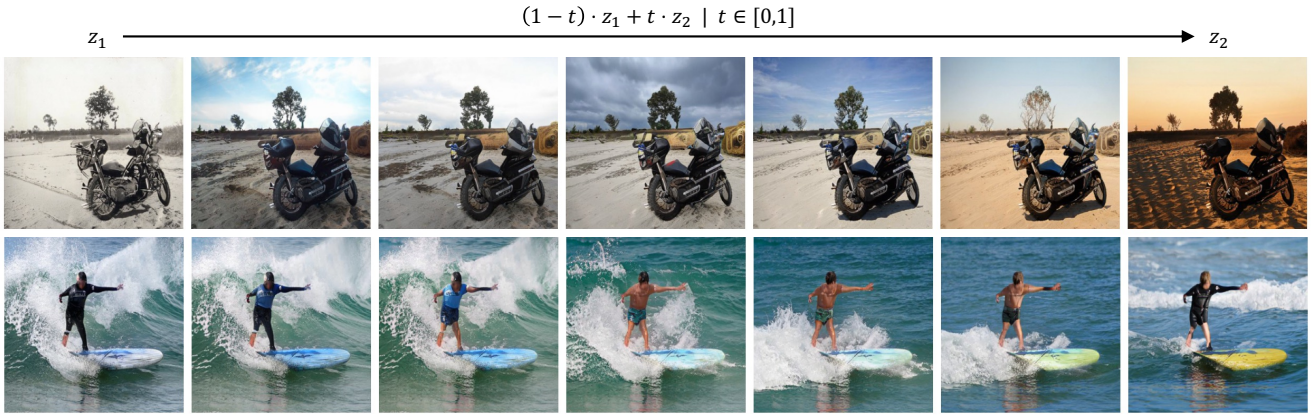

Figure S3. Noise vector interpolation. By interpolating the noise vector between two different values, we can identify the factors of variation in the image which correspond to differences in colors, textures as well as object structures.

the noise vectors $\mathbf{z_1}, \mathbf{z_2}$, therefore our diversity loss with the hinge term is akin to encouraging a minimal amplitude τ_{div} of the gradients with respect to the noise conditioning. Second, while prior work computed the distance between images either in image space or the discriminator's feature space, we found that neither of these two choices was optimal in our experiments. The discriminator's feature space works well for class-conditional synthesis because the discriminator's underlying feature representation is semantically richer than for semantic image synthesis where the dense prediction task of the discriminator yields very localized embeddings.

We obtain the diversity cutoff threshold τ_{div} by computing the mean distance between different generated images in a batch and averaging across the training set:

$$\tau_{\text{div}} = \frac{1}{|\mathcal{B}|^2} \cdot \sum_{i,j \in \mathcal{B}} \frac{\left\| G^f(x_i, \mathbf{z_i}) - G^f(x_j, \mathbf{z_j}) \right\|_1}{\|\mathbf{z}_i - \mathbf{z}_j\|_1}. \quad (7)$$

The distance is computed in the feature space induced by the penultimate layer of the generator. It is then normalized by the distance between the noise vectors.

We conduct a more in-depth analysis on the impact of the diversity loss on the image quality and diversity. We train our model with a ConvNext-L backbone with different values for the diversity loss λ_{div}. These results are reported in Table S9. Without the diversity loss, the generator ignores the noise input, which translates to a low LPIPS score. Improving diversity with a weight of $\lambda_{\text{div}} = 10$ results more diversity (LPIPS), better image quality (FID), and in put consistency (mIoU$_{\text{MF}}$).

Additionally, we experiment with different distances for the diversity loss: based either on the generator features, or on the RGB image space directly as in [38, 63]. As reported in Table S10, we find that the diversity loss in image space is less effective. It reaches an LPIPS score of 0.09 while the feature space loss achieves an LPIPS of 0.47. Both FID and mIoU$_{\text{MF}}$metrics are also improved by this choice. By inspecting example generations, we find that using image space distances results in variations in the overall contrast and brightness of the image only, while using feature space distances results in more high-level variations as illustrated in Fig. S3 and Fig. S4.

