Supplementary material for Full-Glow: Fully conditional Glow for more realistic image generation

Moein Sorkhei[1 [0000−0001−6204−0778]], Gustav Eje Henter[1 [0000−0002−1643−1054]], and Hedvig Kjellström[1,2 [0000−0002−5750−9655]]

[1] KTH Royal Institute of Technology, Sweden
[2] Silo AI, Sweden
{sorkhei,ghe,hedvig}@kth.se

In this supplementary material we provide:

- additional examples of samples generated by different models and high-resolution examples (see Section A with Figures 1, 2, and 3),
- experiments with additional conditioning information and with deeper models (see Section B with Figures 5 and 6 as well as Table 5),
- visual examples of the effect of varying the temperature (see Section C with Figures 7 and 8),
- further implementation details of the models (see Section D),
- and finally an example loss curve from training our proposed model (see Section E with Figure 4).

A Additional samples

In the experiments in the paper, we compared the performance of the models quantitatively and showed a visual example of the samples generated by different models. Figure 1 here shows another visual example of samples generated by different models.

We also showed that Full-Glow is able to generate plausible high-resolution images given a segmentation map as condition. Figures 2 and 3 show additional synthesized images in resolution 512×1024 along with the conditions.

B Effect of boundary maps and deeper models

This part illustrates the effect of using boundary maps and model depth on the performance of the model.

Boundary maps. When using segmentations as condition, the boundaries between objects of the same class are not visible when the objects are placed next to each other. This is because they all belong to the same class in the segmentation mask. As observed by [12], in the Cityscapes [1] dataset, each object in an image has an instance ID, which could be used to derive the boundary maps between the objects. For each pixel, if any of the 4 neighbors around it corresponds to a different instance ID than that of the pixel, the pixel will take value 1, otherwise it will be 0 (see Figure 5).

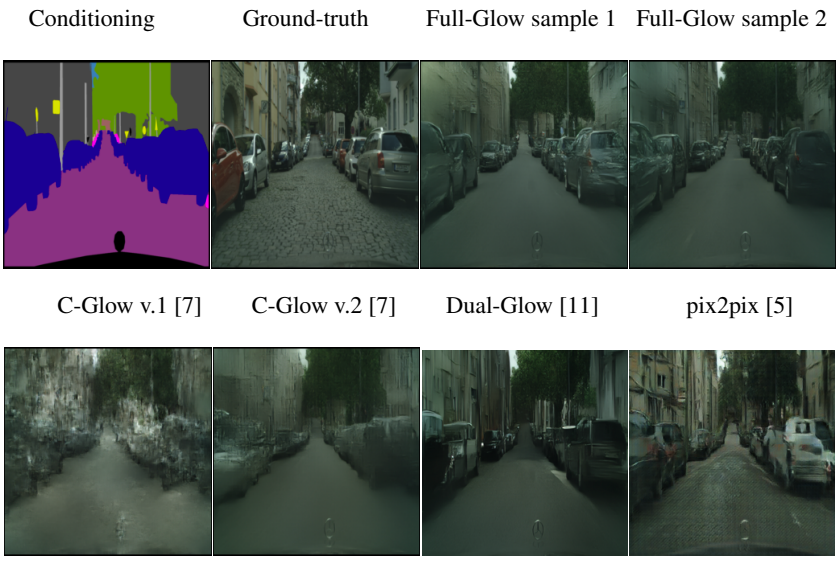

Fig. 1: Visual samples from different models. Samples from likelihood-based models are taken with temperature 0.7. Please zoom in to see more details.

Using boundary maps as side information enables the model to better distinguish between the objects available in the segmentation mask. In our experiments, we used boundary maps in all conditioning networks of the model by concatenating them along the channel dimension with the input to the conditioning networks [3]. In order to do so, we down-sampled the boundary maps so they match the spatial dimension of each Block (see Figure 6). We used two versions of boundary maps. In the first version, when down-sampling the boundary maps (using bi-linear interpolation), the pixel values that were previously either 0 or 1 will take float values between 0 and 1. In the second version, we forced the values to strictly be either 0 or 1, but that did not improve the performance. The results could be seen in Table 1 (Configs. B and C of our model).

