CoordGAN: Self-Supervised Dense Correspondences Emerge from GANs Supplementary Materials

Jiteng Mu[1] *, Shalini De Mello[2], Zhiding Yu[2], Nuno Vasconcelos[1],
Xiaolong Wang[1], Jan Kautz[2], Sifei Liu[2]
[1]UC San Diego, [2]Nvidia

1. Overview

In this supplementary material, we provide more details of the submission: We show quantitative results of structure swapping in Section 2; We provide ablation studies on the objectives, as described in Section 3; We conduct a user study on disentangling structure and texture by swapping the attributes of the generated images, as described in Section 4; We answer the question of how an MLP models explicit transformation between canonical and warped coordinate frames in Section 5; More implementation details are discussed in Section 6; We show at last in Section 7 the application of the finetuning the CoordGAN on other domains.

2. Quantitative Results on Structure Swapping

We present quantitative comparisons to the SOTA structure swapping method DiagonalGAN. We sample 5000 pairs of images, each pair with the same texture code and different structure codes. Each pair is evaluated by both the *LPIPS* (to measure how the structure varies) and the *ArcFace* (to evaluate whether identity changes) scores. CoordGAN has better performance: **0.75 over 0.65 for ArcFace**, and **0.55 over 0.50 for LPIPS**. The results indicate that, with the same texture code, different structure codes of CoordGAN produce images of larger structural variations, whereas DiagonalGAN tends to generate images with similar identities. This demonstrates the structure and texture are better disentangled with the proposed method.

3. Ablation Studies

Ablation for image synthesis. In this part, we study the effect of different loss functions for training the generator. As shown in Table 1, it is observed that the combination of the warp loss, structure swapping constraint and texture swapping constraint achieve the best performance. Without the texture swapping constraint, the texture code tends to take the majority of the variances while the warped coordinates are similar across different samples. Without the warp loss, the correspondence performance drops significantly. This suggests that regularizing the correspondence maps is crucial to extracting dense correspondence accurately.

	Disentanglement		Label Propagation	
	LPIPS↓	Arcface ↓	CelebA-HQ	DGAN-face
CoordGAN	0.10	0.32	52.25	23.78
w/o warp loss	0.11	0.22	24.84	10.52
w/o structure swap	0.18	0.51	46.52	18.53
w/o texture swap	0.64	0.90	45.96	17.66

Table 1. Ablation on generator losses on CelebAMask-HQ. We show that incorporating the all losses is essential to good disentanglement and label propagation performance (measured by IOU).

	Reconstruction			Label Propagation	
	LPIPS↓	Arcface ↓	MSE↓	CelebA-HQ	DGAN-face
CoordGAN	0.25	0.49	0.03	52.25	23.78
w/o latent consistency	0.28	0.59	0.04	46.19	21.50
w/o texture swap loss	0.23	0.47	0.03	50.83	23.53

Table 2. Ablation on encoder losses on CelebAMask-HQ. We show that incorporating all the losses is essential to faithfully reconstructing the input and encoding accurate correspondence.

Ablation for encoder. In this part, we fix the parameters of the generator and study the effects of different loss functions for training the encoder. Table 2 compares the reconstruction performance with respect to different loss combinations. We find that both the latent consistency and the texture loss are essential to achieving the best reconstruction performance. While removing the texture swapping loss results in lower reconstruction errors, we find the correspondence performance slightly decreases. Without the latent consistency loss, both reconstruction and correspondence performance drop significantly. This indicates that encouraging the encoded structure to match the learned distribution plays an important role to model accurate correspondence for real image inputs.

*Work done while an intern at Nvidia

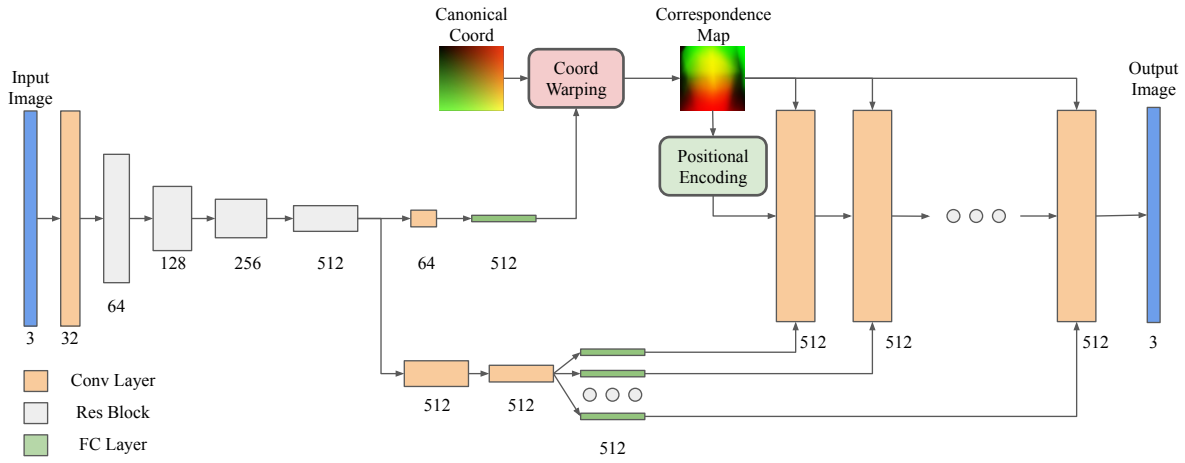

Figure 1. Auto-encoder architecture. The encoder takes an image as input and outputs a latent structure code and a latent texture code. Then the generator takes the predicted structure and texture latent codes and outputs images.

