Collaborative Neural Rendering using Anime Character Sheets

Transpchan, et al.

http://www.github.com/traspchan ; @transpchan

Fig. 1. CoNR takes multiple arbitrarily posed reference images designed by artists as the character sheet input, and an Ultra-Dense Pose (UDP) representation as the pose input. The UDP can also be detected from another image providing the desired pose. Then, the model renders an image of the given character with the desired pose

Abstract. Drawing images of characters at desired poses is an essential but laborious task in anime production. In this paper, we present the Collaborative Neural Rendering (CoNR) method to create new images from a few arbitrarily posed reference images available in character sheets. In general, the high diversity of body shapes of anime characters defies the employment of universal body models for real-world humans, like SMPL. To overcome this difficulty, CoNR uses a compact and easy-to-obtain landmark encoding to avoid creating a unified UV mapping in the pipeline. In addition, CoNR's performance can be significantly increased when having multiple reference images by using feature space cross-view dense correspondence and warping in a specially designed neural network construct. Moreover, we collect a character sheet dataset containing over 700,000 hand-drawn and synthesized images of diverse poses to facilitate research in this area. The code and dataset will be released.

1 Introduction

Animation is one of the essential carriers of art and entertainment and shows human creativity. Human artists commonly use character sheets to show their virtual character design. A character sheet is the image collection of a specific character with multiple postures observed from different viewpoints. Thus it covers all the appearance details and is widely used to assist image creation of

animations or their derived media. Moreover, the sheet allows other artists to collaborate while maintaining the consistency of the design of this character.

Creating images of the characters with new poses is still an uphill task during most animation, comic, and game production. This is especially true for anime, a prominent art form that traditionally requires human imagination and expertise to draw every character image manually. Drawing a sequence of anime frames with desired poses is extremely time-consuming, and therefore does not scale easily for interactive applications like games or live streaming with virtual avatars. Due to the semantic gap between the character sheets and the desired pose, it is very challenging for computers to draw character images using character sheets like human artists automatically. Non-photorealistic rendering [21], one of the computer graphics techniques, enables fast and on-demand anime frame generation. However, it requires significantly more expertise and effort to design special textured 3D models with complicated shading programs to resemble the drawing style of different characters. Thus it is less accessible to the anime industry and the vast majority of the fan-art communities who are more familiar with traditional hand-drawing processes.

We formulate the task of rendering a particular character in the desired pose from character sheets. Instead of modeling character sheets as sequences, which suffer from the ordering issues, we consider them as dynamical-sized sets, better matching the established convention in the anime industry.

Based on this formulation, we develop a **Collaborative Neural Rendering (CoNR)** model. CoNR fully exploits the information available in a provided set of reference images by using feature space cross-view dense correspondence and warping. In addition, CoNR uses the Ultra-Dense Pose (UDP), an easy-to-construct compact landmark designed specifically for anime characters to avoid requiring a unified UV texture mapping [54,55,18] in the pipeline. It can represent the fine details of characters, such as rabbit ears, hairstyles or clothing, thus allowing better artistic control and adjustment over the desired pose for anime production purposes. It can also be easily generated with existing computer graphics pipelines, allowing a wide range of interactive applications like anime-based games or virtual assistants. Moreover, we collect a character sheet dataset containing over $700,000$ hand-drawn and synthesized images of diverse poses and appearance (which will be released). Training on this dataset, CoNR achieves impressive results both on hand-drawn images and synthesized images. We provide a demo video in the **supplementary material**.

To sum up, our main contributions are:

- We introduce a UDP representation designed specifically for anime characters and build a large character sheet dataset containing diverse poses.
- We explore the collaborative inference method of feed-forward neural networks to model character sheets as a dynamically-sized set of images.
- We present a baseline method, CoNR, for rendering anime character images with desired pose using character sheets. We show that CoNR has considerable potential to assist in anime video creation.

2 Related Works

2.1 Image Generation and Translation for Anime

Recent years have seen significant advancement in applying deep learning techniques to assist the creation of anime. For example, [19,59] propose to apply realistic lighting effects and shadow for 2D anime automatically; [52] propose to transfer photos to anime style; [45] propose to interpolate in-between frames for anime. There are also attempts to produce vectorized anime images [46], similar to the step-by-step manual drawing process. The generative modeling of anime faces has achieved very impressive results [26,20,49,28,24]. The latent semantics of generative models has also been extensively explored [42]. A modified StyleGAN2 model [4] is proven to be able to generate full-body anime images, although it still suffers from artifacts, including weirdly connected body parts, because of the high degree of freedom of the body.

2.2 Human Pose and Appearance Transfer

There has been a great success in the human pose or appearance transfer tasks [12,5]. Most of these works create vivid body motions or talking heads from only one single image [54,17,41,50,55,43,29].

The learned prior of the human body [31], head [8], or real-world clothing shape and textures [2] enable the model to solve ill-posed problems like imagining and inpainting the back view even if only the frontal view of the human is available. As a result of human imagination, anime has long been featuring a flexible character design leading to high diversity in clothing, body shapes, hairstyles, and other appearance features. A model trained on a huge dataset (*e.g.*, CLIP [39]) might be able to encode some popular anime character designs implicitly, but it is generally more challenging to establish priors or styles in the domain of anime characters than in the domain of human.

