Pose Guided Multi-person Image Generation From Text

Soon Yau Cheong, Armin Mustafa, Andrew Gilbert

University of Surrey

Abstract. Transformers have recently been shown to generate high quality images from texts. However, existing methods struggle to create high fidelity full-body images, especially multiple people. A person’s pose has a high degree of freedom that is difficult to describe using words only; this creates errors in the generated image, such as incorrect body proportions and pose. We propose a pose-guided text-to-image model, using pose as an additional input constraint. Using the proposed Keypoint Pose Encoding (KPE) to encode human pose into low dimensional representation, our model can generate novel multi-person images accurately representing the pose and text descriptions provided, with minimal errors. We demonstrate that KPE is invariant to changes in the target image domain and image resolution; we show results on the Deepfashion dataset and create a new multi-person Deepfashion dataset [1] to demonstrate the multi-capabilities of our approach.

Keywords: Text-to-image, image generative models, multimodal, human pose, transformer

1 Introduction

Generating an image from a text prompt or caption, also known as text-to-image, is a challenging problem. Image pixels are considered continuous values across the image, while the text is represented as discrete values. The difficulty in mixing continuous and discrete representation has hampered the progress in this field. However, the recent improvement in multimodal research, most notably DALL-E [2], led to a dramatic improvement in the quality in text-to-image generation. In the DALLE-E architecture, the image is encoded into discrete tokens with discrete variational autoencoder dVAE [3]. The relationship between the discrete image tokens and discrete text tokens can be modelled by a transformer [4]. The text and image tokens are concatenated, and the transformer is trained to predict each of them in an autoregressive manner.

DALL-E does not include the image results of people, but results of subsequent models adopting similar transformer architecture [5] [6] show poor performance in generating people images. The people can have blurry faces, distorted bodies, out of proportion bodies and missing or additional limbs. We hypothesise that the text’s inherent ambiguity partly makes it difficult for the model to learn the mapping to the human body’s high degree of freedom. This challenge includes describing body shape, size, pose, clothing details, position in the images, camera view and the number of people, all difficult to describe in words
fully. Thus, we propose to add human pose as an additional input constraint to improve the correctness and fidelity of people being generated. To our knowledge, our paper is the first to use both text and human pose in the generative image model. We can see this as enforcing disentanglement of content and style of image \cite{7} where the content is the pose, and the text depicts the style. Furthermore, motivated by the inadequacy of existing metrics to measure image errors specific to people, we devised a People Count Error (PCE) metric to measure the false positive rate when generating images of multiple people. Therefore, we can empirically and qualitatively show that the disentanglement improves the fidelity of people generated with a significantly reduced number of false positive or erroneous body parts.

![Image of text-and-pose-to-image model](image.png)

Fig. 1: Our text-and-pose-to-image model takes in text prompt and the target pose as inputs to generate photo-realistic people images. Our method supports partial pose view, multiple people, all at different scales.

The current method of representing pose for transformer is to convert skeleton image into discrete image token with dVAE. This is the method adopted by VQGAN \cite{8} which frames pose-to-image as an image-to-image problem. However, this image pose encoding has shortcomings, primarily high computational complexity, which increases quadratically with image resolution, limiting the output resolution of images at around 256 x 256. Therefore, we have devised a novel, efficient and accurate pose representation suitable for a transformer - Keypoint Pose Encoding (KPE). Instead of using the high dimensional skeleton image containing redundant information, we use body joint keypoints for pose representation. The low dimensional representation is invariant to changing the target image resolution or domain, e.g. from the natural landscape to synthetic objects. It is also 10x more memory efficient and increases computational inference speed by over 73%. Finally, we also demonstrate that KPE can generate video by using pose information to offer consistency in people’s appearance from frame to frame. This is infeasible with existing text-to-image models. The advantages of our proposed method against other methods are summarised in Table 1. In summary, our contributions are:
1. **Pose and Text Guided multi-person Image Generation** We are the first to use both text and pose guided models to generate images of multiple people. In addition, our method can be used for image sequence generation.

2. **Novel Keypoint Pose Encoding (KPE)** To enable the *tokenisation* of a human pose representation that is computationally efficient and invariant to changes in target images such as the resolution.

3. **Introduction of a person centric metric** A new metric to measure the false error rate of generated humans in multi-person images in generated images. Helping to illustrate that the disentanglement of pose and text leads to better higher quality results with reduced false positive humans.

