Supplemental Material: Enhancing Person Synthesis in Complex Scenes via Intrinsic and Contextual Structure Modeling

Xi Tian[1]
xt275@bath.ac.uk
Yong-Liang Yang[1]
strongyang@gmail.com
Qi Wu[2]
qi.wu01@adelaide.edu.au

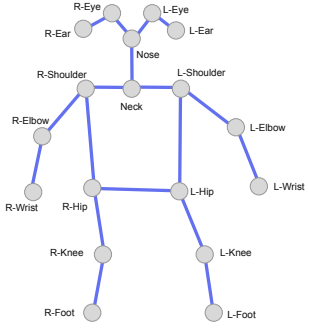
[1] Department of Computer Science
University of Bath
Bath, UK

[2] Department of Computer Science
University of Adelaide
Adelaide, Australia

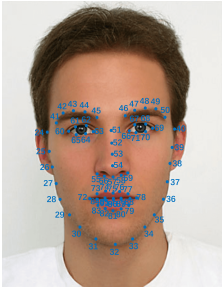
1 Overview

This supplemental material provides the details for: (1) **Person Structure Representation**: the forms of pose and face keypoints (Sec. 2); (2) **Models**: detailed architectures, losses, and implementations (Sec. 3); (3) **Additional Results**: more comparison results and discussions (Sec. 4); (4) **Ethics Statements**: potential person ethical issues and measures (Sec. 5).

2 Structure Representation



(a) Pose keypoints and segments.



(b) Face keypoints as defined in [6].

Figure 1: The person pose and face structures used in our paper. **R-** or **L-** denotes right or left points in the person's perspective, respectively.

The person keypoints, including pose keypoints and face keypoints, are extensively used in our work. This section presents the details of the keypoint/segment representation for describing pose and face structure. The original COCO dataset [1] provides 17 pose keypoints, while we add another point *neck* as the mean position of *left shoulder* and *right shoulder*, and connect the *neck* with *nose*. The addition of the new point leads to a more natural connection between the body and the head. The way to connect the pose keypoints as bone segments is shown in Figure 1a, which also reflects the natural pose structure of a person. As a result, a person pose is represented by 18 key points and 18 segments. Each face has 68 keypoints, and we follow the original form defined in COCO WholeBody dataset [6], as shown in Figure 1b.

3 Models

3.1 Intrinsic Structure Model

The model architecture is illustrated in Figure 2. It includes three parts: structure encoder, structure memory with vector quantization, and structure decoder. The model is built with Linear layers considering the processed data are the coordinates of keypoints. Specifically, the prior input channels (**Ch**) are 36 (18×2) for pose keypoints and 136 (68×2) for face keypoints, which are flattened and normalized x and y coordinates of each keypoint. The memory size N_M is set to 64 for pose keypoints and 8 for face keypoints in our experiments, but it is flexible to change the memory size according to different data varieties.

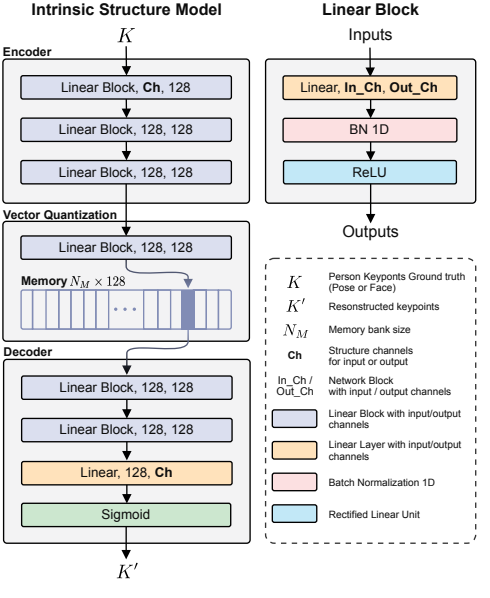


Figure 2: **Intrinsic Structure Model**. The input K and prior input channels **Ch** depend on the input types, such as pose keypoints (36) or face keypoints (136).