There are also some attempts [30] to extend the pose transfer task by utilizing SMPL [31], a parametric 3D human model, to combine appearance from different views. Using multiple reference images would, in principle, allow the model to follow the original appearance of the given person instead of finding a plausible posterior estimate, and better suit the needs of anime production.

Some recent works utilize NeRF [33], a category of neural rendering models which are trained using photometric reconstruction losses and optional camera poses over multiple images of a 3D object. Due to their ray-marching nature and capability to in-paint in 3D, they are promising methods in modeling real-world 3D data [37,36], which are not depending on or being influenced by any prior knowledge other than the object to be modeled. However, they have not yet made much progress in modeling hand-drawn data like anime character sheets, which are less following strict geometric and physical constraints.

2.3 Representation of Human Pose

Stick-figure of skeletons [12], SMPL vectors [31], and heat maps of joints [11,44] are well-defined and widely-used representations that can be obtained from objective data sources, including motion capturing equipment. However, noisy manual annotations, occlusion caused by diverse styles of garment or other body decorations, and ambiguity caused by diverse body shapes impose significant challenges when migrating these sparse representations from human to anime [1].

Anime characters usually require flexible artistic control over fine details or poses, exaggerating additional moving parts like floating hair or flowing skirt, which are not directly driven by human joints. The aforementioned human pose representations are very abstract and fuzzy in the inverse task of pose-guided anime rendering.

Human parsers or clothing segmenters [56,55,17,14] are robust to the uncertainty of joint positions. However, the provided semantic masks are not informative enough to represent the pose or even the orientation of a person. DensePose [22] and UV texture mapping [54,55,18], greatly enhance the detail of pose representation on the human body or face by imposing a universal definition that essentially unwarps the 3D human body surface into a 2D coordinate system. However, three problems may emerge when anime characters start to annotate themselves accordingly. Anime girls have trouble finding the precise location visually where they should cut their skirts and flatten this cone-like object in the same way as others. Anime boys get stuck as they are unsure how to consistently handle jeans, kilts, and other non-homeomorphic body shapes. Meanwhile, they have no idea of the number of key points they should use for rabbit or cat ears. Due to the diverse body shapes of anime characters, every body region can require more key points than others. Therefore, existing dense representations that are not designed for anime-related tasks may still not serve as an off-the-shelf solution.

3 Method

3.1 Task Formulation

We explore this task by first observing real artists drawing anime images. While drawing different body parts on the canvas, artists will usually refer to multiple images in the character sheets. Because appearance details required at the desired pose are usually distributed across different reference images.

Given a sequence of reference images $\mathbf{I}_1 ... \mathbf{I}_n$ from the character sheet, human artists drawing anime images can be seen a sequence of operations on the canvas after seeing each $\mathbf{I}_t$, similar to the widely adopted formulation of tasks about drawing or painting art [58,25,46]. As $\mathbf{I}_t$ differ from each other only by the pose of the character, the order of sequence $\mathbf{I}_t$ can also be seen as the order of poses. However, existing pose representations prevent an easy definition of their order or a satisfying way to discretize them into finite canonical categories. Even if an algorithm or a human successfully finds a place in the sequence for a 45-degree

left side view, it may still struggle at a standing character with his head looking back, which is a frequently seen pose in the domain of anime. Therefore, the sequence formulation is not favorable for character sheets. For freeing the users from putting $\mathbf{I}_t$ into a sequence during inference, arbitrary ordering should be allowed in the character sheet.

Here we present a formal task formulation by considering one character sheet $\mathbf{S}_{ref}$ in a whole as an input sample and ignoring the order of reference images $\mathbf{I}_n \in \mathbf{S}_{ref}$. To provide rendering directives to the model, a target pose $\mathbf{P}_{tar}$ representation is also required in the input. The task can be formulated as mapping input sample $\mathbf{S}_{ref}$ to target image y that follows the desired target pose $\mathbf{P}_{tar}$:

$$y = f(\mathbf{P}_{tar}, \mathbf{S}_{ref}). \tag{1}$$

We also notice that complicated poses, motions, or characters may require a larger collection of references in $\mathbf{S}_{ref}$ than others. A dynamically sized $\mathbf{S}_{ref}$ should therefore also be allowed.

3.2 Modeling of Character Sheets

To address the task in 3.1, we utilize a Collaborative Inference for convolutional Neural Networks (CINN) inspired by PointNet [38] and Equivariant-SFM [34].

Usually, multiple images can be fed in arbitrary order into multiple copies of the same classical convolutional neural network (CNN) to obtain corresponding inference results. In CINN, however, multiple images in a set are defined in a whole as one single input sample. Adding feature-averaging on outputs of all corresponding blocks in multiple copies of a CNN, we obtain a network of a dynamical number of sub-networks. The sub-networks share the same weight and are inter-connected by message passing mechanisms. Due to the commutative nature of addition, changing the order of the sub-networks will not affect the inference results.

When performing a collaborative inference on such a network, n reference images (or views) in a set are fed into n weight-shared sub-networks, respectively. The sub-networks form a fully connected graph, as illustrated in Figure 4, so that each block of a sub-network would share part of its outputs as messages to corresponding successive blocks in all other sub-networks, in addition to forwarding other outputs to its following blocks like in a classical neural network. To further modulate the cross-view message sent after each block, we apply weighted averaging on messages, where the weights are predicted by CNN and normalized by the number of views (inspired by [60]).

