The Visual Computer (2024) 40:4635–4646
https://doi.org/10.1007/s00371-024-03447-7

REVIEW

# Few-shot anime pose transfer

Pengjie Wang[1,3] · Kang Yang[1,4] · Chengzhi Yuan[2] · Houjie Li[2] · Wen Tang[3] · Xiaosong Yang[3]

Accepted: 1 May 2024 / Published online: 24 May 2024


**Abstract**

In this paper, we propose a few-shot method for pose transfer of anime characters—given a source image of an anime character and a target pose, we transfer the pose of the target to the source character. Despite recent advances in pose transfer on real people images, these methods typically require large numbers of training images of different person under different poses to achieve reasonable results. However, anime character images are expensive to obtain they are created with a lot of artistic authoring. To address this, we propose a meta-learning framework for few-shot pose transfer, which can well generalize to an unseen character given just a few examples of the character. Further, we propose fusion residual blocks to align the features of the source and target so that the appearance of the source character can be well transferred to the target pose. Experiments show that our method outperforms leading pose transfer methods, especially when the source characters are not in the training set.

**Keywords** Generative adversarial networks · Anime generation · Image generation · Video generation · Meta-learning

## 1 Introduction

Recent years, researchers [1, 5, 7, 8, 26, 27, 30, 32, 34, 43] have proposed numerous algorithms for pose transfer—given a source image and a target image, transfer the pose from the target to the source. Those methods have been conducted on real people and have not taken into account anime characters, which have quite different visual appearance and structure from real people. Real people dataset can be easily constructed by collecting a large number of samples through videos and images, but the anime characters are drawn and are not easy to collect in the same way as real people. Specifically, before training samples can be collected for a character, that character must be created by artist and 3D modeled by animator. This entire process is more expensive and less convenience compared to collecting real people's image or video. Furthermore, dataset for the real people can be very huge, considering huge amount of image/video on the Internet. However, constructing a similarly very large dataset for anime characters is challenging. This motivate us to develop few-shot pose transfer method for anime characters. This brings great challenges to the pose transfer of anime characters. [13] generate images of full-body anime characters with generative adversarial networks (GANs) [12]. They can change the character's clothes and pose. But their method can only adapt to one specific character and limited poses and fail to give satisfying results when source characters have not been observed in training.

In view of these challenges, we find that model-agnostic meta-learning (MAML) [3, 11] provides a learning strategy with which a unseen character in training set can be initialize by fine-tuning in inference based on meta-learned model. In light of this, we propose a few-shot method for anime pose transfer that can learn with small anime character data and generalize well to unseen characters with a few examples of the characters. At the core of our method is a novel pose transfer framework that is especially tailored for anime characters. Figure 1 shows the results of our method.

Our framework solves pose transfer by training conditional GANs containing a generator and a discriminator. The generator and discriminator are trained on multiple tasks per

✉ Xiaosong Yang
xyang@bournemouth.ac.uk

1 School of Computer Science and Technology, Dalian Minzu University, Dalian, China

2 School of Information and Communication Engineering, Dalian Minzu University, Dalian, China

3 National Centre for Computer Animation, Bournemouth University, Bournemouth, UK

4 School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China


**Fig. 1** Given an image of a source anime character and a target pose, our method transfer the pose of the target to the source automatically

batch to gain generalization ability to adapt to unseen characters. We also propose the fusion residual blocks (FRBs) to align the features of the source and the target to generate more accurate textures. With different tasks representing different characters, each task has a support set and a query set. The support set has four samples from different poses, which allow the network to adapt to multi-view appearances of the anime character. The query set is to verify the network's ability of adapting to the new pose and the appearance of the character. One sample of the support set and query set consists of a color image of the character and its corresponding pose image.

The proposed training method has two stages. (a) Character adapting stage: fine-tuning the parameters of the generator on the support set. (b) Character refining stage: the fine-tuned generator is trained on the query set to adapt to new poses and different views. During testing, given the support set of a source character and a target pose, we first fine-tune the parameters of the generator on the support set, and then use the fine-tuned generator to generate a pose transfer result from the source image and the target pose.

Our contributions are as follows:

- We propose the first meta-learning framework that is especially designed pose transfer of anime characters, which trains with multiple stages to gain superior generalization ability.
- Our proposed method, for the first time, can achieve high-quality pose transfer results on unseen anime characters with just a few examples of them.
- Extensive experiments show that our method outperforms baselines and state-of-the-art pose transfer methods both in terms of visual quality and quantitative metrics.

## 2 Related work

### 2.1 Pose transfer

U-net [31], Pix2Pix [17] and Pix2PixHD [35] provide a good network architecture foundation for many pose transfer work [2, 5, 8, 9, 27, 34]. Ma et al. [27] proposed the novel network ($PG^2$) that allows to generate person images in arbitrary poses, on the basis of that person's image and a new pose. Zhu et al. [45] proposed making the resulting features consistent with the characters by adding pose attention to the generator so that the characters are transferred into the pose and can be done with a single image. Chen et al. [8] proposed a novel pose transfer method, pro-