2 **Related Work**

2.1 **Text-to-Image Generation**

In GAN-based text-to-image models, early works such as StackGAN\[9\], AttenGAN\[10\], or XMC-GAN\[5\] are forms of conditional GAN \[11\] where the text sequence is projected to an embedding vector as a conditioning feature, i.e. the text is modelled as continuous variables. More recently, researchers turned their attention to the transformer architecture that has proven to excel in natural language processing (NLP) tasks. Instead of using a single embedding vector for the entire word sequence, it is encoded into a sequence of discrete text tokens. Each token is then projected into the transformer’s embedding space. The transformer is trained to predict the text tokens autoregressively in sequence.

However, there are challenges in applying the transformer to computer vision tasks. The high dimensionality of image pixels makes it computationally expensive as the transformer computational complexity is $O(N^2)$. Moreover, image pixels are considered continuous real values and do not fit into the discrete token needed by the transformer. DALL-E \[2\] addressed these problems by using VQ-VAE \[12\] which is a variant of dVAE to encode the image into discrete tokens. Hence, both text and image are represented as discrete tokens, allowing the adoption of generic transformers to predict the image tokens from text tokens. The resulting image is generated by feeding the image tokens to the decoder part of dVAE. This method forms the basis of many modern transformer-based text-to-image models such as CogView \[6\] and NUWA \[13\]. We employ a similar method to tokenise the text in our models but condition our transformer on a further additional modality, i.e., pose, to enrich the image quality and precision.

2.2 **Pose Guided Image Generation**

There are a few ways of representation pose in the context of image generation; including, body keypoint heatmap \[14\] [15] [16], segmentation maps \[17\] and a skeleton image \[18\] [19]. These representations can be generalised as a 2D spatial tensor $\mathbb{R}^{H \times W \times C}$ where $H$ and $W$ are the image height and width. $C$ is the number of image colour channels or classes for segmentation and heatmap. In
other words, pose-to-image is framed as image-to-image where an input image is a form of 2D spatial tensor representing pose, and the output image is the person image. However, these representations have many redundancies; for example, a segmentation map includes every body pixel and background. The same goes for skeleton images, dominated by background pixels that do not carry any useful information about the pose. A skeleton image and the keypoint heatmap are derived from Cartesian coordinates of body joints key points that contain all the pose information. The reason for using the 2D spatial tensor for pose representation is that the spatial information is required for convolutional layers in GANs [20], but this is no longer a pre-requisite for transformer. Despite this, recent generative transformer models [8,13] continue using an image for pose representation by encoding image into discrete image tokens. The image tokenisation can take be very slow. [21] attempts to reduce the training time by pre-encoding the images into tokens, but this prohibits the use of augmentation onto the images and poses during training. Thus, several papers [22, 23] came to realise the shortcoming of using skeleton images and started to use keypoint for pose estimation regression. However, we are the first to use pose keypoint to guide the image generation.

Table 1: Comparing features with other person image generation models. Our method is the only one that support both text and pose input. *NUWA uses only text or pose but not both at the same time. Our pose encoding method is fast, memory efficient and invariant to changes in image data (e.g., resolution, domain) or image encoder.

As shown in Table 1, our proposed method is the only model that uses both text and pose to guide the generation of people in images. While NUWA supports text guidance and pose guidance, they do not co-exist in the same model. Furthermore, NUWA is a 3D transformer that is more computationally expensive to run than our model that is based on a conventional 2D transformer.

3 Method

Figure 2 shows the overall architecture of our pose guided text-to-image model. The first stage is to convert the text, pose keypoints and image into tokens with their respective encoders. Then the tokens are projected into embedding before
adding positional encoding. We use learnable positional encoding for text tokens, and axial positional encoding [24] for image tokens due to its 2D structure. We do not use positional encoding for keypoint tokens as they are equally important for all positions of the image tokens. The transformer is made of 12 transformer encoder blocks adopted from [25], and there is no transformer decoder block. We will now describe the details of token encoders.

![fig2.png](image)

**Fig. 2:** Architectural diagram of our pose-guided text-to-image generation model. The text, pose keypoints and image are encoded into tokens and go into an encoder-only transformer.