3.3 Ultra-Dense Pose

We present UDP, a compact landmark representation designed specifically for anime characters. A UDP specifies a character's pose by mapping 2D viewport coordinates to feature vectors, which are 3-tuple floats that continuously and consistently encode body surfaces. In this way, a UDP can be represented as a

Fig. 2. UDP representation of a random anime character (Kurei Kei). UDP uses 3D coordinates at same **A-pose** as landmarks. When the anime body changes its pose, the landmark at the corresponding body part will remain the same. Compared to existing dense human-pose representations, UDP by-passes the step of unwarping 3D surfaces onto a 2D UV texture mapping, which may not be done in a consistent manner for anime characters

color image $\mathbf{P}_{tar} \in \mathbb{R}^{H \times W \times 3}$ with pixels corresponding to landmarks $L_{(x,y)} \in \mathbb{R}^3$. Non-person areas of the UDP image are simply masked. It allows better compatibility across a broader range of anime body shapes and enables better artistic control over body details like garment motions.

3D meshes are widely used data representations for anime characters in their game adaptations. Vertex in a mesh usually comprises corresponding texture coordinate (u, v) or a vertex color (r, g, b). Interpolation over the barycentric coordinates allows triangles to form faces filled by color values or pixels looked up from textures coordinates.

Taking a bunch of anime body meshes standing at the center of the world, we ask them to perform the same **A-pose** (a standard pose) to align the joints. To construct UDP, we remove the original texture and overwrite the color (r, g, b) of each vertex with a landmark, which is currently the world coordinate (x, y, z). When the anime body changes its pose, the vertex on the mesh may move to a new position in the world coordinate system, but the landmark at the corresponding body part will remain the same, shown as the same color in Figure 2.

To avoid the difficulty of down-sampling and processing meshes, we convert the modified meshes into 2D images, which are friendly to CNNs. This is done by introducing a camera, performing culling on occluded faces and projecting only the faces visible from the camera into an image. The resulting UDP representation is a $H \times W \times 4$ shaped image recorded in floating-point numbers ranging from 0 to 1. The four channels include three body landmark encodings and one occupancy for indicating whether the pixel is on the body.

Three properties of UDP could alleviate the difficulties mentioned in 2.3:

1) UDP is a detailed 3D pose representation since every tiny piece of surface on the anime body, no matter if it is from the hair or the garment, could be automatically assigned with a unique encoding without hand-crafted annotations.

2) UDP is a widely compatible pose representation with acceptable exchangeability since the anime characters with similar body shapes will also get out-fits that are consistently pseudo-colorized.

3) UDP serves a role of describing the local 3D shape of the human body, which could provide additional geometric information to downstream tasks.

Fig. 3. Random characters with random backgrounds

3.4 Data Preparation

As character sheets used in the anime-related industries are not yet available to the computer vision community, we build a dataset containing more than 20,000 hand-drawn anime characters by selecting human-like characters from public datasets [3,27]. We manually perform matting to remove the background from the character with the help of the watershed algorithm. We also construct a 2D background dataset containing 4000 images with a similar method.

Manually annotating hand-drawn anime images with UDP involves intolerable levels of hardship. To alleviate the problem of label scarcity, we constructed a synthesized dataset from anime-styled 3D meshes in the way described in 3.3.

Finally, we randomly split a synthesized dataset with high-quality UDP labeling and a hand-drawn dataset, which has higher diversity in styles and characters, by a 16 : 1 ratio into the training and validation sets. The split is done on a per-anime-character basis so that the validation set contains characters unseen during training. The whole dataset contains over 700,000 hand-drawn and synthesized images of diverse poses and appearances. We manually exclude content that is not suitable for public display. Random characters with the random background are shown in Figure 3.

3.5 Collaborative Neural Rendering

Overview. A CoNR pipeline consists of a renderer and an optional UDP detector. Figure 4 shows the pipeline of the proposed approach. The renderer generates character images of the desired pose, taking the target pose's UDP representation $\hat{\mathbf{P}}_{tar}$ and a character sheet $\mathbf{S}_{ref}$, as the inputs.

The input UDP representation can be produced by a UDP detector from reference images or videos. For interactive applications like games, the existing physics engine can be used as a drop-in replacement for the UDP detector to directly compute body and cloth dynamics for the anime character.