Deepening the model. Gradient-checkpointing [10] is an effective technique that allows fitting deeper models in GPU memory by storing activations only at specific checkpoints in the model and recompute activations between checkpoints in the backward pass (lower memory usage at the cost of longer iteration time). We used this technique to train deeper models with the same training procedure as before. A checkpoint is inserted after each Flow in the model. The deeper models have 120 Flows in total (as opposed to 64 for the shallower models). We experimented with different versions of deep models. In Config. D, the model has equal Flows in each Block. In Config. E, more Flows are concentrated in the initial Blocks (closer to data space) since these Blocks are known to be responsible for lower-level details of the synthesized image as opposed to higher Blocks, closer to latent space, that are responsible for higher-level, abstract concepts [2]. We also observed the effect of number of Blocks while keeping total number

[3] We used implementation from [12] to derive the boundary maps.

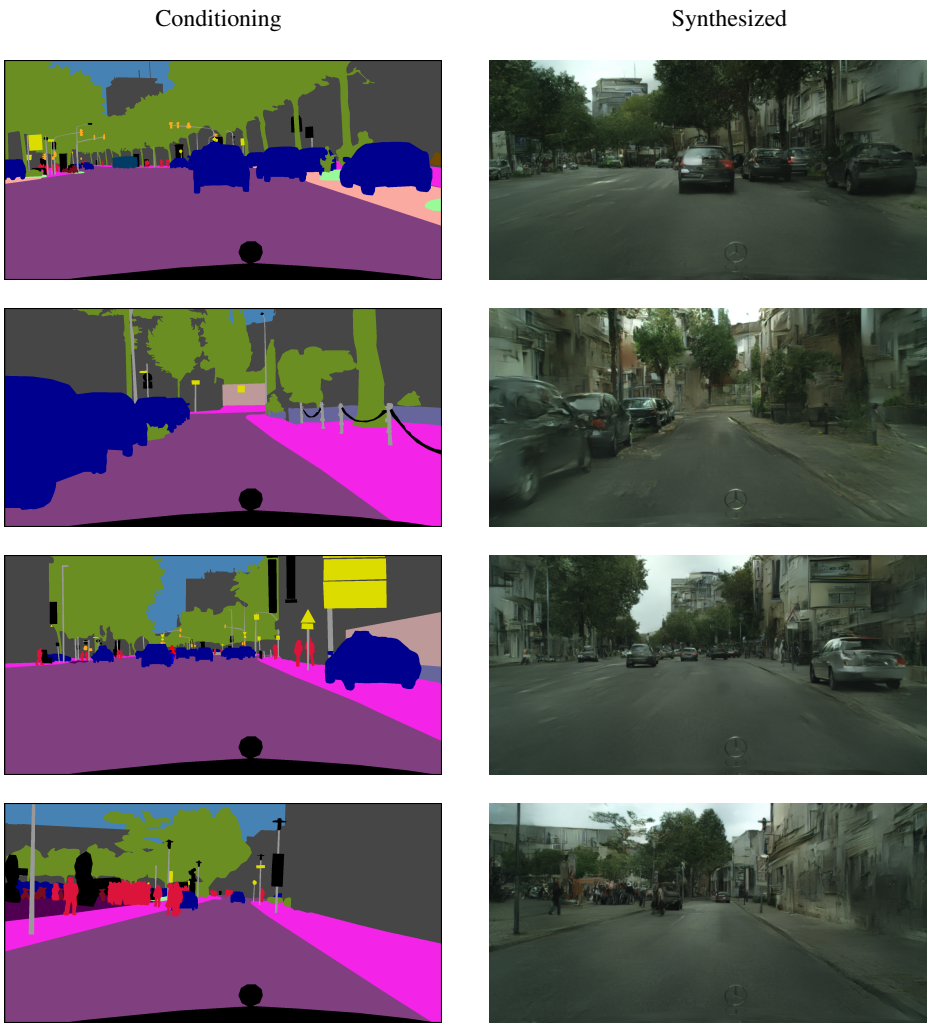

Fig. 2: Higher resolution image synthesis by our model taken with temperature 0.9. Please zoom in to see more details.

of Flows unchanged. While the number of Blocks and Flows can be seen as a choice of hyper-parameters, we observe that the optimal Blocks and Flows in this task seems to be in Config. H which has fewer Blocks with more concentrated Flows.

