Collaborative Neural Rendering using Anime Character Sheets

Anonymous CVPR 2022 submission

Paper ID 6818

Figure 1: **Collaborative Neural Rendering** (CoNR) takes multiple arbitrarily ordered and posed reference images as the **Character Sheet** input, and an **Ultra-Dense Pose** (UDP) representation as the pose input. The UDP is generated from another image providing a desired pose. Then, CoNR outputs 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 3D body models for real-world humans, like SMPL. To overcome this difficulty, CoNR uses a novel and compact form of landmark encoding to avoid requiring 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 data will be released.

1. Introduction

Artists commonly use character sheets to show their design of a character. Character sheets are the image collections of a specific character with multiple postures observed from different viewpoints. Thus they cover all the appearance details and are widely used to assist image creation of animations or their derived media. Moreover, these sheets allow other artists to draw together while maintaining the consistency of the design of this character.

However, 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 generalize 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 automatically draw character images using character sheets like human artists. Non-photorealistic rendering, a recently emerged computer graphics technique, enables fast and on-demand anime frame generation. However, it requires significantly more expertise and effort to design special 3D models with textures and 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 an image of a particular character in the desired pose from character sheets.

Instead of modeling character sheets as sequences, which suffer from the ordering issues in various aspects, our formulation allows them to be dynamically-sized sets, better matching the established convention in the anime industries.

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, a novel and compact form of landmark encoding designed specifically for anime characters to avoid requiring a unified UV mapping in the pipeline. It can represent the fine details of characters, such as 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.

The contributions in this work are three-fold:

- We present a baseline method, CoNR, for a novel task of rendering anime character images with desired pose using character sheets.
- We explore the collaborative inference method of feed-forward neural networks to model character sheets as a dynamically-sized set of images.
- We introduce an Ultra-Dense Pose representation designed specifically for anime characters and build a character sheet dataset containing images of diverse poses.

2. Related Works

2.1. Image Generation & Translation for Anime

Recent years have seen significant advancement in applying deep learning techniques [17, 43, 49] to assist the creation of anime. The generative modeling of anime faces has achieved exciting results [23, 18, 40, 25, 21]. There are also attempts to produce vectorized anime images [37], similar to the step-by-step manual drawing process. A modified StyleGAN2 model [4] is proven to be able to generate full-body anime images. However, it still suffers from artifacts, including weirdly connected body parts, and it is not efficient or straightforward to control the generated poses of the character.

2.2. Human Pose & Appearance Transfer

In the real world, however, there has been a great success in the human pose or appearance transfer tasks [11, 5]. Most of these works create vivid body motions or talking heads from only one single image [44, 15, 34, 41, 45, 35, 26].

The learned prior of the human body [28], 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. Unfortunately, anime has long been featuring a flexible character design leading to high diversity in clothing, body shapes, hairstyles, and other appearance features, making it challenging to adopt real-world human priors in the domain of anime characters.

There are also some attempts [27] to extend the pose transfer task by utilizing SMPL [28], 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 similar posterior estimates, and better suit the needs of anime production.

Some recent works utilize NeRF [30], 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 [32, 31], 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

We analyze the representation of the human body posture in the real world in multiple data modes.

In the real world, stick-figure of skeletons [11], SMPL vectors [28], and heat maps of joints [10, 36] 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 pose exaggerating on additional moving parts like floating hair or flowing skirt, which are not directly driven by human joints. The aforementioned human pose representations, however, are very abstract and fuzzy in the inverse task of pose-guided anime rendering.

Human parsers or clothing segmenters [46, 45, 15, 12] are robust to the uncertainty of joint positions. However, the provided semantic masks are not informative enough for representing pose, or even the orientation of a person.