3.2 Contextual Structure Prediction

The predictor used to infer the person keypoints is a multi-layer Graph Convolution Network $\mathcal{G}_g$. It captures the contextual person representations from relationships among objects, and uses a classifier head $\mathcal{Q}$ to query the compatible pose/face keypoints from the pre-trained memory bank (Section 3.1).

Node Feature Embedding. The objects and relationships features are encoded using the linear embedding layers. Specifically, each object has three types of attributes including the category $o^c \in \mathbb{R}^{64}$, the location $o^{loc} \in \mathbb{R}^{25}$ defined using a 5×5 grid, and the size $o^{size} \in \mathbb{R}^{10}$ which is categorized to 10 levels. The object's location is specified by the center of the bounding box, and the size attribute is represented by the ratio to the whole image. The final concatenated object embedding size is thus $f_{obj} \in \mathbb{R}^{99}$. The relationships are defined by relative object locations and sizes. The *surrounding* and *inside* relationships are defined if one object's area can totally contain another object's area, and vice versa. Otherwise, using the polar coordinate system by setting a subjective object's location as origin, the *left of*, *below*, *right of* and *above* relationships are defined when the other object's relative radian falls in the range $[0, \pi/4] \cup (7\pi/4, 2\pi)$, $(\pi/4, 3\pi/4]$, $(3\pi/4, 5\pi/4]$ and $(5\pi/4, 7\pi/4]$, respectively. The size of the relationship embedding $f_r \in \mathbb{R}^{64}$.

Information Propagation. The GCN model processes the triples in the form (o_i, r, o_j), where o_i and o_j are object nodes and r is their relationship. Figure 3 illustrates a detailed example of how a *person* node's feature is calculated when the person acts as the *objective* in the triple *(cloud, above, person)* and as the *subjective* in another triple *(person, left of, skateboard)*. Each triple is first concatenated and then processed by two shared linear blocks into hidden states (state feature is 512 in our work). The linear blocks are similar as defined in Figure 2 except the batch normalization layer. Then, three subsequent linear heads g'_s, g'_r, g'_o are adopted for calculating the features for *subjective*, *relationship*, and *objective*, respectively. The output channels are 512 for both g'_s and g'_o, and 64 for g'_r. As the person exists in two triples, the final features are pooled using the average function. A final linear block is used to obtain the final object features. We use three layers in our work. At the end of the last layer, a linear-layer-based classifier head $\mathcal{Q}$ is introduced to map the contextual person features to classification features. In practice, we use the classifier $\mathcal{Q}$ to query the pose keypoints and introduce another MLP-based module to infer the face keypoints conditioned on the pose.

Person-centred GCN In the final layer, we only take *person* nodes' features as inputs for $\mathcal{Q}$. But it is necessary to feed all the objects as inputs because other object-object relationships will also be updated and contribute to the person nodes as information flows in the graph. In training, an object is randomly set to *subjective* or *objective*.

3.3 Appearance Refinement Modules

PRM and FRM The architectures for pose and face refinement modules (PRM and FRM) follow the structure of [11]. For the PRM, we introduced two heads where one is for generating person masks S_p^M and the other one for learning semantic maps S_p, as shown in Figure 4. The FRM has a similar structure as PRM except that there is only one output for face