4. User Study on Attribute Swapping

We conduct a user study to further evaluate the disentanglement of structure and texture for the proposed CoordGAN and DiagonalGAN. Given a pair of images generated with the same structure code but diverse texture codes, we ask users on AMT to rate the pairs of images based on their structural similarity with a score from 1 to 5. A higher score indicates that the pair of images are more similar in terms of structure. Likewise, we ask users to rate the texture similarities of images generated with the same texture code but diverse structure codes. For each dataset, we randomly sample 200 image pairs and each pair of images is rated independently by three individuals.

Table 3 shows that CoordGAN significantly outperforms DiagonalGAN in terms of texture-swap ratings on both CelebAMask-HQ and Stanford Cars datasets. This further suggests that the proposed approach of modeling the structure with a coordinate space effectively disentangles fine-grained structure from texture. While both methods perform similarly in terms of structure-swap studies, we emphasize that many structure-swapped pairs from DiagonalGAN are just slightly different as the learned structure code is only responsible for coarse viewpoint. More visualization results are shown in Figures 3 to 9.

	CelebAMask-HQ		Stanford Cars	
	Struc-swap ↑	Text-swap ↑	Struc-swap ↑	Text-swap ↑
DiagonalGAN	3.39	2.83	3.76	3.11
CoordGAN	3.32	3.68	3.58	3.77

Table 3. User study on attribute swapping. Struc-swap denotes the setting where the pair of images are generated with the same texture code but different structure codes; Text-swap denotes the setting where the pair of images are generated with the same structure code but diverse textures.

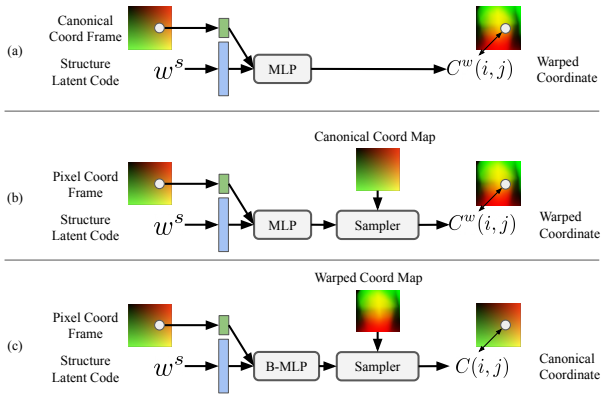

Figure 2. Coordinate warping network design. Sampler indicates the grid sampling operation.

5. Coordinate Warping Network Analysis

In this section, we validate that the coordinate warping network, designed as an MLP conditioned on the sampled structure code, formulates an explicit geometric transformation between the canonical coordinate frame and a warped coordinate frame. Formally, a geometric transformation between two coordinate frames should satisfy two properties: (1) one-to-one mapping exists between each element of two sets; (2) the transformation is invertible.

In the following, we show that the coordinate warping network equivalently outputs a flow w.r.t. each input coor-

	CelebA-HQ			Stanford Cars	
	LPIPS ↓	Arcface ↓	FID ↓	LPIPS ↓	FID ↓
CoordGAN	0.22	0.38	16.16	0.21	24.27
CoordGAN-B	0.19	0.38	15.45	0.18	23.95

Table 4. Image generation results on Coordinate Warping Network with backward MLP (CoordGAN-B).

	CelebA-HQ	DGAN-face	DGAN-car
CoordGAN	52.25	23.78	13.23
CoordGAN-B	54.51	25.44	12.59

Table 5. Label propagation results on Coordinate Warping Network with backward MLP (CoordGAN-B). Measured by IOU.

dinate, which satisfies the aforementioned property (1). We begin by defining another pixel coordinate frame $\mathcal{P}$ denoting pixel locations. This is numerically similar to the canonical coordinate frame, where coordinates are normalized to the range [-1, 1]. For example, $\mathcal{P}(1,1) = (1,1)$ indicates the bottom right pixel is of value (1,1). It then follows that the proposed coordinate warping network, as shown in Figure 2 (a), is equivalent to the architecture in Figure 2 (b). This comes from two facts: (i) the pixel coordinate frame is constructed exactly the same as the canonical coordinate frame; (ii) the grid sampling operation in Figure 2 (b) outputs exactly the same value as the MLP output as the MLP is constrained to output values from -1 to 1. Therefore, we show that, given a structure code, the MLP learns a transformation from the canonical coordinate frame to the warped coordinate frame.

In addition, we build another backward MLP to satisfy the second property, such that warped coordinates can be back to canonical coordinates with the same structure code. Specifically, as shown in Figure 2 (c), we construct another three-layer MLP to map warped coordinates to canonical coordinates. To distinguish the MLP mapping from canonical coordinates to warped coordinates, we refer to this one as the Backward MLP (B-MLP). We train CoordGAN with the additional B-MLP (CoordGAN-B) from scratch, where the B-MLP is supervised with an additional L1 loss between the predicted canonical coordinate frame and ground-truth canonical coordinate frame. As show in Table 4 and Table 5, CoordGAN-B achieves on average better performance in both image synthesis and label propagation.

To this end, we prove that the proposed MLP models an explicit geometric transformation between the canonical coordinate frame and a warped coordinate frame. We opt for an MLP as it preserves the order of the coordinates in the canonical coordinate frame due to its continuity. Since an explicit transformation is learned, it ensures that, when the MLP outputs the same coordinate given two different structure codes, these two positions are corresponding to the same coordinate in the canonical frame.

6. Implementation Details

We introduce the training details and specify the architecture for each module of our network.