DensePose [19], UV texture mapping [44, 45, 16], 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 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. Last but not least, they have no idea of the number of key points they should use. Due to the diverse body shapes of anime characters, every region of the body 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 $\mathbf{I}_1$ ... $\mathbf{I}_n$ in 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 [48, 22, 37]. 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, the underlying mathematics of existing pose representations prevents 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 novel 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 give 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 properly model the task in 3.1, we propose a Collaborative Inference method for convolutional Neural Networks (CINN).

Usually, multiple images can be fed in arbitrary order into multiple copies of the same classical convolutional neural network 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 an existing convolutional neural network, we obtain a network of a dynamical number of sub-networks that 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 won't affect the inference result of the entire pipeline.

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 3, 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 message sent at each block of each view, we use half of the messaging outputs of each block as edge weights to perform a weighted averaging.

3.3. Ultra-Dense Pose

Here we present Ultra-Dense Pose (UDP), a compact landmark representation designed specifically for anime characters. 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 T-pose to align the joints, as shown in the Figure 2(a). To construct UDP, we remove the original texture and overwrite the color (r, g, b) of each vertex with a landmark, which is the current world coordinate(x, y, z) of this vertex, as shown in Figure 2(a). 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(b).

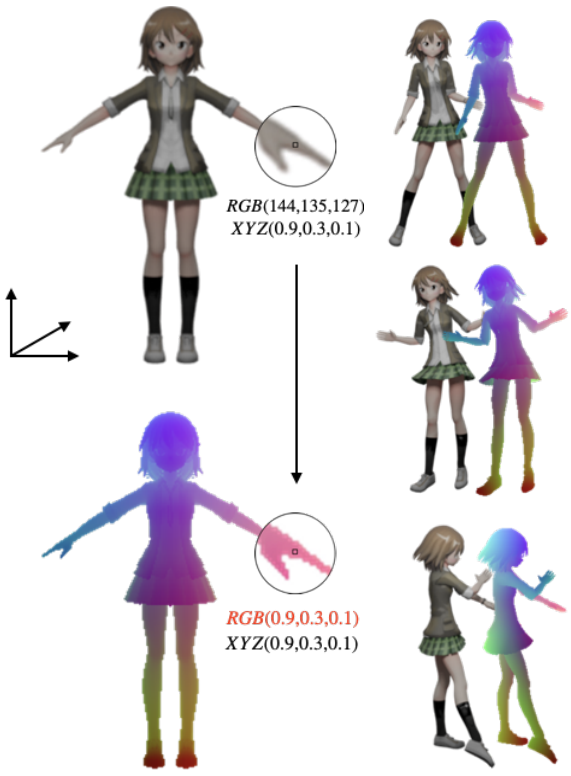

Figure 2: **UDP of Kurei Kei, an anime character.** (a) UDP uses 3D coordinates as landmarks to avoid the difficulties of unwarping 3D surfaces into a 2D UV map. (b) When the anime body changes its pose, the landmark at the corresponding body part will remain the same.

To avoid the difficulty of down-sampling and processing meshes, we convert the modified meshes into 2D images, which are more friendly to neural networks. 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 $u \times v \times 4$ shaped image recorded in floating-point numbers ranging from 0 to 1. The four channels include three body landmark encodings and one occupancy for whether the pixel is body or empty.

We find three properties of this representation that 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 complicated handcrafted annotations.

2) UDP is also 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) the encoding defined in UDP also serves a dual role of describing the local 3D shape of the human body, which could provide additional geometric information to downstream tasks, although it may not be necessary for the task of this work.

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, 24]. We manually perform matting to remove the background from the character with the help of the watershed algorithm. With a similar method, we also obtain a background dataset for use in augmentations.

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 from a synthesized dataset, which has 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-character basis so that the validation set contains characters unseen during training.

3.5. Collaborative Neural Rendering

Overview CoNR consists of a renderer and an optional Ultra-Dense Pose detector. Figure 3 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 compute body and cloth dynamics for the anime character directly.

Renderer The renderer is based on a simple U-Net [33]. We apply the following modifications.