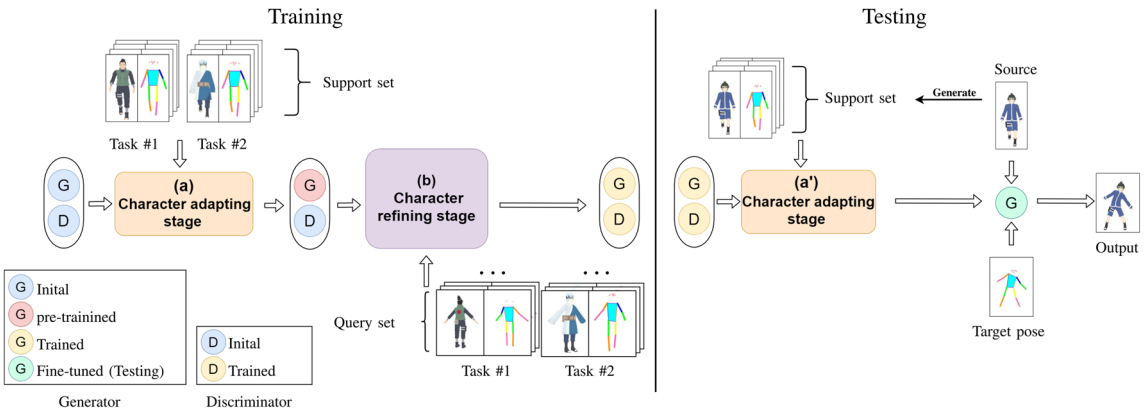



**Fig. 2** Architecture of the proposed method, with the training (left) and testing (right) pipelines. During training, we jointly train a generator ($G$) and a discriminator ($D$). The parameters of the generator is first updated on the support set of each character so that it adapts to the character appearance in the character adapting stage (a). Then, the parameters of the generator and discriminator are updated jointly on the query set of each character in the character refining stage (b). In testing, the generator is fine-tuned on the support set of a source character and then used to generate the image of the character in a target pose

gressive multiattention network (PMAN), which is built on many multiattention transfer blocks with two different attention mechanisms, pose-conditioned batch normalization and cooperative attention mechanism. Tang et al. [32] used two generation branches that modeled the person's shape and appearance, respectively. Yu et al. [30] proposed a globalflow local-attention framework to reassemble the inputs at the feature level. Zhang et al. [43] synthesized a human parsing map aligned with the target pose and then used joint global and local per-region encoding and normalization to generate the final image. Chan et al. [7] proposed pose transfer of target video with the appearance of source video, and normalization of source pose and target pose enables the source character to perform pose transfer in case of changing position. This process relies on pose estimation model, and in order to complete the pose transfer of a particular character, a large number of training sample from the character is required to train the model.

However, these previous methods have difficulty in generalizing well on the test set if the training samples are insufficient. They failed generate high-quality images when the source characters are not in the training set, as demonstrated in Sect. 5. Furthermore, these methods focus on pose transfer on real people and did not take into account of anime characters, which dataset is harder to collect.

To address this issue, we introduce meta-learning to pose transfer of anime characters, enabling our method to generate high-quality pose transfer results for an arbitrary character given just a few samples of the character. Further, we propose a specialized module to accommodate the potential misalignment between the source character and the target pose in the image space.

### 2.2 GANs

Compared to variational auto-encoders (VAE) [23] and PixelRNN [33] generation models, GANs have a broader application. GANs such as Pix2Pix [17], Pix2PixHD [35], and CycleGAN [44] have laid solid foundation for the state-of-the-art GAN study. And the subsequent progressive GAN [20], StyleGAN [21] goes even further to enhance the generation of GANs. Based on these foundations, researchers proposed GAN-based methods for image inpainting [38], text-to-image synthesis [39], unsupervised video summarization [4], single image de-raining [42] and de-snowing [18]. These previous works provide a good reference for the design of our network architecture.

### 2.3 Few-shot image-to-image translation

Finn et al. [11] proposed the model-agnostic meta-learning method (MAML) which can carry out meta-learning training without changing the network structure. Antoniou et al. [3] improve MAML by using multi-step loss optimization and derivative annealing. Zakharov et al. [40] proposed a method of face animation with few samples using AdaIN to control the generated image feature network. Liu et al. [25] proposed a network (FUNIT) based on AdaIN to control image features for unsupervised image-to-image translation. They all used a small number of samples to generate a specific image, verifying the generalization ability of GAN with a small number of samples, and controlled AdaIN to generate an affine transformation, thus effectively guiding the conditional generation.

While the above methods have achieved good results on few-shot image-to-image translation, in this paper, we try to apply few-shot learning to a new scenario, pose transfer of anime characters.

## 3 Method

We aim to train a pose transfer that could be fine-tuned in a few example images of an unseen anime character to generate the image of the characters in any target pose. To this end, we propose to use a conditional GANs setup, and train the generator ($G$) and discriminator ($D$) under a meta-learning framework based on MAML [11] and show our learning framework in Fig. 2. The difference is that we treat a task as a character pose transfer problem. Each task's support set contains $K$ samples of the corresponding character, and thus each task is a 1-way $K$-shot pose transfer problem. Each task's query sets contains a 2D projections of the character that represents a new pose. Each sample is represented by a color image and its corresponding pose image.

In training, the generator is first trained on the support sets of different characters. Then, the generator and discriminator are trained simultaneously on the query set to enable the generator to adapt to new poses. At test time, given an image of a source character along with a target pose, we first generate a support set for the character by pairing up the source image with its pose image generated using an off-the-shelf pose detector. Then, we fine-tune the generator on the support set so that it adapts to the appearance of the source character. Finally, the generator takes as input an image of the source character and the target pose and synthesize an output image of the character in the target pose.

### 3.1 Generator

The generator $G$ has two inputs, a pose image $x$ and a source image $z$, and produces an output image $G(x, z)$. The target pose is represented by a colored stick figure where joint keypoints are connected by lines according to a human skeleton.