λ_{div}	0	10	100
FID (↓)	22.9	22.7	23.3
mIoU$_{\text{MF}}$(↑)	67.7	67.8	67.7
LPIPS (↑)	1.5e-5	0.47	0.36

Table S9. Influence of diversity loss weight on model performance. We evaluate image quality using FID and mIoU$_{\text{MF}}$ metrics while diversity is evaluated using LPIPS.

Distance space	FID (↓)	mIoU$_{\text{MF}}$	LPIPS (↑)
Feature	**22.7**	**67.8**	**0.47**
Image	23.2	64.2	0.09

Table S10. Comparing different distances for our diversity loss.

B.4. Sampling strategy

We quantify the influence of the balanced sampling strategy with respect to standard uniform sampling on COCO-Stuff and Cityscapes datasets. We report these results in Table S11, and find that balanced sampling yields performance gains in both FID and mIoU for both the datasets. In Figure S5, we present qualitative examples of images generated with the model trained on Cityscapes. Balanced sampling clearly leads to improvements in the visual quality of objects such as scooters, buses and trams.

Figure S4. Additional examples of diversity in generated images.

Dataset	Sampling strategy	FID (↓)	mIoU$_{MF}$(↑)
COCO-Stuff	Uniform	14.1	62.9
	Balanced	**13.6**	**65.2**
Cityscapes	Uniform	38.7	75.6
	Balanced	**38.3**	**78.3**

Table S11. Influence of sampling strategy for models trained on the COCO-Stuff and Cityscapes datasets.

B.5. Influence of pixel-wise loss function

In Figure S6, we compare the per-class mIoU values when training using different loss functions: weighted cross-entropy (as in OASIS), focal loss, and weighted focal loss. This extends the class-aggregated results reported in Table Table 8 in the main paper. These experiments were conducted on the Cityscapes dataset using a pre-trained ConvNext-L backbone for the discriminator. Our use of the weighted focal loss to train the discriminator results in improved IoU for most classes. The improvements tend to be larger for rare classes. Class weighting is still important, as can be seen from the deteriorated IoU for a number of classes when using the un-weighted focal loss.

B.6. Influence of instance-level annotations

Since some works do not use the instance masks [33, 52, 58], we provide an additional ablation in Table S12 where we train our models on COCO-Stuff and Cityscapes without the instance masks to isolate the gains in performance they may bring. For both these datasets, we observe deterioration in the model's performance when not using instance masks. The difference is less noticeable on COCO-Stuff where the labels are already partially separated, FID only increases by 0.3 points. On the other hand, this difference is more acute in Cityscapes where FID increases by 1.9 points while mIoU$_{MF}$reduces by 2.2 points. In Cityscapes, instances are not separated in the semantic label maps, this adds more ambiguity to the labels presented to the model which makes it more difficult to interpret them in a plausible manner.

Dataset	Instance masks	FID (↓)	mIoU$_{MF}$(↑)
COCO-Stuff	✗	13.9	65.0
	✓	**13.6**	**65.2**
Cityscapes	✗	40.1	76.3
	✓	**38.2**	**78.5**

Table S12. Influence of instance masks on model perofrmance.

	FID	mIoU$_{MF}$
DP-SIMS (ConvNext-L)	13.6	65.2
DP-SIMS (ConvNext-XL)	**13.3**	**68.0**

Table S13. Models with different ConvNext backbones on COCO-Stuff.

B.7. Larger discriminators

For larger datasets, scaling the backbone architecture could prove beneficial in capturing the complexity of the dataset. Accordingly, we train a model on COCO-Stuff using a ConvNext-XL model. It is approximately 1.76 times bigger than ConvNext-L used in our main experiments, with 350M parameters. In Table S13, we report its performance as a pre-trained feature encoder in our discriminator. The larger ConvNext-XL encoder further improves results in terms of both FID and mIOU.

B.8. Qualitative samples

We provide qualitative samples of the images generated with our DP-SIMS model using different pre-trained backbones for the discriminator in Figure S7. In Figure S8, Figure S9, and Figure S10 we provide examples of images generated with our DP-SIMS model and compare to other state-of-the-art models on the ADE-20K, COCO-Stuff, and Cityscapes datasets, respectively.