6.1. Architecture

Generator. Both the sampled structure and texture codes are 512-dimensional. The structure and texture mapping networks are implemented with an 8-layer MLP with a latent dimension of 512. The coordinate warping network, conditioned on a latent structure code, is implemented with a three-layer MLP. A tanh function is used at the output of the coordinate mapping network to ensure that the output is within a valid coordinate space. The dense correspondence map is passed to a positional encoding layer where, a Fourier embedding with 512 channels is obtained by the application of a 1×1 convolution followed by a sine function. In all experiments, the canonical coordinate map and the correspondence map are defined with a spatial resolution of 128. The modulated generator consists of 10 layers and all layers are with 512 channels. The design of each layer is similar to StyleGAN2. We follow StyleGAN2 to inject the latent texture code into different layers of the modulated generator via weight modulation/demodulation. The dense correspondence map is concatenated with all 10 layers of the modulated generator, as shown in Figure 1. To generate higher resolution images, another two upsampling blocks are added to the last layer of the modulated generator. Note that the correspondence map is not concatenated to these upsampling blocks. Skip connections are used to combine features for every two layers from intermediate feature maps to RGB values.

Patch Discriminator. The patch discriminator architecture for the structure-swapping constraint is designed following Swapping Autoencoder. The patch discriminator consists of a feature extractor of 5 downsampling residual blocks, 1 residual block, and 1 convolutional layer, and a classifier. Specifically, 8 randomly cropped patches from the same image are used as reference. Each patch is cropped randomly from $\frac{1}{8}$ to $\frac{1}{4}$ of the image dimensions for each side. All cropped patches are resized to $\frac{1}{4}$ of the image size and then input to the patch discriminator. Each patch is passed to multiple downsampling blocks to obtain a feature vector. The feature vectors of all reference patches are averaged and then concatenated with a feature vector from a real or fake patch. The real patches are patches from the same image as the reference patches and fake patches are from a structure-swapping image. The classifier finally determines whether the concatenated feature vector is real or fake.

Encoder. Given an image, the encoder produces two 512 dimensional vectors. As shown in Figure 1, our encoder network design follows Swapping Autoencoder. The difference is that instead of outputting a feature map for the structure code, the proposed design outputs a 512 dimen-

sional structure code. Specifically, 4 downsampling residual blocks are first applied to produce an intermediate tensor, which then produces separate features for the structure code and texture codes. The structure code is produced by first applying 1-by-1 convolutions to the intermediate tensor, reducing the number of channels and then applying a fully-connected layer. The texture codes in the W+ space are produced by applying stride convolutions, average pooling, and then different dense layers.

6.2. Training Details

To train the generator, we follow StyleGAN2 and use the non-saturating GAN loss and lazy R1 regularization. The R1 regularization is also applied to the patch discriminator. The weight of the R1 regularization is 10.0 for the image discriminator and is 1.0 for the patch discriminator. We use the ADAM optimizer with a learning rate of 0.002 and with $\beta_1 = 0.0$ and $\beta_2 = 0.99$. The batch size is set to 16 with 8 GPUs. Coefficients for different losses are set as following: $\lambda_{cham} = 100$, $\lambda_{GAN} = 2$, $\lambda_t = 5$, $\lambda_{warp} = 5$, $\lambda_s = 1$. To warm up training, for the first 20k iterations, λ_{warp}, λ_t, and λ_s are linearly increased from 0. For celebAMask-HQ, we train the generator for 300k iterations at the resolution 128×128 and then train at a high resolution for another 200k iterations. For the Stanford Cars and AFHQ-cat datasets, we train the generator for 300k iterations at the resolution 128×128. The hyper parameters for training the encoder are selected as following: $\lambda_{rec} = 10$, $\lambda_{con} = 10$, $\lambda_t = 5$. The encoder is trained for 200k iterations.

7. Application in Other domains

In this section, we show that the CoordGAN can handle structure texture transfer on other domains, e.g., paintings. Specifically, we finetune the CelebAMask-HQ pretrained model at the resolution of 512×512 on the metfaces dataset [1]. The metfaces dataset contains 1336 high-quality images at 1024×1024 resolution. Following [2], we freeze the first three high resolution layers of the discriminator during finetuning. Furthermore, to enable texture swapping across different domains, we fix the weights of the structure mapping network and coordinate mapping network. As show in Figure 10, we qualitatively demonstrate that, CoordGAN can generate arts with high quality by combining the structure representation learned from real images with texture codes learned from arts. Note that the structure-texture disentanglement is still well maintained.

References

[1] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020. 4

[2] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze discriminator: A simple baseline for fine-tuning gans. *CoRR*, abs/2002.10964, 2020. 4

