Synthesizing Anyone, Anywhere, in Any Pose

Håkon Hukkelås Frank Lindseth
Norwegian University of Science and Technology
hakon.hukkelas@ntnu.no

A. Experimental Details

All models are trained with Pytorch 1.12 [11] on 4 NVIDIA A100-80GB. FID and FID_{CLIP} are computed with Torch Fidelity [10]. For qualitative examples, we use multi-modal truncation [8] for sampling diverse high quality samples. TriA-GAN is computationally efficient, where Config E processes $\sim$ 25 images per second on an NVIDIA RTX 3090 for single-image inference with unoptimized Pytorch [11]. For computing OKS, we use the VITPose-H* trained on COCO [6], AI Challenger, MPII and CrowdPose

Discriminator Architecture We use identical discriminators architectures for the different resolutions. Each D_ℓ (inputting features from the projection P_ℓ) consists of three convolutions with 512 channels, where the output of D_ℓ is half the spatial resolution of P_ℓ. We use spectral normalization for each convolution, and each convolution is followed by BatchNorm2d [4] and LeakyReLU [7], except the last. In total, the discriminator has 5.3M trainable parameters per feature network. For convolutional feature networks, we upsample the image to 288×160, whereas for ViT we upsample/downsample to 224×224. —

Generator Architecture Our generator architecture is similar to the architecture used by [2] with the modifications stated in the main paper and the following. We change the operation order of each convolution to instance normalization $\rightarrow$ style modulation $\rightarrow$ convolution. We use exponential moving average (EMA) [1] for the generator parameters with a warmup period following [5].

Training Hyperparameters Experimental hyperparameters are given in Table 1.

B. Cleaning the FDH Dataset

We clean the FDH dataset by refining the keypoint annotations with the top-down pose estimation model VITPose [12] [1]. VITPose [12] estimates 17 keypoints following the COCO [6] format given the image from the FDH dataset and the minimal enclosing bounding box of the embedding mask (named E_mask in the FDH dataset). Given the original keypoints and the new keypoints from VITPose, we select one of them given how well the annotation matches the DensePose annotation (from CSE [9]) in the FDH dataset. Specifically, by using pixel-to-vertex correspondences we segment each surface pixel into a semantic body part [2]. Then, we count the number of keypoints matches to the correct body part (e.g. the keypoint "eye" should match to the body part "head"). The annotation with the highest percentage correct matches is selected. From this selection, 19,722 of 30K images are updated for the validation dataset, and 1,199,927 out of 1,829,496 images in the training dataset are updated.

C. Qualitative Examples of Different Discriminator Feature Networks

The generated images are given in Figure 1.

D. Random Generated Examples and Comparison Surface-Guided GANs

Randomly selected images comparing TriA-GAN to SG-GAN [2] are given in the following figures: Figure 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, and 18.

E. TriA-GAN *vs*. SG-GANs for Anonymization

We have integrated TriA-GAN in DeepPrivacy2 [2] to support anonymization. Figure 21 compares TriA-GAN to Surface-Guided GANs [3] (DeepPrivacy2 variant [2]). Note that the majority of pedestrians are not anonymized by SG-GAN, as DensePose fails to detect pedestrians further away from the camera. In addition, we note that the synthesis quality of TriA-GAN is notably better for all pedestrians in the scene.

[1] We use the VITPose-H* trained on COCO [6], AI Challenger, MPII and CrowdPose

[2] We use an open source semantic segmentation of the SMPL model, found here.

Table 1. Training hyperparameters. * Batch size/channel size is given per resolution, where "18" refers to the resolution 18×10. † Decoder is symmetric.

	Config A-D	Config E
Adam parameters	lr=0.002, $\beta_1 = 0.0, \beta_2 = 0.099$	Same
GPUs	4x A100-80GB	8x A100-80GB
Batch size*	18: 512, 36: 512, 72: 512	18: 1024, 36: 1024, 72: 1024, 144: 512, 288: 128
EMA	0.9976	Same
Discriminator trainable parameters	5.3M per feature network	Same
Data Augmentation	Horizontal flip	Same
Number of images seen by the discriminator	50M each resolution	18: 300M, 36: 200M, 72: 160M, 144: 110M, 288: 110M
Generator parameters (72×40) (Config A-D)	62.2M	110.4M
Generator parameters (288×160)	Not trained	124.2M
Convolution Channels*	18: 512, 36: 512, 72: 512, 144: 256, 288: 128	Same
Number of residual blocks per generator encoder block †	1	2

References

[1] Yasin Yazıcı Chandrasekhar, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, and Vijay. The Unusual Effectiveness of Averaging in GAN Training. In *International Conference on Learning Representations*, 2018. 1

[2] Håkon Hukkelås and Frank Lindseth. DeepPrivacy2: Towards Realistic Full-Body Anonymization. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1329–1338. IEEE, jan 2023. 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23

[3] Håkon Hukkelås, Morten Smebye, Rudolf Mester, and Frank Lindseth. Realistic Full-Body Anonymization with Surface-Guided GANs. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1430–1440. IEEE, jan 2023. 1, 23

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*, pages 448–456, 2015. 1

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116. IEEE, jun 2020. 1

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *European conference on computer vision*, volume 8693 LNCS, pages 740–755. Springer, Cham, 2014. 1

[7] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013. 1

[8] Ron Mokady, Omer Tov, Michal Yarom, Oran Lang, Inbar Mosseri, Tali Dekel, Daniel Cohen-Or, and Michal Irani. Self-Distilled StyleGAN: Towards Generation from Internet Photos. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9. ACM, aug 2022. 1

[9] Natalia Neverova, David Novotny, Vasil Khalidov, Marc Szafraniec, Patrick Labatut, and Andrea Vedaldi. Continuous Surface Embeddings. In *Advances in Neural Information Processing Systems*, volume 33, pages 17258–17270. Curran Associates, Inc., nov 2020. 1

[10] 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. 1

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and Others. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[12] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. *arXiv preprint arXiv:2204.12484*, 2022. 1