Firstly, 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 shown in Figure 3. This allows us to checkpoint the evaluated results from the encoder that can be reused when inference on multiple target UDPs in a video.

Secondly, inspired by [13, 14], 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.

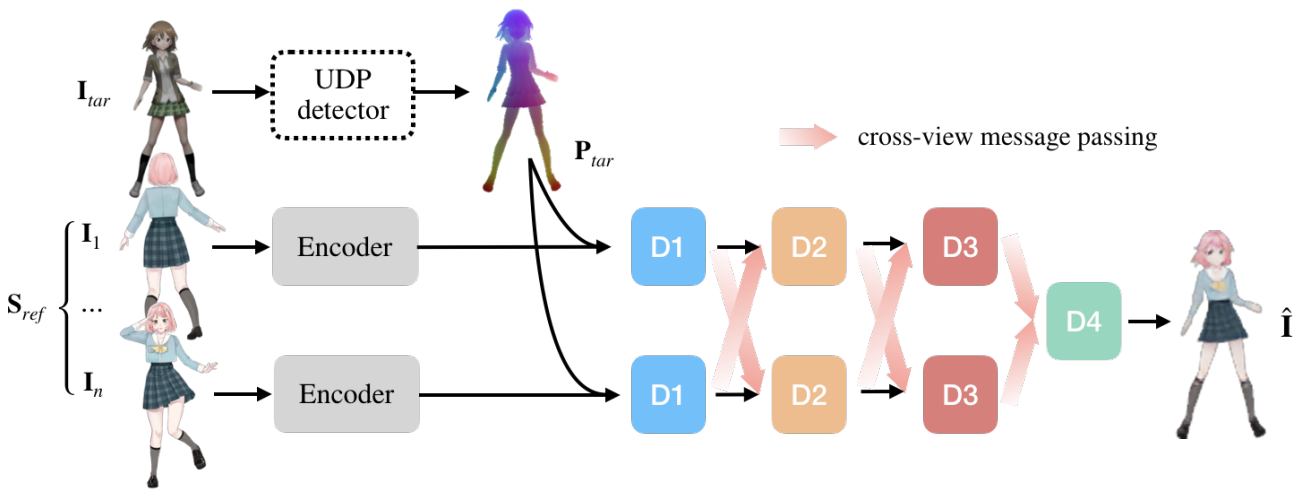

Figure 3: **Overview of CoNR.** Reference images $\mathbf{I}_1 \cdots \mathbf{I}_n \in \mathbf{S}_{ref}$ from the input character sheet are feed into a CINN using modified U-Nets as sub-networks. The same UDP representation $\mathbf{P}_{tar}$ detected from $\mathbf{I}_{tar}$ is resized and concatenated into each scale of the encoder outputs in all sub-networks. Blocks with the same color share weights. D1 to D4 refer to block 1 to block 4 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.