***Encoders:*** Our generator $G$ has two encoders, a pose encoder and a texture encoder. The pose encoder encodes $x$ into a pose feature map $P$, and the texture encoder encodes

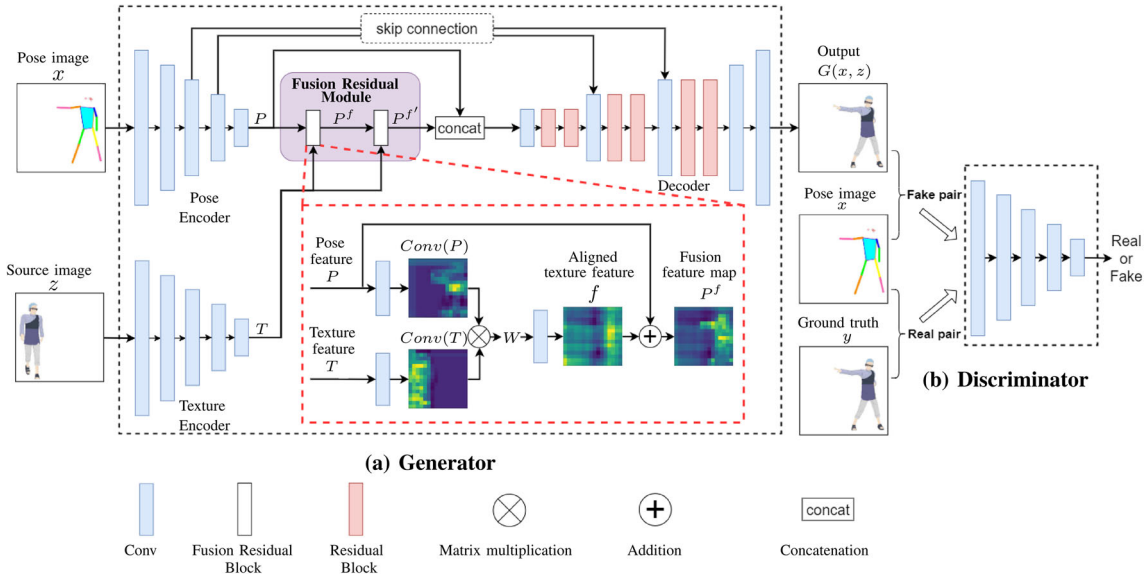



**Fig. 3** Generator and discriminator architectures

z into texture feature map $T$. Both encoders share the same network architecture, which is a five-layer convolutional neural networks with kernel sizes $7 \times 7$, $3 \times 3$, $3 \times 3$, $3 \times 3$, and $3 \times 3$, as shown in Fig. 3.

***Fusion Residual Module:*** We propose fusion residual module (FRM) to align the features of the source and target so that the appearance of source character can be well transferred to the target pose. The FRM consists of two fusion residual blocks (FRBs), whose structure is based on ResNet [14], as shown in the red box of Fig. 4. Given the texture and pose feature maps, we introduce matrix multiplication into FRB to establish direct relationship between the row elements of the texture feature map and the column elements of the pose feature map to bring the source and pose features into alignment. As a result, such matrix multiplication operation has an effect of shifting the features of the source character to align with those of the target pose. In this way, the multiplication result will contain weighted texture features around the position of the target pose. It is then convolved to obtain an aligned texture feature $f$, which is finally added onto the pose feature map to fuse the aligned texture and pose features, giving a fusion feature map $P^f$.

Concretely, we feed $T$ and $P$ into two $3 \times 3$ convolutions, each followed by the BatchNorm [16] and ReLU [29], and get $Conv(P)$ and $Conv(T)$. Then, we perform matrix multiplication in a channel-wise manner to obtain the $W$:

$$W = Conv(T) \otimes Conv(P). \tag{1}$$

Let $e^m_{i,n}$ be the activation at position $(i, n)$ on channel $m$ of $Conv(T)$ and $u^m_{n,j}$ on channel $m$ of $Conv(P)$. $w^m_{i,j}$ is calculated as:

$$w^m_{i,j} = \sum_{n=1}^{N} e^m_{i,n} u^m_{n,j}. \tag{2}$$

Since the input source image and pose image have pure white background, their feature maps only activate around locations where the source character and pose stick figure exist.

In the FRM, the FRB is applied two times in succession, as shown in Fig. 4. The first FRB takes as input $P$ and $T$ to output $P^f$, which is fed into the second FRB along with $T$ to produced the final output of the FRM.

Figure 4 visualizes aligned texture feature maps generated by the FRB when given a single source image and a set of target poses with the stick figures at different positions in the image space. As can be seen, as the position of the stick figure changes across the pose images, the resulting aligned feature map can shift the features of the source (i.e., the vertical rectangle of high activations) accordingly to align with the stick figure in the image space.

***Decoder:*** The output of the FRB is concatenated with the pose feature map, and then fed into a decoder to obtain an output image $G(x, z)$. The decoder is formed by stacks of a convolutional layer and two residual blocks, followed by two convolutional layers, as shown in Fig. 3. The first convolutional layer has $3 \times 3$ kernel size. Through the three conv-residual stacks, the spatial resolution of the feature map is progressively doubled while the output channels are halved. Changes in spatial resolution are achieved via upsampling. The final two convolutional layers have kernel sizes of

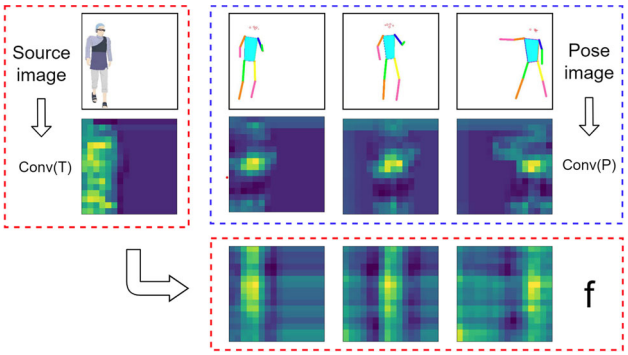



**Fig. 4** Visualization of the FRB feature maps. Left: a source image. Right: a set of target pose images. Bottom: output aligned texture feature maps

$3 \times 3$ and $7 \times 7$. We introduce skip connections as in the U-net from the encoder to the decoder so that the decoder can easily access important low-level pose information.