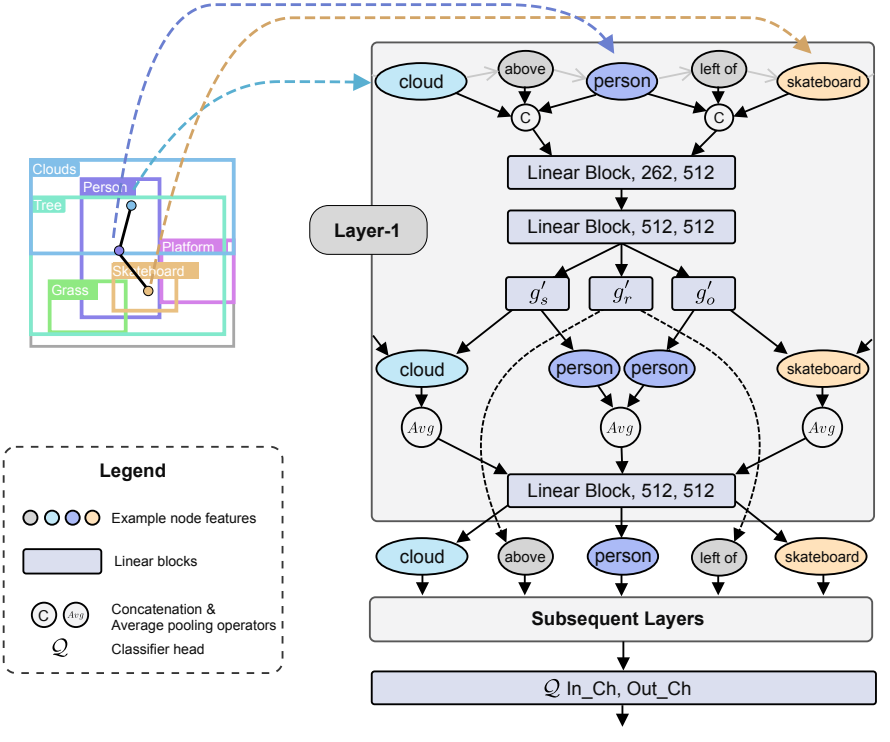


Figure 3: **Contextual Structure Predictor**. Left is the input layout. Taking the triple *(cloud, above, person)* and *(person, left of, skateboard)* for example, the right shows the information propagating among the three nodes with their relationship. The *person* acts as the *subjective* and *objective* object and features are finally averaged using the pooling function.

region semantic map generation. As the FRM is first pretrained for face generation and its parameters are frozen, we add an extra adaption linear layer at the end for fine-tuning. The PRM or FRM output person pose or face semantic maps of spacial size 64×64 and channel 180.

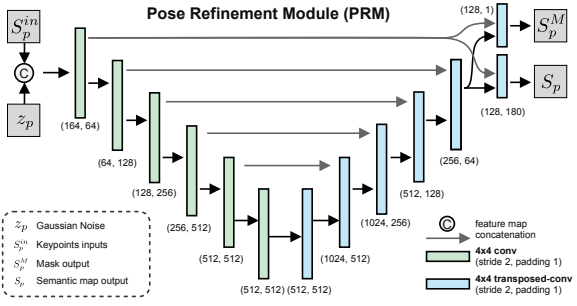


Figure 4: The illustration for PRM module.

Image Generation Figure 5 illustrates the overview of the image generation in training phase. We use bold symbols to denote data in batch form. The generator is built by a linear layer followed by several ResNet Blocks (ResBlock). Each ResBlock contains two modulation modules which adopt the *semantic map* to modulate (or denormalize) the intermediate feature maps.

Semantic Map Construction. To construct the *semantic map*, we first use a mask regressor $\mathcal{M}$ to generate the non-person objects' masks $m_{obj} \in \mathbb{R}^{32\times32}$ from their category embedding c_{obj} concatenated with Gaussian noise z_{obj}. The masks are further placed to their layout locations defined by the bounding boxes. This is done by the function *ToLayout* operation which performs bilinear interpolation on the masks and transforms the them to the corresponding boxes in the layout. The layout-level object masks are denoted as m'_{obj}.

