Supplementary Material

Overview of Supplementary Material. This supplementary material consists of 6 sections: (1) related literature and context of our paper (Section 1), (2) details of data creation process (Section 2), (3) additional AnimeCeleb samples and experimental results (Section 3), (4) implementation details of the *AniMo* and the baselines (Section 4), (5) additional head reenactment results of *AniMo* (Section 5), and (6) discussions and future work (Section 6).

1 Related Work

With abundance of digital contents, numerous animation datasets collected from different media are released to community. Focusing on animation head datasets, there exist multiple studies [1, 2, 4, 16, 26] that provide the pre-processed animation heads. Based on these datasets, early animation-related research [14, 20, 24] mainly focused on recognizing and detecting an animation character in animation scenes. However, an extension of animation research to generative modeling is non-trivial. One major bottleneck is that the released datasets are collected from unlisted online source, thereby containing unexpected and noisy images (*e.g.*, an occluded head). In this regard, existing datasets are forced to narrow their application scope; for example, current animation datasets are not suitable to train *head reenactment models* [3, 5, 18, 21, 22, 23].

Head reenactment aims to drive a source image to mimic a motion of a target image while preserving identity of the source image. Most approaches [3, 5, 18, 21, 22, 23] use two frames from the same video during training; an image conveys the identity-related information while the other provides the motion-related information, which are combined to produce a final output. Also, multiple pose representations (*e.g.*, keypoints and 3DMM parameters) play vital roles to deliver the head motion in previous literature [5, 15, 22, 23]. In fact, the pose representation is an important aspect for head reenactment approaches as shown in previous work [3] when training a high-performing model.

Undoubtedly, the collected animation images of current datasets [1, 2, 4, 16, 26] do not include its detailed pose annotations, and obtaining accurate pose representations is also non-trivial. In opposition, AnimeCeleb provides numerous groups of images that have the same identity, and detailed pose annotations, which bear a potential to be used for various generation tasks.

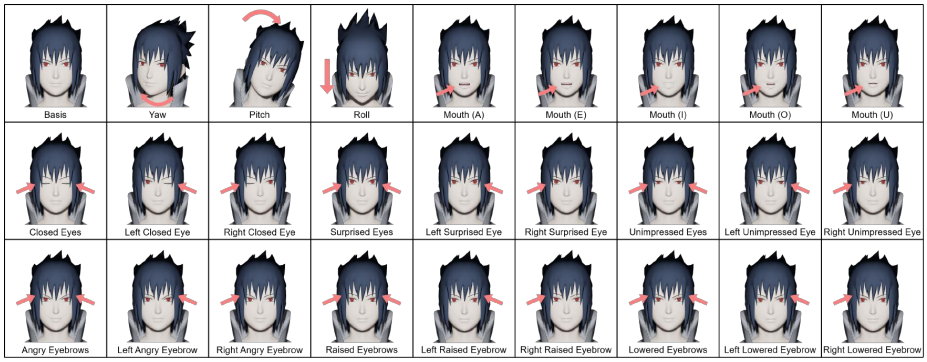

Fig. 1: Visualizations of all target morphs and 3D head angles. Given a neutral image (*Top-left*), we apply every annotated target morph independently with the maximum intensity (*i.e.*, 1.0) to obtain the morph-applied images. We highlight the locations, where manipulations occur with pink arrows.

2 Details of Data Creation Process

In this section, we present the details of the data creation process as follows:

- The visualizations of entire pre-defined target morphs that a single character has (Fig. 1).
- Detailed user interfaces of the annotation system: statistics, group annotation, and individual inspection (Fig. 3) and mapping relationships between the source morphs and the target morphs after the annotation (Table 1).
- Detailed description of pose sampling process for generating a pose vector (Algo. 1).
- Sampling examples from a 3D animation model (Fig. 2).

Visualizations of Target Morphs. Fig. 1 shows the visualizations of the manipulated poses and their corresponding target morphs, which are responsible for annotating the source morphs. For head rotation and mouth annotation, there is a single value to control each semantic, respectively. On the other hand, for eyes and eyebrows annotation, we consider left and right part separately and define three different target morphs: left-related, right-related and both-related semantics. Note that although we have defined 23 target morphs including six morphs that control both parts (*e.g.*, closed eyes and raised eyebrows), during constructing a pose vector, the both-related morphs simultaneously determine two values of the pose vector (*i.e.*, left and right part). Therefore, the dimensions of a pose vector become 20 (=17+3) with three additional head angle dimensions.
Semantic Annotation System. Fig. 3 shows the components of semantic annotation system developed with Vue.js [1]. Given a group of neutral images and morph-applied images, our system aims at visualizing the images and the source morph names. Through the annotation, the source morphs are annotated as the target morphs, considering a semantic match.

[1] https://vuejs.org/

The system consists of three views: statistics, group annotation and individual inspection. In statistics view, there are the number of models and unique morph names that the models contain, and annotation progress shows the ratio of the annotated models to the total models. During annotation, we match the source morphs (*e.g.*, 困る and なごみ) as their corresponding target morphs (*e.g.*, lowered eyebrows and closed eyes) by considering given sample images as seen in the group annotation view of Fig. 3. Next, we manually check the validity of a single morph one-by-one by examining its corresponding morph-applied image as shown in individual inspection view of Fig. 3. If the morph-applied image has an unmatched semantic, we exclude that source morph marking it as X. We present the annotation results in Table 1.

Pose Sampling Process. Algorithm 1 depicts a detailed process for sampling a pose vector $\mathbf{p} \in \mathbb{R}^{20}$. Note that the annotated target morphs can be different depending on the 3D animation model. Given the annotated target morphs $\{e_n\}_{n=1}^{N}$, we first select a semantic of each part: eye s_{eye}, eyebrow $\mathrm{s}_{eyebrow}$, and mouth s_{mouth}. For example, if there exist Mouth (A), Mouth (E) and Mouth (O) as mouth semantics, we randomly sample one of them as s_{mouth}. Similar to this, the pre-defined target morphs are randomly sampled for s_{eye} and $\mathrm{s}_{eyebrow}$, respectively. The difference is that we check whether a 3D animation model contains independent morphs that can control left and right part separately or a single morph to adjust both parts. If there exist the independent morphs, they are used with priorities. Then, the values sampled from a uniform distribution are assigned to the selected semantics as well as head angles (*i.e.*, roll, pitch, and yaw). This results in a pose vector $\mathbf{p}$ that works for manipulating a pose of an animation character.

Sampling Examples. Fig. 2 shows the example pairs of generated images and pose vectors from a 3D animation model. The output data consists of *frontalized-expression* and *rotated-expression* images and their corresponding pose vectors that contain 17 different morphs and 3D head angles. In addition, we provide four different shader styles: (S.1), (S.2), (S.3) and (S.4) to boost the diversity of images and consider various drawing styles of animation creators.

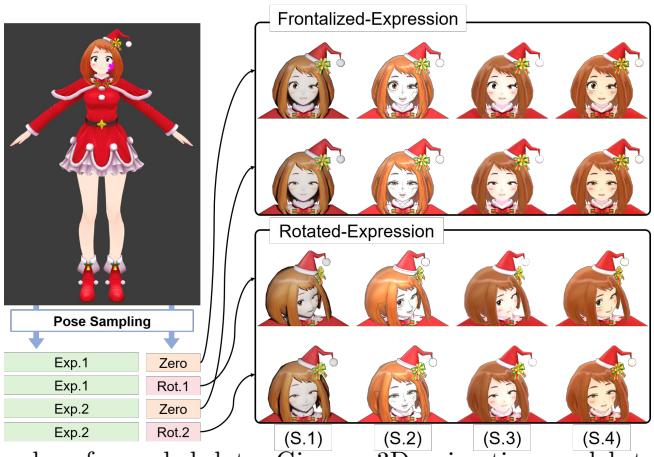

Fig. 2: Examples of sampled data. Given a 3D animation model, two groups of images are generated: (1) *frontalized-expression* images using the sampled target morphs and zero head angles (*Top-right*), and (2) *rotated-expression* images after adding the sampled head angles (*Bottom-right*). Note that four different shading styles are applied for image rendering.