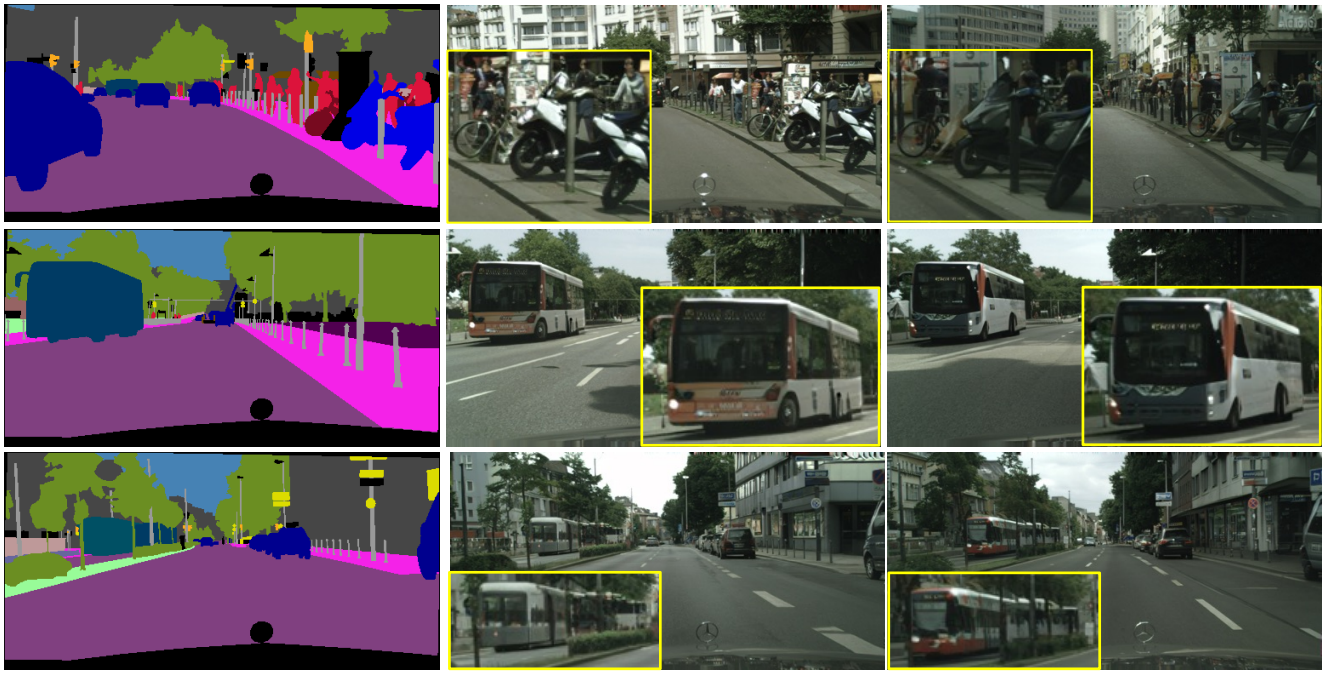

Figure S5. Qualitative examples of images generated with and without balanced sampling to train models on Cityscapes.

Figure S6. Top: Per-class IOU_{MF} on Cityscapes with models trained with different loss functions using ConvNext-L backbone. Labels are sorted according to their frequency in the validation images, which is written below the class name. Bottom: Per-class difference in IOU_{MF} of models trained with weighted and non-weighted focal loss w.r.t. the model trained with weighted cross-entropy (CE) loss.

Figure S7. Qualitative comparison of DP-SIMS on ADE-20K using Swin-B, Resnet50 (R50), EfficientNet-34, and ConvNext-L backbones.