C Examples of effect of temperature

This part provides visual examples of the effect of the temperature. Figures 7 and 8 show how the colors and structures change with different temperature values. We can see that samples generated with higher temperatures seem to have more vibrant colors

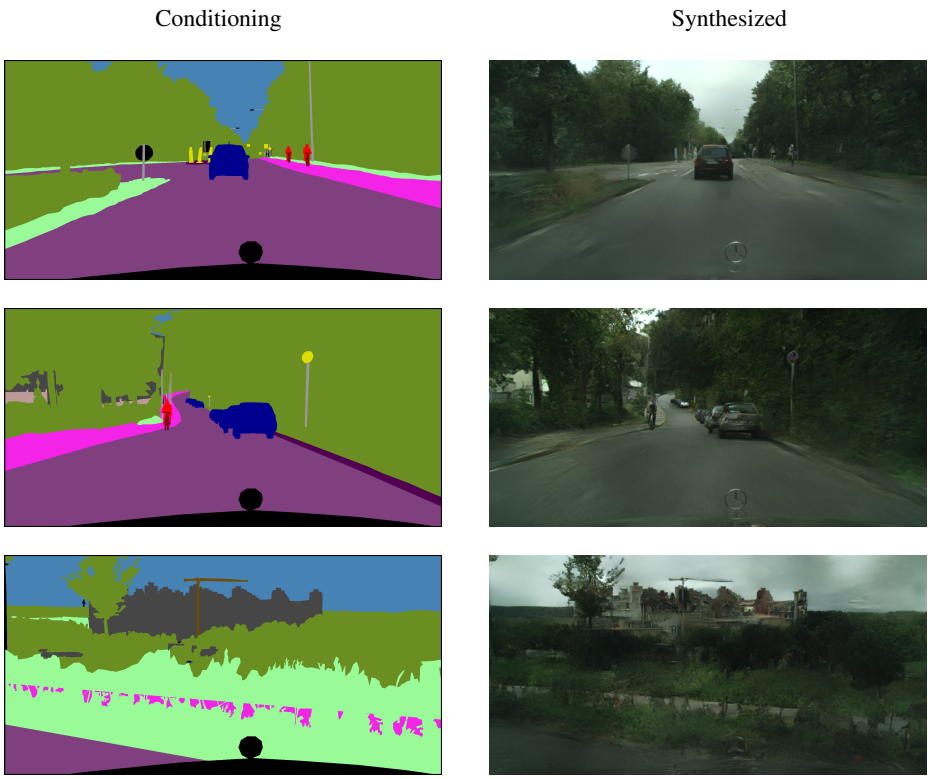

Fig. 3: Higher resolution image synthesis by our model taken with temperature 0.9. Please zoom in to see more details.

and show more diversity, while samples generated with lower temperatures have dim colors with small variations, but the structures are generally better well-maintained.

It can be observed that synthesis quality is not as good for very low temperatures, close to 0. Such low-temperature samples are known to be atypical and nonrepresentative, as discussed and exemplified in [3, 4] and [9, App. A.1]. This is a consequence of the geometry of high-dimensional spaces, where the highest-density regions tend to contain negligible mass overall (see again [4]), a notion formalized by the asymptotic equipartition property from information theory. Samples drawn at the low-temperature extreme are thus not well indicative of model quality, and flow-based models tend to use a temperature closer to 1 when generating samples in practical applications, with the optimal temperature being dependent on the model and the task.

D Other details of the models

This part elaborates on the details of the models used in the paper.