### 3.2 Discriminator

Our discriminator follows the discriminator architecture of Pix2Pix [17], which consists of four convolutional layers of kernel size $4 \times 4$. The discriminator takes as input fixed size patches randomly sampled from input images and classify them as real or fake. The discriminator implicitly drives the performance of the generator, because the generator needs to generate more realistic images to confuse the discriminator.

### 3.3 Loss functions

Our network is trained through two stages, as shown in (a) and (b) of Fig. 2. We use the GAN loss and $L1$ loss from Pix2Pix [17] for those two stages. We also use the perceptual loss introduced by [19] which calculates $L1$ distance between feature maps of a pretrained network. Let $\mathcal{R} = \{T_i\}_{i=1}^{N}$ be our training dataset with $N$ tasks, where $T_i = (\mathcal{Z}_i, \mathcal{S}_i, \mathcal{Q}_i)$ denotes the data for the $i$-th character (task) and $z^i$ is a source image of the character $\mathcal{S}_i = \{(x_s^i, y_s^i)\}$ is the support set, where $(x_s^i, y_s^i)$ is a support set sample, with $x_s^i$ and $y_s^i$ being a pose image and ground-truth image, respectively. $\mathcal{Q}_i = \{(x_s^i, y_s^i)\}$ is the query set. During training, we sample a batch of characters per iteration and update the parameters of our model using the losses defined as follows.

***Character Adapting Loss:*** The character adapting stage trains $G$ on the support sets of the sampled characters. For each sampled character $i$ whose source image is $z^i$, we iterate over all the samples in its support set $\mathcal{S}_i$ and update $G$ once using each sample. The GAN loss $\mathcal{L}_A^{GAN}$, $L1$ loss $\mathcal{L}_A^{L1}$ and perceptual loss $\mathcal{L}_A^{prec}$ for a sample $(x_s^i, y_s^i)$ are written as:

$$\mathcal{L}_A^{GAN} = \log(D(x_s^i, y_s^i)) + \log(1 - D(x_s^i, G(x_s^i, z^i))), \quad (3)$$

$$\mathcal{L}_A^{L1} = \|y_s^i - G(x_s^i, z^i)\|_1, \quad (4)$$

$$\mathcal{L}_A^{prec} = \sum_n \|\phi_n(y_s^i) - \phi_n(G(x_s^i, z^i))\|_1. \quad (5)$$

where $\phi_n$ is the activation map of the $n$-th layer of a pretrained network. The full loss for this stage is:

$$\mathcal{L}_{CA} = \mathcal{L}_A^{GAN} + \lambda_l \mathcal{L}_A^{L1} + \lambda_p \mathcal{L}_A^{perc}. \quad (6)$$

***Character Refining Loss:*** With the trained models in the previous stage, we further train $G$ and $D$ with a character refining loss on the query set of this batch of sampled characters. For ease of explanation, we define the loss in terms of a single character. In practice, we need to compute the mean over all the characters. The character refining loss $\mathcal{L}_{CR}$ encourages the generator $G$ to handle a wide range of different target poses and different viewpoints in pose transfer. For a sampled character $i$ with the source image $z^i$ and the query set $\mathcal{Q}_i$, we draw a query sample $(x_q^i, y_q^i)$ from $\mathcal{Q}_i$ and define $\mathcal{L}_{CR}$ as:

$$\mathcal{L}_{CR} = \mathcal{L}_R^{GAN} + \lambda_l \mathcal{L}_R^{L1} + \lambda_p \mathcal{L}_R^{perc}, \quad (7)$$

$$\mathcal{L}_R^{GAN} = \log(D(x_q^i, y_q^i)) + \log(1 - D(G(x_q^i, z^i), x_q^i)), \quad (8)$$

$$\mathcal{L}_R^{L1} = \|y_q^i - G(x_q^i, z^i)\|_1, \quad (9)$$

$$\mathcal{L}_R^{prec} = \sum_n \|\phi_n(y_q^i) - \phi_n(G(x_q^i, z^i))\|_1. \quad (10)$$

where $\mathcal{L}_R^{GAN}$, $\mathcal{L}_R^{L1}$ and $\mathcal{L}_R^{perc}$ are GAN loss, $L1$ loss and perceptual loss, respectively. Character refining loss can force the generator to adapt to the appearance of new characters at multiple different views and different poses.

***Multi-step Loss:*** Considering that the model is iterated over the support set several times during the character adapting stage, calculating the query set loss using only the final trained model may result in losing some of the optimization information. In light of this, we leverage multi-step loss (MSL) [3] during the training phase. Specifically, we calculate the query set loss $\mathcal{L}_{CR}$ after each iterative update on the support set, and use the weighted sum of all query set losses as the final character refining loss, which allows a more accurate optimization of the model parameters $\theta$:

$$\theta = \theta - \beta \cdot \nabla_\theta \sum_{i=1}^{N} \sum_{s=1}^{S} v_s \mathcal{L}_{CR}^{\mathcal{Q}}. \quad (11)$$

where $\beta$ is the learning rate, $N$ is the number of tasks in each batch size, and $S$ is the number of iterations of each task on the support set. $\mathcal{L}_{CR}^{\mathcal{Q}}$ denotes the query set loss of all sampled characters computed after $s$-step iterations of each task on the support set. $v_s$ denotes the importance weight of $\mathcal{L}_{CR}^{\mathcal{Q}}$, which is calculated in the same way as [3] and is gradually increased during the iterations of each task.