Then, inspired by the ISLA-Norm [14], we perform cross product between c_{obj} and m'_{obj} to construct the *semantic map* base. The cross product operation is shown as *Mapping* function in Figure 5. We then use PRM and FRM for generating person pose and face semantic maps and adopt the same *ToLayout* function to place person-related semantic maps to layout and add them with the base *semantic map*. Finally, the *semantic map* $\in \mathbb{R}^{180\times H\times W}$, where H and W is image height and width, respectively.

Modulation. The modulating in GAN has been widely used [2, 8], and our implementation can be formulated as:

$$\hat{x} = x' \cdot \gamma + \beta \tag{1}$$

where x' is the Batch Normalized feature of input x and $\hat{x}$ is the output of modulation layer. γ and β are modulation parameters learnt from the *semantic map*. Note this part is different form ISLA-Norm [14] because the *semantic map* aggregates all the objects and the persons features before sending them to the modulation layer, while the ISLA-Norm separately learn each object's modulation parameters and averaging them later.

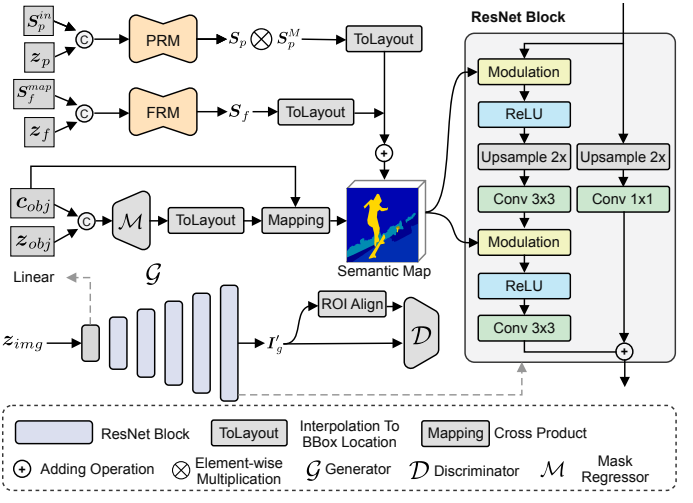


Figure 5: The image generation illustration.

Learning and losses. We adopts a discriminator with stacks of standard ResBlocks [4] for image-level and object-level discrimination. The object-level features are extracted by ROI-Align [8] from their bounding boxes. The losses in training include image-level GAN loss, object-level GAN loss, Total Variation (TV) loss, and face loss. The TV loss is formulated as: $\sum_{i,j} \left|m_{i+1,j} - m_{i,j}\right| + \left|m_{i,j+1} - m_{i,j}\right|$, where $m_{x,y}$ denotes pixel values at (x,y) on the mask. The total loss can be formulated as follows:

$$L_{total} = \lambda_1 L_{image} + \lambda_2 L_{objects} + \lambda_3 L_{TV} + \lambda_4 L_{face}, \tag{2}$$

where the loss coefficients $\lambda_1 = 0.1$, $\lambda_2 = 1.0$, $\lambda_3 = 0.1$, and $\lambda_4 = 0.3$. We also test $\lambda_4 = 0.1$ for face loss in the ablation study of the main paper.

3.4 Implementations.

We implemented our models using PyTorch [9]. The image generator is implemented on top of LostGAN-v1 [14] with the aforementioned modifications. In training, the batch size is set to 32 and 12 for 128 × 128 and 256 × 256 resolution, respectively. We use Adam optimization method [7] with learning rate $1e^{-4}$ for both generator and discriminator in image generation. We set the batch size to 128 for training the Intrinsic Structure Model and the Contextual Structure Predictor. We use the cosine annealing to schedule the learning rate in range $[0, 1e^{-3}]$ at the end of each epoch.