Source morphs	*Target morphs*
あ, ああ, あ2	Mouth(A)
え, ええ, え2, え	Mouth(E)
い, いい, い2, い	Mouth(I)
お, おお	Mouth(O)
う, うう	Mouth(U)
ばたき, 笑い, なごみ	Closed Eyes
ウィンク, ウィンク.001, ウィンク2, なごみ左	Left Closed Eye
ウインク右, なごみ右, ウインク2右, 2右	Right Closed Eye
半目, じと目, ジト目	Unimpressed Eye
じと目左	Left Unimpressed Eye
じと目右	Right Unimpressed Eye
びっくり2, びっくり, 驚き	Surprised Eyes
びっくり左, びっくり2左	Left Surprised Eye
びっくり2右, びっくり右	Right Surprised Eye
怒り眉, 怒り2, 怒り	Angry Eyebrows
怒り左, ’怒り眉左, 怒りL	Left Angry Eyebrow
怒り眉右, 怒り右, 怒りR	Right Angry Eyebrow
上	Raised Eyebrows
上左, 上L	Left Raised Eyebrow
上右, 上R	Right Raised Eyebrow
下, 困る	Lowered Eyebrows
困るL, 下L, 困る左, 下左	Left Lowered Eyebrow
困る右, ’下R, 下右, 困るR	Right Lowered Eyebrow

Table 1: Examples of mapping relationships between the source morphs and the target morphs.

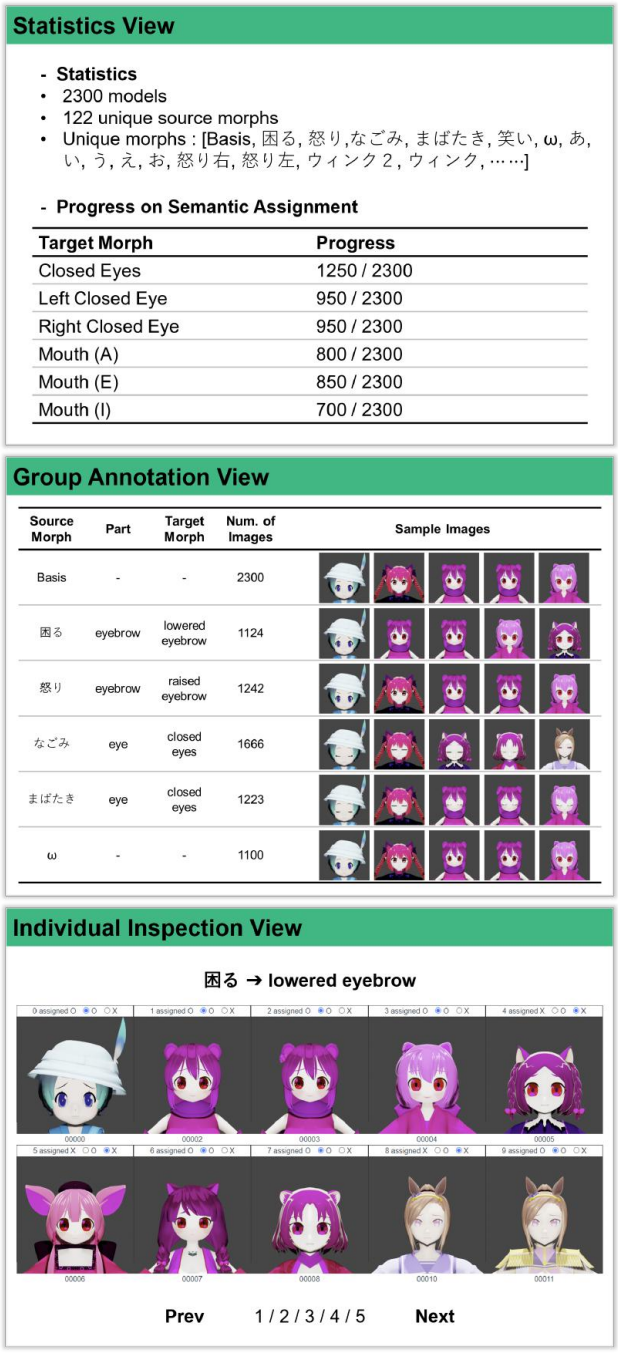

Fig. 3: Simplified semantic annotation system overview.

Algorithm 1: Pseudo Codes for Pose Sampling

Data: Annotated target morphs $\{e_n\}_{n=1}^{N}$
`/* N indicates the number of source morphs of a 3D animation model. */`
Result: A sampled pose $\mathbf{p} \in \mathbb{R}^{20}$
`/* Select eye, eyebrow, mouth semantics and sample the values from a uniform distribution. */`
$\mathrm{s}_{eye}, \mathrm{s}_{eyebrow}, \mathrm{s}_{mouth} \leftarrow \mathrm{sample}(\{e_n\})$
if $\exists\, left\text{-}s_{eye}, right\text{-}s_{eye} \in \{e_n\}$ **then**
  $u_1, u_2 \sim \mathcal{U}(0,1)$;
  $left\text{-}\mathrm{s}_{eye}(v) \leftarrow u_1$;
  $right\text{-}\mathrm{s}_{eye}(v) \leftarrow u_2$;
else
  $u \sim \mathcal{U}(0,1)$;
  $left\text{-}\mathrm{s}_{eye}(v) \leftarrow u$;
  $right\text{-}\mathrm{s}_{eye}(v) \leftarrow u$;
end
if $\exists\, left\text{-}s_{eyebrow}, right\text{-}s_{eyebrow} \in \{e_n\}$ **then**
  $u_1, u_2 \sim \mathcal{U}(0,1)$;
  $left\text{-}\mathrm{s}_{eyebrow}(v) \leftarrow u_1$;
  $right\text{-}\mathrm{s}_{eyebrow}(v) \leftarrow u_2$;
else
  $u \sim \mathcal{U}(0,1)$;
  $left\text{-}\mathrm{s}_{eyebrow}(v) \leftarrow u$;
  $right\text{-}\mathrm{s}_{eyebrow}(v) \leftarrow u$;
end
$u \sim \mathcal{U}(0,1)$;
$\mathrm{s}_{mouth}(v) \leftarrow u$;
`/* Sample roll, pitch and yaw from a uniform distribution. */`
$\mathrm{roll}(v), \mathrm{pitch}(v), \mathrm{yaw}(v) \sim \mathcal{U}(-20^\circ, 20^\circ)$;
`/* Fill p with sampled values. p[·] denotes an index of each semantic. */`
initialize $\mathbf{p} = \{p_m\}_{m=1}^{20} = \{0, 0, ..., 0\}$;
$\mathbf{p}[left\text{-}\mathrm{s}_{eye}] = left\text{-}\mathrm{s}_{eye}(v)$;
$\mathbf{p}[right\text{-}\mathrm{s}_{eye}] = right\text{-}\mathrm{s}_{eye}(v)$;
$\mathbf{p}[left\text{-}\mathrm{s}_{eyebrow}] = left\text{-}\mathrm{s}_{eyebrow}(v)$;
$\mathbf{p}[right\text{-}\mathrm{s}_{eyebrow}] = right\text{-}\mathrm{s}_{eyebrow}(v)$;
$\mathbf{p}[\mathrm{s}_{mouth}] = \mathrm{s}_{mouth}(v)$;
$\mathbf{p}[\mathrm{roll}] = \mathrm{roll}(v)$;
$\mathbf{p}[\mathrm{pitch}] = \mathrm{pitch}(v)$;
$\mathbf{p}[\mathrm{yaw}] = \mathrm{yaw}(v)$;

3 Additional AnimeCeleb Samples and Experimental Results

This section presents additional results as follows:

- Other examples sampled from the AnimeCeleb.
- Qualitative head reenactment results on other animation head images obtained from Waifu Labs [2] and Danbooru 2019 [2].
- Other applicable tasks using the AnimeCeleb: animation colorization and image harmonization.

Additional Examples from AnimeCeleb. Fig. 4 shows the sampled images of eight different characters. As aforementioned, we present two image groups: *frontalized-expression* (the first row) and *rotated-expression* (the second row), and a difference between two groups lies in whether a head rotation is applied to the animation heads or not. As seen in Fig. 4, the images rendered with different shaders are generated with the exact same pose vector ((S.2-4) in Fig. 4) for the purpose of providing multiple styles of images.