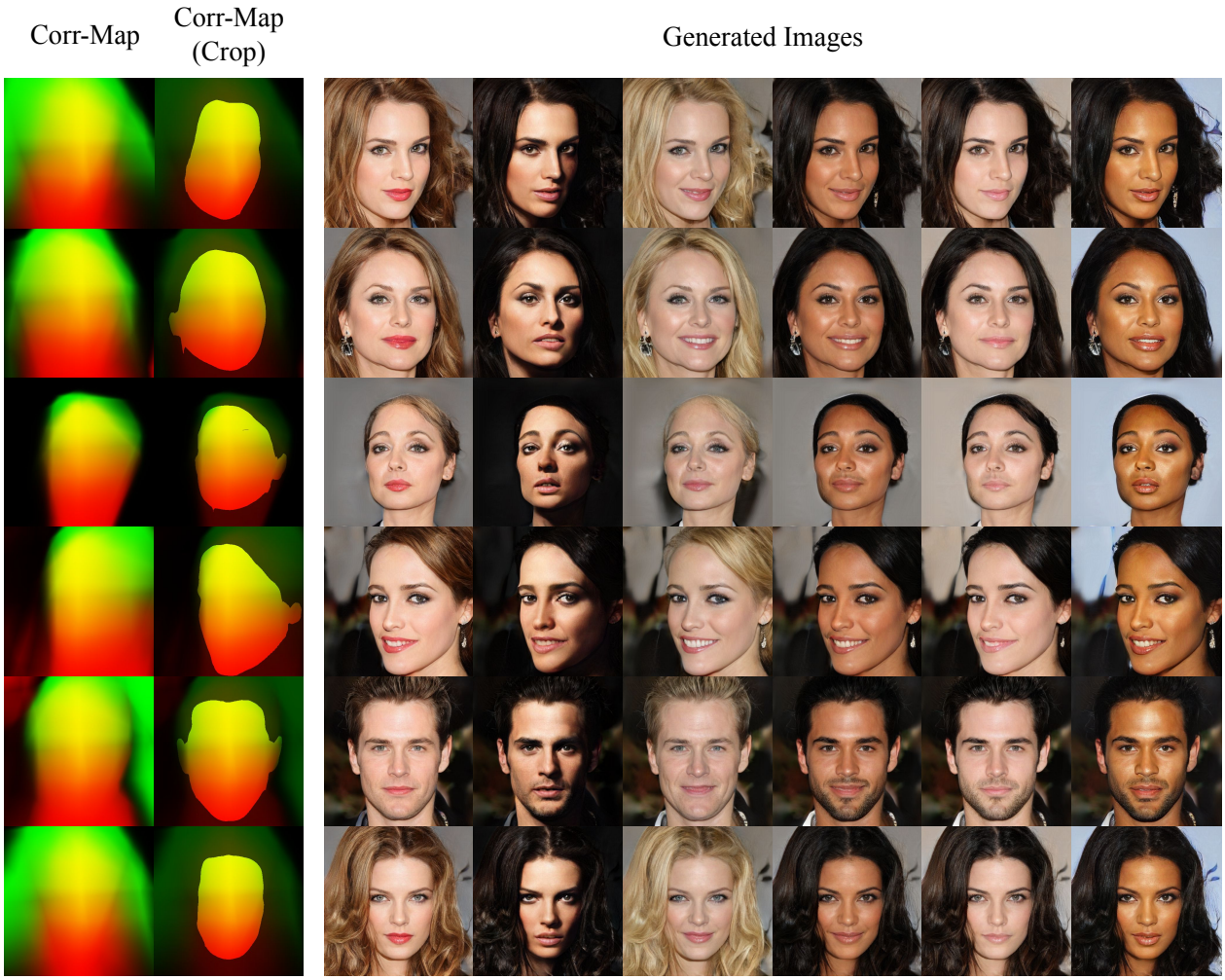

Figure 3. Images synthesised by the proposed CoordGAN model: each row displays images with the same structure but different textures; in each column, structure varies while texture is fixed. The correspondence maps (Corr-Map) controlling the structure of the synthesized images are shown in the first column of each row. For better visualization, we use off-the-shelf segmentation models to highlight the foreground areas of all the predicted correspondence maps, as shown with Corr-Map (Crop).