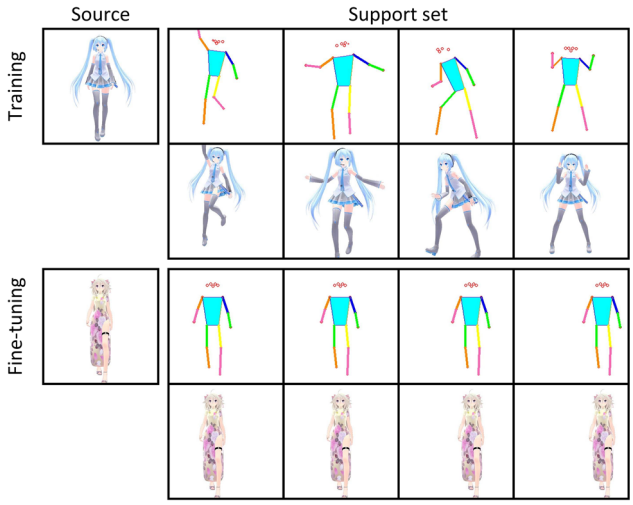



**Fig. 5** One sample of the source image and the support set ($K = 4$)

## 4 Dataset

To train and evaluate our model, we use Unity to create a dataset by rendering a set of virtual anime characters. All of these characters are common animated characters with a wide variety of clothes and actions with common size and body proportions. We render the color image of the characters and create the corresponding pose images by detecting 17 joint keypoints on each color image with an off-the-shelf pose detector, AlphaPose [10, 24, 37]. We render using a fixed camera position and focus on the character.

Specifically, for each character, we use four different orientation views of front, back, left, and right to render the character's pose as the basic pose. In addition, we use dancing animation sequences of the characters that contain different poses to generate samples. For each animation sequence, we sample multiple poses over time, ending up with about 140 samples for each character. The training set contains a total of 37,289 samples from 214 anime characters, and the testing set contains a total of 7,918 samples from 46 anime characters. These samples vary in terms of pose, viewpoint, character etc. Each sample is in resolution $256 \times 256$ with white background. We select samples based on $K$ to construct support set, and one of the data samples is shown in Fig. 5.

## 5 Experiments and results

### 5.1 Implementation details

In the training phase, we iterated over 96 epochs. The weight $\lambda_l$ is 75.0 and $\lambda_p$ is 0.5. We set the size of the support set $K = 4$, the number of tasks in a batch is 4, and the number of iterations in the character adapting stage per task is 4. For the character refining stage, we use Adam [22] optimizer to optimize the model parameters. We set the initial learning rates of the generator and discriminator as 0.0002 and 0.0004 according to the TTUR update method [15]. We also use the cosine annealing algorithm to update the learning rate of the generator. Specifically, we use the cosine annealing algorithm to set a decreasing learning rate for the generator with a minimum of 0.00005 during the first 64 epochs of the training phase and maintain the minimum learning rate for the next 32 epochs. For the character adapting stage, the generator are updated by the gradient descent algorithm, with a learning rate of 0.001. The convolutional layers for both the generator and the discriminator use spectral normalization [28, 41]. We train with an image resolution of $256 \times 256$.

In the testing phase, we used a learning rate of 0.03 for the character refining stage to utilize the basic pose fine-tuning generator for each character, then test on the test dataset.

When using multi-step loss optimization, we set the size of the support set $K = 2$, the number of tasks in a batch is 4 and all others remain unchanged. Please note that some results were obtained from a model trained with flexible setting for efficiency, like without perceptual and MSL losses, and 2 tasks per batch, etc. These include Figs. 7, 8, 10 and 11.

### 5.2 Compared methods

Compared Methods. We compare with an image-to-image translation baseline, Pix2Pix [17]. Furthermore, we compare with three leading pose transfer methods, $PG^2$ [27], PATN [45] and XingGAN [32]. For fair comparison, we used the training split of our dataset to train on these networks.

### 5.3 Evaluation metrics

We use the Fréchet inception distance (FID) [15] to compare the feature statistics of generated images and the real images. To compute the FID, we use features from a model trained on a dataset of anime characters, i.e., Danbooru2018 [6]. We also adopt the structural similarity index (SSIM) [36] to measure the perceptual distance between generated images and their ground-truth image.

### 5.4 Comparison with prior methods

For this experiment, we set support set size $K = 1$ during test for fair comparison. Given a source image of a character, and the sample in its support set is formed by the ground-truth image and a pose image obtained by applying the AlphaPose [10, 24, 37] to the ground-truth image.

Figures 6 and 7 show the qualitative results of different methods on our test dataset and in-the-wild YouTube video frames of real people with the background removed, respec-


**Fig. 6** Visual comparison of our method (Ours), PATN, $PG^2$, Pix2Pix and XingGAN on our test dataset. The support set size in our method is set to 1 during test and the MSL is used

tively. We find that while all the methods respect the target poses well, our method produce results with much better visual quality than the other methods on the anime characters in Fig. 6. The Pix2Pix generates blurry results, while the PATN, XingGAN and $PG^2$ synthesize wrong textures particularly at the face and clothing regions. Specifically, in second row of Fig. 6, our method preserves more detailed feature such as clothing and hair, while other methods have difficulties in reconstructing these fine details. In the addition, in the last row of Fig. 6, it is evident that our method has more accurate color transformation. This suggests that the other methods fail to generalize to the unseen characters, while our method can handle them favorably given only a single example of each character. The superior generalization ability of our method is further demonstrated in Fig. 7, where all the source images contain real people. Despite

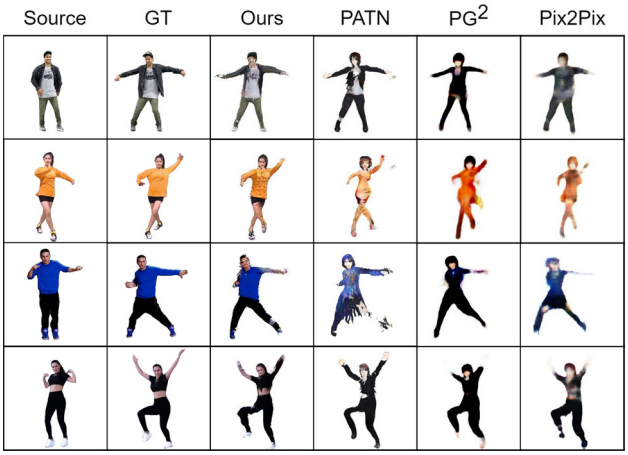



**Fig. 7** Visual comparison of our method (Ours), PATN, $PG^2$, and Pix2Pix on YouTube video frames. The support set size in our method is set to 1 during test


**Fig. 8** Results of our method under 1-shot, 4-shot and 10-shot

being trained only on anime characters, our method is able to synthesize good appearance on the target poses successfully, while the other methods struggle with giving reasonable results. Table 1 shows the FID and SSIM scores for different methods. We can see that our method outperforms the other methods in terms of both metrics. More example results by our method on both anime character and real people can be found in Fig. 10.