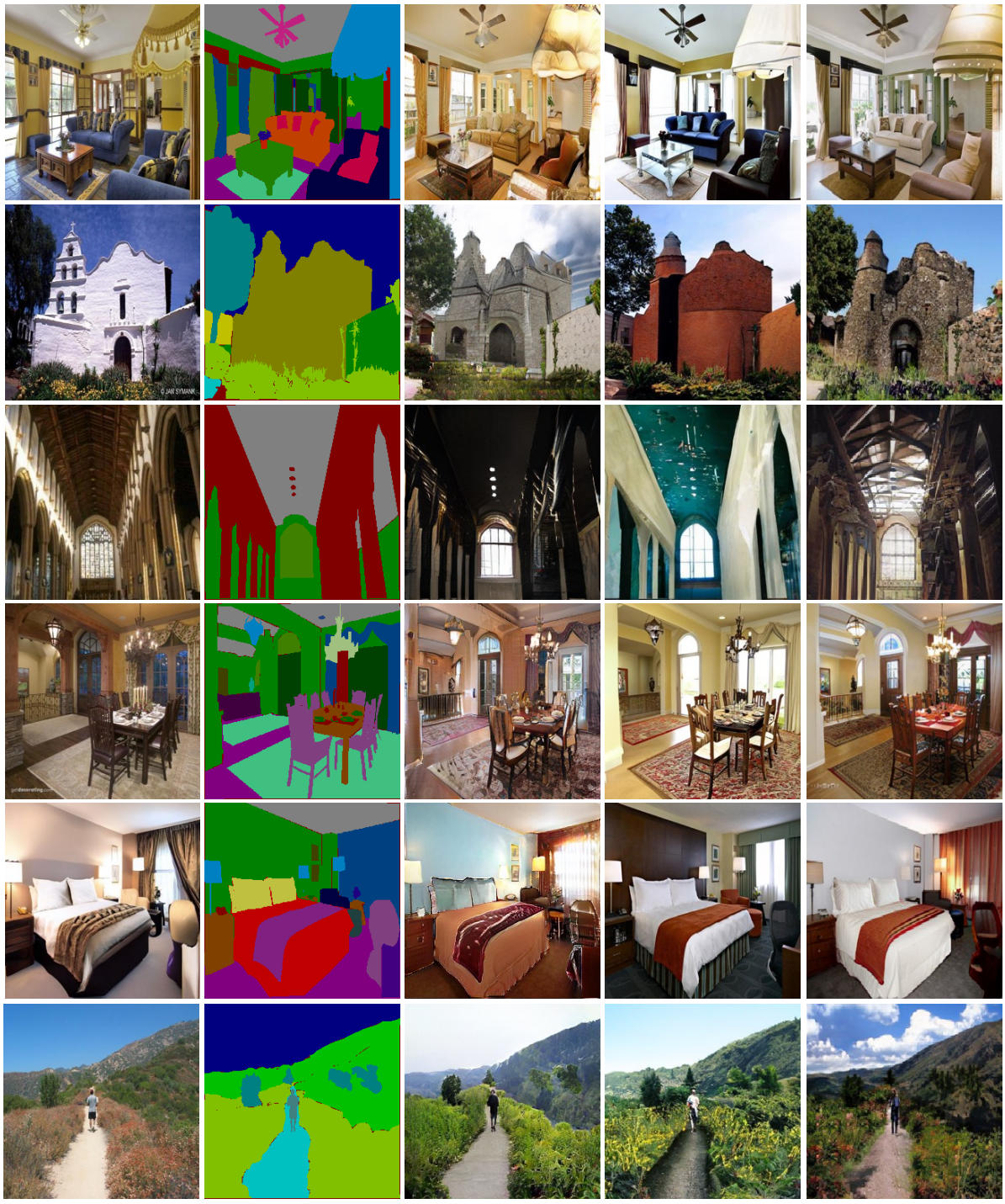

Figure S8. Qualitative comparison with prior work on ADE-20K, using a ConvNext-L backbone for DP-SIMS (Ours).