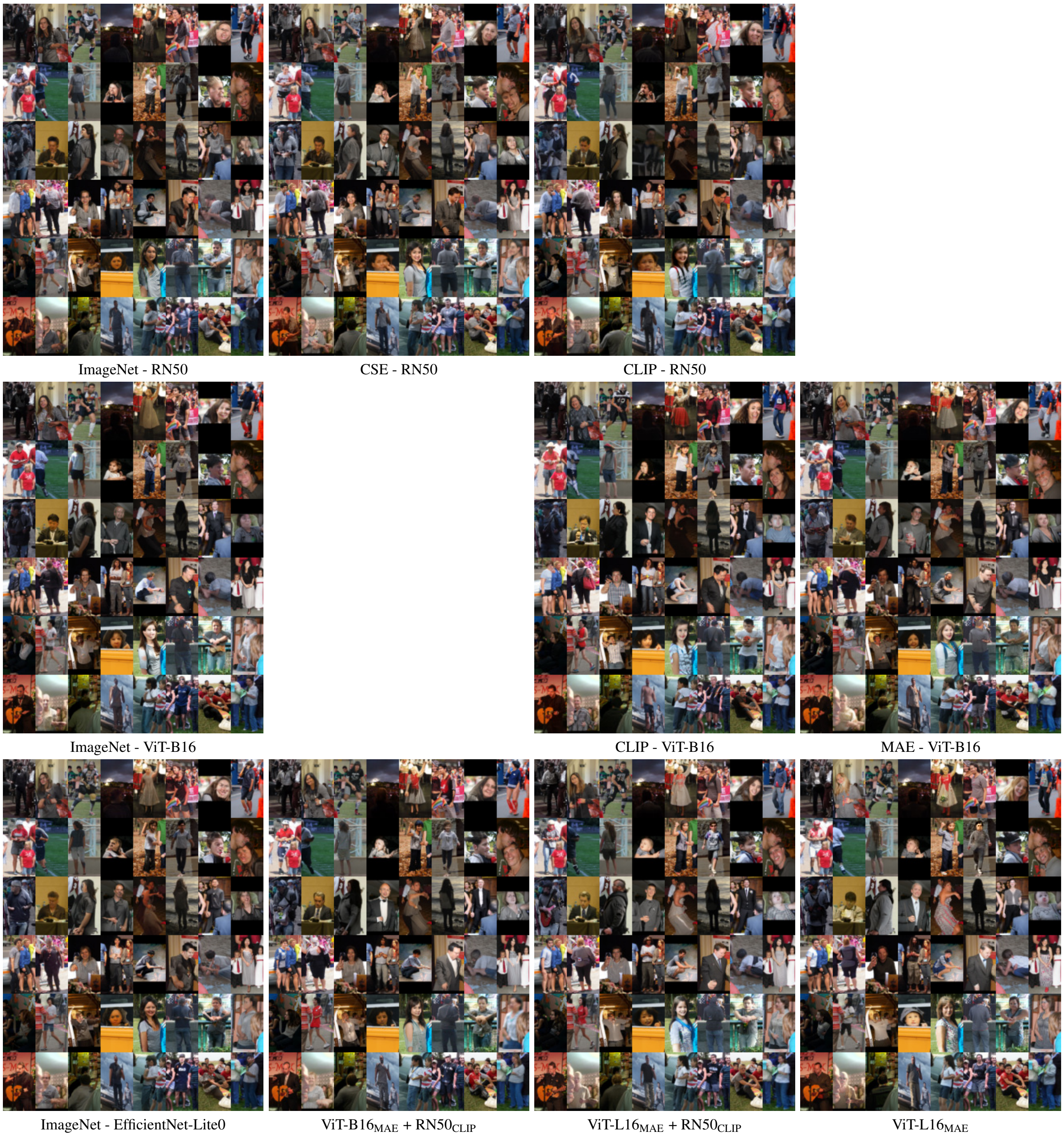
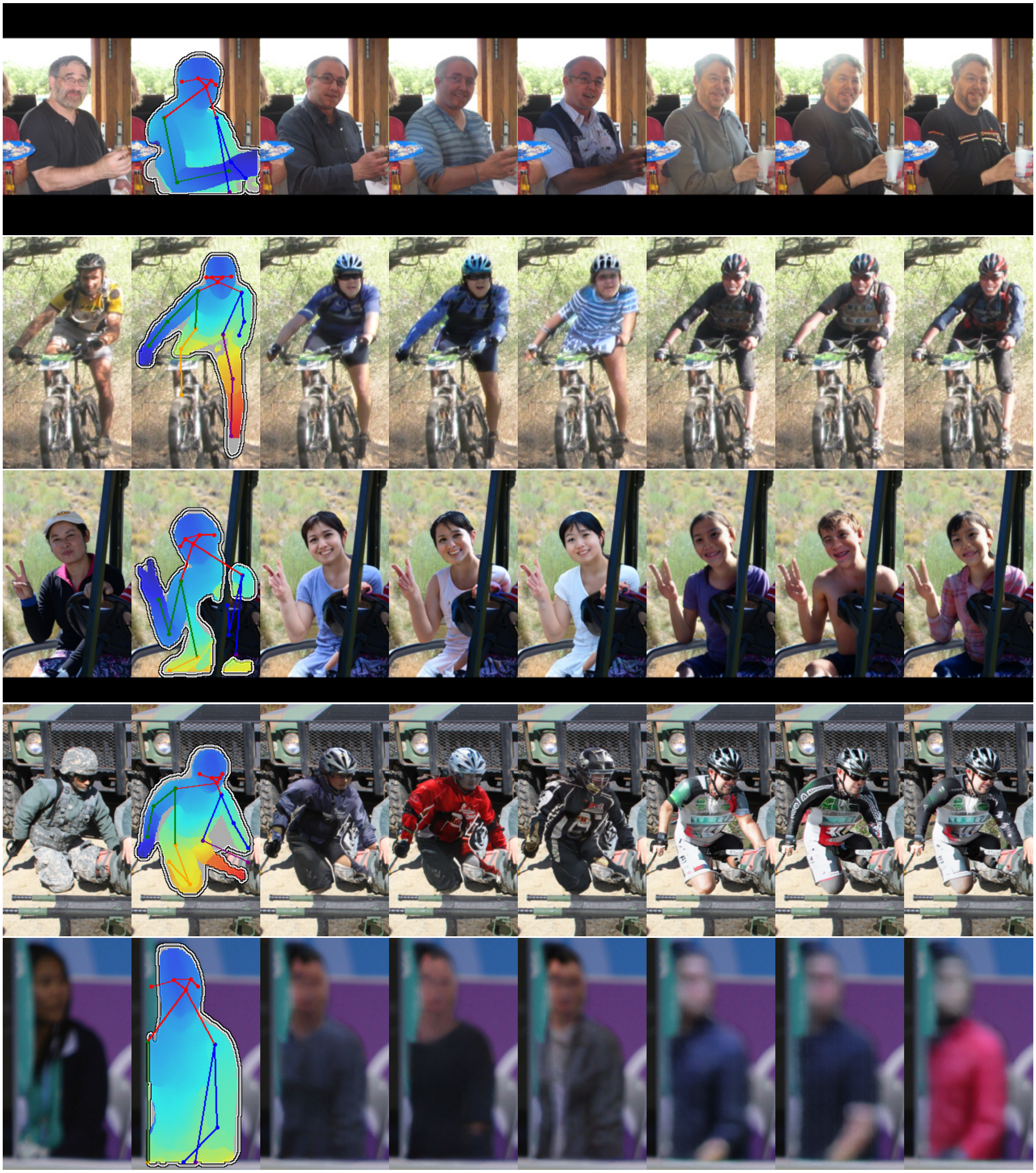
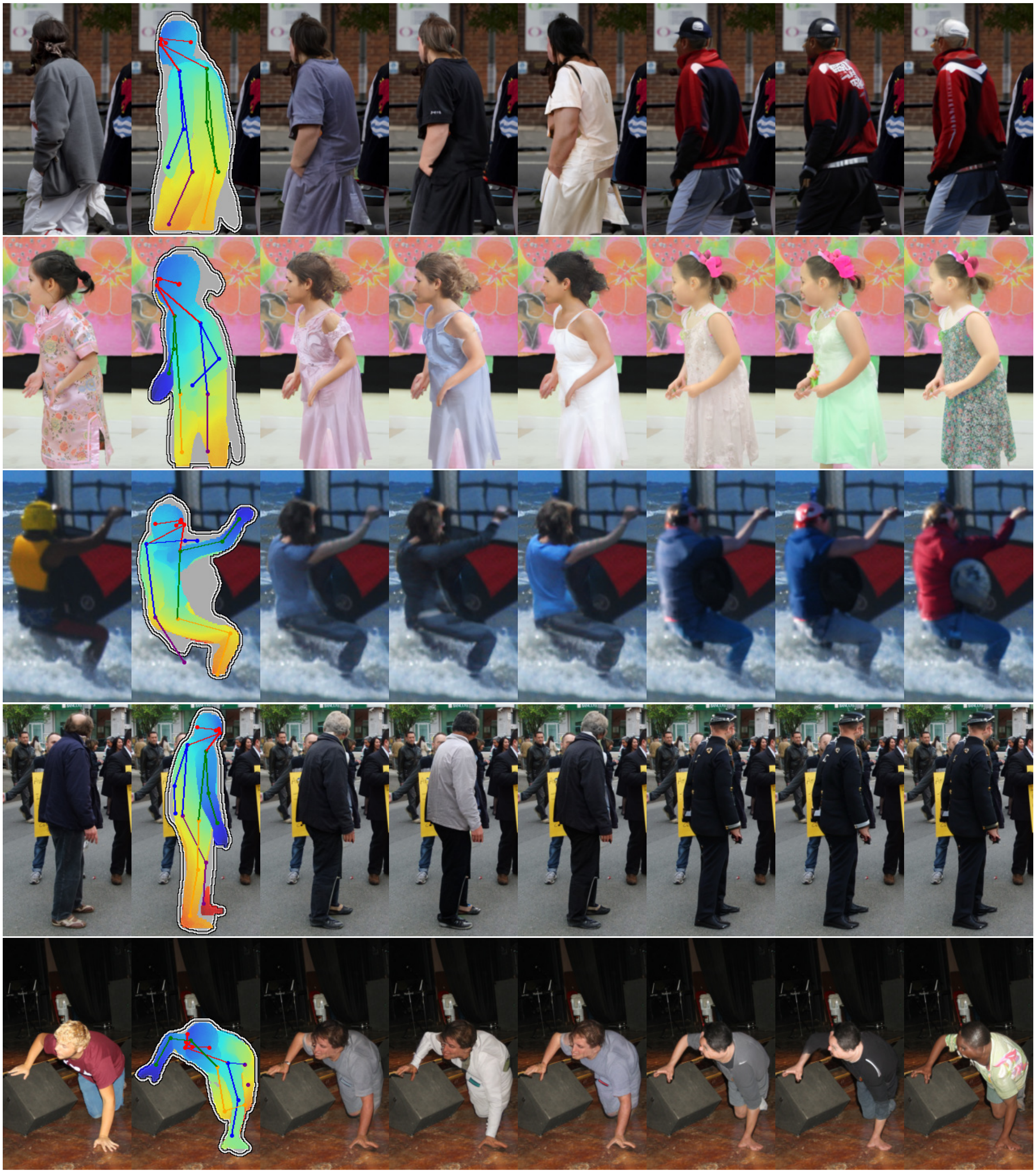


Figure 1. Generated images with Config B using different feature networks stated below each image. Images are given in full resolution (72×40), and we recommend the reader to zoom in.