Renderer. Based on the U-Net [40], we apply the following modifications:
1) To allow efficient inference on videos, we remove the UDP input from the encoder side but instead concatenate the UDP input rescaled with the nearest-sampling method into each skip channel from the encoder to the decoder, as

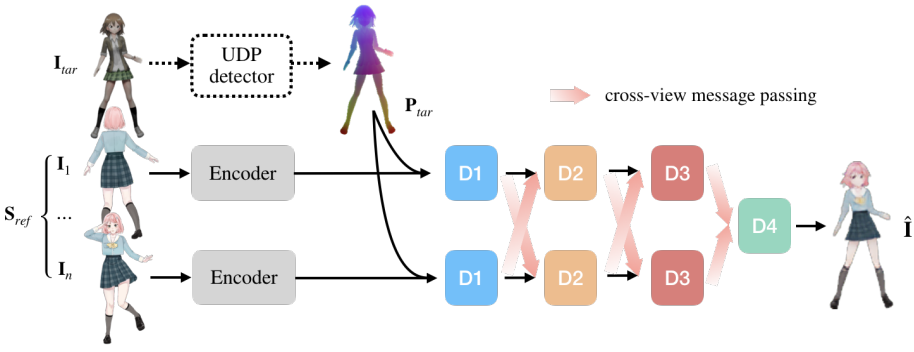

Fig. 4. Overview of CoNR. Reference images $\mathbf{I}_1 \cdots \mathbf{I}_n \in \mathbf{S}_{ref}$ from the input character sheet are fed into a CINN using modified U-Nets [40] as sub-networks. The same UDP representation $\mathbf{P}_{tar}$ detected from target pose $\mathbf{I}_{tar}$ is resized and concatenated into each scale of the encoder outputs in all sub-networks. Blocks with the same color share weights. "D1 to D4" refers to 4 blocks of the decoder. The sub-networks form a fully connected graph using cross-view message passing. Each block will receive the averaged message from corresponding blocks in all other sub-networks

shown in Figure 4. This allows us to checkpoint the evaluated results from the encoder that can be reused when inference on multiple target UDPs in a video.

2) Inspired by [15,16], we use two extra channels in each decoder block to generate a flow-field and perform a grid sampling over other output features of the block, for enhancing the long-range look-up ability for CNNs.

3) We apply the CINN method to the decoders of the network. We split the original up-sampling output feature channels by half, one as the remote branch and the other for the local branch. The warping is first performed only on the remote branches. Then the remote branch of output features from all sub-networks are averaged and passed to the next block, as described in 3.2. The local branch remains unchanged. The local branch output and the remote branch output from the previous block are concatenated with the encoder output and fed into the next block (illustrated in **Appendix**). The last decoder block will collect averaged output features from all previous decoder blocks in all sub-networks and decode them as the final RGB output $\hat{\mathbf{I}}_{tar}$.

UDP Detector. To prepare the UDP representation $\widehat{\mathbf{P}}_{tar}$ of target pose, we use a simple U-Net [40] consists of a ResNet-50 [23] encoder and a decoder with 5 Residual Blocks to detect it from an RGB image $\mathbf{I}_{tar}$, The UDP detector is evaluated on the target $\mathbf{I}_{tar}$ to obtian the four-channel UDP representation, as formulated in 3.3.

The detector can be trained independently on the synthesized dataset or trained jointly with the renderer in an end-to-end manner. In the second case, the detector could provide the renderer with augmentations on UDPs, which gradually decrease when the detector converges. To further reduce the memory

footprint of the model, we can share the weight of the ResNet-50 backbone in the encoder of the UDP detector with the encoder of the renderer.

4 Experiments

4.1 Training Strategy

We train CoNR models of m sub-networks (views) on our dataset. To create one training sample, we randomly select a character from the dataset and then randomly select $m+1$ arbitrary poses of that character. These images are split as m character sheet inputs $\mathbf{I}_1 \cdots \mathbf{I}_m \in \mathbf{S}_{ref}$ and one image of the target pose as the ground truth of CoNR's final output.

We apply random image augmentations to the target pose image, paste them onto k random backgrounds and run them through the UDP detector. We use the average of the k UDP detection results of the same target pose, $\widehat{\mathbf{P}}_{tar} = 1/k \sum_{j=1}^{k} \widehat{\mathbf{P}}_j$, as the final UDP detection results. We also obtain the corresponding UDP of the target pose from the dataset as the ground truth to train the UDP detector. We compute losses at both the output of the detector and the end of the CoNR pipeline. We use L1 loss on the 3 landmark encodings and BCE loss on the mask to train the detector if the ground truth is available:

$$\mathcal{L}_{udp} = ||\widehat{\mathbf{P}}_{tar} - \mathbf{P}^{GT}_{tar}||_1, \tag{2}$$

$$\mathcal{L}_{mask} = BCE(\widehat{\mathbf{P}}_{tar\_mask}, \mathbf{P}^{GT}_{tar\_mask}). \tag{3}$$

We use a consistency loss $\mathcal{L}_{cons}$ by computing the standard deviation of k UDP detector outputs:

$$\mathcal{L}_{cons} = \sqrt{\frac{1}{k-1}\sum_{j=1}^{k}(\widehat{\mathbf{P}}_j - \widehat{\mathbf{P}}_{tar})^2}. \tag{4}$$

At the end of the collaborated renderer Φ, we use L1 loss to ensure a correct photometric reconstruction of the character in the desired target pose,

$$\mathcal{L}_{photo} = ||\Phi(\widehat{\mathbf{P}}_{tar}, \mathbf{S}_{ref}) - \mathbf{I}^{GT}_{tar}||_1. \tag{5}$$

Optionally, we compute the perceptual loss $\mathcal{L}_{perc}$ using LPIPS [57]:

$$\mathcal{L}_{perc} = LPIPS(\Phi(\widehat{\mathbf{P}}_{tar}, \mathbf{S}_{ref}), \mathbf{I}^{GT}_{tar}). \tag{6}$$

The UDP detector and renderer are trained end-to-end simultaneously. The total loss function is a weighted sum of all these losses:

$$\mathcal{L} = \mathcal{L}_{udp} + \alpha\mathcal{L}_{mask} + \beta\mathcal{L}_{perc} + \gamma\mathcal{L}_{photo} + \theta\mathcal{L}_{cons}, \tag{7}$$

where $\alpha = 1.0, \beta = 0.05, \gamma = 1.0, \theta = 1.0$.