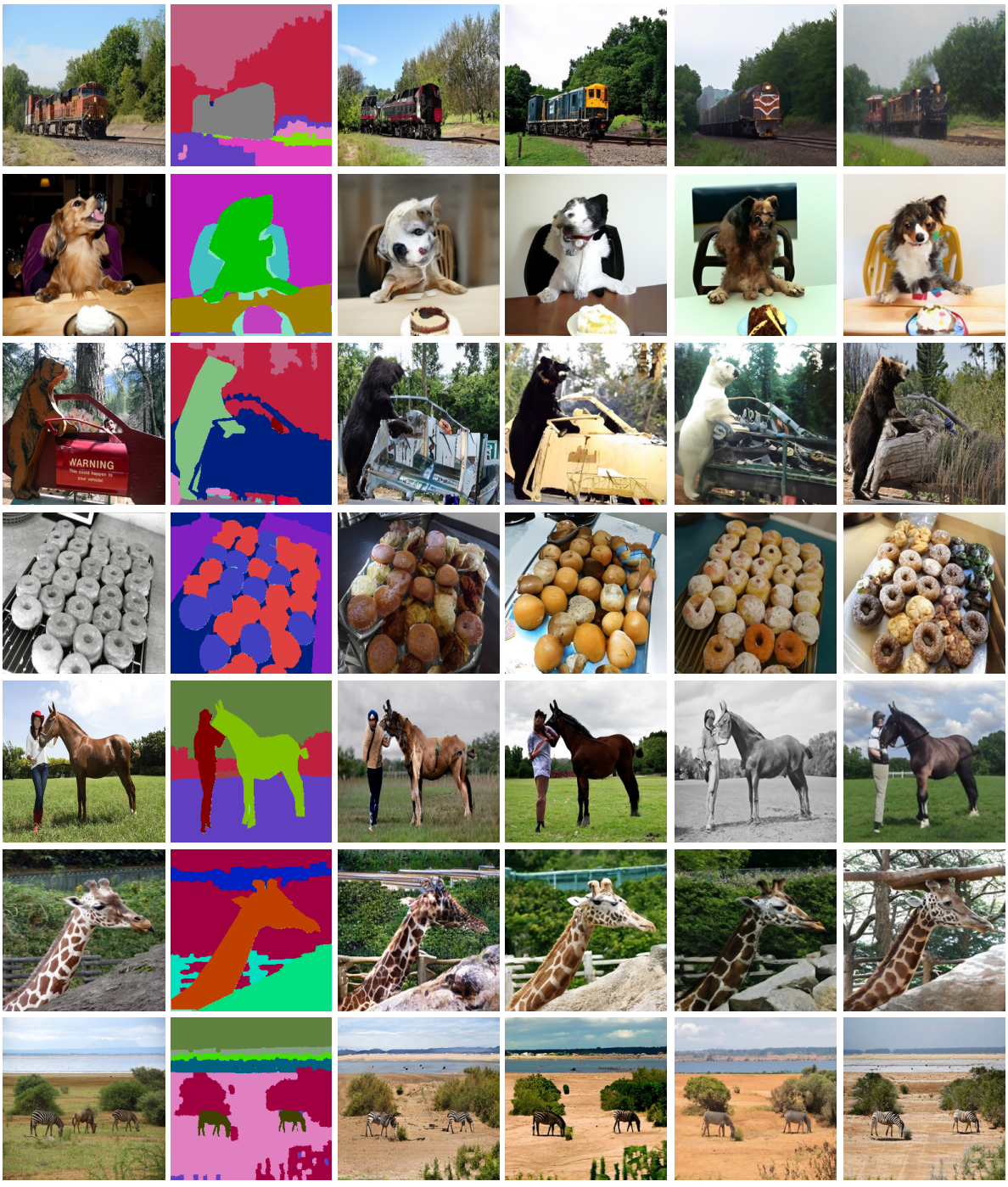

Figure S9. Qualitative comparison with prior work on COCO-Stuff, using a ConvNext-L backbone for DP-SIMS (Ours).