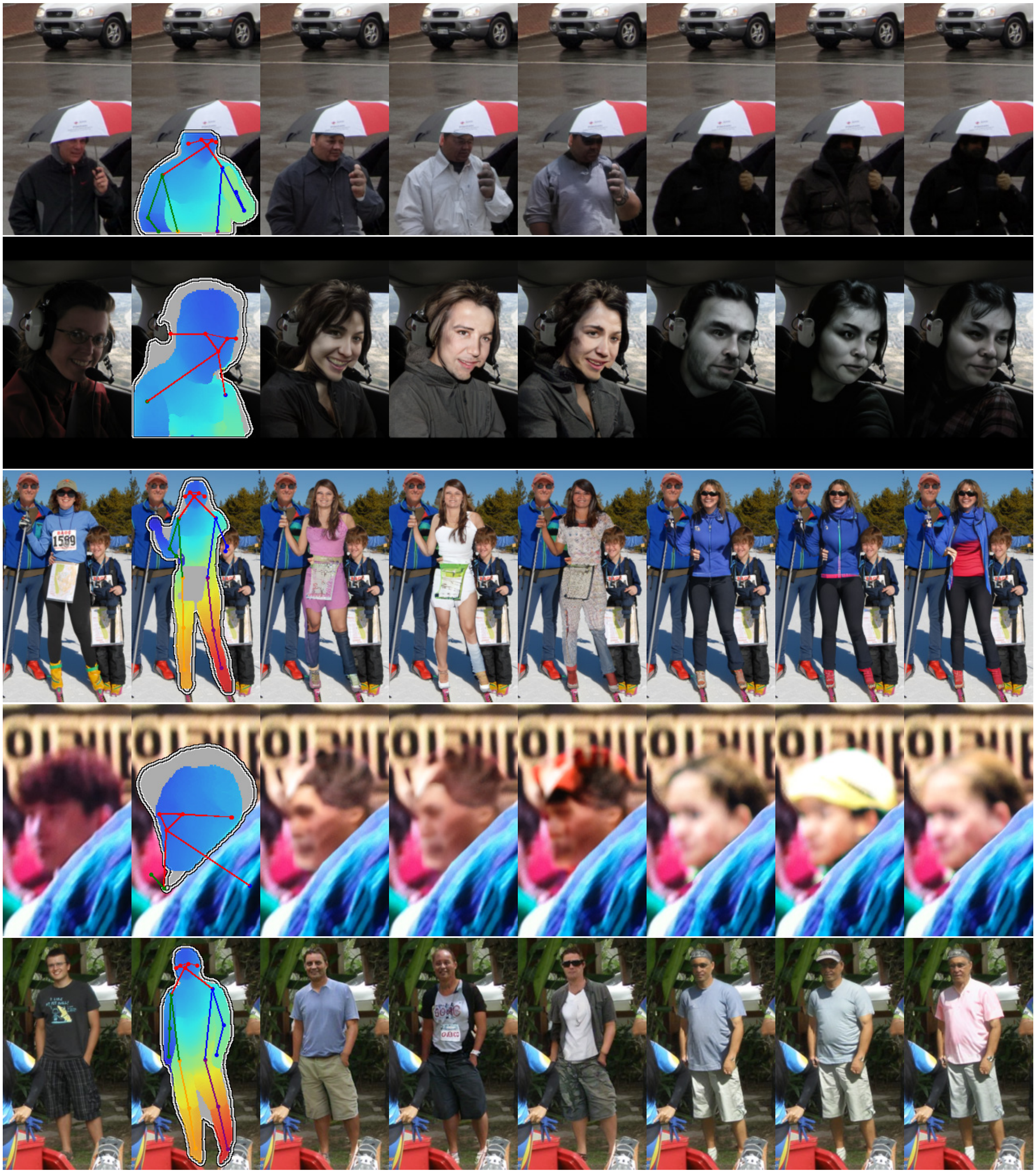
(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 2. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.



(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 3. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.



(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 4. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.

(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 5. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.

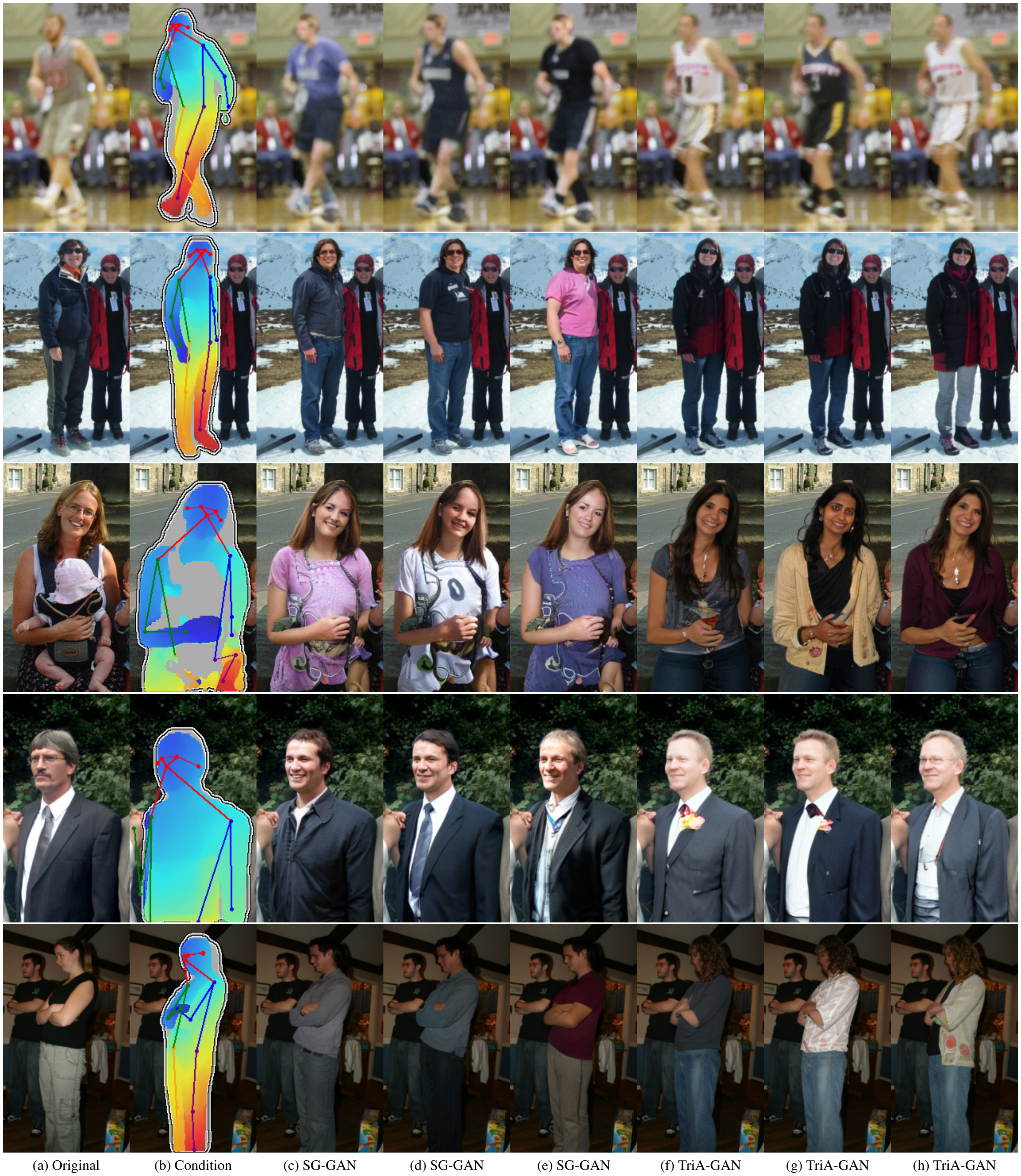
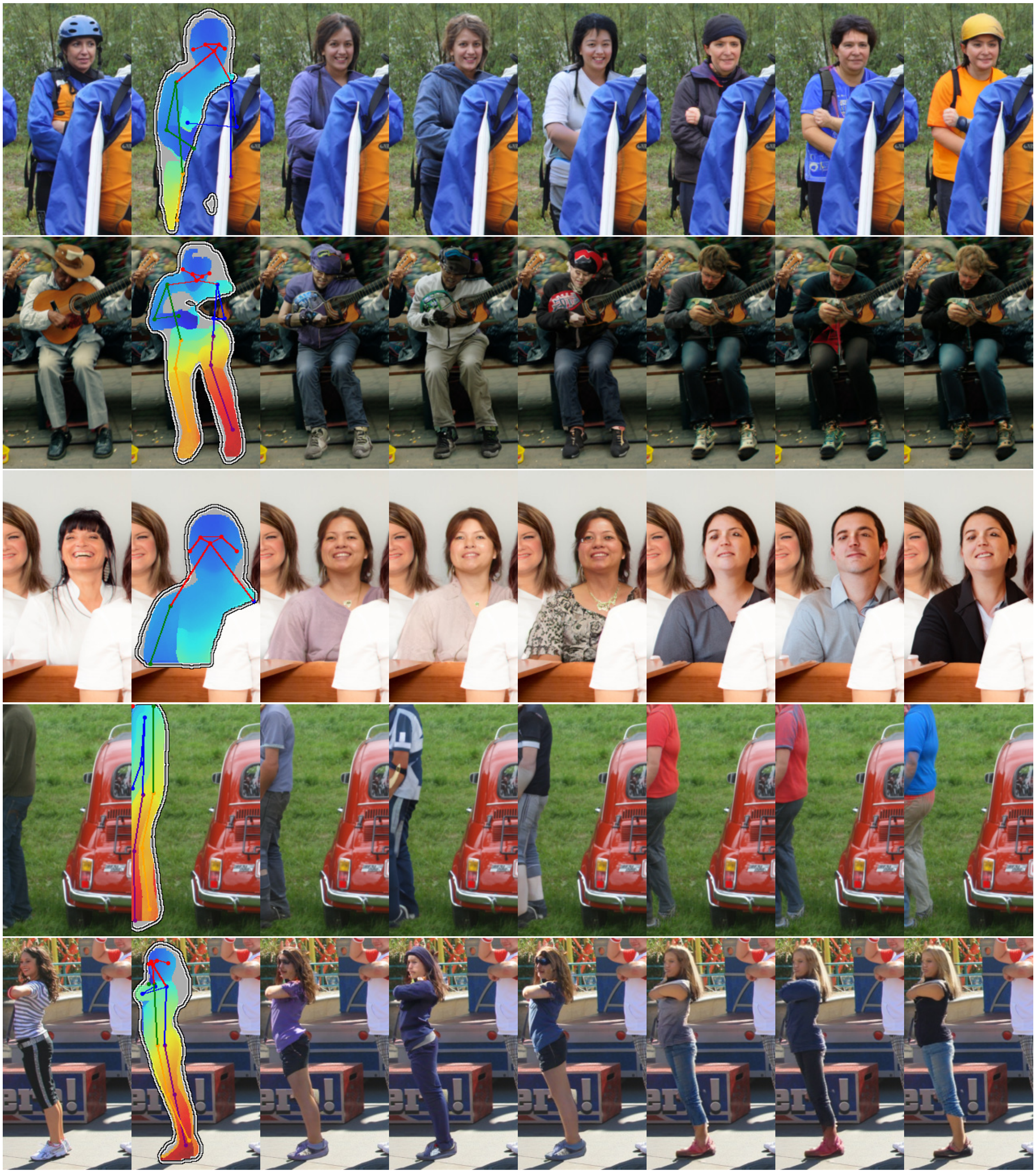