Our model is optimized by AdamW [32] with weight decay 10^{-4} for $224k$ iterations. We choose $m = 4, k = 4$ during training and $m = 4, k = 1$ during inference in all experiments unless otherwise specified. The training process uses a batch size of 24 with all input resolution set to 128×128. Model training takes about a week on four GPUs. For quantitative evaluation, we measure the losses on the validation split.

Table 1. Comparison on the number of input reference images. We use character sheets of m reference images to train the CoNR model, and then use character sheets of n reference images to evaluate the trained model

Setting	112k iter		224k iter	
	$\mathcal{L}_{photo}$	$\mathcal{L}_{perc}$	$\mathcal{L}_{photo}$	$\mathcal{L}_{perc}$
$m = 1, n = 1$	0.0247	0.0832	0.0238	0.0801
$m = 4, n = 1$	0.0249	0.0865	0.0237	0.0827
$m = 1, n = 4$	0.0219	0.0798	0.0211	0.0764
$m = 4, n = 4$	**0.0187**	**0.0659**	**0.0179**	**0.0612**

Fig. 5. left: Inference results on validation dataset. **right**: Inference results with the same character sheet input $\mathbf{S}_{ref}$ on different body structure $\mathbf{P}_{tar}$ (difference in skirts)

4.2 Result

Inference Visual Effects. CoNR can cope with the diverse styles of anime, including differences between synthesized images and hand-drawn images. Figure 5 list some random CoNR outputs on the evaluation split. With the same character sheet input $\mathbf{S}_{ref}$, we replaced the provided UDP $\mathbf{P}_{tar}$, the results of CoNR changed accordingly. Notably, CoNR goes beyond a naïve correspondence by absolute UDP value (transfer between different cloths).

Effectiveness of the Collaboration. Unlike most previous methods for similar tasks that allow only one reference image as input, CoNR uses a dynamically-sized set of reference inputs during training and inference.

The experiment results in Table 1 show that using an additional number of views $m > 1$ during training will enhance the accuracy of generated images. On the opposite, removing images from the character sheet will reduce coverage of the body surface. When CoNR is trained with $m = 1$, the chances are high that the provided character sheets do not allow a valid solution, such as providing the character's backside and forcing the model to imagine the frontal side. In this case, the photo-metric reconstruction and perceptual losses may encourage the network to learn the wrong solution. Therefore even if enough information in the $n > 1$ images is provided during inference, the network may not generate the target image accurately. Similarly, keeping $m = 4$ while reducing the number of inference view ($n = 1$) will also harm the quality of the resulting image.

CoNR does not require the number of inference view n to match the m during training. Adding additional views during inference will enhance the quality of the generated images as shown in Figure 6. As illustrated in 3.2, CoNR models the input character sheet as a dynamically-sized set so that it can leverage

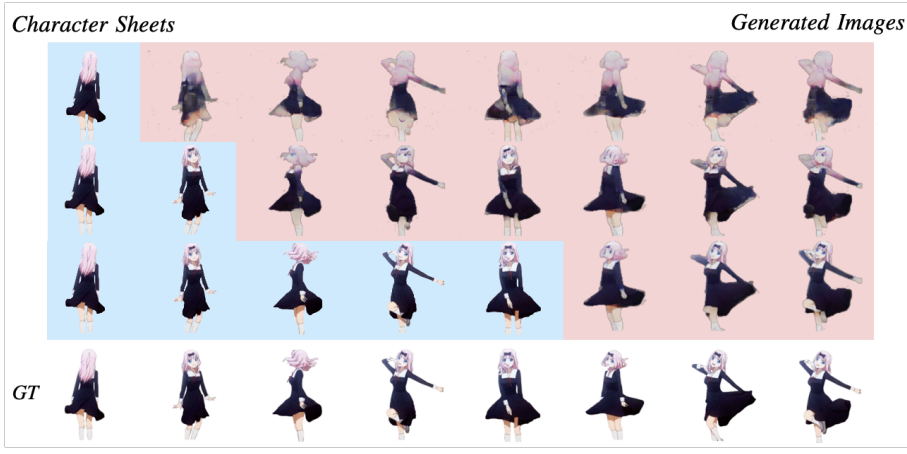

Fig. 6. Effectiveness of the collaboration. We perform a reconstruction test with random video in the wild, with unseen pose and appearance. The last row shows 8 ground truth frames $\mathbf{I}_j^{GT} \in \mathbf{S}_{vid}^{GT}$ from a video. First three rows shows the inputs and outputs of CoNR. The used subsets of character sheet $\mathbf{S}_{ref} \subset \mathbf{S}_{vid}^{GT}$ are marked using blue background. Generated images for novel poses are marked using red background

information distributed across all images without favoring the first m inputs. This allows CoNR to scale seamlessly from image synthesis, when only a few shots of the character are given, to image interpolating when a lot of footage of a character are available. The example in Figure 6 shows the behavior of CoNR when it does not have enough information to draw missing parts correctly during the inference. Even if very few reference image is provided, the target pose that finds a similar reference in the character sheets will be accurate, as expected. User can also iterate on the results and feed them back into CoNR to accelerate anime production.