Other Animation Images Head Reenactment. We present qualitative results using the PIRenderer [15] trained with the AnimeCeleb on two other animation sample images obtained from the Waifu Labs and the Danbooru 2019. We choose to use the PIRenderer(w/ pose vector) because it has strong generalization capacity compared to other models as seen in main manuscript. As shown in Fig. 5, the trained model successfully generates the head reenactment results given a source and a driving image. The PIRenderer(w/ pose vector) produces favorable outputs, imitating the head poses of driving images. Furthermore, Fig. 6 shows the outcomes on the Danbooru 2019. Due to the distribution gap between the AnimeCeleb and the Danbooru 2019, we confirm slight performance degradation for the samples from the Danbooru 2019.

Additional Applications of AnimeCeleb. To reveal the benefits of the AnimeCeleb, we implement additional two tasks: an *animation colorization*, and an *image harmonization*. The third shader (*i.e.*, S.3) styled images are used to train the colorization and the harmonization models. We clarify an importance of each task in the animation domain and show experimental results in the following paragraphs.

First, the animation colorization is a practical task for animation creators to reduce their effort during the labor-intensive painting process. Given a trained colorization model, creators are able to obtain colorized images given sketch images. We conduct character colorization tasks using both unconditional and conditional colorization baselines [7, 10]. As can be seen in Fig. 7, the colorization models trained with the AnimeCeleb show a promising performance at painting the animation character sketch images, producing plausible colorization outputs in an automatic manner or following a given animation reference image. To demonstrate the broad generalization capacity of the reference-based colorization

[2] https://waifulabs.com/

model [10] trained with the AnimeCeleb, we also use the reference images crawled from online cartoons. We find that not limited to the AnimeCeleb reference images, the model achieves high-quality colorization outputs based on other animation head images.

Second, the image harmonization aims to generate natural composite images given two images from different domains, achieving a visually pleasing match for both content and style. We implement a representative optimization-based approach [25] to explore the applicability of the AnimeCeleb and generate more realistic animation images. Since the AnimeCeleb images only contain a foreground object (*i.e.*, an animation head), a composition with suitable background is a natural extension of the AnimeCeleb. Not limited to the background composition, decorative objects (*e.g.*, sunglasses, caps and masks) are available assets to be exploited for the composition. We can employ an optimization-based composition model [25] that requires a foreground segmentation mask because the AnimeCeleb includes the segmentation mask. As shown in Fig. 8, both background and decorative object composition with the AnimeCeleb produce plausible results, demonstrating a potential extension of the AnimeCeleb in that it can provide the images with diverse backgrounds and multiple objects.

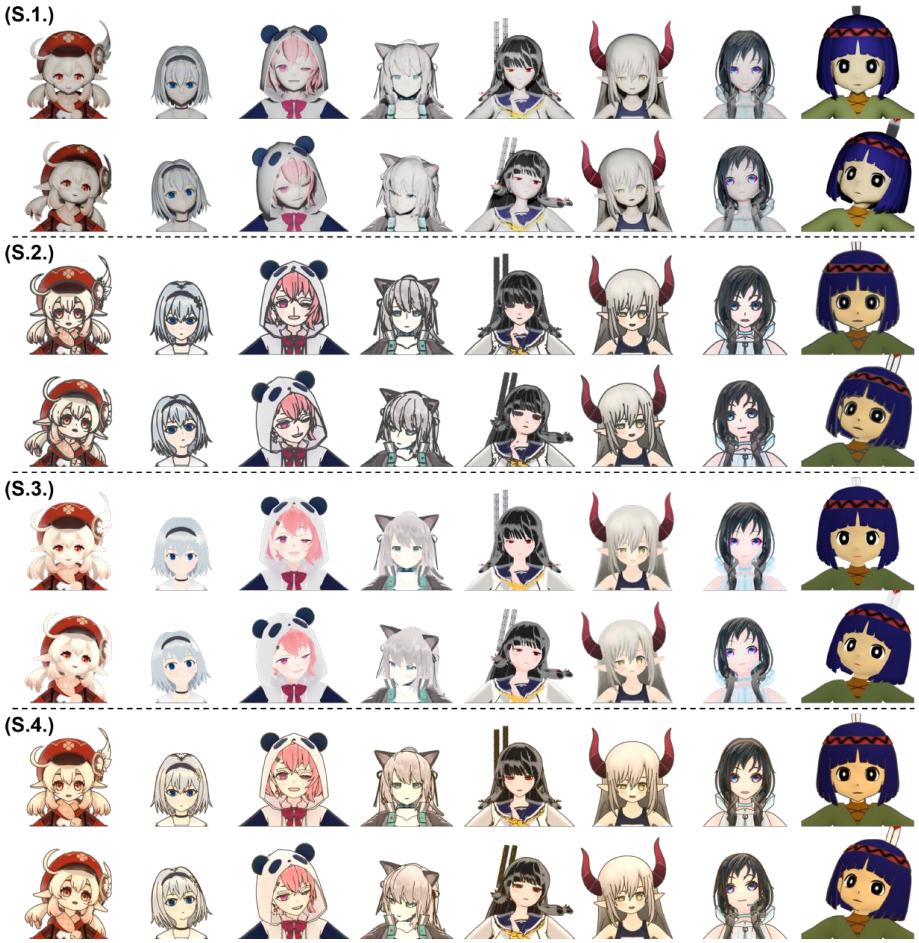

Fig. 4: Examples of the created images from the AnimeCeleb.

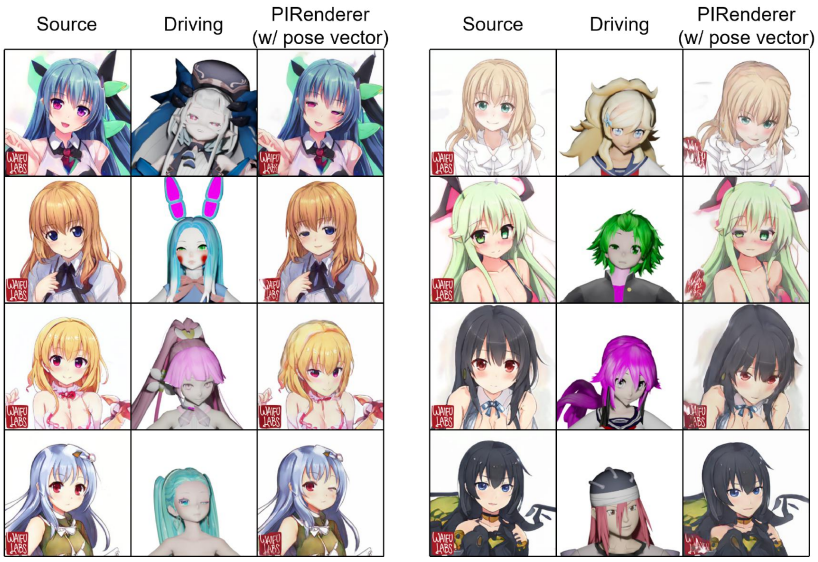

Fig. 5: Additional animation head reenactment results on the images from Waifu Labs.

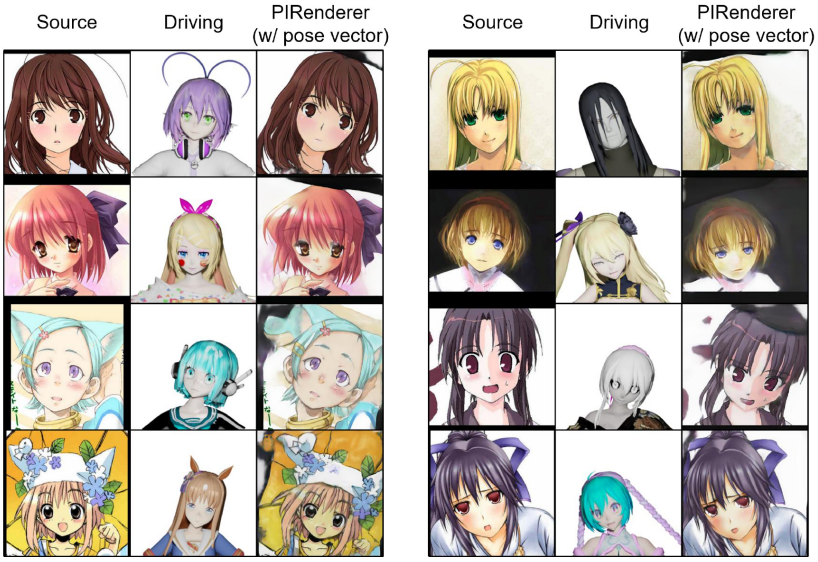

Fig. 6: Additional animation head reenactment results on the Danbooru 2019.