- The conditioning networks for actnorm and 1×1 convolution have three convolutional layers with 8, 4, and 2 output channels respectively, all with stride 1. The kernel size is 1 in the first layer and 3 in the next two layers.
- The first three fully connected layers in the conditioning networks have a depth of 32, 64, and 48 respectively. The depth of the last fully connected layer is determined by the number of parameters of the conditional operation. For a $c \times h \times w$ tensor, it generates $2c$ parameters for the actnorm operation and c^2 parameters for the 1×1 operation.
- The conditioning network for the coupling layer has two convolutional layers, both of which have 128 output channels with kernel size of 3 and stride of 1.
- The first version of C-Glow [7] has 3 Blocks each having 8 Flows. The number of filters in the convolutional layer of the coupling operation is 104, the convolutional layer of the conditioning network has 52 channels, and the width of the linear layer in the conditioning network is 26. These numbers are chosen to reflect the same ratio as the original numbers used by the authors while being feasible to fit in GPU memory.
- The second version of C-Glow [7] has 4 Blocks each having 16 Flows. The number of channels of the coupling layer, convolutional layer of condition networks and the width of the linear layer of the conditioning network are 512, 20, and 10 respectively. These were chosen so that its Glow is as similar as possible to our model and Dual-Glow (in terms of depth) while keeping the conditioning networks as deep as possible.
- Both Dual-Glow [11] and pix2pix [5] were trained using the default hyper-parameters specified in their official implementations.
- For synthesizing higher-resolution images, we trained a model with 4 Blocks each with 10 Flows for 100 epochs. In the first 50 epochs we used a learning rate of 10^{-4}. Next, we linearly decreased the learning rate to 0 in the remaining 50 epochs.

E Loss curve

By initializing all parameters using the same principles described in the original Glow paper [6], we found Full-Glow training to be very stable, as illustrated in Figure 4. In contrast, non-smooth training behavior was evident when training the pix2pix benchmark using the official code. This is consistent with mathematical findings that widely-used GAN setups such as WGAN need not converge since monotonicity is not assured [8].

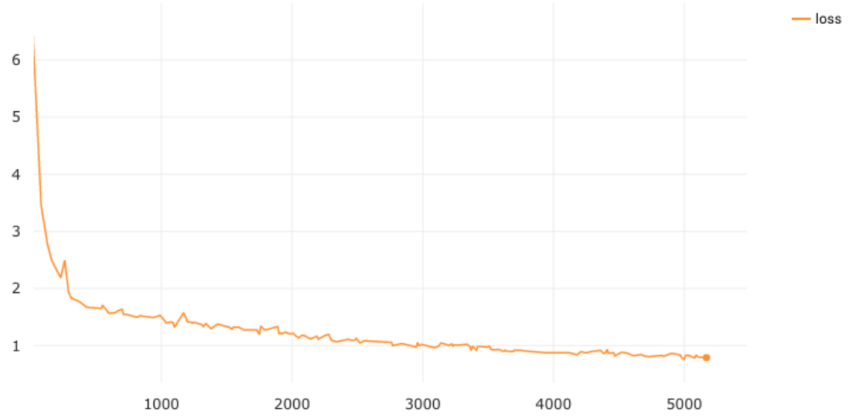

Fig. 4: Example Full-Glow loss curve demonstrating stable training.