4.3 Comparison with Related Works

As CoNR provides the baseline for a novel task, we admit that direct comparison with related works can either be impossible or unfair. We still try to include comparisons, only for showing how our task is related to other existing tasks.

Human Pose Synthesis. We compare the results produced by CoNR to a real-world digital human system [30] using the same target poses [1] as used in their demo. We use two characters sheets [2] with both pose and appearance unseen during training. One contains the same 4 images as in Figure 5, the other character sheet taken from a random YouTube video contains $\mathbf{I}_{1\sim5}$ as used in

[1] Video from download.impersonator.org/iper_plus_plus_latest_samples.zip
[2] A random anime character from a random video www.youtube.com/watch?v=m6k_t8yEyvE and another character illustrated by an amateur artist for this paper.

Fig. 7. Comparison with the SMPL-based method. We compare results of our method (the 3nd row) with results of [30] (the 2rd row) when trying to resemble the target poses shown in the first row. **The real person is only used to show the pose**. Further diagnosis shows that parametric 3D human models like SMPL may not represent diverse clothing in anime. The short skirt of the second character seems to be tightened on the legs instead of sagging naturally

Table 2. Ablation on UDP Detector with different backbones

Setting	112k iter		224k iter	
	$\mathcal{L}_{udp}$	$\mathcal{L}_{mask}$	$\mathcal{L}_{udp}$	$\mathcal{L}_{mask}$
Original U-Net	0.1247	0.0856	Divergence	Divergence
U-Net+ResNet-34	0.1051	0.1068	0.1004	0.0747
U-Net+ResNet-50	**0.0969**	**0.0792**	**0.0971**	**0.0736**

Figure 6. The results in Figure 7 indicates that the long skirt prevents a high accuracy estimation of the leg joints and that parametric 3D human models like SMPL may not handle the body shape of anime characters correctly. CoNR pipeline can produce images at desired target poses with better quality.

Image synthesis pipelines starting with human pose representations is theoretically inapplicable on anime data, as the diverse body structure, clothing and accessories of the character cannot be reasonably represented. It should be noted that here we deliberately hand-pick anime characters without rabbit ears or other body shapes uncommon for human, in favor of existing methods instead of CoNR. It should also be noted that using human pose synthesis methods on anime, we may lose the artistic control over garments and fine details which is crucial in anime creation workflows.

Style Transfer. Style transfer typically refers to applying a learned style from a certain domain (anime images) to the input image taken form the other domain (real-world images) [13,53]. The models usually treat target domain in a whole as a kind of style and require extensive training to remember the style. There are also methods that uses a single image to provide a style hint during the inference. For example, one could use Swapping Auto Encoders [35], a recent style-transfer method to swap the textures or body structures between two

Fig. 8. Comparision between detection results of hand-drawn anime character images using OpenPose [11], the UDP Detector and SMPLify [9]. The images are from the validation split of the hand-drawn dataset [3] with the corresponding ID number. The detected SMPL body can not fully handle the diverse body shapes of anime characters. UDP detector seems to be theoretically possible

Table 3. Ablation on CoNR Renderer. The grid-sample operation (Sec 3.5) over the output features of each decoder block. All comparisons and are done with character sheets of $m = 4$ images

U-Net	grid-sample	CINN	ResNet-50	112k iter		224k iter	
				$\mathcal{L}_{photo}$	$\mathcal{L}_{perc}$	$\mathcal{L}_{photo}$	$\mathcal{L}_{perc}$
✓				0.0313	0.1100	0.0311	0.1038
✓	✓			0.0309	0.1090	0.0308	0.1036
✓		✓	✓	0.0315	0.1097	0.0286	0.0977
✓	✓	✓		0.0194	0.0673	0.0180	0.0612
✓	✓	✓	✓	**0.0187**	**0.0659**	**0.0179**	**0.0612**

characters. Although the model has a lot of parameters (3× the size of CoNR), our comparison shows that it is still not enough to remember and reproduce the diverse sub-styles of the textures, pose, and body structure in the domain of anime.

CoNR, as a neural rendering method, can also be used to achieve results similar to what style transfer methods could provide, when running a textured image at the reference-pose through the UDP detector, as shown in 4. The CoNR inference pipeline for the anime (or game) production will usually be without the UDP detector.