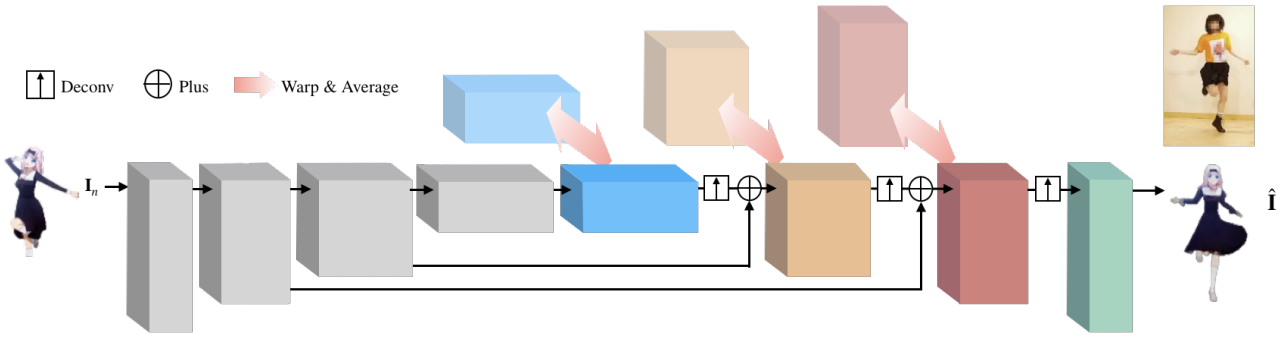

Figure 4: **Sub-network structure of CoNR**. Every input $\mathbf{I}_n \in \mathbf{S}_{ref}$ goes into a U-Net shaped sub-network. The warping and averaging operation are performed on the output of every decoder block. The semi-transparent blocks represent corresponding blocks in all other sub-networks. Blocks with the same color share weights. Each block in a sub-network would pass part of the outputs as messages to corresponding successive blocks in all other sub-networks, in addition to forwarding outputs to its following blocks in the current sub-network. The path of UDP is omitted.

Finally, 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, as shown in Figure 4. The last decoder block will collect averaged output features from all previous decoder block in all sub-networks and decode them as the final RGB output $\hat{\mathbf{I}}_{tar}$.

Ultra-Dense Pose Detector To prepare the UDP representation $\widehat{\mathbf{P}}_{tar}$ of target pose, we use a simple U-Net [33] consists of a ResNet-50 [20] encoder and a decoder with 5 Residual Blocks to detect it from an RGB image $\mathbf{I}_{tar}$, The detector gives out four channels of UDP values, in the same way as the UDP representation and the raining dataset.

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 optionally share the weight of the ResNet-50 backbone in the en-

coder of UDP detector with the encoder of the renderer. The UDP detector is only evaluated for on the target $\mathbf{I}_{tar}$.

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 lanmark encodings and BCE loss on the mask to train the detector if the ground truth label is available.

$$\mathcal{L}_{udp} = ||\widehat{\mathbf{P}}_{tar} - \mathbf{P}^{gt}_{tar}||_1 + BCE(\widehat{\mathbf{P}}_{tar\_mask}, \mathbf{P}^{gt}_{tar\_mask}) \tag{2}$$

In addition, we evaluate a consistency loss by computing the standard deviation of k UDP detector outputs.

$$\mathcal{L}_{cons} = \sqrt{\frac{1}{k-1}\sum_{j=1}^{k}(\widehat{\mathbf{P}}_j - \widehat{\mathbf{P}}_{tar})^2} \tag{3}$$

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,

$$\mathcal{L}_{photometric} = ||\Phi(\widehat{\mathbf{P}}_{tar}, \mathbf{S}_{ref}) - \mathbf{I}^{gt}_{tar}||_1 \tag{4}$$

Optionally, we compute the perceptual loss using LPIPS [47],

$$\mathcal{L}_{perception} = lpips(\Phi(\widehat{\mathbf{P}}_{tar}, \mathbf{S}_{ref}), \mathbf{I}^{gt}_{tar}) \tag{5}$$

The UDP detector and renderer are trained from scratch simultaneously in an end-to-end manner. The total loss function is evaluated as a weighted sum of all these losses, where $\alpha = 0.5, \beta = 0.05, \gamma = 1.0, \delta = 1.0$.

$$\mathcal{L} = \alpha\mathcal{L}_{perception} + \beta\mathcal{L}_{photometric} + \gamma\mathcal{L}_{udp} + \delta\mathcal{L}_{cons} \tag{6}$$

Our model is optimized by AdamW [29] with weight decay 10^{-4} for 2 epochs on the training set. 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 6 with all input resolution set to 128×128. CoNR pipeline is implemented in PyTorch and trained on four 2080Ti GPUs for about four days. For quantitative evaluation, we measure the training 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	epoch1		epoch2	
	$\mathcal{L}_1$	LPIPS	$\mathcal{L}_1$	LPIPS
$m=1, n=1$	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**

4.2. Comparative Study

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 2 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 asking the network 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 the training view count $m = 4$ while reducing the inference view count $n = 1$ will also harm the quality of the resulting image.