Fig. 7: Colorization results in an automatic and reference-based manner on the AnimeCeleb and other collected images. A Pix2Pix [7] trained with the AnimeCeleb successfully outputs a plausible colorized image. Also, a reference-based model [11] successfully fills a given sketch image with the color maps extracted from reference images.

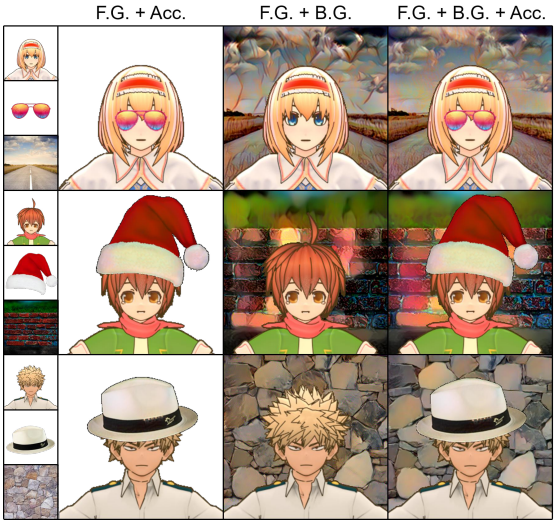

Fig. 8: Image harmonization results. F.G., B.G. and Acc. denotes a foreground object, a background, and an accessory, respectively. The components for image harmonization (the *1st column*) are well-blended, where the backgrounds and the accessories are refined with similar styles with an animation character.

4 Implementation Details of the *AniMo* and Baselines

In this section, we describe the architectures of the motion network, the warping network, and the editing network in detail, and objective functions for training. Then, we elaborate the baselines [18, 15, 3] and implementation details of them, respectively.

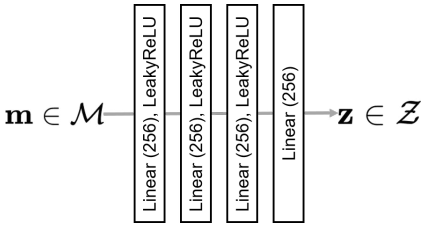

Fig. 9: The architecture of the motion network.

Motion Network. As shown in Fig. 9, the motion network has a multi-layer perceptron structure, which consists of four fully-connected layers that are responsible for resulting in a latent motion code $\mathbf{z} \in \mathbb{R}^{256}$ given the 3DMM parameters $\mathbf{m} \in \mathbb{R}^{70}$. The latent motion code $\mathbf{z}$ are transformed to estimate the affine parameters for adaptive instance normalization (AdaIN) [6] operations in the warping network and the editing network.

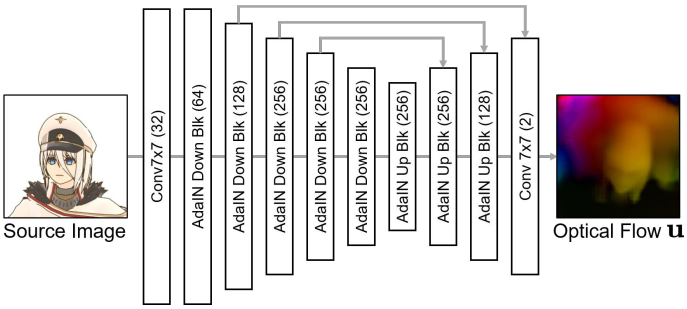

Fig. 10: The architecture of the warping network.

Warping Network. As shown in Fig. 10, the warping network has a encoder-decoder architecture. In addition, we employ the skip-connection as U-Net [17] to preserve the spatial information as well as AdaIN operation to inject the motion information. The optical flow $\mathbf{u} \in \mathbb{R}^{64\times64\times2}$ is upsampled or downsampled to fit the sizes of feature maps in the editing network.

Editing Network. Fig. 11 shows the architecture of the editing network. The editing network employs the structure of a hourglass network [12], in which in-

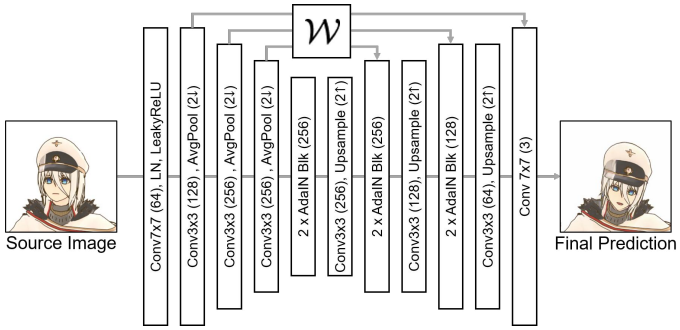

Fig. 11: The architecture of the editing network.

termediate encoder feature maps are passed to the decoder layers by an element-wise addition operation. When propagating the multi-scale feature maps of the encoder to the decoder, the optical flow $\mathbf{u}$ is applied to the multi-scale feature maps. In addition, as similar to the warping network, we utilize the AdaIN operation to inject the motion information.

Objective Functions. In order to train the *AniMo*, we follow the PIRenderer [15] objective functions as follows.

First, a reconstruction loss encourages the warping network to estimate an accurate optical flow. For the sake of this, we apply the estimated optical flow to a source image s, and encourage the warped output to reconstruct a driving image d. Instead of pixel-level loss, we employ the perceptual loss [8] to minimize the ℓ_1 distances in latent feature space between the warped source image $\mathbf{u}(s)$ and driving image d. Formally, this can be written as:

$$\mathcal{L}^{w}_{perc}(s, d) = \sum_{j} \left\| \phi_j(\mathcal{W}(s, \mathbf{u})) - \phi_j(d) \right\|_1, \tag{1}$$

where ϕ_j represents the activation map of j-th layer of the pre-trained VGG-19 network [19] and $\mathcal{W}$ denotes a warping operation. This leads to reliable optical flow prediction of the warping network.

Second, our editing network is trained with two losses: a reconstruction loss $\mathcal{L}^{g}_{perc}$ and a style loss $\mathcal{L}^{g}_{sty}$. The reconstruction loss is designed to reduce the errors between the final prediction $\hat{d}$ and the ground-truth driving image d. This can be formulated as:

$$\mathcal{L}^{g}_{perc}(d, \hat{d}) = \sum_{j} \left\| \phi_j(\hat{d}) - \phi_j(d) \right\|_1. \tag{2}$$

Next, the style loss is introduced to match the statistics between the ground truth driving image d and the final prediction as follows:

$$\mathcal{L}^{g}_{sty}(d, \hat{d}) = \sum_{j} \left\| C^{\phi}_{j}(\hat{d}) - C^{\phi} j(d) \right\|_1, \tag{3}$$

where C_j^{ϕ} denotes the gram matrix calculated from the activation maps ϕ_j.

In summary, our full objective function is given as:

$$\begin{aligned}\mathcal{L}_{total} = &\lambda^w_{perc}(\mathcal{L}^w_{perc}(s^{(a)}, d^{(a)}) + \mathcal{L}^w_{perc}(s^{(r)}, d^{(r)})) \\ &+ \lambda^g_{perc}(\mathcal{L}^g_{perc}(d^{(a)}, \hat{d}^{(a)}) + \mathcal{L}^g_{perc}(d^{(r)}, \hat{d}^{(r)})) \\ &+ \lambda^g_{sty}(\mathcal{L}^g_{sty}(d^{(a)}, \hat{d}^{(a)}) + \mathcal{L}^g_{sty}(d^{(r)}, \hat{d}^{(r)})),\end{aligned}$$

where $\lambda^w_{perc}, \lambda^g_{perc}$ and λ^g_{sty} are hyperparameters that control the relative importance of three different losses. We set $\lambda^w_{perc}, \lambda^g_{perc}$ and λ^g_{sty} as 2.5, 4 and 250, respectively. Note that our framework is jointly trained on both the AnimeCeleb and VoxCeleb.