### 3.1 Text Encoder

Like DALL-E [2], we use the BPE (Byte Pair Encoding) tokeniser [26] for text tokenisation, the encoder breaks the word into subwords, and they are assigned discrete identifier numbers which become the text tokens. The text tokens are then projected into embedding $\mathbb{R}^{t \times d}$, where $t$ is the fixed text token input length and $d$ is the transformer dimension. The BPE tokeniser is pretrained with the vocabulary of the dataset.

### 3.2 Image Encoder

In our model, we use VQ-VAE [12] from VQGAN [8] for image encoding and decoding. Its GAN training pipeline has produced better image quality than the dVAE used by DALL-E. The encoder $E$ first converts the continuous image $x \in \mathbb{R}^{H \times W \times C}$ with height $H$, width $W$ and colour channel $C$ into code $\hat{z} = E(x) \in \mathbb{R}^{h \times w \times n_z}$ with $h$ and $w$ the spatial dimension and $n_z$ the dimension
of the code. Then, each of the codes is quantised $q(\cdot)$ to its closest discrete codebook entry $z_k$ using equation:

$$z_q = q(\hat{z}) := \arg\min_{z_k \in Z} \|\hat{z}_{ij} - z_k\|$$

where $Z = \{z_k\}_{k=1}^K \in \mathbb{R}^{n_z}$ is the discrete codebook. The decoder $G$ reconstructs the discrete code into image $\hat{x}$:

$$\hat{x} = G(z_q) = G(q(E(x)))$$

The model and codebook can be trained end-to-end using the loss function:

$$L_{VQ}(E, G, Z) = \|x - \hat{x}\|^2 + \|sg[E(x) - z_q]\|^2$$

where $sg[\cdot]$ denotes stop-gradient operation.

Overall, the image is tokenised into $h \times w$ grid of discrete image tokens. Although $h$ and $w$ are hyperparameters, they have a linear correlation with image resolution $H$ and $W$ to maintain the same quality of image texture details. Therefore, increasing image resolution will lead to more extended image token lengths, resulting in a quadratic increase in computational complexity. The image encoder is pretrained with the target images like a text encoder. The discrete image tokens are then projected into transformer embedding $\mathbb{R}^{h \times w \times d}$.

### 3.3 Keypoint Pose Encoder (KPE)

KPE is our method for pose representation. It converts detected pose positions for multiple people into keypoint tokens and then encodes them into a keypoint embedding.

A single 2D keypoint is defined as a tuple of $(x, y, v)$ where $x$ and $y$ are the normalised Cartesian coordinates in $[0, 1]$ and $v$ is the visibility score in $[0, 1]$. We denote multi-person 2D keypoint format as $(x, y, v)_{i,j}$ where $i$ is the person index, $j$ is the keypoint index from 0 to $N - 1$ where $N$ is the total number of keypoint defined. Different pose estimation models use different keypoint schemes, but the common keypoints are the nose, neck, shoulder, elbow, wrist, hip, knee, ankle, eye, ear, big toe, and heel.

Figure 3 shows the process of KPE. The process is similar to the method in Vision Transformer (ViT) [27] of grouping image pixels into image patches, except that we group keypoints into keypoint tokens. Each keypoint token corresponds to a skeleton joint; in this illustration, keypoint token 0 is designated for the right eye and keypoint token 1 is for the left wrist. The right eye keypoint for all the people in the image are stacked together and moved into token 0. In this example, we define the system to support up to four people, and if there are fewer than the maximum number of people, they will be padded with zeros in the keypoints. The same goes for keypoints that are not visible. The resulting keypoint tokens will have a length of $N$. In other words, the number of fixed with the keypoint definition does not change with the number of people in the
Fig. 3: Block diagram showing KPE encoding multi-person pose to keypoint tokens by using a maximum of three people as an example. If the image contains fewer people, the tokens will be padded with zeros. The tokens are flattened and projected into keypoint embedding within the transformer. The skeleton image is for illustration purposes as we use the keypoints directly from the pose estimation model’s outputs.

image or changes in image resolution. One thing worth noting is that keypoint token is not discrete; it is a continuous value within the range \([0, 1]\).