However, CoNR does not require the inference view count n to match the m during training. Adding additional views during inference will, on the contrary, enhance the quality of the generated images as shown in Figure 5. As illustrated in 3.2, CoNR models the input character sheet as a dynamically-sized set so that it can leverage information distributed across all images without favoring the first m inputs. This allows CoNR to scale seamlessly from a pose transfer task, when only a few shots of the character are given, to frame compiling or interpolating when a lot of footage of a character are available.

The example in Figure 5 shows the behavior of CoNR when it does not have enough information to draw missing parts during the inference correctly. Even if very few ref-

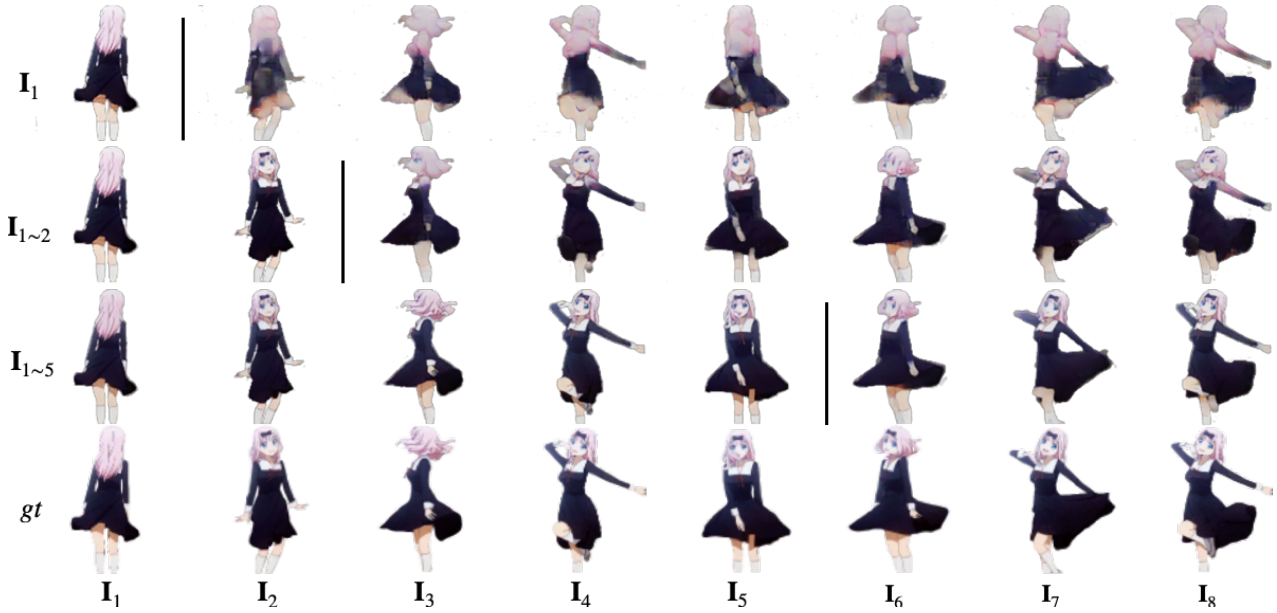

Figure 5: **Effectiveness of the collaboration.** We perform a reconstruction test with random video in the wild, with unseen pose and appearance, to understand the effectiveness of the collaboration. The last row shows 8 frames $\mathbf{I}_j^{gt} \in \mathbf{S}_{vid}^{gt}$ from an Youtube video. Starting form the first row shows the result of $\Phi(\widehat{\mathbf{P}}_j, \mathbf{S}_{ref})$. $\widehat{\mathbf{P}}_j$ are UDP detected from $\mathbf{I}_j^{gt}$. The used subsets of character sheet $\mathbf{S}_{ref} \subset \mathbf{S}_{vid}^{gt}$ are shown in the first column. Even if the provided reference images are not sufficient for drawing the target pose, CoNR will generate a conservative guess by trying to in-paint the unprovided area. Generated images for novel poses unseen in the character sheets are shown on the right of vertical lines per row.