We train the *AniMo* in two stages, where the motion network and the warping network are trained for 100 epochs, and we train the entire network for the additional 100 epochs. We employ the Adam [9] optimizer, one of the widely-used optimization methods, with the learning rate of 0.0001. The learning rate is set initially as 0.0001, then decreased to 0.00002 after 150 epochs. The batch size is set to 12 for all experiments.

Head Reenactment Baselines. We compare the *AniMo* with state-of-the-art models [3, 18, 15]. Since we leverage two datasets during training, comparable baselines are trained on either the VoxCeleb following their original implementations or both the VoxCeleb and AnimeCeleb.

In the following, we describe each baseline and experimental settings:

- **First-Order Motion Model (FOMM)** [18] is an unsupervised landmark-based approach, which internally detects the spatial positions to transform the source image. We implement two versions of this model: a VoxCeleb-trained and a jointly-trained model using both the AnimeCeleb and the VoxCeleb.
- **PIRenderer** [15] takes the 3DMM parameters to represent a driving motion and employs the AdaIN operation to inject the motion information. Similar to FOMM, we first implement a VoxCeleb-trained model. Also, we apply our pose mapping $\mathcal{T}$ to use a shared pose representations (*i.e.*, 3DMM parameters) for the purpose of achieving joint training.
- **Latent Pose Descriptor (LPD) [3]** relies on the AdaIN operation to inject a motion information, where the driving image is encoded as latent pose vector in unsupervised manner. To handle an unseen identity during inference, a trained model is fine-tuned with the same-identity images to infer. For evaluation, we utilize a model trained on the VoxCeleb, and fine-tune it using a group of the same-identity images in the AnimeCeleb.

For the implementations of existing baselines, we follow the hyper-parameters given in the original papers and codes.

5 Additional Head Reenactment Results of *AniMo*

This section contains additional head reenactment results with the *AniMo* and the baselines as follows:

- Qualitative results on self-identity (VoxCeleb and AnimeCeleb), cross-identity (VoxCeleb and AnimeCeleb), and cross-domain head reenactment (Vox. → Anime. and Anime. → Vox.) tasks.
- Intuitive pose editing of an animation and human head images.
- Qualitative results on cross-domain head reenactment using various unseen head images.
- A user study to compare the characteristics with iCartoon and head angle distribution comparison between VoxCeleb.

Additional Qualitative Head Reenactment Results of *AniMo*. In the experiments, we utilize two different training source: single dataset (VoxCeleb) and joint datasets (AnimeCeleb and VoxCeleb). We use the single dataset (VoxCeleb) to compare the original experimental setup of the previous studies [3, 18, 15]. For qualitative comparisons, we show the results of three tasks: (1) **self-identity head reenactment** where the identical being provides both a source and a driving image, (2) **cross-identity head reenactment** where the identities of a source and driving image are different within the same dataset, and (3) **cross-domain head reenactment** where two frames of different identities sampled from the AnimeCeleb and the VoxCeleb alternatively for the sake of serving as a source and a driving image; for example, Vox. → Anime. denotes a source and driving image are sampled from the AnimeCeleb and the VoxCeleb, respectively. Note the warping and the editing network for each domain: W_A, G_A and W_V, G_V are responsible for producing an animation and a real human head image, respectively.

Fig. 13 shows qualitative comparisons on self-identity head reenactment using the VoxCeleb. As seen in Fig. 13, our model produces the outputs that are perceptually realistic, as good as the baselines. Although the baselines show similar results on the task, there is a performance gap between the models when it comes to handling cross-identity inputs. As shown in Fig. 14, the FOMM [18] often fails to produce photo-realistic results because a head structure of a driving image is involved to generate results (the *3rd* and the *5th* columns). Compared to these results, the models which rely on the 3DMM parameters successfully handle cross-identity inputs (the *4th*, the *6th* and the *last* columns in Fig. 14).

Meanwhile, when performing on self-identity head reenactment using the AnimeCeleb, it is obvious that the models trained only with the VoxCeleb do not work well (the *3rd* and the *4th* columns in Fig. 15). In contrast, the models trained with the VoxCeleb and the AnimeCeleb show a promising performance (the *6th* and the *last* columns in Fig. 15), yet the FOMM still has difficulty in synthesizing vivid textures of a source image (the *5th* column in Fig. 15). In addition, Fig. 16 shows similar results on cross-identity head reenactment, where the models trained with the VoxCeleb have performed poorly (the *3th* and the *4th* columns). In contrast, the others trained with the AnimeCeleb successfully synthesize the outputs (the *6th* and the *last* columns[3]) except for the FOMM

[3] Note that both models use our pose mapping method.

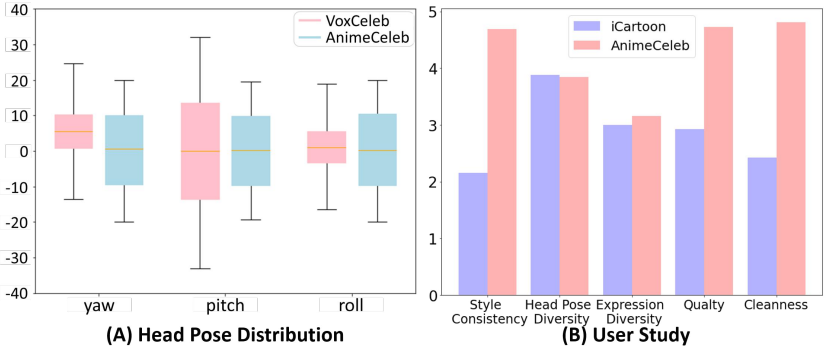

Fig. 12: (A) Comparison of head pose statistics between VoxCeleb and AnimeCeleb. (B) User study results for comparison between iCartoon and AnimeCeleb. The higher score is better.

(the *5th* column). Furthermore, Fig. 17 and 18 demonstrate that our model generates photo-realistic results compared to the baselines for cross domain head reenactment.

Intuitive Image Editing. One of the important applications of our model is to explicit control of a facial expression and head rotation on both the animation and human domain. As shown in Fig. 19, the *AniMo* is capable of generating high-quality images steered by diverse semantics. For example, an animation and human head can be controlled along roll, pitch and yaw axis (the *1st* row in Fig. 19), and manipulating the facial expressions (*i.e.*, eyes and a mouth) is achievable (the *2nd* row in Fig. 19).

Head Reenactment of Other Animation Images. In this experiment, we evaluate our model on multiple head image samples collected from different sources, including Waifu Labs, Naver Webtoon [4], Face Sketches [13], 2D Disney [5] as seen in Fig. 20. Given the trained W_A and G_A of the *AniMo*, the poses of other animation images can be controlled with the guidance of driving poses. However, we also find that there exist problems such as a background distortion and a lack of detailed expressions. We discuss such problems in Section 6.

Head Angle Comparison and User Study. Fig. 12 (A) shows the ranges of head angles of 10K samples from each dataset. As can be seen, we determine the ranges of head poses in the scope of covering most samples of VoxCeleb. For purpose of quantitative comparison with iCartoon, we conduct a user study to compare the properties of datasets after see- ing 100 samples from each dataset. As shown in Fig. 12 (B), users positively evaluate the style consistency, quality[6] and cleanness[7] of AnimeCeleb. Also, the users respond that AnimeCeleb has a comparable diversity of head pose and expression.