The next step is to ensure that the pose embedding has the same dimension as the transformer embedding. We propose two methods; the first one is to pad the keypoint tokens with zeroes to match the transformer dimension. This method is the fastest as it does not require any arithmetic computation. It can accommodate many people up to constraint within \(3M \leq d\) where \(M\) is the maximum number of people, and \(d\) is the transformer dimension. We tested this method to be working faster. However, for the generality of the unbounded number of people, we use a linear layer to project the keypoint tokens into keypoint embedding. Overall, the KPE converts multi-person keypoints \(\mathbb{R}^{M \times N \times 3}\) into embedding \(\mathbb{R}^{N \times d}\).

3.4 Training

To train, the text tokens \(T\), keypoint tokens \(K\) and image tokens \(I\) are concatenated to be fed into the transformer. The transformer output has the same length as the input, and it aims to generate the same tokens as the input tokens. As the text tokens and image tokens are discrete, prediction of them become a multiclass classification problem, and we use the cross entropy loss \(\mathcal{L}_{ce}\) as is common with transformer training. However, the keypoint tokens are continuous values, and we use an \(L2\) loss \(\mathcal{L}_{L2}\) on the keypoint embedding. Therefore, the overall loss function is:

\[
\mathcal{L} = \mathcal{L}_{ce}(T) + \lambda_I \mathcal{L}_{ce}(I) + \lambda_K \mathcal{L}_{L2}(K)
\] (5)
\( \lambda_I \) and \( \lambda_K \) are constants. Higher value encourages more accurate people image and pose respectively, and their values are discussed in Section 4.3.

### 3.5 Inference of A Pose Guided Image

The image token input is left blank at inference, only providing values for the text and keypoint inputs. After a forward pass, we sample the first image token at the transformer output. If we select the image token with the highest probability, the model will generate the same image every time and become deterministic. Therefore, to introduce variation in generating images, we select several tokens with the highest probability and sample them uniformly. The size of the sampling pool is a hyperparameter; setting it too low can give more variation but may also introduce more errors in the image. The sampled token is used in the input image token at the same position and used as a condition together with text and keypoint tokens to predict the next image token. The process then repeats for subsequent image tokens for the given text and pose inputs.

### 3.6 Generating Image Sequence

While our model is not a fully fledged video generation model that trains using video samples like NUWA, we can use a trained model to generate a sequence of images. This is difficult with DALL-E or similar text-to-image models as the person will appear in random poses across video frames. While pose-to-image models like VQGAN could use pose sequences to ensure consistent poses, they suffer from an inconsistent appearance like clothing colour. We can constrain the appearance by using the same text caption and reducing the sampling pool of image tokens during inference. To anchor the top of the image result with the previous frame, we use the image tokens from initial image rows from the first video frame as image input for subsequent frames.

### 4 Experiments

In this paper, we implement our KPE model and train it on a multi-person dataset derived from the DeepFashion [1] dataset. In addition, we implemented two baseline models for comparison. DALL-E [2] as an ablation study to understand the effect of using pose guidance on generated image quality. Moreover, we implement a DALL-E [2] + VQGAN [8] model to compare the performance of KPE against VQGAN’s method of using skeleton images for pose representation. We highlight the advantages of KPE in Section 5.1.

#### 4.1 DeepFashion Dataset

We use DeepFashion’s fashion synthesis benchmark dataset for the experiments. The original dataset contains 78.5K images of a single person and a brief description of the gender, clothing colour, and type. However, most of the samples
in this dataset contain a single person standing in the centre with a neutral pose. We believe that text-to-image models will produce satisfactory results even without pose guidance. This motivated us to derive a multiscale and multi-person dataset from the original images. Therefore, we randomly sampled the single-person images, randomly resized them by 10%, cropped them, and concatenated them into a single $256 \times 256$ image. As a result, the new images will have between 1 to 3 people in various locations, sizes and poses. This adds about 49.5K of images for 2 and 3 people, bringing the total training dataset to 177.5K. In contrast to some generative approaches, we create a train/test split to ensure that the models are not exposed to the reserved test dataset in the training process. Therefore, all image results in the paper were generated using the test set. The test dataset contains 446, 299 and 255 samples for 1, 2 and 3 people. Then we use the OpenPose [28] pose estimation approach to obtain the multi-person pose keypoints. We will release code to enable the generation of the multi-person training and test data and splits.