Figure 6. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.

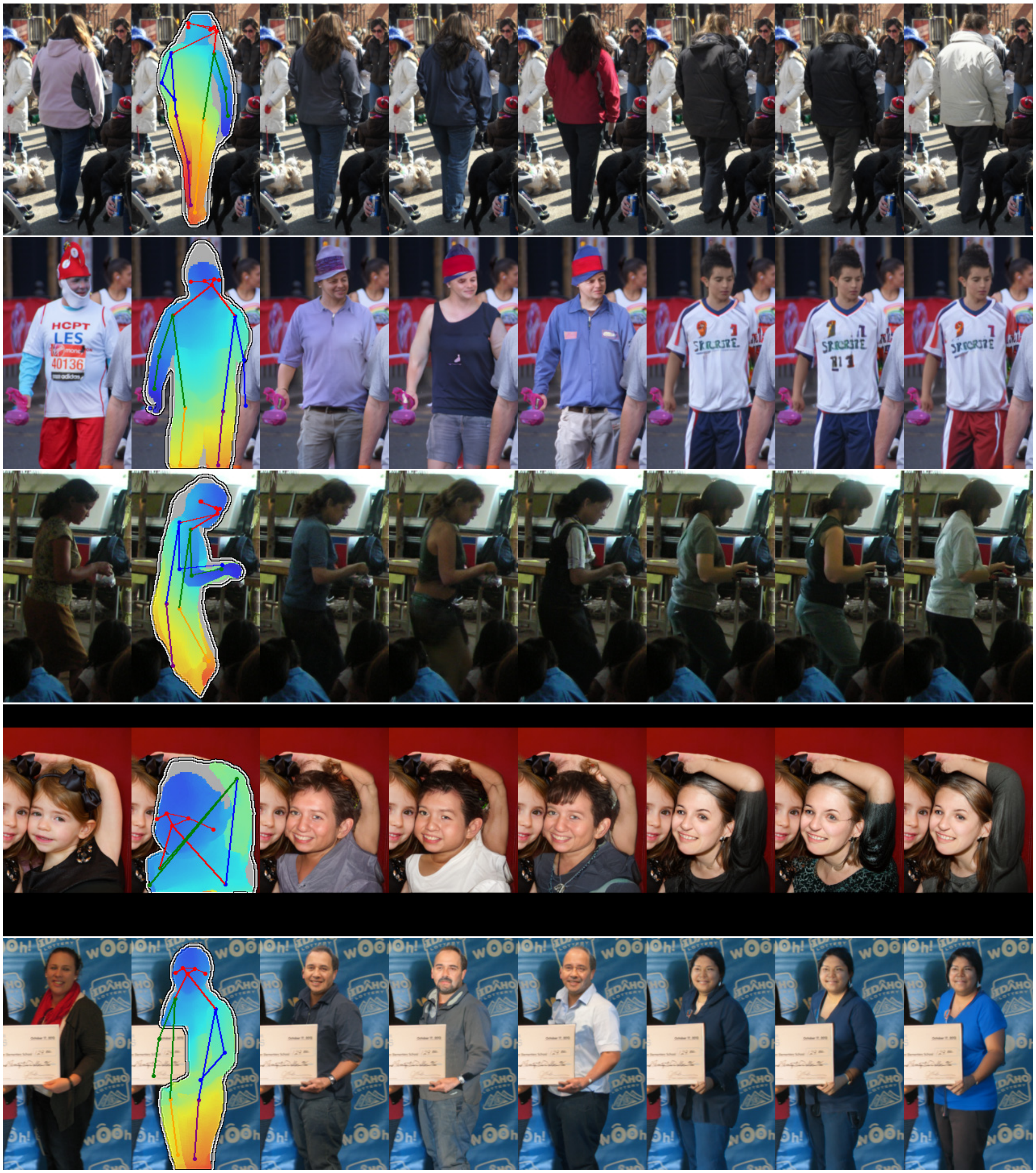


(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 7. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.

(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 8. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.



(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 9. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.