[4] https://comic.naver.com/
[5] https://toonify.photos/
[6] A low-resolution or defocused image is considered as low-quality one.
[7] If a face is occluded with an object or incompletely cropped, then it is considered as a noisy image

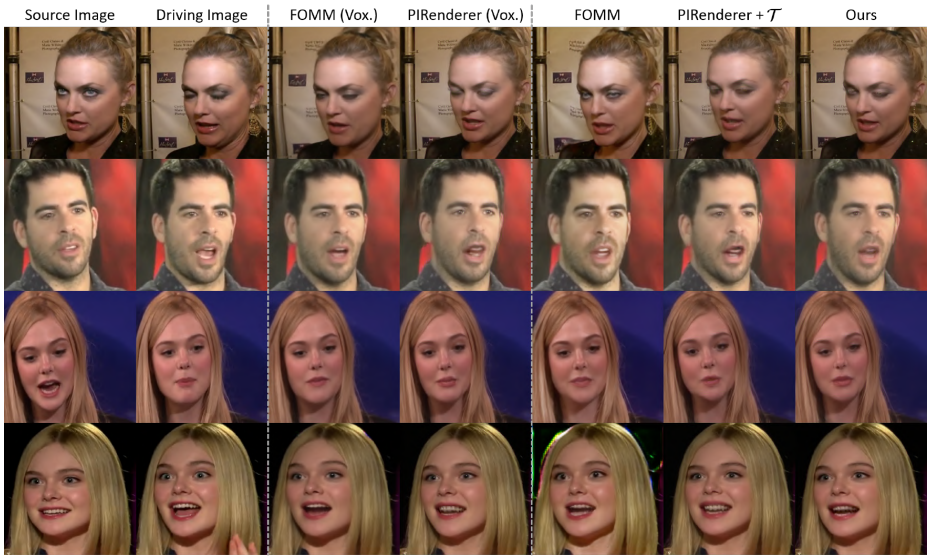

Fig. 13: Qualitative comparison between our model and the baselines on self-identity head reenactment given the images of the Voxceleb.

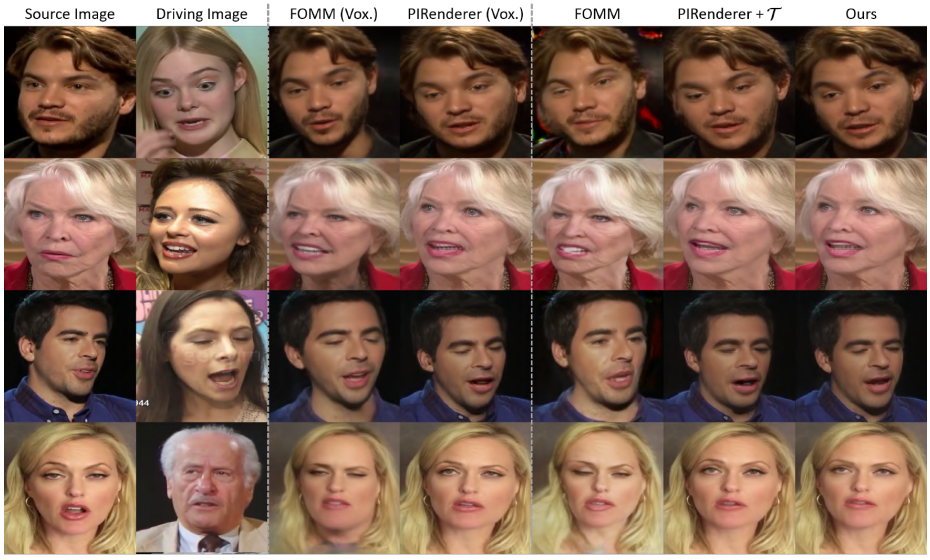

Fig. 14: Qualitative comparison between our model and the baselines on cross-identity head reenactment given the images of the Voxceleb.

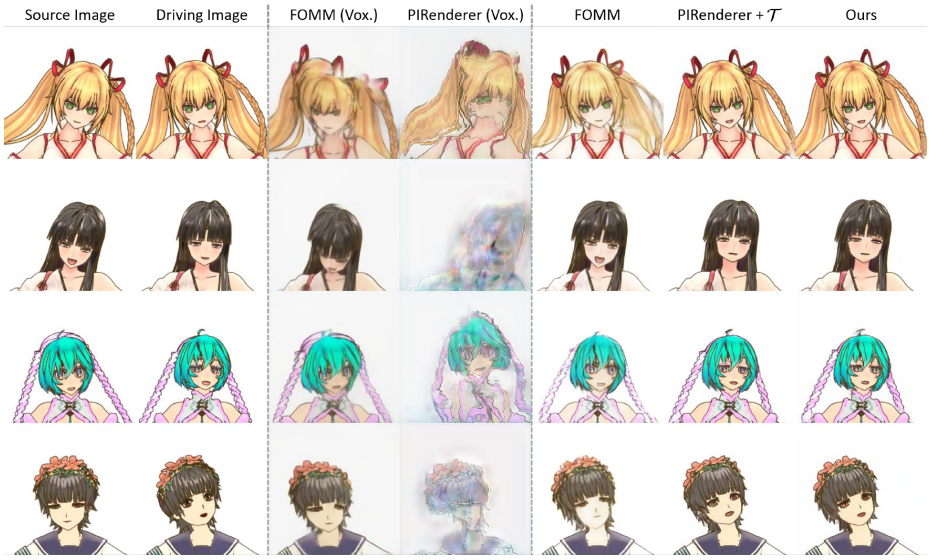

Fig. 15: Qualitative comparison between our model and the baselines on self-identity head reenactment given the images of the AnimeCeleb.

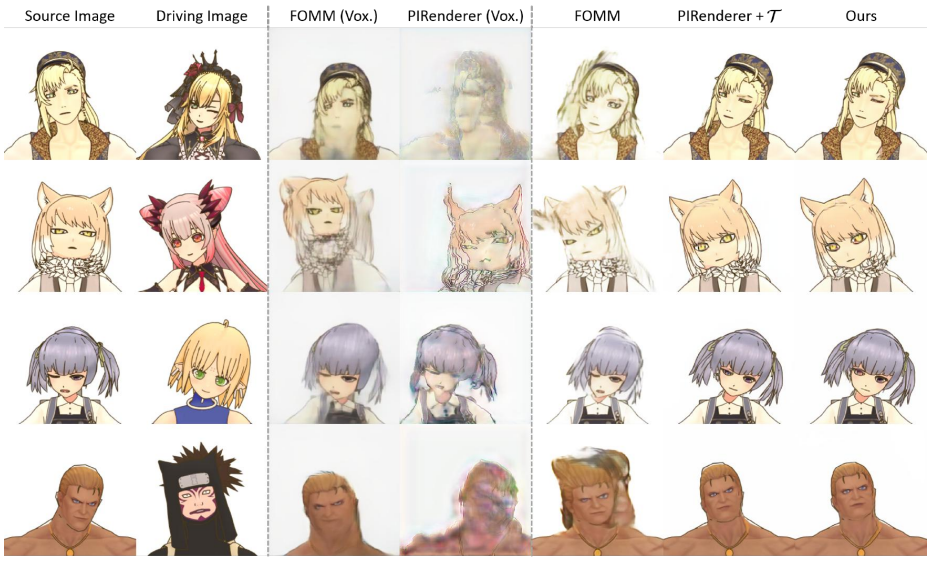

Fig. 16: Qualitative comparison between our model and the baselines on cross-identity head reenactment given the images of the AnimeCeleb.

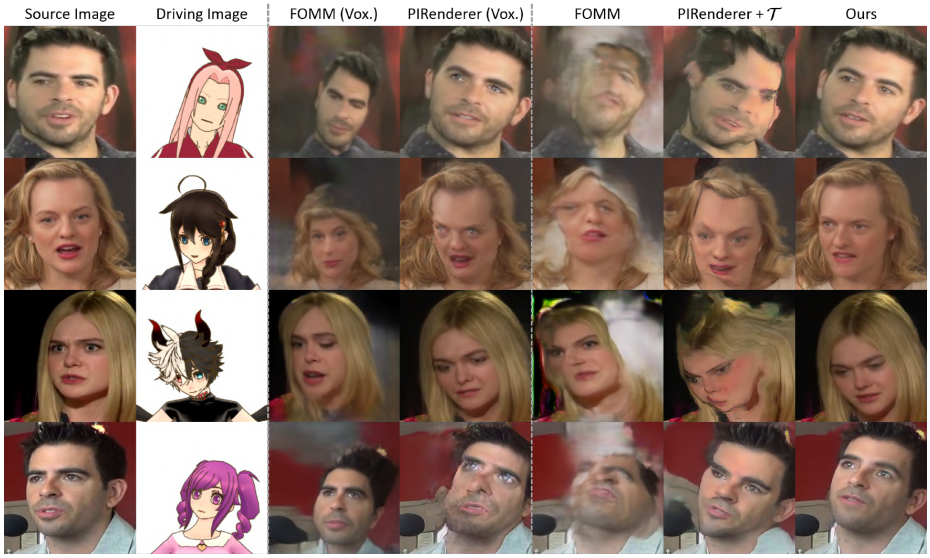

Fig. 17: Qualitative comparison between our model and the baselines on cross-domain head reenactment given the source image from the VoxCeleb and the driving image from the AnimeCeleb (Anime. → Vox.).