4.4 Ablation Study

We performed ablation studies on the UDP detector, renderer and loss functions. Table 2 shows that UDP representation can be inferred from images using a U-Net [40]. An original U-Net, which takes a concatenated tensor of 4 reference images and the target UDP as the input, does not provide acceptable results on this task, as shown in Table 3. The proposed CoNR method with both the

Table 4. Ablation on loss functions

Setting	112k iter				224k iter			
	$\mathcal{L}_{udp}$	$\mathcal{L}_{mask}$	$\mathcal{L}_{photo}$	$\mathcal{L}_{perc}$	$\mathcal{L}_{udp}$	$\mathcal{L}_{mask}$	$\mathcal{L}_{photo}$	$\mathcal{L}_{perc}$
w/o $\mathcal{L}_{mask}$	0.162	0.880	-	-	0.153	0.773	-	-
w/o $\mathcal{L}_{photo}$	-	-	0.023	0.084	-	-	0.022	0.076
w/o $\mathcal{L}_{perc}$	-	-	0.028	0.158	-	-	Divergence	Divergence
Baseline	**0.097**	**0.079**	**0.019**	**0.066**	**0.097**	**0.074**	**0.018**	**0.061**

feature warping and the CINN method significantly increases the performance, thus establishing a stronger baseline for the proposed task. We further ablate the loss functions in Table 4.

5 Limitations

CoNR is unable to model the environmental lighting effects. The inputs of CoNR, neither the character sheet nor the UDP representation, could provide any information about the environmental or contextual information that could be utilized to infer any lighting effects. Even if CoNR may be used as a drop-in deferred shader for a portion of interactive applications, users still have to look for sketch relighting techniques [59,19,47].

CoNR is unable to model the dynamics of the character. The CoNR model accepts target poses detected from a video of a character with a body shape similar to $\mathbf{S}_{ref}$, and could therefore inherit the dynamics. However, it requires such video to exist beforehand. To bypass the UDP detector, users have to rely on additional technologies like garment captures [10], physics simulations [6] learning-based methods [51,7,48] or existing 3D animation workflows to obtain a synthesized UDP $\mathbf{P}_{tar}$. CoNR focuses on the rendering task. Obtaining a suitable 3D mesh with all the body parts rigged and all clothing computed with proper dynamics, is beyond the scope of this paper. Using $\mathbf{P}_{tar}$ from a inappropriate body shape may lead to incorrect CoNR results as shown in Figure 5.

The dataset may not fully follow the distribution of anime characters in the wild. The collected dataset contains only human-like anime characters from the year 2014 to 2018. As the character meshes are aligned using joints, the models trained on this dataset may not be applied to animal-like characters. Research on broader datasets will be the future work.

6 Conclusion

This work introduces a new task to render anime character images with the desired pose from multiple images in character sheets. We developed a simple feed-forward baseline method, CoNR, for this task. The effectiveness of the method is encouraging. We hope the method and the datasets presented in this paper will inspire further research.

References

1. Pose estimation of anime/manga characters: A case for synthetic data. In: Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding (2016) 4
2. Alldieck, T., Pons-Moll, G., Theobalt, C., Magnor, M.: Tex2shape: Detailed full human body geometry from a single image. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019) 3
3. Anonymous, community, D., Branwen, G.: Danbooru2020: A large-scale crowdsourced and tagged anime illustration dataset, https://www.gwern.net/Danbooru2020 7, 13
4. aydao: This anime does not exist, https://thisanimedoesnotexist.ai/, https://thisanimedoesnotexist.ai/ 3
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 (CVPR) (2018) 3
6. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98. pp. 43–54. ACM Press, http://portal.acm.org/citation.cfm?doid=280814.280821 14
7. Bertiche, H., Madadi, M., Escalera, S.: Pbns: Physically based neural simulator for unsupervised garment pose space deformation. arXiv preprint arXiv:2012.11310 (2020) 14
8. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '99. pp. 187–194. ACM Press, http://portal.acm.org/citation.cfm?doid=311535.311556 3
9. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In: European conference on computer vision (ECCV) (2016) 13
10. Bradley, D., Popa, T., Sheffer, A., Heidrich, W., Boubekeur, T.: Markerless garment capture. In: ACM SIGGRAPH 2008 Papers on - SIGGRAPH '08. p. 1. ACM Press, http://portal.acm.org/citation.cfm?doid=1399504.1360698 14
11. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: realtime multi-person 2d pose estimation using part affinity fields. IEEE transactions on pattern analysis and machine intelligence (TPAMI) **43**(1), 172–186 (2019) 4, 13
12. Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody dance now. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019) 3, 4
13. Chen, J., Liu, G., Chen, X.: Animegan: A novel lightweight gan for photo animation. In: International Symposium on Intelligence Computation and Applications. pp. 242–256. Springer (2019) 12
14. Chou, C.L., Chen, C.Y., Hsieh, C.W., Shuai, H.H., Liu, J., Cheng, W.H.: Template-free try-on image synthesis via semantic-guided optimization. IEEE Transactions on Neural Networks and Learning Systems (TNNLS) (2021) 4
15. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017) 8
16. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015) 8