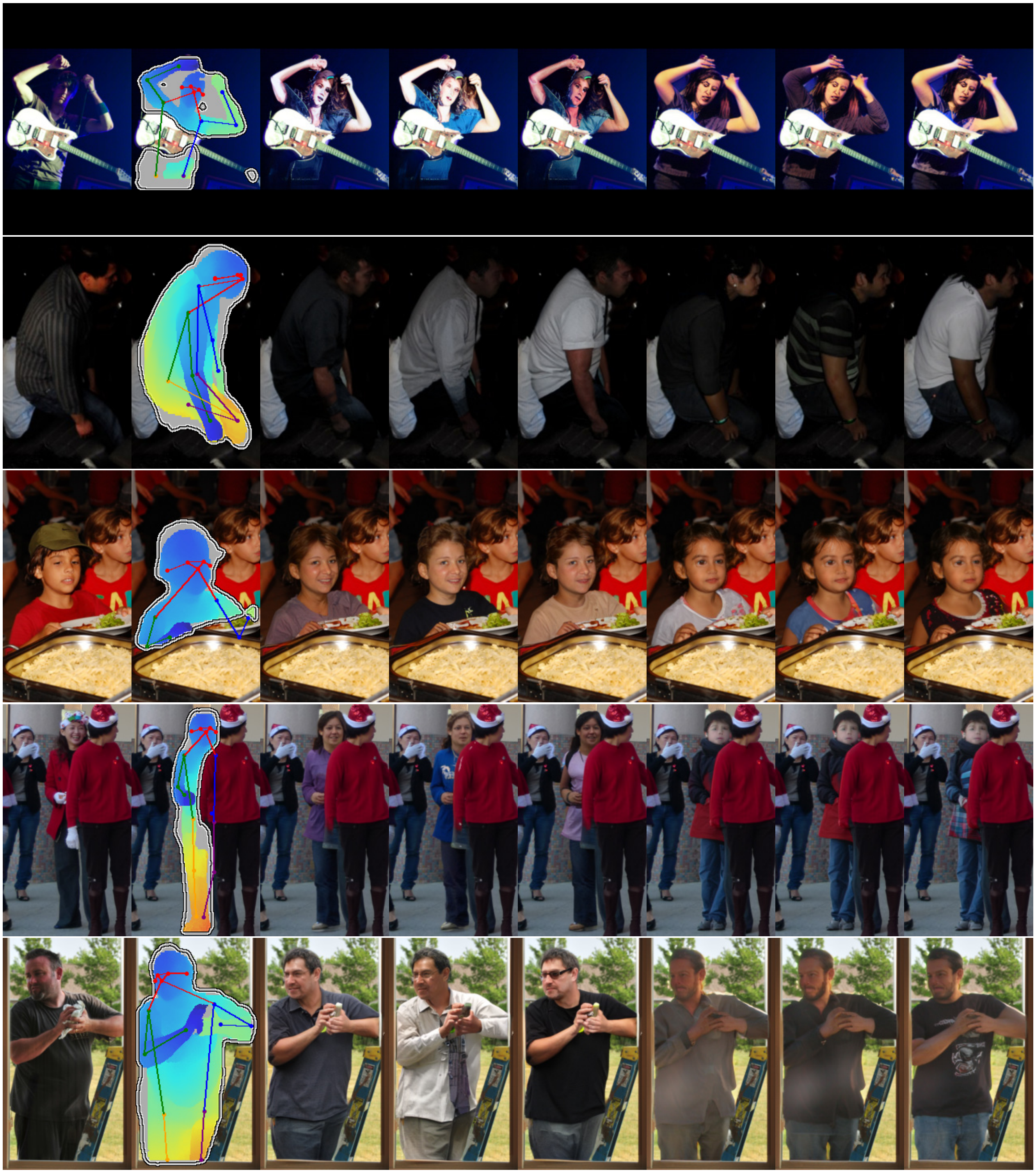
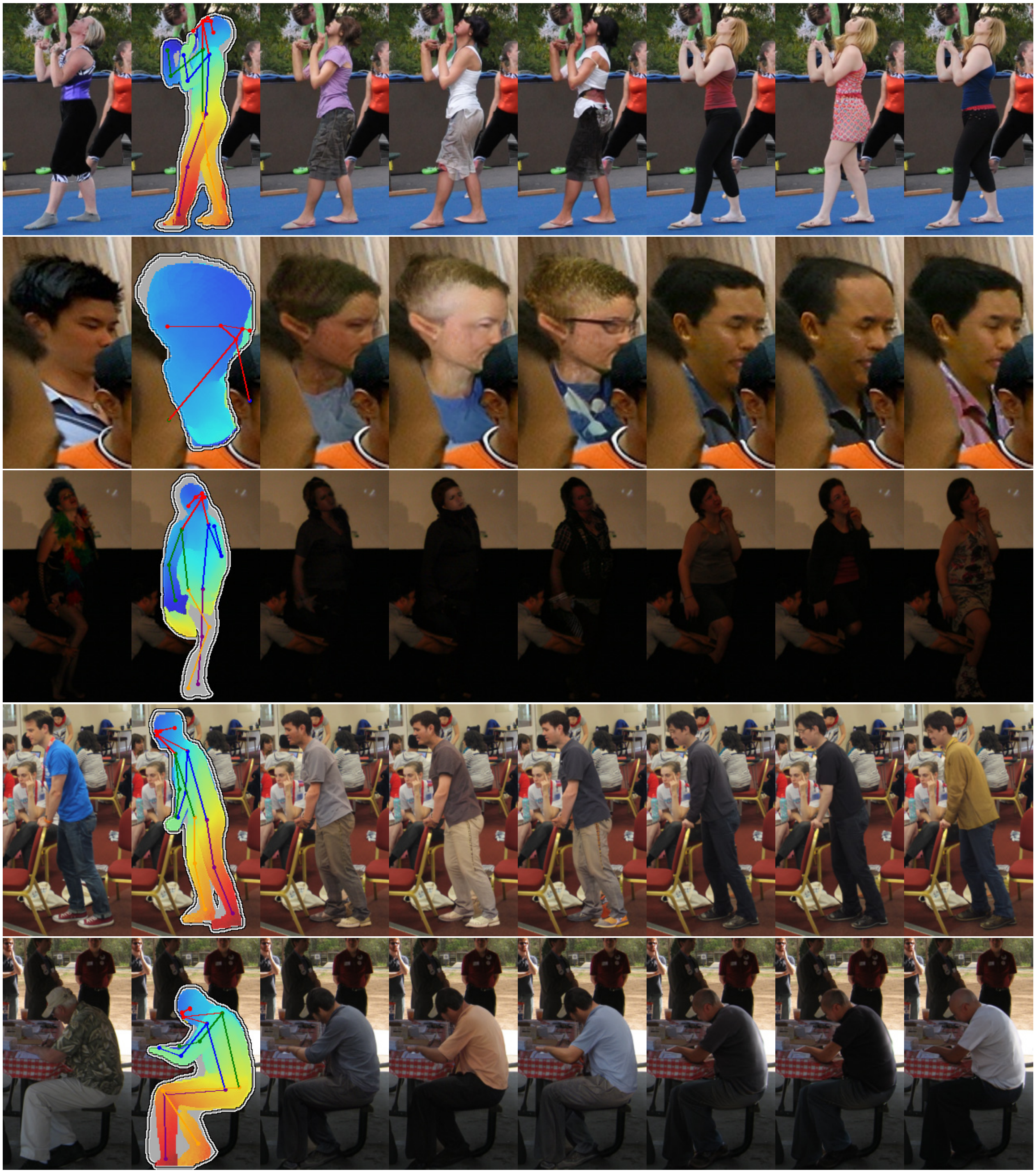


Figure 10. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.

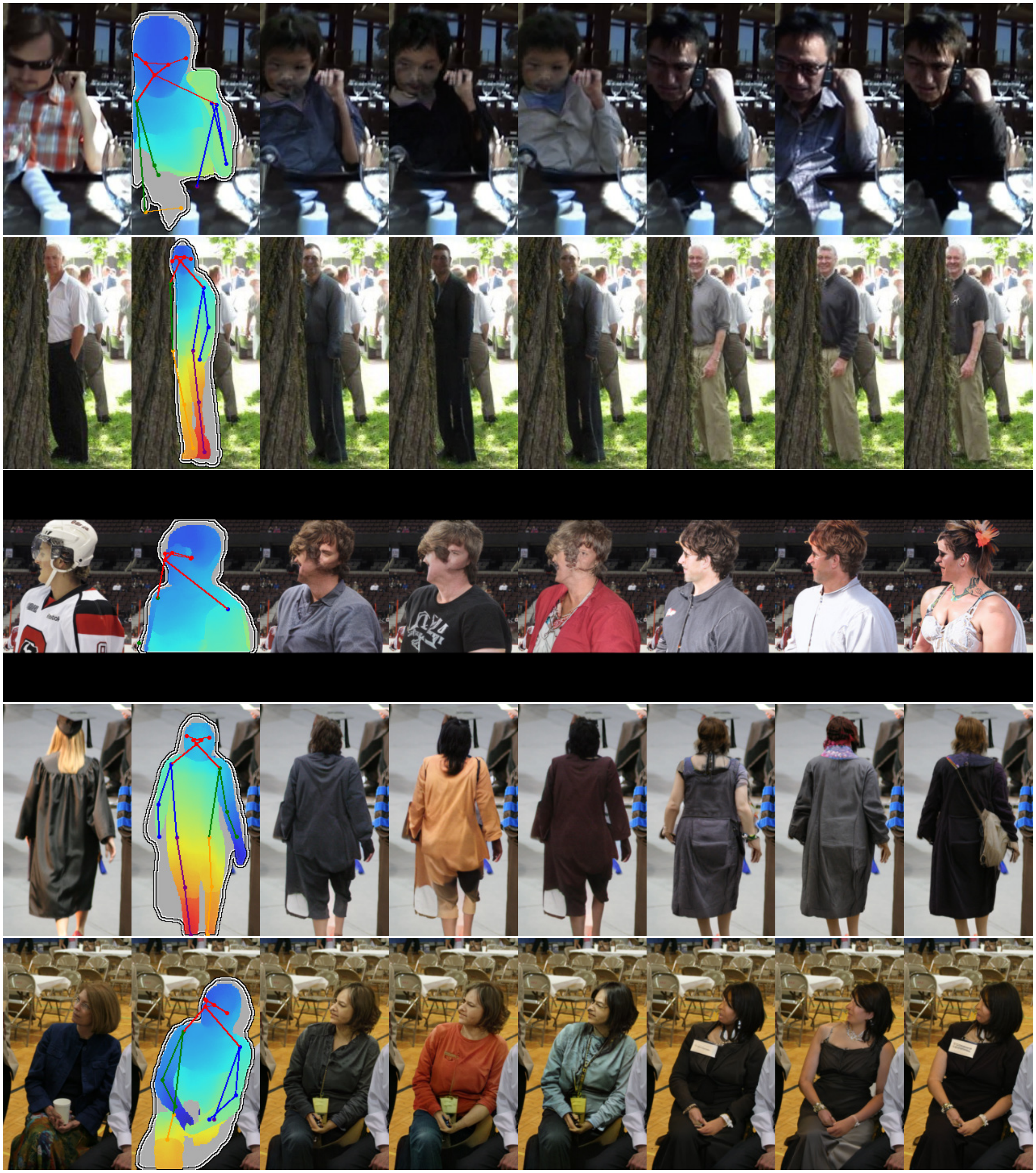
(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 11. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.