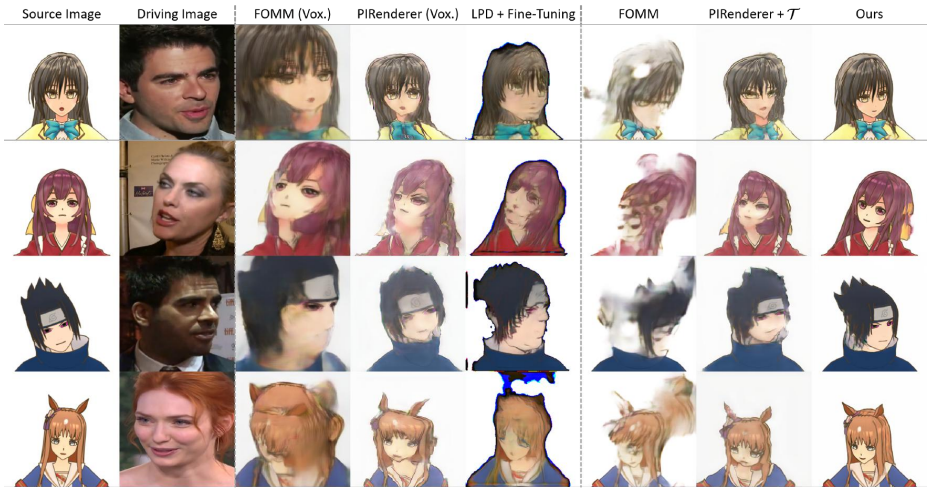

Fig. 18: Qualitative comparison between our model and the baselines on cross-domain head reenactment given the source image from of the AnimeCeleb and the driving image from the VoxCeleb (Vox. → Anime.).

Fig. 19: Intuitive image editing results on animation and human heads via controlling the semantics and the head angles.

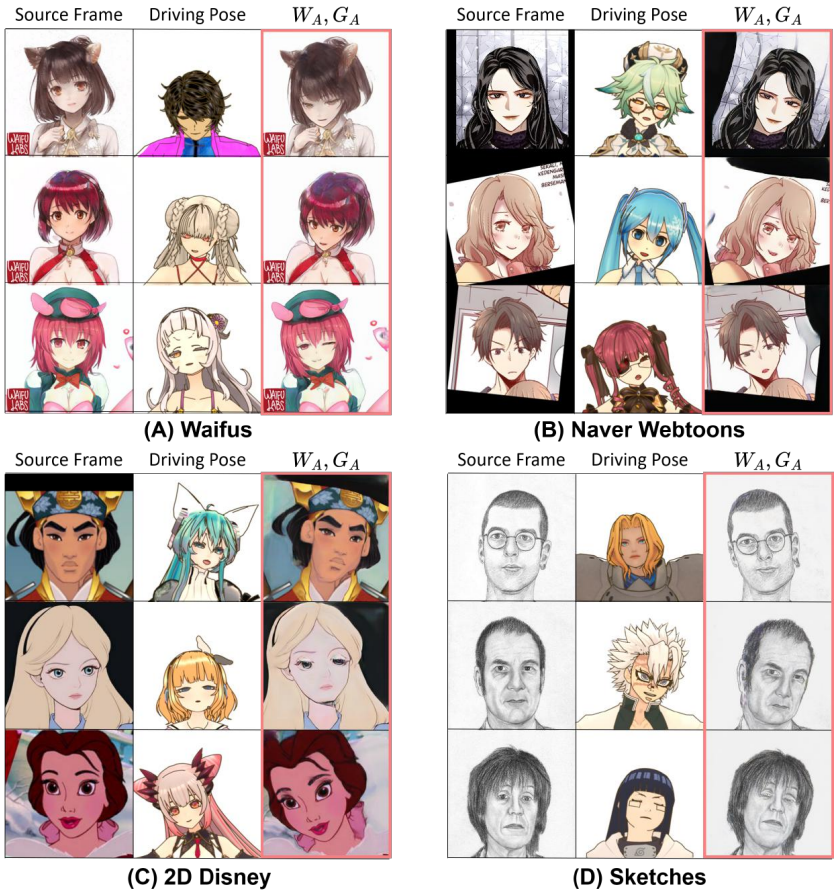

Fig. 20: Additional head reenactment results on head images from various animation head samples.

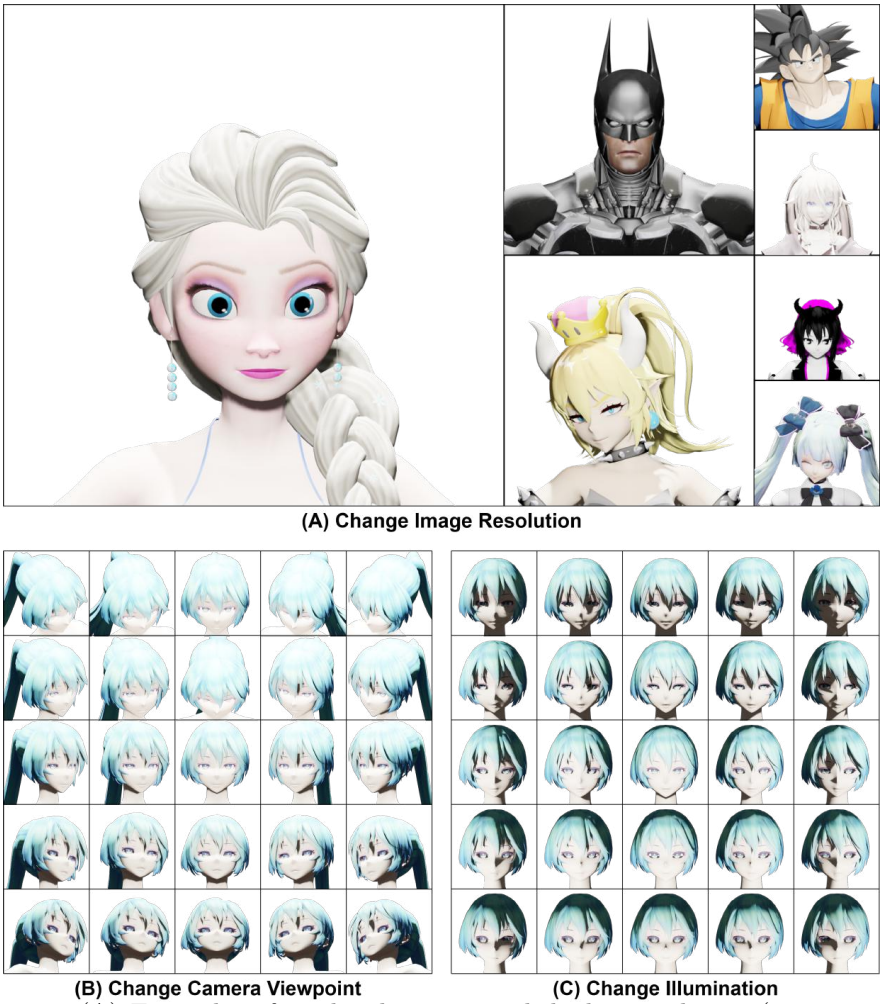

Fig. 21: (A) Examples of rendered images with higher resolution (*i.e.*, 1024 × 1024, 512 × 512, and 256 × 256) in order. (B) We generate additional examples under different camera viewpoints from spherical coordinate system where the neck bone is the origin, ranging azimuth [-40°, 40°] and elevation [-40°, 40°]. (C) Similar to (B) we render the images by relocating a light source position, ranging azimuth [-40°, 40°] and elevation [-40°, 40°] with setting the neck bone as the origin.