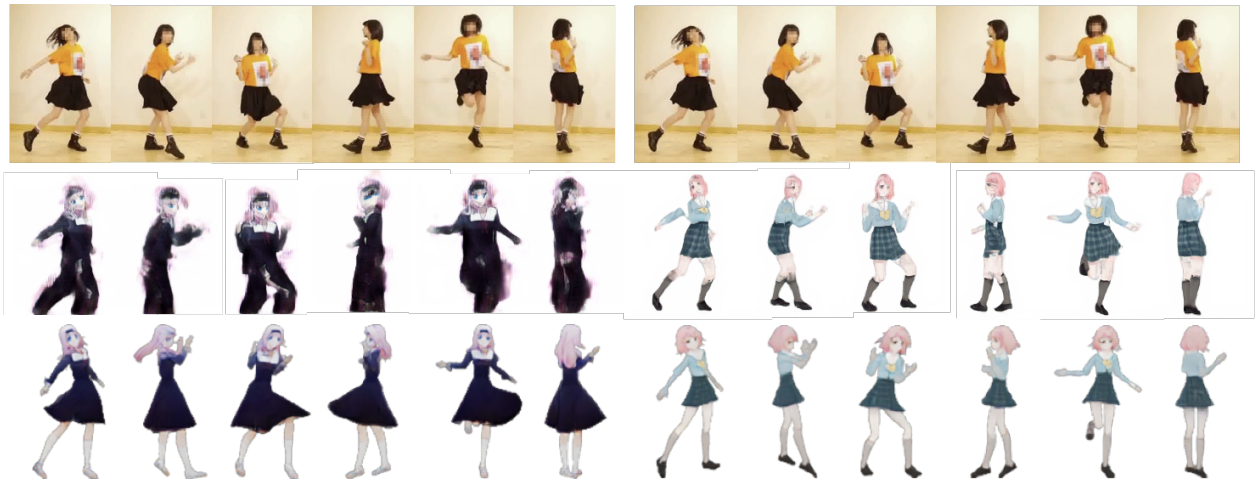

Figure 6: **Comparison with the SMPL-based method.** We compare results of our method (the 3nd row) with results of [27] (the 2rd row) when trying to resemble the target poses shown in the first row. Further diagnosis shows that parametric 3D human models like SMPL may not represent diverse clothing in anime. The long skirt of the first character is incorrectly mapped to the legs, and the short skirt of the second character seems to be tightened on the legs instead of sagging naturally.

erence image is provided, the target pose that finds a similar reference in the character sheets will be accurate, as expected.

Comparison with SMPL-based pose representations. We compare the results produced by CoNR to a real-world

Table 2: **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	epoch1		epoch2	
	L1	LPIPS	L1	LPIPS
$m=1, n=1$	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**

digital human system [27] using the same target poses [1] as used in their demo. To perform the test, we use two characters sheets[2] with both pose and appearance unseen during training. One contains the same 4 images as in Figure 1, the other character sheet taken from a random Youtube video contains $\mathbf{I}_{1\sim5}$ as used in Figure 5. Figure 6 shows that the long skirt prevents a high accuracy estimation of the leg joints and that parametric 3D human models like SMPL may not may not handle the body shape of anime characters correctly.

The visual comparison of inference results in Figure 6 also indicates that CoNR can produce images at desired target poses with better quality.

4.3. Ablative Studies

We performed ablation studies on the UDP detector and renderer. Table 3 shows that UDP representation can be inferred from images using a U-Net [33]. A näive 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 4. The proposed CoNR method with both the feature warping and the CINN method significantly increases the performance, thus establishing a baseline for this task.