4 Additional Results and Discussion

4.1 Intrinsic Structure Modeling.

Reconstruction Results. The structure reconstruction results are shown in Figure 7. The ground truth keypoints for both pose and face are not always complete, as part of the key-

points are likely to be missing which makes the prior learning challenging. For example, the pose keypoints include standing (first column), sitting (3rd column), half-body (4th column), and upside-down (7th column), *etc*. The face keypoints are also various containing more points. It shows that the proposed memory-bank-based model can successfully restore the person pose and face keypoints. The restoration may not be exactly the same as the inputs, but considering the real-world keypoints are in the continuous space, the discrete memory space has caught the significant pose/face structures. It is flexible to change the memory bank size, the reconstruction results become finer when using larger sizes. However, we found it is enough to use 64 for pose memory bank size and 8 for face memory bank size to represent person structures, otherwise there would be more relatively repetitive keypoints structures. There is also a trade-off between reconstruction precision and memory bank querying accuracy because larger memory bank means harder to query.

(a) Pose keypoints reconstruction. (b) Face keypoints reconstruction.

Figure 6: The person keypoints reconstruction results of our Intrinsic Structure Model. Images in grey boxes are ground truth and in blue boxes are restored keypoints.

Effectiveness of Vector Quantization. In the main paper, we compared with the vanilla auto encoder (VanillaAE), which has the same purpose as our Intrinsic Structure Model. The main difference is vanilla auto encoder uses the continuous latent space versus our discrete memory space implemented by Vector Quantization. The main paper's ablation study quantitatively demonstrates that our model performed better, while this part evaluates by comparing the visualized results in terms of the structure reconstruction and prediction. Note the continuous nature of the AE makes it need a regressor head for the Contextual Structure Predictor rather than a classifier.

Taking the face keypoints for example, Figure 7a shows the reconstruction results of the vanilla AE. The reconstruction is convincing, and the reconstructed keypoints are almost the same as the original ones. However, when adopting the trained decoder with the Contex-

(a) VanillaAE face keypoints reconstruction results. (b) Face keypoints generation using VanillaAE.

Figure 7: **VanillaAE results.** In comparison with our memory-based Intrinsic Structure Model, the VanillaAE performed better in reconstruction. However, it was hard to generate the contextual keypoints conditioned on the layout inputs.

tual Structure Predictor, the results become *distorted*, as shown in Figure 7b. Theoretically, regressing the right face keypoints is much harder, especially the input is only the layouts of the persons and other objects. Thus, the condition is not strong enough to supervise the generation of the keypoints in desirable distribution. In comparison, the usage of vector quantization makes it possible to project keypoints to some latent points, which can be considered as the representative samples of the underlying manifolds.

4.2 Contextual Structure Predictor.

Contextual Keypoints Inference. Here we discuss more about how our method can infer context-aware person keypoints. In Figure 8 (D), the layout indicates a person near a baseball bat, and our method infers the holding pose that is corresponding with the context. For method that does not consider context, [LS] can only generate a person but is not matching with the layout. Figure 9 (D) shows our method can also infer precise keypoints. The layouts indicate four persons near a dining table, the predicted keypoints for the two persons behind the dining table are half-body poses without legs as the legs might be obscured by the tables. However, the inferred keypoints for the left and right persons near chairs have half-leg segments, implying the persons might sit in the outer chairs and more body parts can be seen. Figure 10 (D) shows a layout mainly containing a person near a horse. Our method not only infers a riding pose but also predicts there should be only one leg appear and another one is obscured by the horse.

Keypoints Completion Ability. Inferring person keyponints from the layout also brings an advantage that is Keypoints Completion. Even the ground truth person keypoints are not complete and the face keypoints only exist in part of the images as most of the faces are very small. As shown in Figure 9 (A), there is only one person that has relatively complete keypoints while others' keypoints are just several segments. In this case, it is hard to generating better persons even using the ground truth person keypoints. However, our method is able to infer relatively complete keypoints complying with the context, which is beneficial for person generation. We speculate this is also the reason that the person classification accuracy (PAcc) achieves the highest when using the predicted keypoints (please refer to the quantitative results in main paper).