4.2 Evaluation Metrics

We evaluate the performance of our approach with a range of quantitative metrics in addition to qualitative results. To evaluate the realness of the images, we use the perceptually trained Inception Score (IS) [29] in line with literature image generative models [14] [19] [2] [13] [6]. However, as studied by [30], IS that uses [31] pretrained on Imagenet [32] does not work well for data that does not fall into Imagenet’s 1000 classes. Unfortunately, Imagenet does not include a human category despite its many classes of clothing, e.g., t-shirts, jeans, sweatshirts, miniskirts etc., that may have a human in it. Therefore, images with perfect clothing but unrealistic looking people may still get high IS. This is worse for the multi-person dataset, spreading the Inception model’s prediction to multiple clothing classes, lowering IS.

To overcome this limitation, we propose a new evaluation metric that uses a pose estimation model such as OpenPose [28] that was trained on the multi-person dataset. We call this metric People Count Error (PCE). Given an image of people $x$, $gt(\cdot)$ indicates ground truth function that returns the labelled number of people in the image, and $h(\cdot)$ is the function that returns the number of people detected by the pose estimation algorithm; therefore, PCE is defined as:

$$ PCE(x) = \begin{cases} 1, & \text{if } h(x) \neq gt(x) \\ 0, & \text{otherwise} \end{cases} \quad (6) $$

Generated images can have false positive errors, such as having additional detected joints or arms in an impossible pose. As most pose estimation approaches such as OpenPose have explicitly learnt the human body anatomy, it would assign the additional body parts in doubt to a new person, increasing the people count and flagging PCE as 1. Unlike IS, which requires a large amount of data, PCE applies to a single image, making it usable to find an error in an individual image. Visual examples of results that PCE would return as 1 are shown in the results in Fig 5.
A further metric to measure the accuracy of the generated images is Object Keypoint Similarity (OKS) from the MSCOCO keypoint challenge [33]. OKS measures the $L_2$ distance between the ground truth joints labels and the estimation from the generated image, giving higher weights to keypoints perceptually more important like shoulders, knees, hips to eyes, nose, and ears. To evaluate the pixel accuracy against the held out test data, we perform Structural Similarity (SSIM) and, as stated in [14] masking out the background. In addition, we crop away the excessive background as we found this can dominate SSIM when people appear small within the image.

4.3 Implementation details

We adopt a two-stage training process like DALL-E. The first stage trains VQ-VAE using the VQGAN pipeline on the target images to encode $256 \times 256$ images into $16 \times 16 = 256$ image tokens where each token can assume 8192 possibilities. We use an open-source implementation of DALL-E [25] in which the transformer dimension $d=512$, with 8 heads and a depth of 12 encoder blocks. The text token length is set as 256, and the input text tokens will be truncated if they exceed this length. We train using the loss function in Equation 5 using OpenPose’s BODY [25, 34] pose format. Therefore, the keypoint token length is 25, corresponding to the 25 keypoints.

The DALL-E+VQGAN model is also a text-and-pose guided model. The difference with our KPE model is that it uses VQGAN’s pose representation method of using VQ-VAE to encode skeleton images into pose image tokens. Like VQGAN, we reuse the VQ-VAE pretrained on target images. The DALL-E+VQGAN’s loss function is:

$$\mathcal{L} = \mathcal{L}_{ce}(T) + \lambda_I \mathcal{L}_{ce}(I) + \lambda_K \mathcal{L}_{ce}(P)$$

where $P$ are the pose image tokens. We use the same VQ-VAE, training configuration and hyperparameters for all the models. For loss constants, We use $\lambda_I=7$ from [25]. We tried 1 and 10 for $\lambda_K$ but did not notice much difference in the results. Eventually, we select $\lambda_K=10$ to have the same order of magnitude as $\lambda_I$. For the optimiser, we use Adam [35], with initial learning rate of 0.0001, $\beta_1=0.9$, $\beta_2=0.999$. The learning rate is reduced by half if the loss has plateaued for 12 epochs until it reaches 1e-6. We use a batch size of 10 and train for 100 epochs on an RTX5000 GPU with 16GB GPU memory.

Due to nondeterministic nature of image generation, we compute 5 images per sample in the test dataset and obtain the mean value of the metrics for the qualitative result. As stated before, we sample image tokens from the top 0.1% highest probability or 8 out of 8192 tokens. This sampling improves the image quality and consistency of metrics values.

5 Results

Table 2 shows that KPE tops all the evaluation metrics. The high IS indicates that KPE can generate realistic looking people. Figure 4 show examples showing
a variety of genders, the number of people, multiscale person, poses and partial pose with missing keypoints. Given the OKS score of 0.97, which indicate a highly accurate pose, the high mask-SSIM score suggests the generated images have gender and clothing appearance matching the text description.