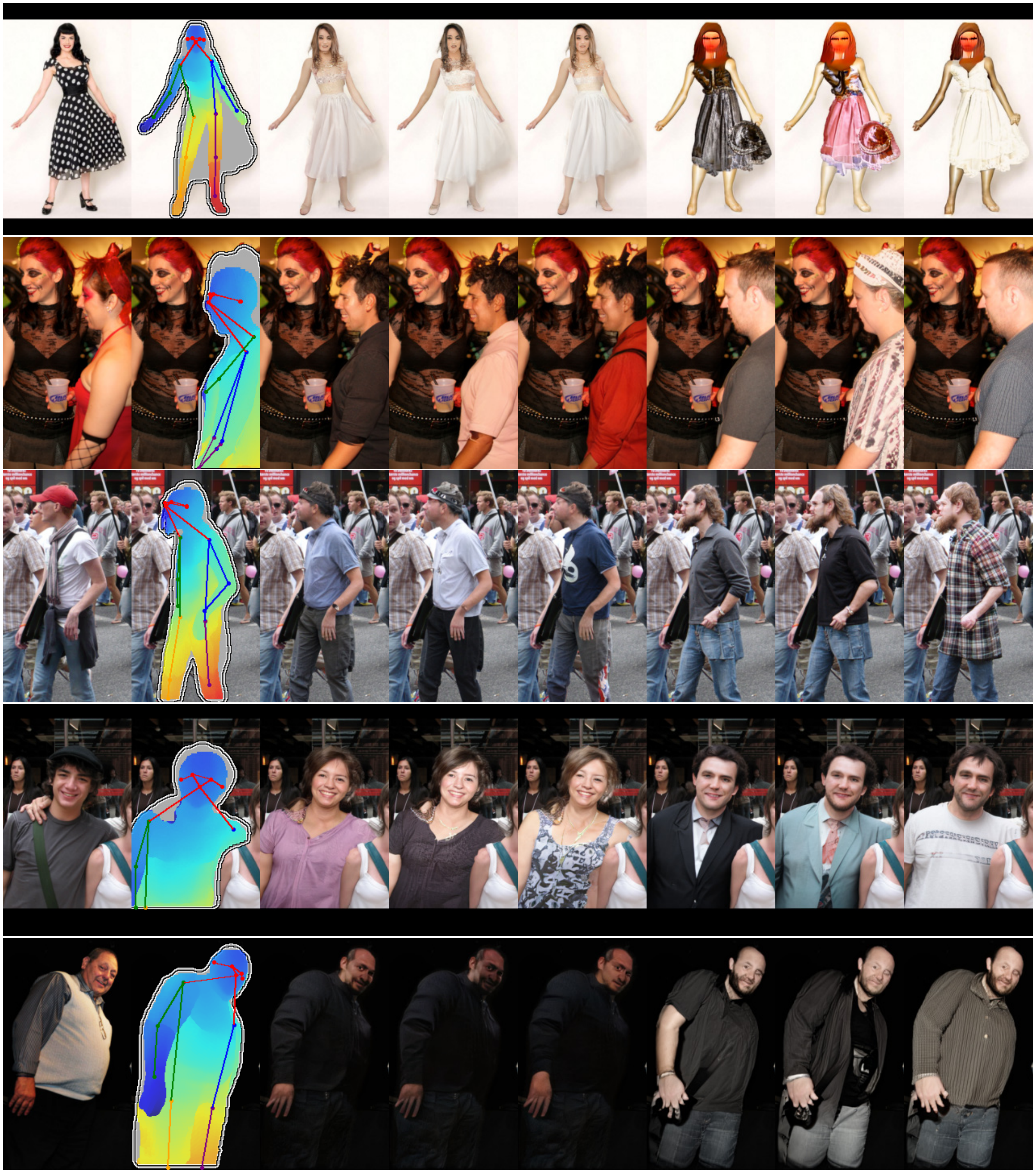
(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 12. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.



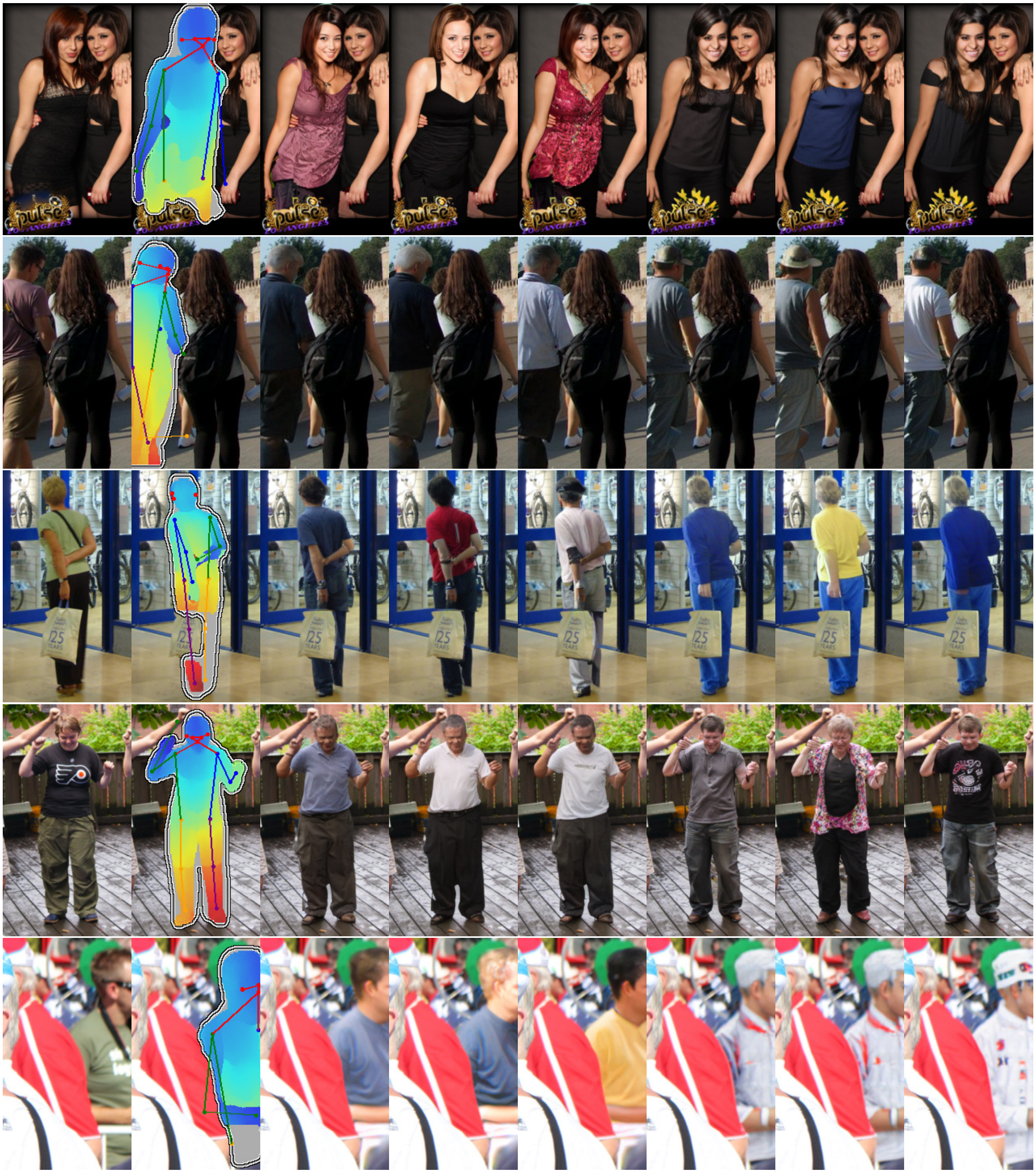
(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 13. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.