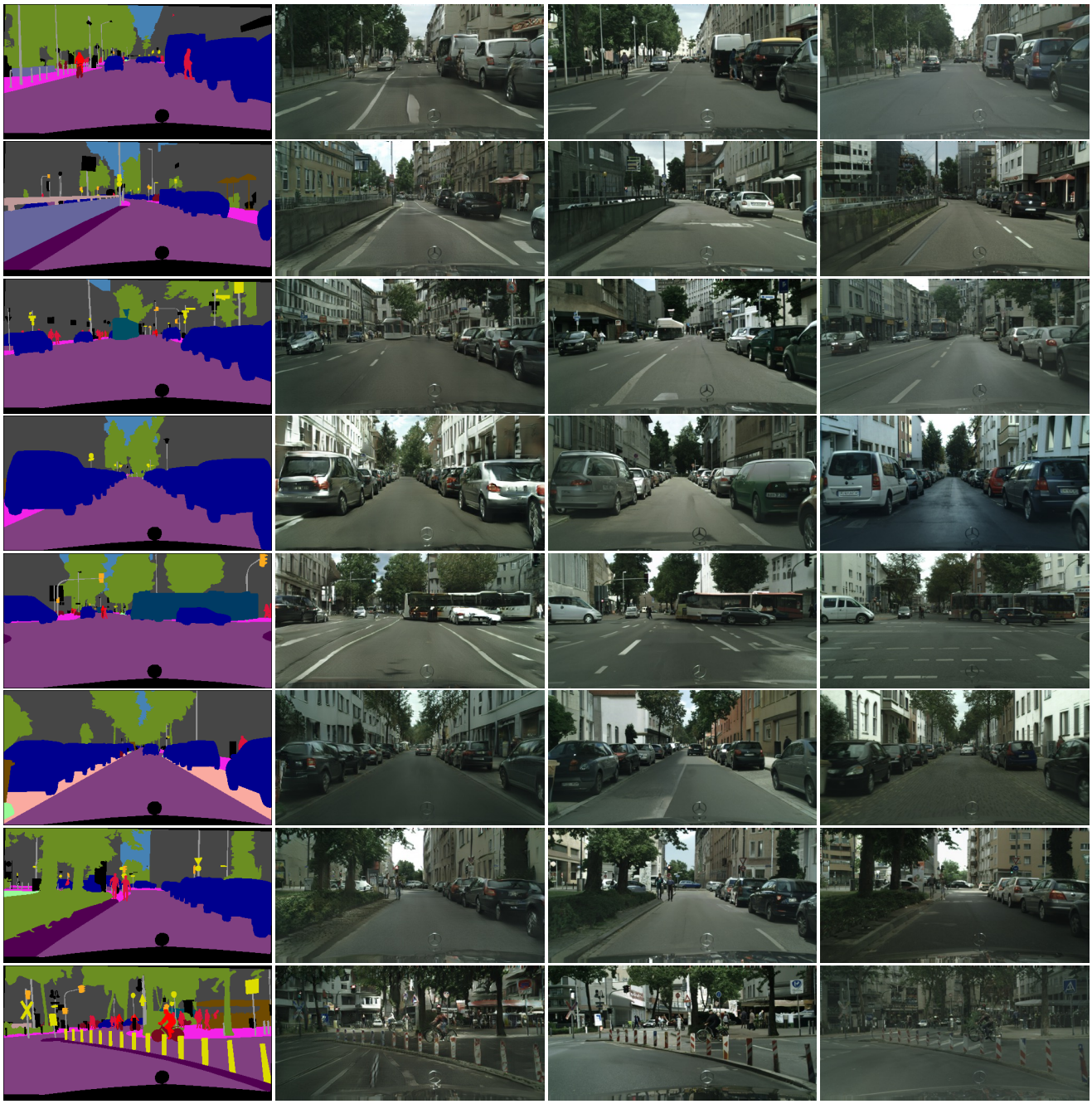

Figure S10. Qualitative comparison with prior work on Cityscapes, using a ConvNext-L backbone for DP-SIMS (Ours).

References

[1] Mohamed Akrout, Bálint Gyepesi, Péter Holló, Adrienn Poór, Blága Kincső, Stephen Solis, Katrina Cirone, Jeremy Kawahara, Dekker Slade, Latif Abid, Máté Kovács, and István Fazekas. Diffusion-based data augmentation for skin disease classification: Impact across original medical datasets to fully synthetic images. In *MICCAI workshop*, 2023.

[2] Chitwan Saharia amd William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

[3] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *CVPR*, 2023.

[4] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves imagenet classification. *TMLR*, 2023.