4.3 Image Generation Results.

The results shown in the main paper are in 256 $\times$ 256 resolution for better viewing. We show more 256 $\times$ 256 resolution results and comparisons with LostGAN-v2 [LS] in Figure 8 and 9, and more of those for 128 $\times$ 128 resolution in Figure 10. The following gives more discussion from different perspectives.

Crowd Generation. There exist images containing many persons, or crowds, as shown in Figure 8 (A) and 10 (C)(E). Existing method [LS] is hard to generate reasonable persons where the synthesized persons are distorted, such as in Figure 8 (A), or incomplete, as shown in Figure 10 (E). Thanks to the person refinement modules, our method refines each person separately by generating their intermediate semantic features, which helps generate each person with clear body shape even when these persons are very close or overlapped in the above images.

Face Generation. As shown in Figure 8 (C) and 9 (A)(C)(D), the generated person faces are clearer and much more recognizable compared with results of existing method. Also, the generated faces match the inferred face keypoints. Actually, the face are almost ignored using existing methods, which reveals it is very challenging to generate reasonable face regions together with persons in complex scenes. Although our method can deliver better face generation, it is still difficult to generate more realistic faces with more details. We discuss this limitation in section 4.4.

User Study Interface. Figure 11a and 11b show the user interface for the comparing results with LostGAN-v2, and whether our predicted person keypoints are compatible with the layout, respectively. The two user experiments are separately conducted, in which each of the 10 users was given about 200 cases. We also shuffle the testing images and present them in random order.

4.4 Limitations and Future Work.

Limitations. Although our experiments demonstrate convincing person generation results qualitatively and quantitatively, we admit there mainly exist two limitations. First, the diversity of inferred keypoints. When the layouts are similar, the inferred keypoints tend to be similar, and this affects the diversity of person keypoints. Second, face details generation. We found when face regions are smaller, it is hard to generate rich face details, as shown in Figure 8(A) and all 128×128 images in Figure 10. By inspecting the datasets, we found there are only about 20% images with face areas larger than 10% of the image. This imbalance might causes biases and lead the model to generate faces with less details.

Future Work. Our future work can be summarized as follows. First, keypoints diversity. The intrinsic model has encoded the most representative person structure information and applying keypoints augmentation [3] can make slight change without changing the essence. Secondly, face details improvement. Further investigation includes incorporating methods from head inpainting methods [12]. Finally, further exploration include applying the proposed method in other tasks, such as text-to-image generation, which would benefit from person structure learning.

5 Ethics statement.

This work depends on the publicly released dataset Microsoft COCO, COCO-Stuff and COCO WholeBody. This work improves the person image generation quality, which can facilitate the content creation industry. This work does not target person identity, for example, learning the face or pose information for personal identification. However, the learnt representations may include biases that are undesirable. In generating persons, our work may unintentionally leak the privacy of persons, generate biased appearance and be used for potentially inappropriate applications, such as synthesizing fake portraits [10]. Viable solutions to prevent misuse and negative effects include: (1) adopting effective methods for detecting fake content generated by deep models [16], (2) using watermarking or obfuscation [13] to protect generated persons.