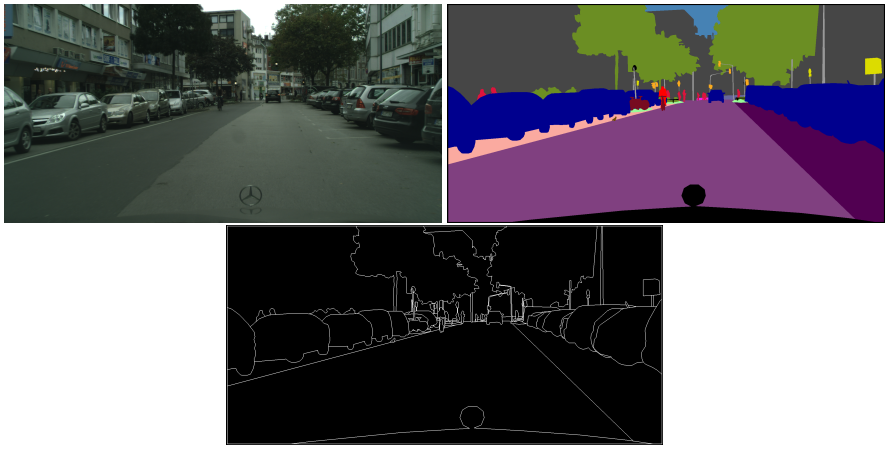

Fig. 5: Different modalities of an image in Cityscapes [1]. Boundary maps are not explicitly available but could be derived from available instance IDs.

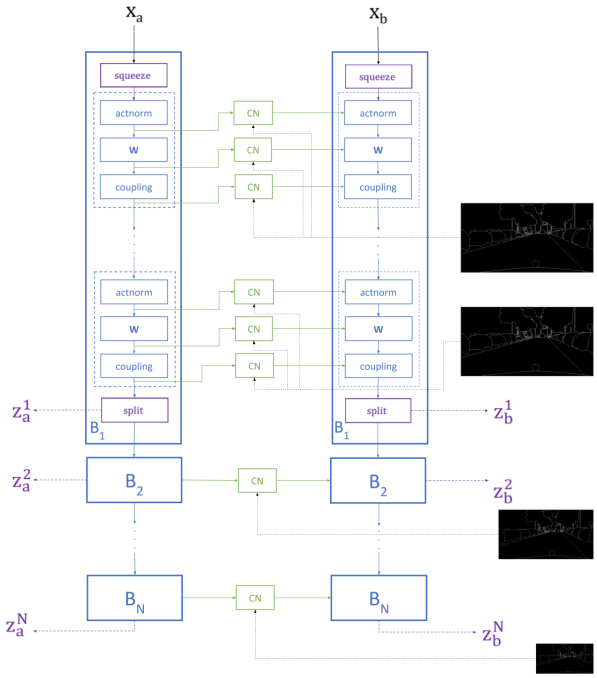

Fig. 6: The proposed architecture, making all the layers conditional by inserting conditioning networks. $\mathbf{x}_a$ and $\mathbf{x}_b$ are the image pairs in the source and target domains respectively. Conditioning networks also allow for using other side information for conditioning (here shown as boundary maps, down-sampled to match spatial dimension of each Block).