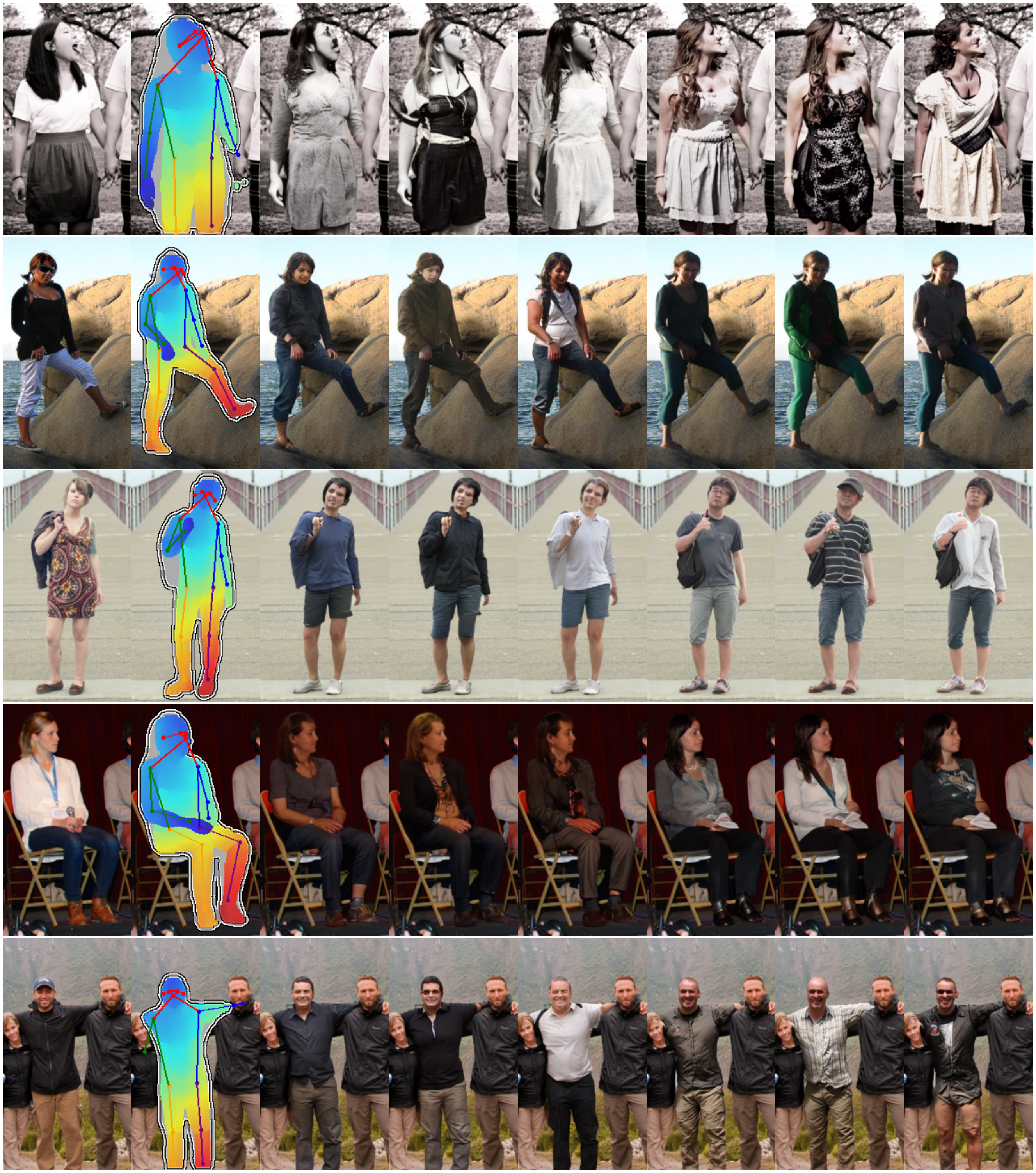
(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 14. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.



(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 15. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.

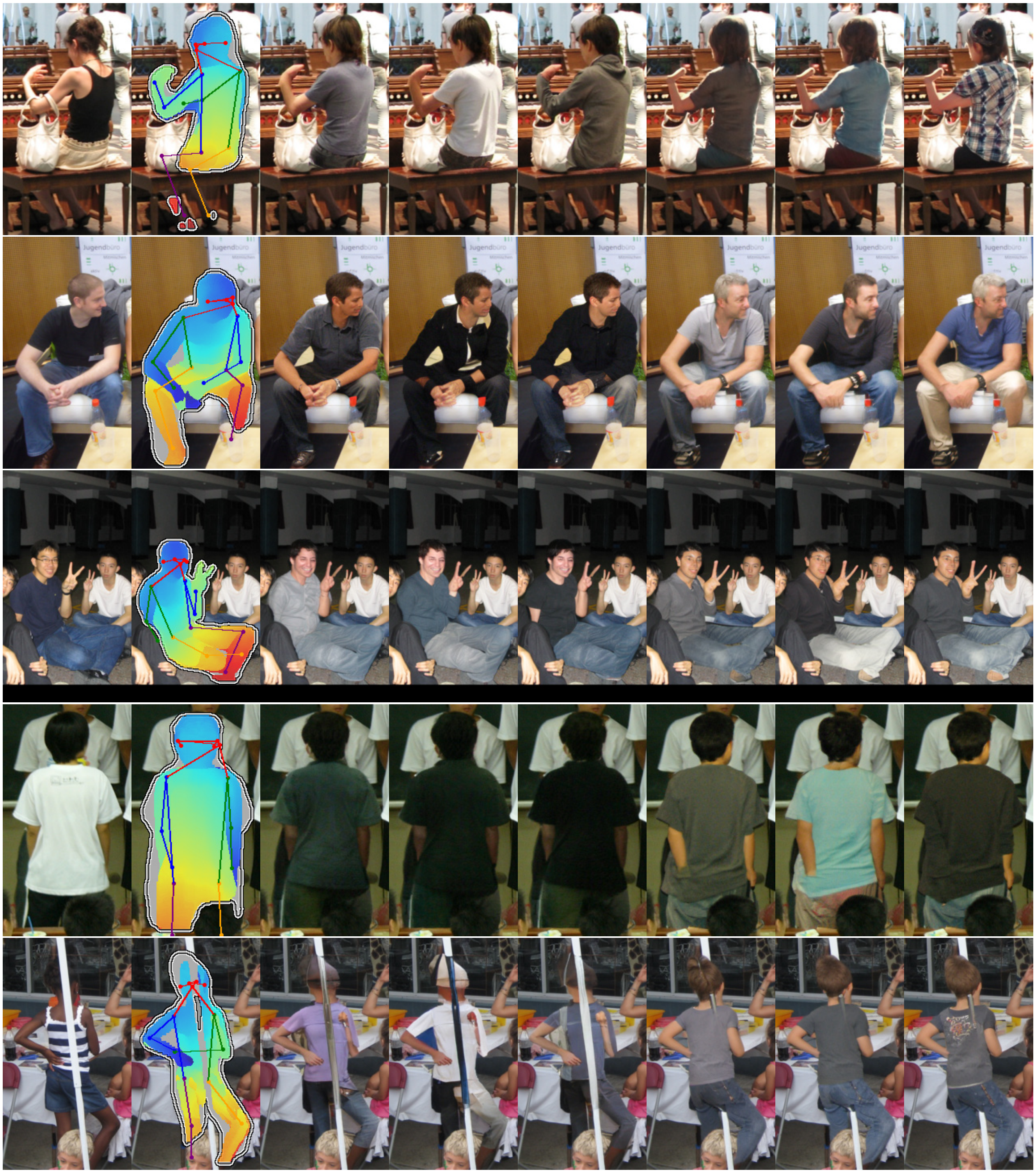


(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 16. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.