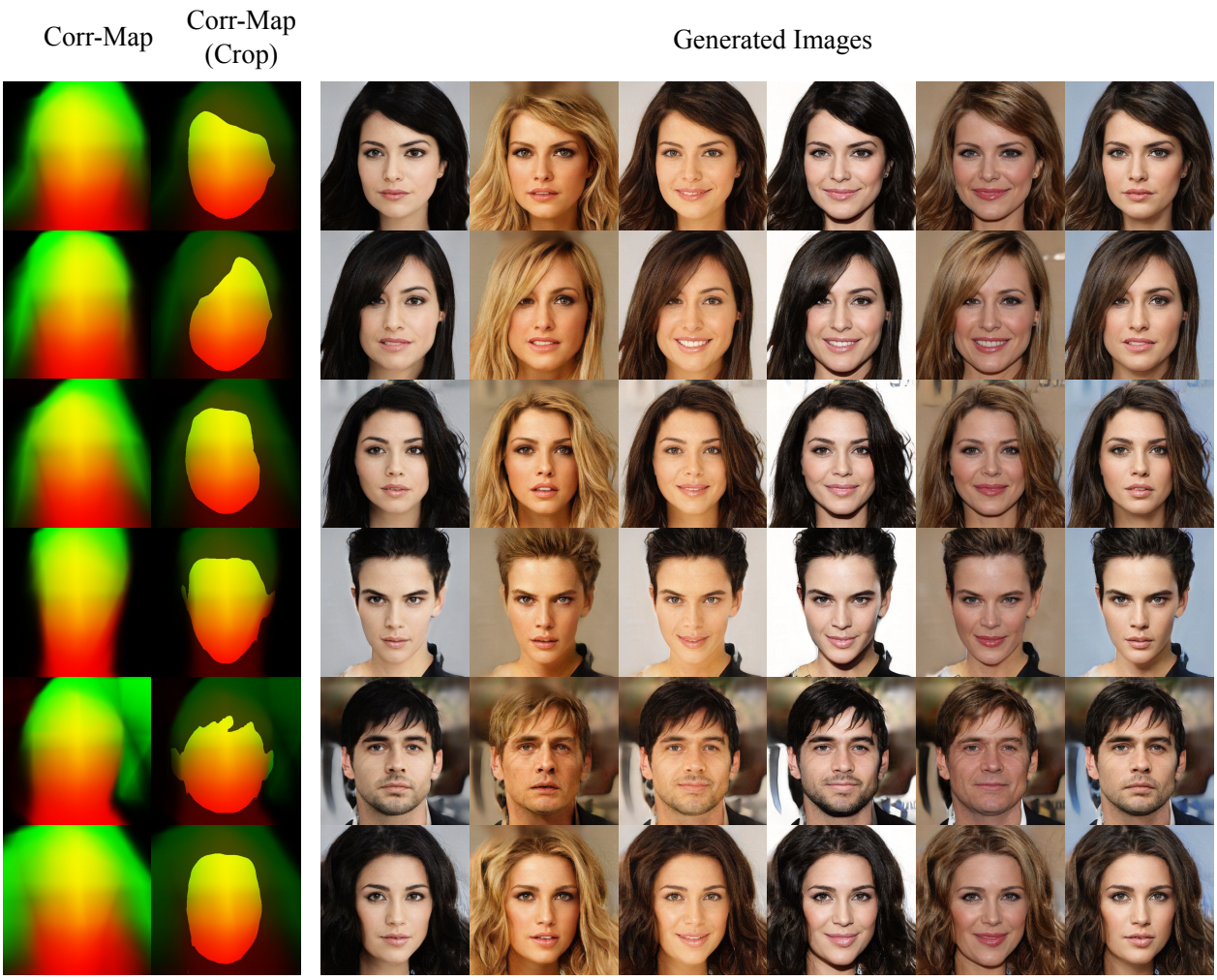

Figure 4. Images synthesised by the proposed CoordGAN model: each row displays images with the same structure but different textures; in each column, structure varies while texture is fixed. The correspondence maps (Corr-Map) controlling the structure of the synthesized images are shown in the first column of each row. For better visualization, we use off-the-shelf segmentation models to highlight the foreground areas of all the predicted correspondence maps, as shown with Corr-Map (Crop).

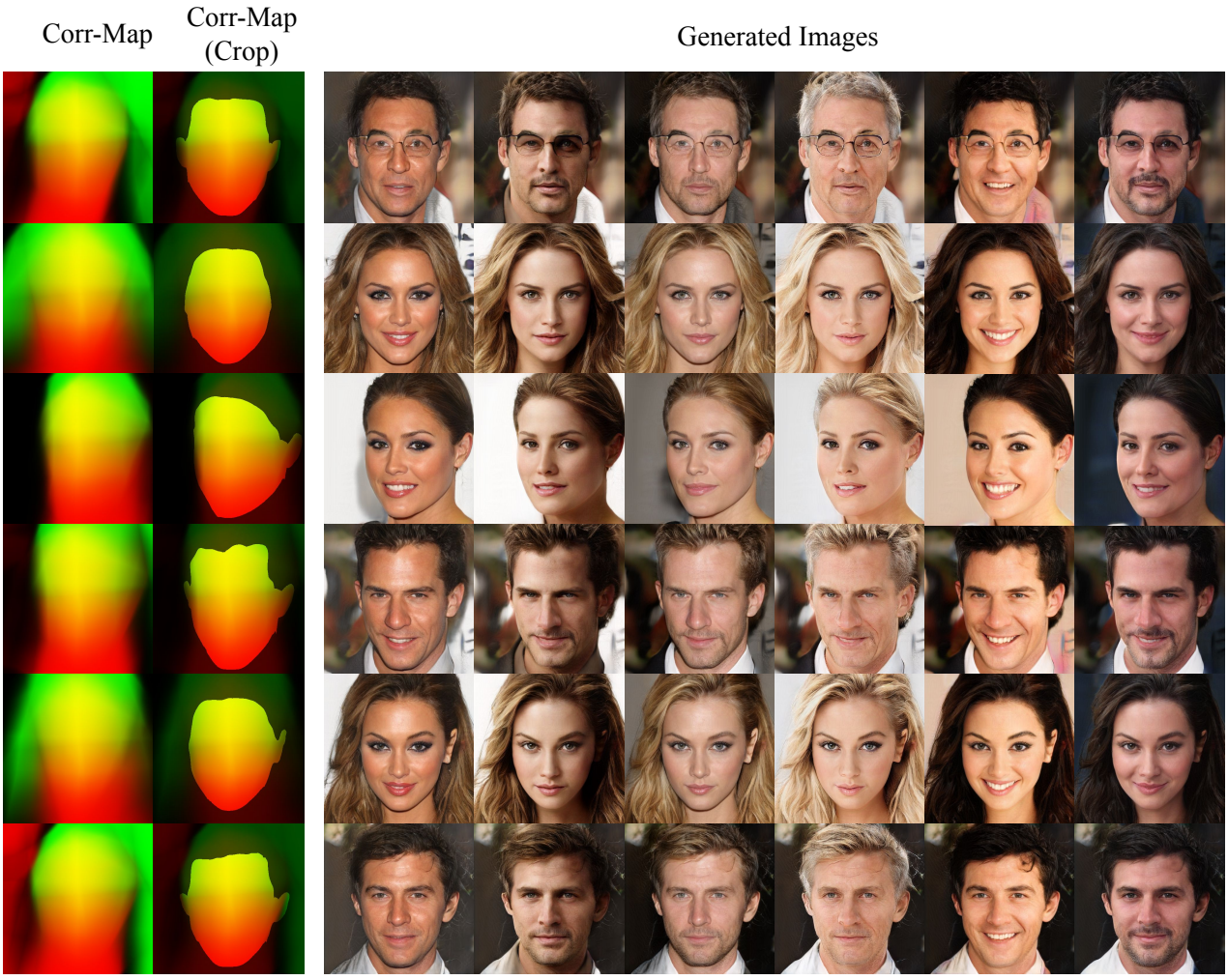

Figure 5. Images synthesised by the proposed CoordGAN model: each row displays images with the same structure but different textures; in each column, structure varies while texture is fixed. The correspondence maps (Corr-Map) controlling the structure of the synthesized images are shown in the first column of each row. For better visualization, we use off-the-shelf segmentation models to highlight the foreground areas of all the predicted correspondence maps, as shown with Corr-Map (Crop).