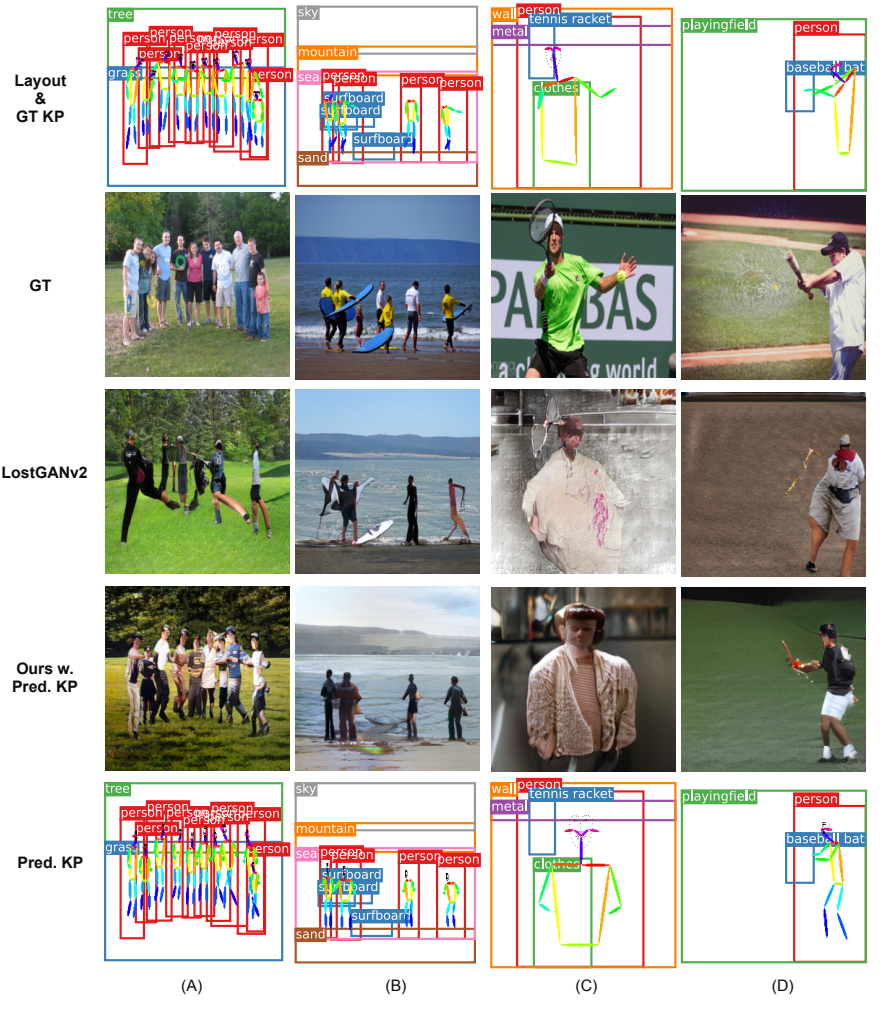


Figure 8: Comparison between our methods using predicted keypoints (Pred. KP) with LostGANv2 in **256 × 256 resolution**. The predicted face keypoints are included in Pred. KP. The ground truth keypoints (GT KP) with the layout are also shown for comparison. Zoom in for better view.

Figure 9: Comparison between our methods using predicted keypoints (Pred. KP) with LostGANv2 in **256 × 256 resolution**. The predicted face keypoints are included in Pred. KP. The ground truth keypoints (GT KP) with the layout are also shown for comparison. Zoom in for better view.