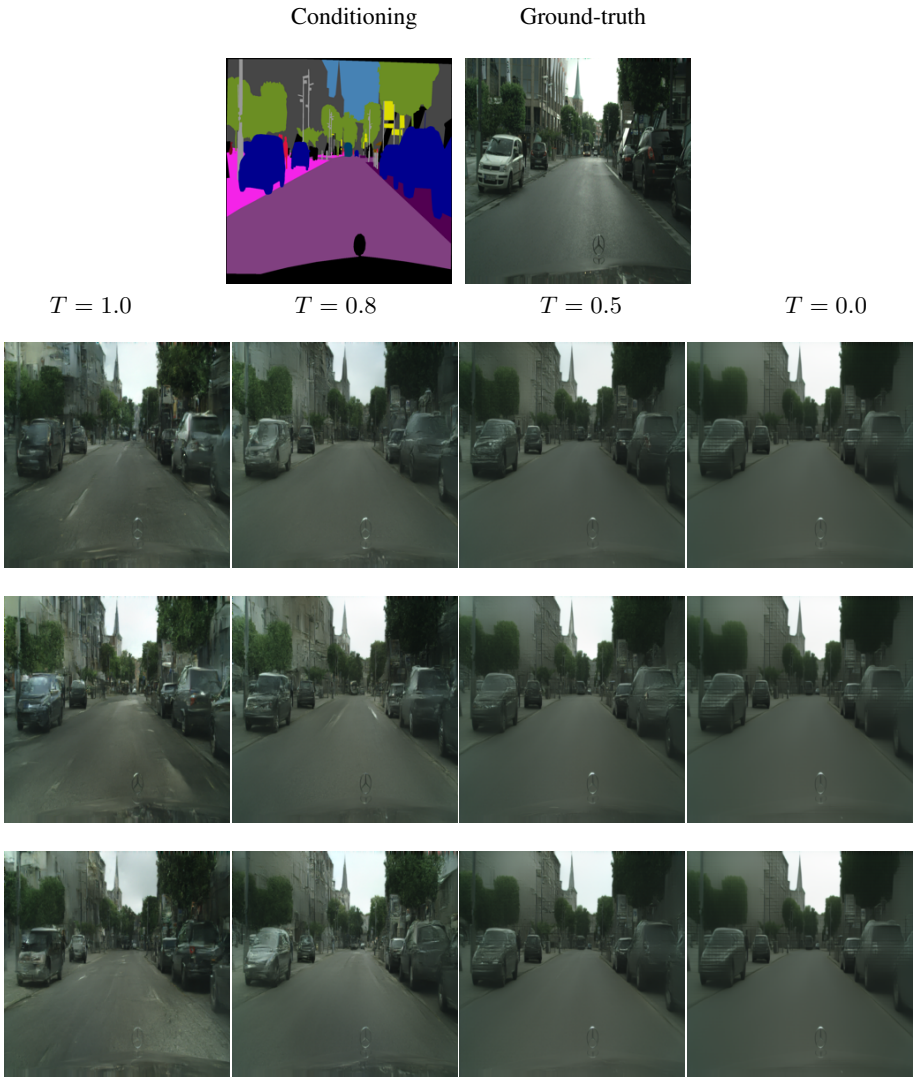

Fig. 7: Effect of temperature: for each row, samples with temperature 1.0, 0.8, 0.5, and 0.0 are shown from left to right. Columns represent 3 samples with the given temperature. It is clear that samples with higher temperature show more diversity.