6 Discussions

In this section, we discuss potential issues and directions for improvement of the AnimeCeleb and the *AniMo* in further research.

Extension of Creation Protocol. Due to the limited budget, the proposed pipeline is designed to generate a group of multi-pose yet single-view animation head images with the limited poses. However, we believe that the AnimeCeleb

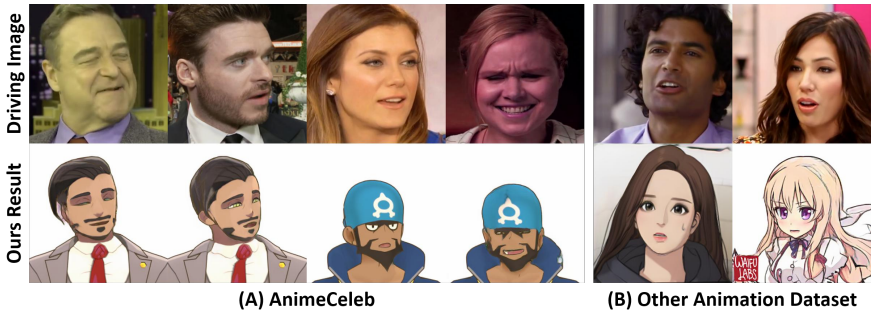

Fig. 22: Additional cross-domain head reenactment results on (A) AnimeCeleb and (B) other animation datasets.

has room for improvement in three aspects: (1) constructing high-quality images higher than 256×256, (2) obtaining multi-view animation head images by rotating the camera, and (3) building a various light-conditioned animation head dataset from changing the light source position. To prove these concepts, we present these samples in Fig. 21. As seen in in Fig. 21 (A), our data creation pipeline is able to render a higher resolution than 256×256 (*e.g.*, 1024×1024). This definitely allow us to construct a high-quality dataset in future research. Next, the images of AnimeCeleb are created based on the frontal face, and thus do not span comprehensive appearances that can be created at various camera angles. This is mainly due to the goal of the AnimeCeleb lies in constructing the public animation dataset, which is suitable for head reenactment. A straightforward method to improve our creation process is to render an animation head at different camera angles in Blender as shown in Fig. 21 (B). Also, as can be seen in Fig. 21 (C), we can control the illumination for the aim of generating animation head images under different light conditions.

Diversity of the AnimeCeleb. One of the AnimeCeleb strengths lies in a wide spanning of animation characters. However, we fixed the camera position with the aim of capturing frontal faces of animation characters during the AnimeCeleb generation process. Although this enables us to extract character face easily, the fixed camera position also constrained dataset diversity especially in terms of a translation. In addition, we uniformly set a background of the generated image as 0 (*i.e.*, white color). Obviously, this weakens the capacity of a head reenactment model trained with the AnimeCeleb when handling a center-unaligned or complicated-background animation head image. Our planned solution to these limitations is to develop a more flexible architecture that can consider translation parameters under this constraint.

Limitations of the *AniMo*. We have found that when using 3DMM parameters obtained from the VoxCeleb, the *AniMo* often fails to reflect the detailed poses (*e.g.*, eye or mouth pose). Indeed, there are successful examples as shown in Fig. 22, our finding is that region sizes of lip and eyes are important to generate diverse images; more dynamics are tend to be entailed when a lip or eyes

are noticeably large. On the other hand, this is not the case when we use 3DMM parameters acquired by our pose mapping method with a pose vector from the AnimeCeleb. We conclude that this behavior mainly stems from the fact that a pose from the VoxCeleb often does not identify the exact position of an eye or a mouth. In future work, we will address this problem by considering expression detail correctness of the outputs during training.

In addition, since the images of the AnimeCeleb are center-aligned and have no background, it is no surprise that there exists a performance degradation when an animation head image does not these conditions(*e.g.*, containing complicated background). To be specific, the generated outputs have an artifact at background and often loss the detailed poses (*e.g.*, eye or mouth pose). This behavior is also observed in previous studies [3, 18, 23] when a position of a given head in an image is far from the training dataset distribution. The solution to alleviate the problem by shifting a head position of a given source and driving image in the inference time [8]. Similar to these approaches, we plan to implement an additional preprocessing pipeline for an animation source image during the inference.

[8] https://github.com/shrubb/latent-pose-reenactment

References

1. Kaggle animation face. https://www.kaggle.com/splcher/animefacedataset
2. Branwen, G., Anonymous, Community, D.: Danbooru2019: A large-scale anime character illustration dataset. https://www.gwern.net/Crops (May 2020), https://www.gwern.net/Crops, accessed: DATE
3. Burkov, E., Pasechnik, I., Grigorev, A., Lempitsky, V.: Neural head reenactment with latent pose descriptors. In: Proc. of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 13786–13795 (2020)
4. Fujimoto, A., Ogawa, T., Yamamoto, K., Matsui, Y., Yamasaki, T., Aizawa, K.: Manga109 dataset and creation of metadata. In: Proceedings of the 1st international workshop on comics analysis, processing and understanding. pp. 1–5 (2016)
5. Ha, S., Kersner, M., Kim, B., Seo, S., Kim, D.: Marionette: Few-shot face reenactment preserving identity of unseen targets. In: Proc. the AAAI Conference on Artificial Intelligence (AAAI). vol. 34, pp. 10893–10900 (2020)
6. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision. pp. 1501–1510 (2017)
7. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proc. of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 1125–1134 (2017)
8. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Lee, J., Kim, E., Lee, Y., Kim, D., Chang, J., Choo, J.: Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In: Proc. of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 5801–5810 (2020)
11. Lee, J., Kim, E., Lee, Y., Kim, D., Chang, J., Choo, J.: Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In: Proc. of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 5801–5810 (2020)
12. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European conference on computer vision. pp. 483–499. Springer (2016)
13. Ojha, U., Li, Y., Lu, J., Efros, A.A., Lee, Y.J., Shechtman, E., Zhang, R.: Few-shot image generation via cross-domain correspondence. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10743–10752 (2021)
14. Qin, X., Zhou, Y., He, Z., Wang, Y., Tang, Z.: A faster r-cnn based method for comic characters face detection. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 1074–1080. IEEE (2017)
15. Ren, Y., Li, G., Chen, Y., Li, T.H., Liu, S.: Pirenderer: Controllable portrait image generation via semantic neural rendering. In: Proc. of the IEEE international conference on computer vision (ICCV). pp. 13759–13768 (2021)
16. Rios, E.A., Cheng, W.H., Lai, B.C.: Daf: re: A challenging, crowd-sourced, large-scale, long-tailed dataset for anime character recognition. arXiv preprint arXiv:2101.08674 (2021)

17. 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. pp. 234–241. Springer (2015)
18. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: First order motion model for image animation. Proc. the Advances in Neural Information Processing Systems (NeurIPS) **32**, 7137–7147 (2019)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
20. Takayama, K., Johan, H., Nishita, T.: Face detection and face recognition of cartoon characters using feature extraction. In: Image, Electronics and Visual Computing Workshop. p. 48 (2012)
21. Wang, T.C., Mallya, A., Liu, M.Y.: One-shot free-view neural talking-head synthesis for video conferencing. In: Proc. of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 10039–10049 (2021)
22. Zakharov, E., Ivakhnenko, A., Shysheya, A., Lempitsky, V.: Fast bi-layer neural synthesis of one-shot realistic head avatars. In: Proc. of the European Conference on Computer Vision (ECCV). pp. 524–540. Springer (2020)
23. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models. In: Proc. of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 9459–9468 (2019)
24. Zhang, B., Li, J., Wang, Y., Cui, Z., Xia, Y., Wang, C., Li, J., Huang, F.: Acfd: Asymmetric cartoon face detector. arXiv preprint arXiv:2007.00899 (2020)
25. Zhang, L., Wen, T., Shi, J.: Deep image blending. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 231–240 (2020)
26. Zheng, Y., Zhao, Y., Ren, M., Yan, H., Lu, X., Liu, J., Li, J.: Cartoon face recognition: A benchmark dataset. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2264–2272 (2020)