Fig. 4: KPE model can generate photorealistic people with an accurate pose. This figure shows the pose illustration, ground truth, and three generated samples.

5.1 Comparison with DALL-E+VQGAN
Qualitatively, both KPE and DALL-E+VQGAN produce high-quality people images with an accurate pose. From Table 2 against the baselines DALL-E and DALL-E+VQGAN, we can see that its OKS score matches KPE and is only marginally behind in IS and mask-SSIM, but PCE is twice the error rate of KPE. Apart from generating improved images, there are several advantages of using KPE that makes it an overall superior method:

- **Less memory.** The keypoint token length is an order of magnitude smaller than the pose image token hence requiring less computational memory. In our
experiment, the image token length is 256 while there are only 25 keypoint tokens, making it at least 90% more memory efficient.

- **Faster to run.** The reduction of token number reduces computational complexity, which is in $O(N^2)$ for transformer. Furthermore, encoding pose image using VQ-VAE is computationally expensive and removing this step can contribute to huge speed improvement. As a result, the KPE model’s speed is 73% faster than DALL-E+VQGAN, there is no speed penalty compared to DALL-E.

- **Invariant pose representation.** The same VQ-VAE is used to encode both the pose and target images. However, as VQ-VAE is normally pretrained on natural images, thus the trained VQ-VAE may not perform well on synthetic skeleton images. In contrast, KPE relies only on the keypoint information and is invariant to the image nor VQ-VAE.

- **Scalable.** Increasing target image resolution or quality will require an increase in image token length hence more memory and slower to run. Since KPE is invariant to the image, the pose processing will not increase computational resources as the image resolution increases. This makes our method easier to scale to higher image resolution.

| Pose Method | DALL-E | DALL-E+VQGAN | KPE (Our Method) |
|-------------|--------|--------------|-----------------|
| Number of pose tokens ↓ | - | 256 | 25 |
| Relative inference speed ↑ | 1.73× | 1.0× | 1.73× |
| IS ↑ | 2.912 | 3.027 | 3.034 |
| PCE ($\times 10^{-3}$) | 8.2 | 1.2 | 0.6 |
| OKS (1 person) ↑ | 0.691 | 0.976 | 0.976 |
| OKS (2 people)↑ | 0.555 | 0.970 | 0.970 |
| OKS (3 people)↑ | 0.484 | 0.965 | 0.964 |
| OKS (all) ↑ | 0.598 | 0.970 | 0.970 |
| Mask-SSIM (1 person) ↑ | 0.244 | 0.373 | 0.384 |
| Mask-SSIM (2 people)↑ | 0.302 | 0.457 | 0.461 |
| Mask-SSIM (3 peoples) ↑ | 0.260 | 0.456 | 0.454 |
| Mask-SSIM (all) ↑ | 0.265 | 0.420 | 0.424 |

Table 2: Evaluation of different models on DeepFashion multi-person dataset. Our method KPE achieves highest scores in all metrics.

5.2 Ablation Study

We did an ablation study comparing KPE against DALL-E, a text-to-image model without pose guidance. Looking at Table 2, apart from having lower IS than the other two pose-guided models, it has the highest PCE rate at 8.2 $\times 10^{-3}$, which is over 13 times higher than KPE. Figure 5 shows an example of images that contains errors and how we can spot the error by using PCE. Figure 5a contains two realistically looking people but with an additional long arm floating in the centre of the image. The floating arm is assigned to the third
Fig. 5: Top row are erroneous images generated using DALL-E, and the bottom row shows their keypoints obtained from the images. PCE can capture image errors by comparing the generated and intended number of people. A person, causing PCE=1 as $h(x) \neq gt(x)$. Although measuring the discrepancy in people's count does not catch every error, it allows a standardised metric across a wide range of examples without the need for a manual visual inspection on each. Figure 5b to 5d show examples of additional body parts where PCE=1, some of which can be difficult to see initially, like the additional face in Figure 5d. Also, we found that DALL-E sometimes generates fewer or more people than the text description. When it happens, it tends to contain some standalone body parts like Figure 5a and Figure 5e. The PCE can pick up the error in Figure 5e despite OpenPose failing to detect the incomplete person in the centre.