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[6] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018.

[7] Marlène Careil, Jakob Verbeek, and Stéphane Lathuilière. Few-shot semantic image synthesis with class affinity transfer. In *CVPR*, 2023.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *PAMI*, 40(4):834–848, 2018.

[9] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020.

[10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[12] Guillaume Couairon, Marlène Careil, Matthieu Cord, Stéphane Lathuilière, and Jakob Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. In *ICCV*, 2023.

[13] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

[14] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *ICLR*, 2017.

[15] Weichen Fan, Jinghuan Chen, Jiabin Ma, Jun Hou, and Shuai Yi. Styleflow for content-fixed image to image translation. *arXiv*, 2207.01909, 2022.

[16] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 2022.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[18] Reyhane Askari Hemmat, Mohammad Pezeshki, Florian Bordes, Michal Drozdzal, and Adriana Romero-Soriano. Feedback-guided data synthesis for imbalanced classification. *arXiv*, 2310.00158, 2023.

[19] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with Gaussian error linear units. *arXiv*, 1606.08415, 2016.

[20] 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 *NeurIPS*, 2017.

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[22] Xun Huang, Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu. Multimodal conditional image synthesis with product-of-experts GANs. In *ECCV*, 2022.

[23] Drew A. Hudson and C. Lawrence Zitnick. Generative adversarial transformers. In *ICML*, 2021.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[25] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up GANs for text-to-image synthesis. In *CVPR*, 2023.

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020.

[28] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *NeurIPS*, 2021.

[29] Diederik Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1×1 convolutions. In *NeurIPS*, 2018.

[30] Diederik Kingma and Max Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

[31] Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Ensembling off-the-shelf models for GAN training. In *CVPR*, 2022.

[32] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. ViTGAN: Training GANs with vision transformers. In *ICLR*, 2022.

[33] Shijie Li, Ming-Ming Cheng, and Juergen Gall. Dual pyramid generative adversarial networks for semantic image synthesis. In *BMVC*, 2022.

[34] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[36] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *CVPR*, 2022.

[37] Thomas Lucas, Konstantin Shmelkov, Karteek Alahari, Cordelia Schmid, and Jakob Verbeek. Adaptive density estimation for generative models. In *NeurIPS*, 2019.

[38] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *CVPR*, 2019.

[39] Stanislav Morozov, Andrey Voynov, and Artem Babenko. On self-supervised image representations for gan evaluation. In *ICLR*, 2021.

[40] 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.

[41] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *NeurIPS*, 2016.

[42] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 1807.03748, 2019.

[43] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

[44] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *ECCV*, 2020.

[45] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint*, 2204.06125, 2022.

[46] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, 2019.

[47] Stephan R. Richter, Hassan Abu AlHaija, and Vladlen Koltun. Enhancing photorealism enhancement. *IEEE TPAMI*, 45(2):1700–1715, 2022.

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[50] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NeurIPS*, 2016.

[51] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected GANs converge faster. In *NeurIPS*, 2021.

[52] Edgar Schönfeld, Vadim Sushko, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. In *ICLR*, 2021.

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[54] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathan Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[55] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

[56] Hao Tang, Guolei Sun, Nicu Sebe, and Luc van Gool. Edge guided GANs with multi-scale contrastive learning for semantic image synthesis. *PAMI*, 45:14435–14452, 2023.

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

[58] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv*, 2205.12952, 2022.

[59] 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. In *CVPR*, 2018.

[60] Weilun Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Dong Chen, Lu Yuan, and Houqiang Li. Semantic image synthesis via diffusion models. *arXiv preprint*, 2207.00050, 2022.

[61] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

[62] Han Xue, Zhiwu Huang, Qianru Sun, Li Song, and Wenjun Zhang. Freestyle layout-to-image synthesis. In *CVPR*, 2023.

[63] Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. In *ICLR*, 2019.

[64] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, 2017.

[65] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019.

[66] R. Zhang, P. Isola, A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[67] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, Weiming Zhang, and Nenghai Yu. X-Paste: Revisiting scalable copy-paste for instance segmentation using CLIP and StableDiffusion. In *ICML*, 2023.

[68] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.