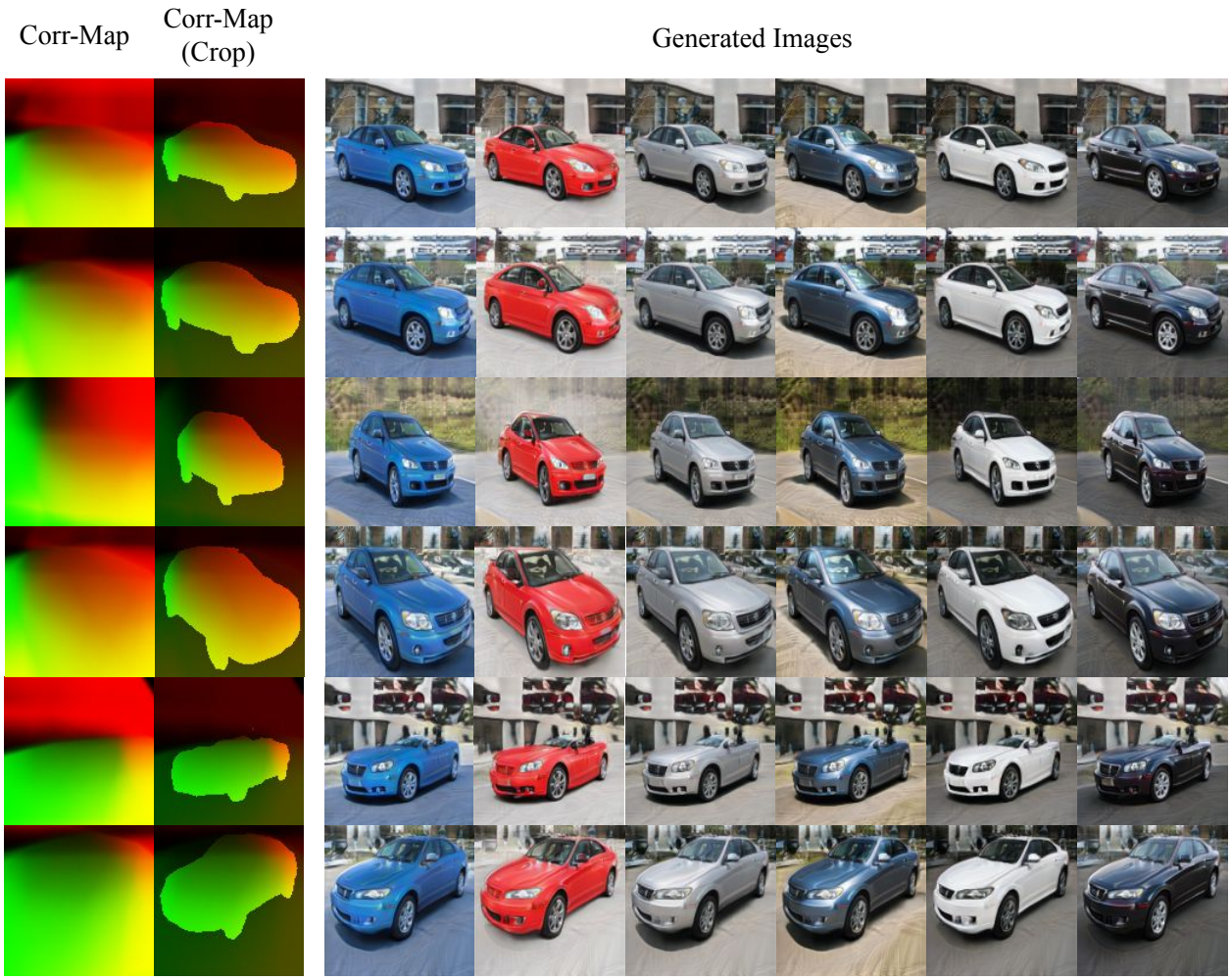

Figure 6. Images synthesised by the proposed CoordGAN model: each row displays images with the same structure but different textures; in each column, structure varies while texture is fixed. The correspondence maps (Corr-Map) controlling the structure of the synthesized images are shown in the first column of each row. For better visualization, we use off-the-shelf segmentation models to highlight the foreground areas of all the predicted correspondence maps, as shown with Corr-Map (Crop).