5. Limitations

The proposed method 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 asserted on the character. Even if the current form of 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 [49, 17, 38] to properly deal with the lighting.

[1]The video is from https://download.impersonator.org/iper_plus_plus_latest_samples.zip

[2]A random anime character from a random Youtube video https://www.youtube.com/watch?v=m6k_t8yEyvE and another character illustrated by an amateur artist for this paper.

Table 3: **Ablation on UDP Detector.** U refers to a näive U-Net without batch-norms, b refers to a ResNet34 Encoder, $R50$ refers to a ResNet50 Encoder. The **red** setting is used in CoNR, the baseline proposed in this paper.

Setting	epoch1		epoch2	
	$\mathcal{L}_1$ †	BCE ‡	$\mathcal{L}_1$ †	BCE ‡
U	0.1247	0.0856	NaN	NaN
$U+R34$	0.1051	0.1068	0.1004	0.0747
$U+R50$	0.0969	0.0792	0.0971	0.0736

NaN: fail to converge, loss become NaN.
†: $\mathcal{L}_1$ of UDP.
‡: BCE of Mask.

Table 4: **Ablation on CoNR Renderer.** U refers to a näive U-Net without backbone or batch-norms. C refers to CINN method, G refers to the gird-sampling operation to perform output feature warping. R50 refers to ResNet50. The **red** setting is used in CoNR, the baseline proposed in this paper.

Setting	epoch1		epoch2	
	$\mathcal{L}_1$	LPIPS	$\mathcal{L}_1$	LPIPS
U	0.0313	0.1100	0.0311	0.1038
$U+G$	0.0309	0.1090	0.0308	0.1036
$U+C+R50$	0.0315	0.1097	0.0286	0.0977
$U+C+G$	0.0194	0.0673	0.0180	0.0612
$U+C+G+R50$	**0.0187**	**0.0659**	**0.0179**	**0.0612**

The proposed method is unable to model the dynamics of the character. The CoNR model requires detecting UDP from the images or videos used as the provider of the target body pose. This will limit the use of the proposed method in the animation or games as CoNR requires creating a storyboard beforehand to drive the generation of character images. To bypass the UDP detector, users have to rely on additional technologies like garment captures [9], physics simulations [6] or use learning based methods [42, 7, 39] to obtain a synthesized UDP.

The dataset may not fully reflect the distribution of anime characters in the wild. The collected dataset contains only human-like anime characters from the year 2014 to 2018. As building UDPs requires aligning character meshes according to joints, the models trained on this dataset may not be applied to animal-like characters, which follow joint hierarchies different from humans. Training with larger datasets may alleviate this limitation.

6. Conclusion

In this work, we introduce a novel task to render anime character images with desired pose using multiple images in character sheets. We developed a simple feed-forward baseline, CoNR, for this task. We anticipate the methods and the datasets presented in this paper would 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. 2

[2] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[3] Anonymous, Danbooru community, and Gwern Branwen. Danbooru2020: A large-scale crowdsourced and tagged anime illustration dataset. 4

[4] aydao. This anime does not exist. https://thisanimedoesnotexist.ai/. 2

[5] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[6] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98*, pages 43–54. ACM Press. 8

[7] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Pbns: Physically based neural simulator for unsupervised garment pose space deformation. *arXiv preprint arXiv:2012.11310*, 2020. 8

[8] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '99*, pages 187–194. ACM Press. 2

[9] Derek Bradley, Tiberiu Popa, Alla Sheffer, Wolfgang Heidrich, and Tamy Boubekeur. Markerless garment capture. In *ACM SIGGRAPH 2008 Papers on - SIGGRAPH '08*, page 1. ACM Press. 8

[10] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 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. 2