## 5.5 Number of shots

The results in Sect. 5.4 show that our method can achieve compelling performance under 1-shot setting. We experiment with varying the number of shots. To this end, given a source image during generation, we generate $K$ support set samples by horizontally shifting the character in the image to create a $K$-shot setting. Figure 8 shows the results generated by our network with the number of shots varying from 1, 4 to 10. As expected, when the number of shots increases, the visual quality of the synthesis results improves gradually, with more sharp texture details. The quantitative results reported in Table 2 also indicates that increasing the number of shots will result in better performance.

**Table 1** Quantitative results of different methods on our test dataset

| Method | FID↓ | SSIM↑ |
|---|---|---|
| PATN | 6.77 | 0.807 |
| $PG^2$ | 6.61 | 0.820 |
| Pix2Pix | 8.37 | 0.843 |
| XingGAN | 7.31 | 0.812 |
| Ours (1-shot) | 5.78 | 0.871 |
| Ours (1-shot) (MSL) | **4.03** | **0.872** |

The best results are in bold

**Table 2** Effects of different shots and multi-step loss

| Method | FID↓ | SSIM↑ |
|---|---|---|
| Ours (1-shot) | 5.78 | 0.871 |
| Ours (4-shot) | 4.31 | 0.872 |
| Ours (10-shot) | 4.01 | **0.878** |
| Ours (1-shot) (MSL) | 4.03 | 0.872 |
| Ours (4-shot) (MSL) | **2.33** | 0.874 |

The best results are in bold

**Table 3** Costs of training and fine-tuning. The MSL is unused, and the time is reported in unit of minutes

| Stag | Method | Time |
|---|---|---|
| Training (one epoch) | PATN | 7 |
| | $PG^2$ | 8 |
| | XingGAN | 12 |
| | Ours (4-shot) | 85 |
| Fine-tuning | Ours (4-shot) | <1 |

**Table 4** Results of the ablation study (4-shot)

| Method | FID↓ | SSIM↑ |
|---|---|---|
| w/o ML | 5.30 | 0.837 |
| w/o FT | 5.92 | 0.827 |
| w/o FRM | 4.31 | 0.852 |
| w/o MSL | 3.71 | 0.862 |
| Ours (MSL) | **2.33** | **0.874** |

The best results are in bold

## 5.6 Multi-step loss optimization

Multi-step loss can provide more accurate loss information. Therefore, we use it in the early stage of training to help the model iterate and optimize quickly and in the later stage of training to save training cost by adopting the previous training strategy.

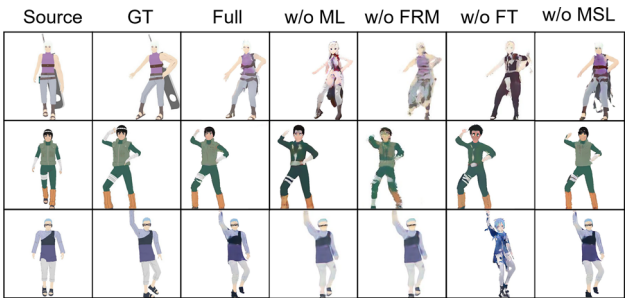



**Fig. 9** Visual comparison of our full method (Full) against its three ablations that train without the meta-learning framework (w/o ML), without the FRM in the generators (w/o FRM), without using fine-tuning (w/o FT), and without the multi-step loss (w/o MSL) during training, respectively. The support set size is set to 4

We compare the results of training with multi-step loss for the case of $K = 4$. The quantitative results in Table 2 show that using multi-step loss optimization leads to slightly better performance.

### 5.7 Training and fine-tuning time

Table 3 shows the time costs used of training our method and previous methods. Due to employing the MAML framework, our method need more training time than previous methods. However, when performing inference on new unseen character, our method needs just fine-tuning instead of training. For each new character, the fine-tuning takes only less 1 min to complete.

### 5.8 Ablation study

To analyze the necessity of each important component in our method, we perform an ablation study by comparing our full method with its several ablations:

- **w/o ML:** To evaluate the effect of our meta-learning framework, we experiment with removing the ML framework. As shown in Fig. 9, without the ML, the outputs do not adapt to the appearance of the characters in the input source images, synthesizing images of random characters. This suggests that the ML is crucial to the excellent generalization ability of our method given only a small number of samples.
- **w/o FT:** In testing, we need to fine-tune (FT) our generator on the support set of the source character for a number of iterations. We remove test-time fine-tuning. In other words, we generate outputs by directly applying the generator obtained after the training. As shown in Fig. 9, without the FT, while the global styles of the source images can be transferred, the results cannot preserve some local and fine-level appearance of the input characters, e.g., at the face regions.
- **w/o FRM:** We evaluate the effect of the FRM by removing it from the generator. As shown in Fig. 9, when there exists large positional discrepancy between the character in the source image and the stick figure in the target pose image, the method without the FRM can fail to properly transfer the texture of the source character onto the target pose. The incorporation of the FRM can solve this issue well.