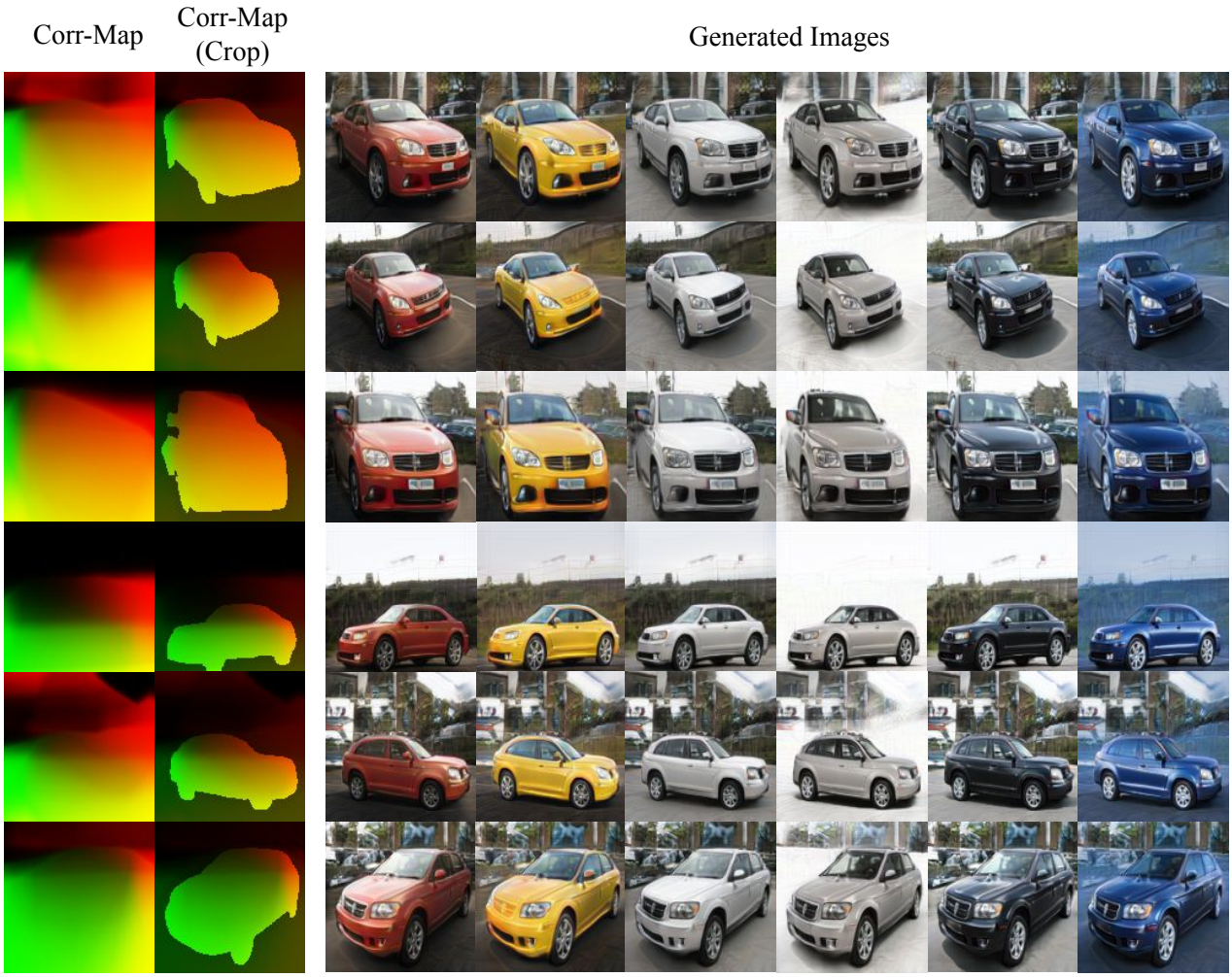

Figure 7. Images synthesised by the proposed CoordGAN model: each row displays images with the same structure but different textures; in each column, structure varies while texture is fixed. The correspondence maps (Corr-Map) controlling the structure of the synthesized images are shown in the first column of each row. For better visualization, we use off-the-shelf segmentation models to highlight the foreground areas of all the predicted correspondence maps, as shown with Corr-Map (Crop).