[11] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[12] Chien-Lung Chou, Chieh-Yun Chen, Chia-Wei Hsieh, Hong-Han Shuai, Jiaying Liu, and Wen-Huang Cheng. Template-free try-on image synthesis via semantic-guided optimization. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2021. 2

[13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 4

[14] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 4

[15] Oran Gafni, Oron Ashual, and Lior Wolf. Single-shot freestyle dance reenactment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[16] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait neural radiance fields from a single image. *arXiv preprint arXiv:2012.05903*, 2020. 2

[17] Zhengyan Gao, Taizan Yonetsuji, Tatsuya Takamura, Toru Matsuoka, and Jason Naradowsky. Automatic Illumination Effects for 2D Characters. In *NIPS Workshop on Machine. Learning for Creativity and Design.*, 2018. 2, 8

[18] Aaron Gokaslan, Vivek Ramanujan, Daniel Ritchie, Kwang In Kim, and James Tompkin. Improving shape deformation in unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[19] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5

[21] Zhenliang He, Meina Kan, and Shiguang Shan. Eigengan: Layer-wise eigen-learning for gans. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[22] Zhewei Huang, Shuchang Zhou, and Wen Heng. Learning to Paint With Model-Based Deep Reinforcement Learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[23] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the Automatic Anime Characters Creation with Generative Adversarial Networks. In *NIPS Workshop on Machine. Learning for Creativity and Design.*, 2017. 2

[24] Jerry Li. Pixiv dataset. 4

[25] Minjun Li, Yanghua Jin, and Huachun Zhu. Surrogate gradient field for latent space manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[26] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. 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. 2

[27] Wen Liu, Zhixin Piao, Zhi Tu, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan with attention: A unified framework for human image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 2, 7, 8

[28] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 6

[30] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[31] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[32] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention (MICCAI)*, 2015. 4, 5, 8

[34] Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. Neural Re-Rendering of Humans from a Single Image. Lecture Notes in Computer Science. 2

[35] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 2

[36] Aliaksandr Siarohin, Oliver J. Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion Representations for Articulated Animation. 2

[37] Hao Su, Jianwei Niu, Xuefeng Liu, Jiahe Cui, and Ji Wan. Vectorization of raster manga by deep reinforcement learning. *arXiv preprint arXiv:2110.04830*, 2021. 2, 3

[38] Wanchao Su, Dong Du, Xin Yang, Shizhe Zhou, and Hongbo Fu. Interactive Sketch-Based Normal Map Generation with Deep Neural Networks. 1(1):1–17. 8

[39] Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Neural-gif: Neural generalized implicit functions for animating people in clothing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 8

[40] Hung-Yu Tseng, Matthew Fisher, Jingwan Lu, Yijun Li, Vladimir Kim, and Ming-Hsuan Yang. Modeling artistic workflows for image generation and editing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[41] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. 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. 2

[42] Tuanfeng Y Wang, Tianjia Shao, Kai Fu, and Niloy J Mitra. Learning an intrinsic garment space for interactive authoring of garment animation. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019. 8

[43] Xinrui Wang and Jinze Yu. Learning to cartoonize using white-box cartoon representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[44] Pengfei Yao, Zheng Fang, Fan Wu, Yao Feng, and Jiwei Li. Densebody: Directly regressing dense 3d human pose and shape from a single color image. *arXiv preprint arXiv:1903.10153*, 2019. 2

[45] Jae Shin Yoon, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt. 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

[46] Mihai Zanfir, Alin-Ionut Popa, Andrei Zanfir, and Cristian Sminchisescu. Human appearance transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

[48] Song-Hai Zhang, Tao Chen, Yi-Fei Zhang, Shi-Min Hu, and Ralph R Martin. Vectorizing cartoon animations. *IEEE Transactions on Visualization and Computer Graphics*, 15(4):618–629, 2009. 3

[49] Qingyuan Zheng, Zhuoru Li, and Adam Bargteil. Learning to shadow hand-drawn sketches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 8