(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 17. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.



(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 18. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.

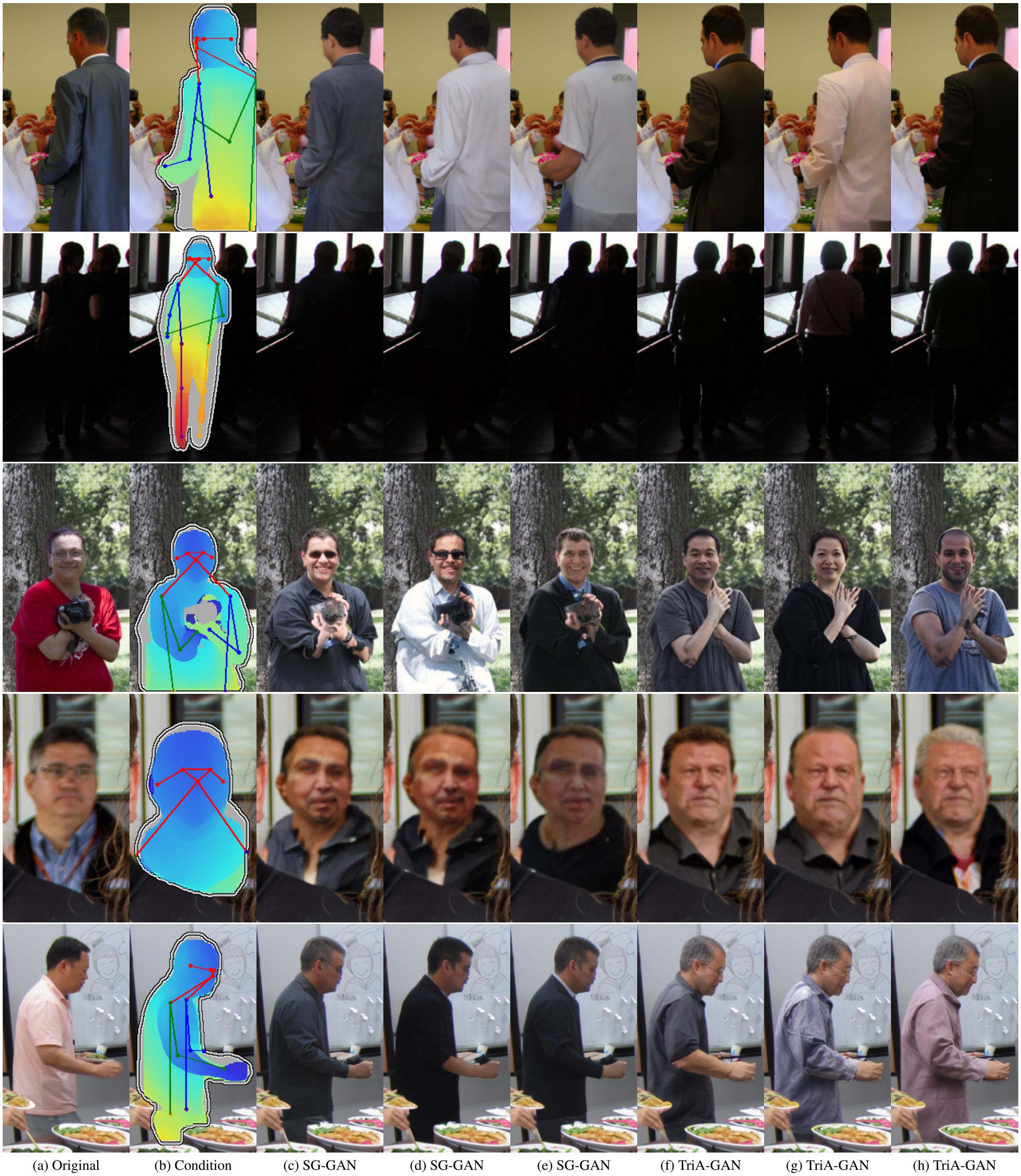
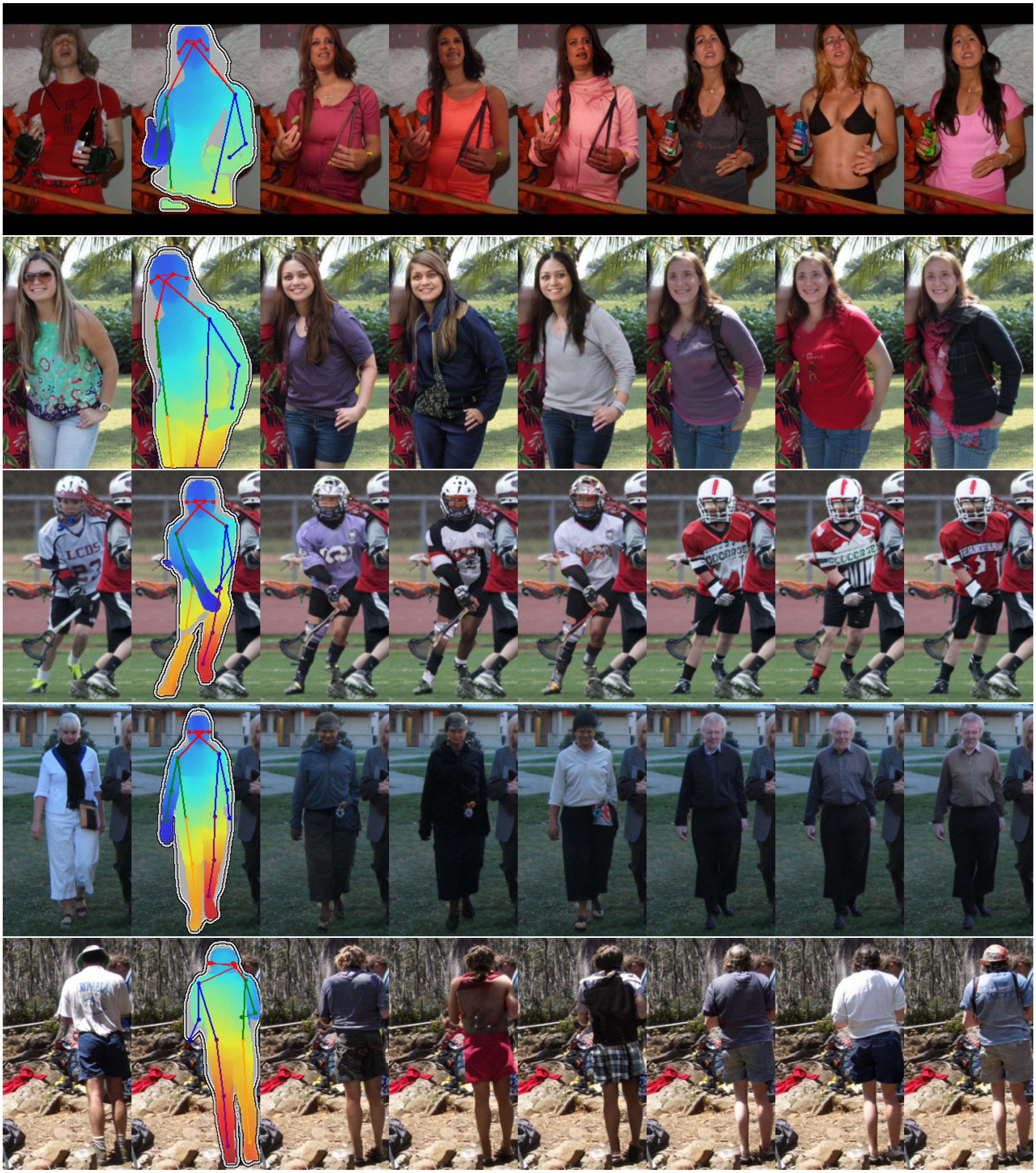


Figure 19. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.



(a) Original (b) Condition (c) SG-GAN (d) SG-GAN (e) SG-GAN (f) TriA-GAN (g) TriA-GAN (h) TriA-GAN

Figure 20. Random generated examples from FDH [2] comparing TriA-GAN to SG-GAN [2]. Note that all examples are generated with multi-modal truncation. Surface map is not used by TriA-GAN.

(a) Original Image

(b) Anonymized with DeepPrivacy2 [2]

(c) Anonymized with TriA-GAN

Figure 21. Comparison of anonymization with TriA-GAN *vs*. SG-GAN [3] trained following DeepPrivacy2 [2].