5.3 Generation of an Image Sequence

Fig. 6: A series of images generated by conditioning on the text and sequence of poses. The top rows of image tokens covering the head are also fixed as conditions to generate subsequent image tokens.
Figure 6 shows how we can animate the image sequence by feeding text input and pose sequence to our KPE model. The text input constraint provides appearance consistency from image to image. Thus, enabling the creation of consistent and highly accurate image sequences driven by an external source, such as a webcam (as was done in this case).

6 Limitations

The DeepFashion dataset is hugely imbalanced, where men form only a tiny fraction of the dataset, and the rest are long-haired white females. Therefore, a pose-only guided model trained on the dataset is more likely to generate female images. To understand the effect of the gender bias, we generated images of various poses using the exact text prompt “a man wore blue shirt” as shown in Figure 7.

Fig. 7: Text of “a man wore blue shirt” was used to generate images from pose that is more masculine (a) towards more feminine pose in (d).

We can see in Figure 7a and 7b that despite the gender bias in the dataset, our model can generate convincing men from neutral poses. However, when presented with poses that are exclusive to females in the dataset, the generated images (Figure 7c and 7d) lean toward feminine appearance. This limitation implies that pose is not entirely disentangled from gender, and the model learned the gender bias from poses containing information about body proportion. Given this insight, in future, we will collect and apply our approach to a more balanced dataset in the future to address this bias in gender and ethnicity.

7 Conclusions

We have proposed the first pose+text to image generative model creating photorealistic multi-person images. We methodologically show that adding pose as guidance improves the image quality over SOTA text only guided generative image model. We also created a novel multi-person pose encoding method suitable for modern transformer architecture, KPE. Not only does KPE generates
a better image and produce a more accurate pose than existing methods, but it is also computationally efficient and does not add inference speed overhead. This allows for easy integration into transformer models. To measure the performance of generative images of people, we propose a suitable metric $\text{PCE}$ to detect the false positive occurrence of generated people, overall creating SOTA performance both quantitatively and qualitatively.
References

1. Ziwei, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Liu Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
2. Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. International Conference on Machine Learning (ICML), 2021.
3. Jason Tyler Rolfe. Discrete variational autoencoders. arXiv preprint arXiv:1609.02200, 2016.
4. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. Conference on Neural Information Processing Systems (NeurIPS), 2017.
5. Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
6. Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. Conference on Neural Information Processing Systems (NeurIPS), 2021.
7. Leon Gatys, Alexander Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
8. Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
9. Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. International Conference on Computer Vision (ICCV), 2017.
10. Tao Xu, Pengchuan Zhang, Qiyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
11. Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. Arxiv:1411.1784, 2014.
12. Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. Conference on Neural Information Processing Systems (NeurIPS), 2017.
13. Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Niùwa: Visual synthesis pre-training for neural visual world creation. arxiv preprint:2111.12417, 2021.
14. Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. Conference on Neural Information Processing Systems (NeurIPS), 2017.
15. Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
16. Lingbo Yang, Pan Wang, Chang Liu, Zhanming Gao, Peiran Ren, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Xiansheng Hua, and Wen Gao. Towards fine-grained human pose transfer with detail replenishing network. *IEEE Transactions on Image Processing*, 2020.

17. Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

18. Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

19. Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

20. Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *NeurIPS*, 2014.

21. Boris Dayma, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phuc Le Khac, Luke Melas, and Ritobrata Ghosh. Dall-e mini, 2021.

22. Tao Jiang, Necati Cihan Camgoz, and Richard Bowden. Skeleton: skeletal transformers for robust body-pose estimation. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021.

23. Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

24. Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *Arxiv:1912.12180v1*, 2019.

25. Phil Wang. https://github.com/lucidrains/dalle-pytorch, 2021.

26. Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *Association for Computational Linguistics (ACL)*, 2015.

27. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference for Learning Representations (ICLR)*, 2020.

28. Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Open-pose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

29. Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv:1606.03498*, 2016.

30. Shane Barratt and Rishi Sharma. A note on the inception score. *International Conference on Machine Learning (ICML)*, 2018.

31. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

32. Olga Russakovsky et al. Imagenet large scale visual recognition challenge. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

33. MSCOCO. https://cocodataset.org/, 2017.

34. OpenPose. https://cmu-perceptual-computing-lab.github.io/openpose/, 2020.

35. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations (ICLR)*, 2014.