17. Gafni, O., Ashual, O., Wolf, L.: Single-shot freestyle dance reenactment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 3, 4
18. Gao, C., Shih, Y., Lai, W.S., Liang, C.K., Huang, J.B.: Portrait neural radiance fields from a single image. arXiv preprint arXiv:2012.05903 (2020) 2, 4
19. Gao, Z., Yonetsuji, T., Takamura, T., Matsuoka, T., Naradowsky, J.: Automatic Illumination Effects for 2D Characters. In: NIPS Workshop on Machine. Learning for Creativity and Design. (2018) 3, 14
20. Gokaslan, A., Ramanujan, V., Ritchie, D., Kim, K.I., Tompkin, J.: Improving shape deformation in unsupervised image-to-image translation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018) 3
21. Gooch, B., Gooch, A.: Non-photorealistic rendering. AK Peters/CRC Press (2001) 2
22. Güler, R.A., Neverova, N., Kokkinos, I.: Densepose: Dense human pose estimation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
23. 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 (CVPR) 8
24. He, Z., Kan, M., Shan, S.: Eigengan: Layer-wise eigen-learning for gans. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2021) 3
25. Huang, Z., Zhou, S., Heng, W.: Learning to Paint With Model-Based Deep Reinforcement Learning. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019) 4
26. Jin, Y., Zhang, J., Li, M., Tian, Y., Zhu, H., Fang, Z.: Towards the Automatic Anime Characters Creation with Generative Adversarial Networks. In: NIPS Workshop on Machine. Learning for Creativity and Design. (2017) 3
27. Li, J.: Pixiv dataset, https://github.com/jerryli27/pixiv_dataset 7
28. Li, M., Jin, Y., Zhu, H.: Surrogate gradient field for latent space manipulation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 3
29. 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 International Conference on Computer Vision (ICCV) (2019) 3
30. Liu, W., Piao, Z., Tu, Z., Luo, W., Ma, L., Gao, S.: Liquid warping gan with attention: A unified framework for human image synthesis. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2021) 3, 11, 12
31. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. ACM transactions on graphics (TOG) **34**(6), 1–16 (2015) 3, 4
32. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2019) 9
33. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020) 3
34. Moran, D., Koslowsky, H., Kasten, Y., Maron, H., Galun, M., Basri, R.: Deep permutation equivariant structure from motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 5

35. Park, T., Zhu, J.Y., Wang, O., Lu, J., Shechtman, E., Efros, A., Zhang, R.: Swapping autoencoder for deep image manipulation. Advances in Neural Information Processing Systems **33**, 7198–7211 (2020) 12
36. Peng, S., Dong, J., Wang, Q., Zhang, S., Shuai, Q., Zhou, X., Bao, H.: Animatable neural radiance fields for modeling dynamic human bodies. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2021) 3
37. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 3
38. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (2017) 5
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021) 3
40. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention (MICCAI) (2015) 7, 8, 13
41. Sarkar, K., Mehta, D., Xu, W., Golyanik, V., Theobalt, C.: Neural Re-Rendering of Humans from a Single Image. Lecture Notes in Computer Science, http://arxiv.org/abs/2101.04104 3
42. Shen, Y., Zhou, B.: Closed-form factorization of latent semantics in gans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 3
43. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: First order motion model for image animation. In: Advances in Neural Information Processing Systems (NIPS) (2019) 3
44. Siarohin, A., Woodford, O.J., Ren, J., Chai, M., Tulyakov, S.: Motion Representations for Articulated Animation, http://arxiv.org/abs/2104.11280 4
45. Siyao, L., Zhao, S., Yu, W., Sun, W., Metaxas, D.N., Loy, C.C., Liu, Z.: Deep Animation Video Interpolation in the Wild, http://arxiv.org/abs/2104.02495 3
46. Su, H., Niu, J., Liu, X., Cui, J., Wan, J.: Vectorization of raster manga by deep reinforcement learning. arXiv preprint arXiv:2110.04830 (2021) 3, 4
47. Su, W., Du, D., Yang, X., Zhou, S., Fu, H.: Interactive Sketch-Based Normal Map Generation with Deep Neural Networks **1**(1), 1–17 14
48. Tiwari, G., Sarafianos, N., Tung, T., Pons-Moll, G.: Neural-gif: Neural generalized implicit functions for animating people in clothing. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2021) 14
49. Tseng, H.Y., Fisher, M., Lu, J., Li, Y., Kim, V., Yang, M.H.: Modeling artistic workflows for image generation and editing. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020) 3
50. Wang, T.C., Mallya, A., Liu, M.Y.: One-shot free-view neural talking-head synthesis for video conferencing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 3
51. Wang, T.Y., Shao, T., Fu, K., Mitra, N.J.: Learning an intrinsic garment space for interactive authoring of garment animation. ACM Transactions on Graphics (TOG) **38**(6), 1–12 (2019) 14

52. Wang, X., Yu, J.: Learning to cartoonize using white-box cartoon representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 3
53. Wang, X., Yu, J.: Learning to cartoonize using white-box cartoon representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8090–8099 (2020) 12
54. Yao, P., Fang, Z., Wu, F., Feng, Y., Li, J.: Densebody: Directly regressing dense 3d human pose and shape from a single color image. arXiv preprint arXiv:1903.10153 (2019) 2, 3, 4
55. Yoon, J.S., Liu, L., Golyanik, V., Sarkar, K., Park, H.S., Theobalt, C.: Pose-guided human animation from a single image in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 2, 3, 4
56. Zanfir, M., Popa, A.I., Zanfir, A., Sminchisescu, C.: Human appearance transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
57. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 9
58. Zhang, S.H., Chen, T., Zhang, Y.F., Hu, S.M., Martin, R.R.: Vectorizing cartoon animations. IEEE Transactions on Visualization and Computer Graphics **15**(4), 618–629 (2009) 4
59. Zheng, Q., Li, Z., Bargteil, A.: Learning to shadow hand-drawn sketches. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 3, 14
60. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016) 5