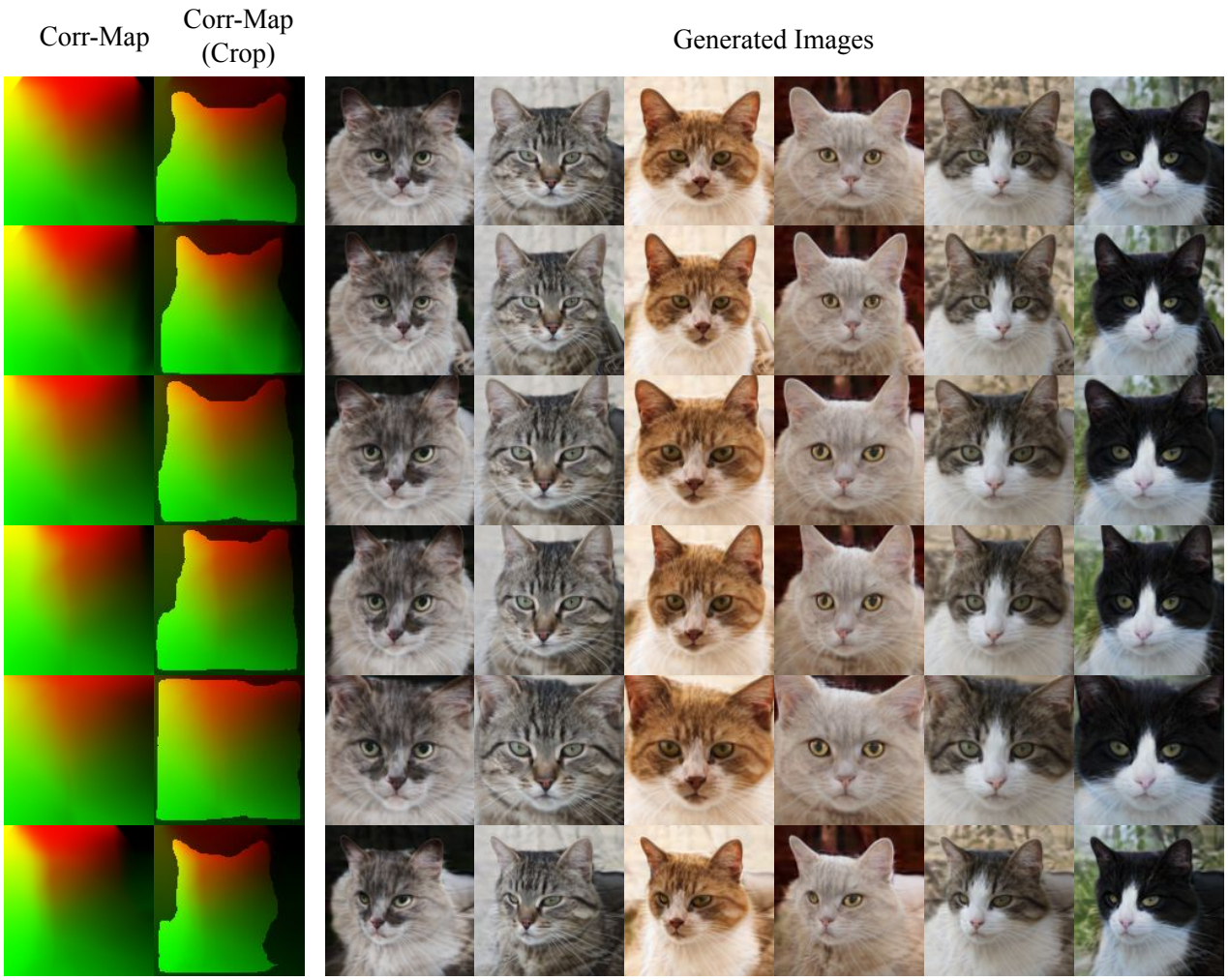

Figure 8. Images synthesised by the proposed CoordGAN model: each row displays images with the same structure but different textures; in each column, structure varies while texture is fixed. The correspondence maps (Corr-Map) controlling the structure of the synthesized images are shown in the first column of each row. For better visualization, we use off-the-shelf segmentation models to highlight the foreground areas of all the predicted correspondence maps, as shown with Corr-Map (Crop).

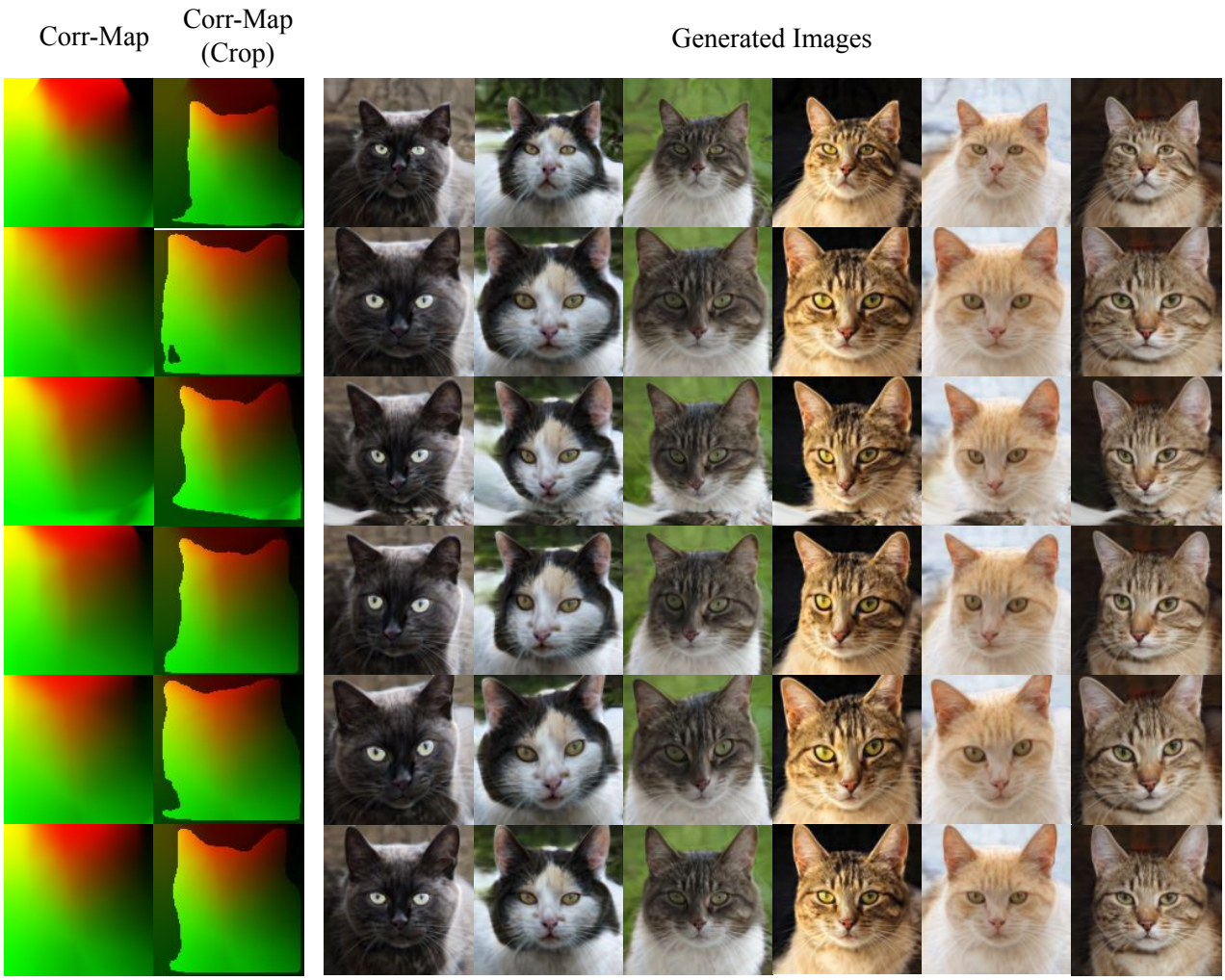

Figure 9. Images synthesised by the proposed CoordGAN model: each row displays images with the same structure but different textures; in each column, structure varies while texture is fixed. The correspondence maps (Corr-Map) controlling the structure of the synthesized images are shown in the first column of each row. For better visualization, we use off-the-shelf segmentation models to highlight the foreground areas of all the predicted correspondence maps, as shown with Corr-Map (Crop).

Figure 10. Images synthesised by the proposed CoordGAN model: the first row displays correspondence maps; from the second row to the bottom, each row displays images with the same texture but different structures; in each column, texture varies while structure is fixed.