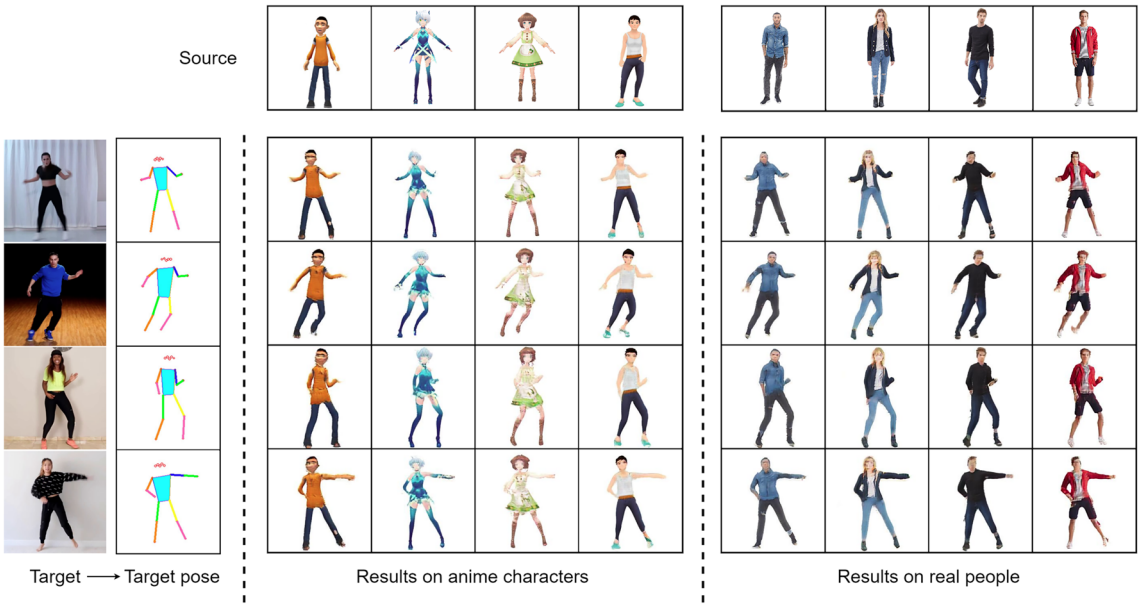



**Fig. 10** Our pose transfer results on anime characters and real people under 10-shot

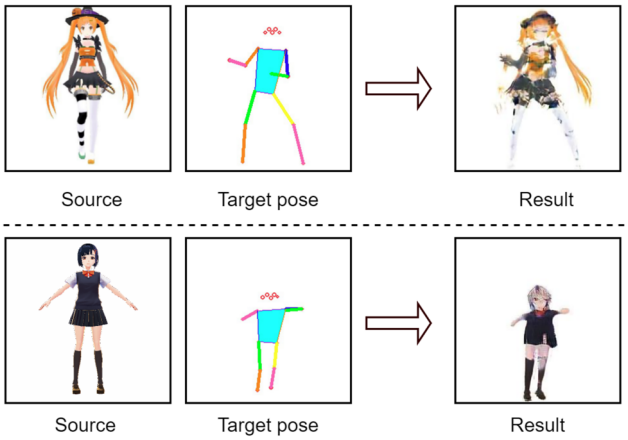



**Fig. 11** Failure cases of our method. Our method fails to give reasonable results when the input source character has very complex texture (top) or the target pose contains significant self-occlusion and pose ambiguity (bottom)

- **w/o MSL:** We evaluate the impact of the multi-step loss (MSL) by removing it in the training phase. As shown in Fig. 9, without the MSL, the color of the target image deviates more compared to the source image, or incorrect textures are generated. This suggests that MSL is important for the generation of details.

We further demonstrate the effectiveness of each component quantitatively in Table 4. Our full method achieves the best performances, indicating the necessity of all the components.

### 5.9 Failure cases

Figure 11 gives some failure cases of our method. Our method may fail to give satisfying results when the input source character have very complex texture, as shown in the top row of Fig. 11. This is perhaps because it is quite challenging to learn how to transfer complex texture patterns to arbitrary pose given only a few examples of a character. This issue can be partially alleviated by using more samples in the test-time support set. How to synthesize complex texture under low-shot setting is a meaningful next step for our problem. Failures can also occur in the presence of some extreme pose (e.g., crouching), which would cause self-occlusion or pose ambiguity in the 2D stick figure. One such example is given in the bottom row of Fig. 11.

## 6 Conclusion

In this paper, we propose a novel approach for anime character pose transfer under few-shot setting. With our proposed meta-learning framework, our method can generate visually compelling pose transfer results on arbitrary unseen anime characters given only a few samples of them. The proposed fusion residual block can learn to align the features of a source character and a target pose, thereby enabling our method to reliably synthesize the character's appearance onto the pose, even when the character and pose are misaligned spatially. Our experiments demonstrate that our method significantly outperforms baselines and previous pose transfer methods in terms of both visual fidelity and quantitative metrics.

**Acknowledgements** Results incorporated in this paper have received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 900025. Thanks to Miaomiao Chen for her carefully editing of the submission. Chenzhi Yuan conducted most of the experiments for the submission, while Kang Yang handled other experiments.