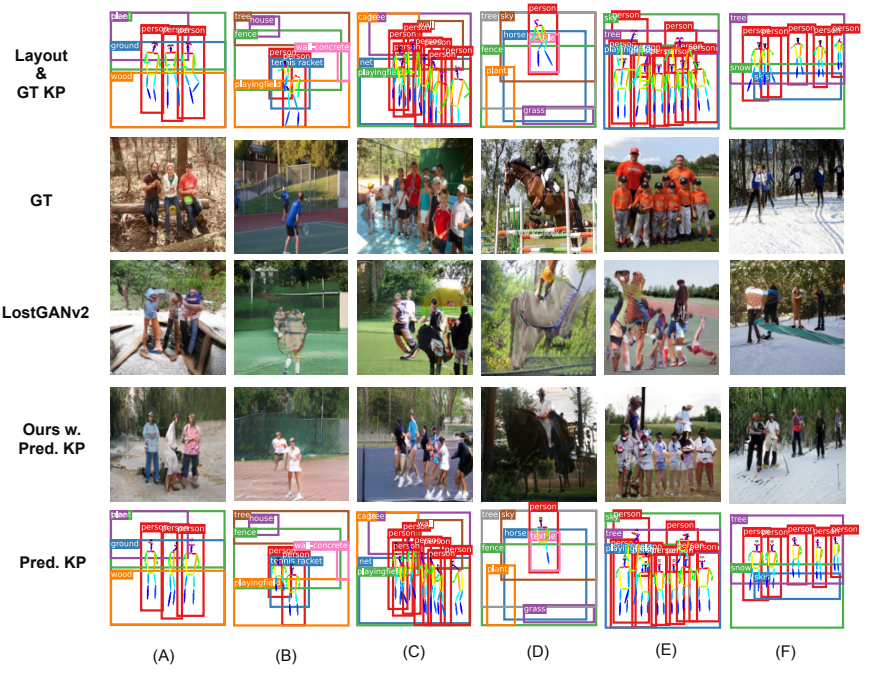
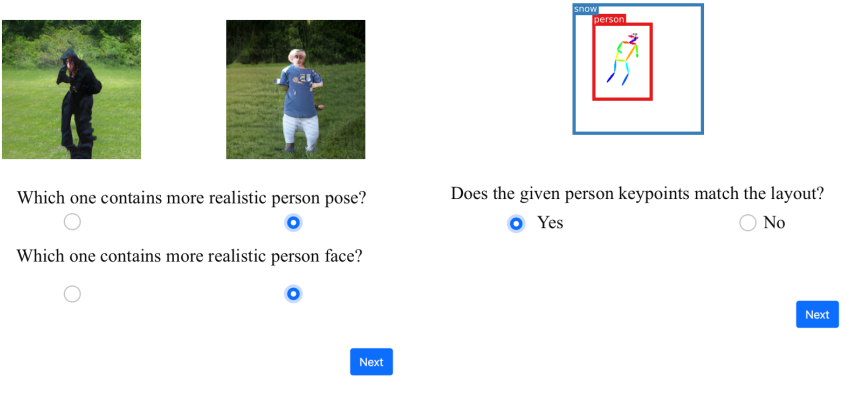


Figure 10: Comparison between our methods using predicted keypoints (Pred. KP) with LostGANv2 in **128 × 128 resolution**. The predicted face keypoints are included in Pred. KP. The ground truth keypoints (GT KP) with the layout are also shown for comparison. Zoom in for better view.



(a) User Study for comparison with LostGANv2.

(b) User Study for whether the predicted person keypoints match the layout or not.

Figure 11: Examples of user interface in our user study.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[2] Ting Chen, Mario Lucic, Neil Houlsby, and Sylvain Gelly. On self modulation for generative adversarial networks. *arXiv preprint arXiv:1810.01365*, 2018.

[3] Kehong Gong, Jianfeng Zhang, and Jiashi Feng. Poseaug: A differentiable pose augmentation framework for 3d human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 8575–8584. Computer Vision Foundation / IEEE, 2021.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[6] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[8] 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*, pages 2337–2346, 2019.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[10] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, et al. Deepfacelab: Integrated, flexible and extensible face-swapping framework. *arXiv preprint arXiv:2005.05535*, 2020.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[12] Qianru Sun, Liqian Ma, Seong Joon Oh, Luc Van Gool, Bernt Schiele, and Mario Fritz. Natural and effective obfuscation by head inpainting. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5050–5059. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00530.

[13] Qianru Sun, Liqian Ma, Seong Joon Oh, Luc Van Gool, Bernt Schiele, and Mario Fritz. Natural and effective obfuscation by head inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5050–5059, 2018.

[14] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10531–10540, 2019.

[15] Wei Sun and Tianfu Wu. Learning layout and style reconfigurable gans for controllable image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (to appear), 2021.

[16] Ruben Tolosana, Sergio Romero-Tapiador, Julian Fierrez, and Ruben Vera-Rodriguez. Deepfakes evolution: Analysis of facial regions and fake detection performance. In *International Conference on Pattern Recognition*, pages 442–456. Springer, 2021.