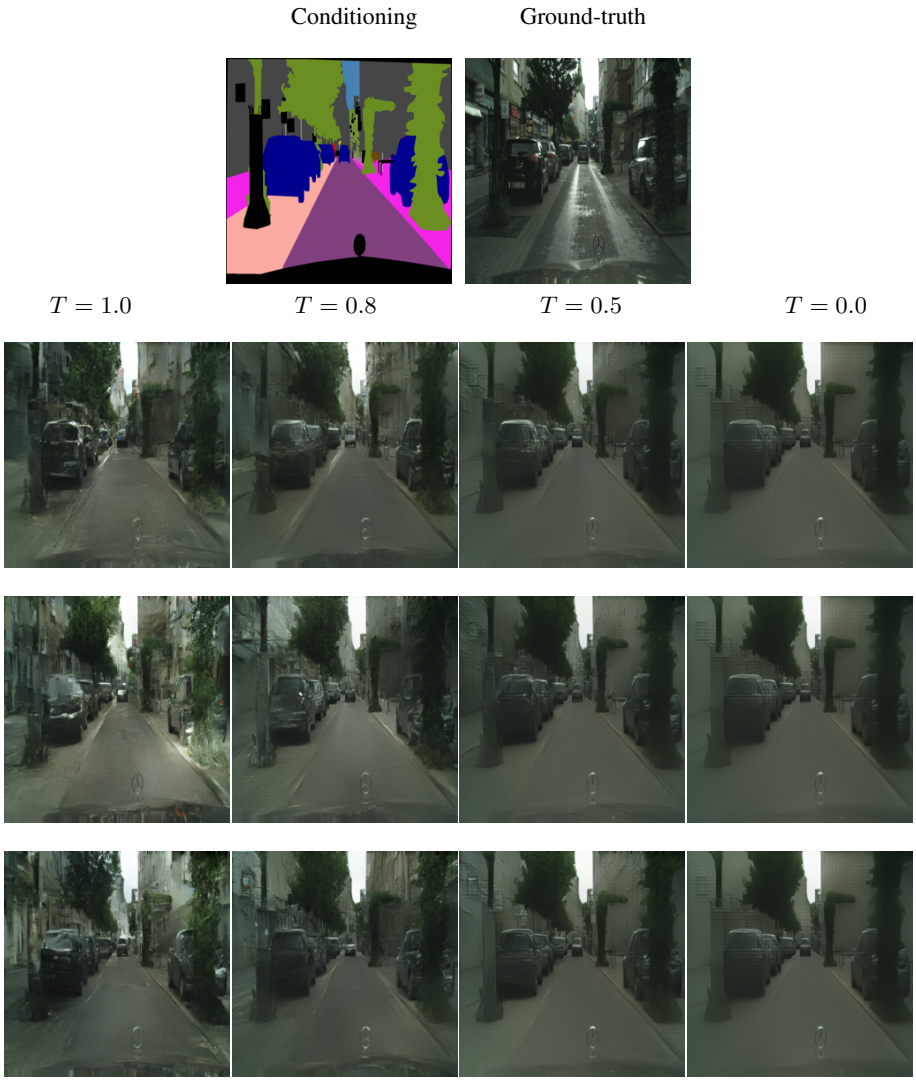

Fig. 8: Effect of temperature: for each row, samples with temperature 1.0, 0.8, 0.5, and 0.0 are shown from left to right. Columns represent 3 samples with the given temperature. It is clear that samples with higher temperature show more diversity.

Table 1: Performance of different configurations of our model, showing the effect of using boundary maps and deepening the models (in 256×256 resolution).

Model	Description	Total Flows	Params	Mean pixel acc.	Mean class acc.	Mean class IoU
Config. A - Baseline	4 Blocks, 16 Flows	64	155.3M	73.50 ± 0.13	29.13 ± 0.39	23.86 ± 0.30
Config. B - Using b-maps v.1	4 Blocks, 16 Flows	64	155.5M	74.95 ± 0.03	30.46 ± 0.27	25.24 ± 0.08
Config. C - Using b-maps v.2	4 Blocks, 16 Flows	64	155.5M	74.12 ± 0.13	28.96 ± 0.04	24.14 ± 0.07
Config. D - Deep model	5 Blocks, 24 Flows	120	431M	75.13 ± 0.28	29.71 ± 0.20	24.32 ± 0.30
Config. E - Deep hierarchical	5 Blocks, [40, 30, 20, 20, 10] Flows	120	348M	77.38 ± 0.36	31.46 ± 0.43	26.30 ± 0.34
Config. F - Deep with fewer Blocks v.1	4 Blocks, 30 Flows	120	290M	74.97 ± 0.01	29.93 ± 0.28	25.07 ± 0.19
Config. H - Deep with fewer Blocks v.2	**3 Blocks, 40 Flows**	**120**	**248M**	**77.90 ± 0.14**	**32.89 ± 0.35**	**27.82 ± 0.27**

References

1. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
2. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using Real NVP. In: International Conference on Learning Representations (2017)
3. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. In: International Conference on Learning Representations (2020)
4. Huszár, F.: Gaussian distributions are soap bubbles (2017), https://www.inference.vc/high-dimensional-gaussian-distributions-are-soap-bubbles
5. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
6. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: Neural Information Processing Systems (2018)
7. Lu, Y., Huang, B.: Structured output learning with conditional generative flows. In: AAAI Conference on Artificial Intelligence (2020)
8. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for GANs do actually converge? In: International Conference on Machine Learning. pp. 3481–3490 (2018)
9. van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., et al.: Parallel WaveNet: Fast high-fidelity speech synthesis. In: International Conference on Machine Learning. pp. 3918–3926 (2018)
10. Salimans, T., Bulatov, Y.: Gradient checkpointing (2018), https://github.com/openai/gradient-checkpointing
11. Sun, H., Mehta, R., Zhou, H.H., Huang, Z., Johnson, S.C., Prabhakaran, V., Singh, V.: DUAL-GLOW: Conditional flow-based generative model for modality transfer. In: IEEE International Conference on Computer Vision (2019)
12. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)