## References

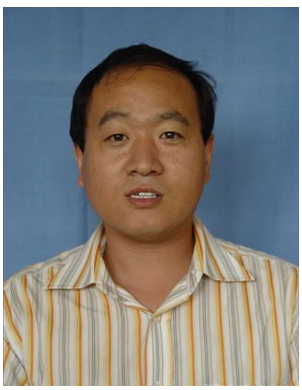
1. Aberman, K., Wu, R., Lischinski, D., Chen, B., Cohen-Or, D.: Learning character-agnostic motion for motion retargeting in 2D. arXiv preprint arXiv:1905.01680 (2019)
2. AlBahar, B., Huang, J.B.: Guided image-to-image translation with bi-directional feature transformation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 9016–9025 (2019)
3. Antoniou, A., Edwards, H., Storkey, A.: How to train your maml. arXiv preprint arXiv:1810.09502 (2018)
4. Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V., Patras, I.: AC-SUM-GAN: connecting actor-critic and generative adversarial networks for unsupervised video summarization. IEEE Trans. Circuits Syst. Video Technol. **31**(8), 3278–3292 (2020)
5. Balakrishnan, G., Zhao, A., Dalca, A.V., Durand, F., Guttag, J.: Synthesizing images of humans in unseen poses. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8340–8348 (2018)
6. Branwen, G., Gokaslan, A.: Danbooru2019: A large-scale crowdsourced and tagged anime illustration dataset (2019)
7. Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody dance now. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 5933–5942 (2019)
8. Chen, B., Zhang, Y., Tan, H., Yin, B., Liu, X.: PMAN: progressive multi-attention network for human pose transfer. IEEE Trans. Circuits Syst. Video Technol. **32**(1), 302–314 (2021)
9. Esser, P., Sutter, E., Ommer, B.: A variational u-net for conditional appearance and shape generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8857–8866 (2018)

10. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: Rmpe: Regional multi-person pose estimation. In: Proceedings of the IEEE international conference on computer vision, pp. 2334–2343 (2017)
11. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning, pp. 1126–1135. PMLR (2017)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
13. Hamada, K., Tachibana, K., Li, T., Honda, H., Uchida, Y.: Full-body high-resolution anime generation with progressive structure-conditional generative adversarial networks. In: Proceedings of the European conference on computer vision (ECCV) workshops, pp. 0–0 (2018)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
15. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Adv. Neural Inf. Process. Syst. 30 (2017)
16. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp. 448–456. PMLR (2015)
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125–1134 (2017)
18. Jaw, D.W., Huang, S.C., Kuo, S.Y.: Desnowgan: an efficient single image snow removal framework using cross-resolution lateral connection and GANs. IEEE Trans. Circuits Syst. Video Technol. **31**(4), 1342–1350 (2020)
19. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14, pp. 694–711. Springer (2016)
20. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
21. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4401–4410 (2019)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
24. Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H.S., Lu, C.: Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10863–10872 (2019)
25. Liu, M.Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., Kautz, J.: Few-shot unsupervised image-to-image translation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 10551–10560 (2019)
26. Liu, W., Piao, Z., Min, J., Luo, W., Ma, L., Gao, S.: Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 5904–5913 (2019)
27. Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., Van Gool, L.: Pose guided person image generation. Adv. Neural Inf. Process. Syst. 30 (2017)
28. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
29. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807–814 (2010)
30. Ren, Y., Li, G., Liu, S., Li, T.H.: Deep spatial transformation for pose-guided person image generation and animation. IEEE Trans. Image Process. **29**, 8622–8635 (2020)
31. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, pp. 234–241. Springer (2015)
32. Tang, H., Bai, S., Zhang, L., Torr, P.H., Sebe, N.: Xinggan for person image generation. In: Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16, pp. 717–734. Springer (2020)
33. Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International conference on machine learning, pp. 1747–1756. PMLR (2016)
34. Wang, M., Yang, G.Y., Li, R., Liang, R.Z., Zhang, S.H., Hall, P.M., Hu, S.M.: Example-guided style-consistent image synthesis from semantic labeling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1495–1504 (2019)
35. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8798–8807 (2018)
36. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
37. Xiu, Y., Li, J., Wang, H., Fang, Y., Lu, C.: Pose flow: Efficient online pose tracking. arXiv preprint arXiv:1802.00977 (2018)
38. Xu, S., Liu, D., Xiong, Z.: E2i: Generative inpainting from edge to image. IEEE Trans. Circuits Syst. Video Technol. **31**(4), 1308–1322 (2020)
39. Yuan, M., Peng, Y.: Bridge-GAN: Interpretable representation learning for text-to-image synthesis. IEEE Trans. Circuits Syst. Video Technol. **30**(11), 4258–4268 (2019)
40. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 9459–9468 (2019)
41. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International conference on machine learning, pp. 7354–7363. PMLR (2019)
42. Zhang, H., Sindagi, V., Patel, V.M.: Image de-raining using a conditional generative adversarial network. IEEE Trans. Circuits Syst. Video Technol. **30**(11), 3943–3956 (2019)
43. Zhang, J., Li, K., Lai, Y.K., Yang, J.: Pise: Person image synthesis and editing with decoupled gan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7982–7990 (2021)
44. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp. 2223–2232 (2017)
45. Zhu, Z., Huang, T., Shi, B., Yu, M., Wang, B., Bai, X.: Progressive pose attention transfer for person image generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2347–2356 (2019)

**Pengjie Wang** is currently a Professor with School of Computer Science, Dalian Minzu University, Dalian, China. His research interests include computer vision and computer graphics.

**Kang Yang** is currently a Master's student with School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include geometric processing for graphics and deep learning image generation.

**Chengzhi Yuan** is currently a Master's student with School of Information and Communication Engineering, Dalian Minzu University, Dalian, China. His research interests include deep learning, computer vision and computer graphics.



**Houjie Li** is currently a Professor with School of Information and Communication Engineering, Dalian Minzu University, Dalian, China. His research interests include deep learning, computer vision, and image processing.

**Wen Tang** is currently a Professor with the National Centre for Computer Animation, Bournemouth University, Bournemouth, UK. Her research interests include interactive virtual reality software technologies, physically based simulation, computer animation algorithms and computer games technology.

**Xiaosong Yang** is currently a Professor, Deputy Head of Department, Programme Leader of MSc AIM at the National Centre for Computer Animation, Bournemouth University, Bournemouth, UK. He has over 30 years' experience of research, education and professional practice in computer animation, machine learning, data mining, digital health, virtual reality and surgery simulation.