DeX-Portrait: Disentangled and Expressive Portrait Animation via Explicit and Latent Motion Representations

Supplementary Material

This document provides additional implementation details and experimental results. We include detailed implementation information for the Motion Trainer, Portrait Animator, dataset usage, and more results. We encourage the reader to examine our visual results for further insights.

A. Implementation Details

In this section, we elaborate on the architectures of Expression Conditioned Decoder and the Flatten & CNN block in the Motion Trainer (Fig. 2 (a)) and provide implementation details in the Diffusion-Based Portrait Animator (Fig. 2 (b)).

A.1. Motion Trainer Architecture

The 3D features (Fig. 2 (a)) are processed through two sequential modules and ultimately decoded into a 256-resolution image, as shown in Fig. S1 (a). In the Flatten & CNN block, we first concatenate the 3D feature and the warped 3D feature (both with spatial sizes $16 \times 64 \times 64$) along the channel dimension. This is followed by a ResBlock3D module [1] that compresses the spatial sizes to $16 \times 32 \times 32$ while reducing the channel count from 64 to 32. Finally, the features are transformed into 2D representations via the flatten module.

Next, we decode these 2D appearance features into an image using the Expression-Conditioned Decoder. The decoder architecture introduces two key enhancements compared to StyleGAN2 [3]:

- Intermediate Feature Injection: To maximize appearance preservation, the 2D features are injected into the intermediate layer (4th layer) of the decoder. The 3×3 StyleConv modules simultaneously perform upsampling and downsampling to capture multi-scale details.
- Progressive Channel Reduction: To better blend multi-resolution images, we modify the toRGB module (*toRGB) to gradually reduce the channel count from 96 to 3 during the upsampling process from 8×8 to 256×256.

Additionally, 3D features are first processed through a 5-layer CNN, then flattened into a 1D 2048-dimensional appearance latent code, which is concatenated with a 1D 512-dimensional expression latent code obtained from the expression encoder. The combined vector serves as a style code for subsequent StyleConv modules.

A.2. Portrait Animator Implementation Details

In the Portrait Animator, both our reference UNet and denoising UNet are based on the Stable Diffusion 1.5 architecture. We now detail the injection mechanisms for reference, head pose, and expression information in our implementation. Head pose and reference information are primarily injected through the spatial-attention module, as illustrated in the Fig. S1 (b). Unlike Animate Any One [2], which relies solely on cross-attention for injection, we further incorporate a residual connection to inject the warped appearance features. Specifically, we modify each BasicTransformerBlock in SD1.5. In the reference unet, the hidden state has a token count of h^2 and channel dimension c, which is reshaped into a 3D appearance feature of size $\frac{h}{2} \times h \times h$ with $\frac{2c}{h}$ channels. This feature is then warped using the source and target head pose $\mathbf{R}, \mathbf{t}, s$, passed through a convolutional layer, and finally injected into the denoising UNet via a residual connection. For expression latent injection, we follow the implementation of X-Nemo [5] by reshaping the 512-dimensional 1D expression latent into 32 tokens of 16-dimensional embeddings. These tokens are then injected into the denoising UNet through an additional cross-attention mechanism. A temporal attention module is further introduced at the final stage to enhance temporal coherence.

B. Training Strategies

B.1. Motion Trainer Losses

We train our motion trainer in a self-supervised manner on video datasets. Specifically, we select a source image $\mathbf{I}_\text{s}$ and a driving image $\mathbf{I}_\text{d}$ (serving as ground truth) from the video dataset and supervise the predicted image $\hat{\mathbf{I}}$ using a combination of multiple losses:

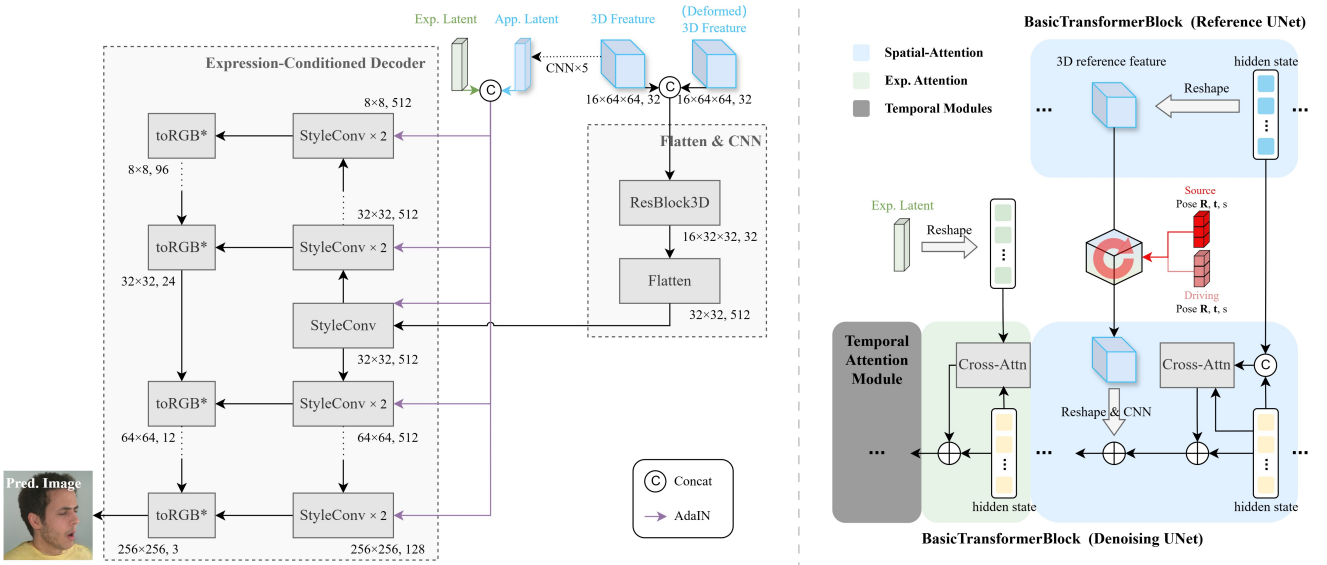
- Reconstruction Loss. We minimize pixel-wise color differences with:

$$\mathcal{L}_1 = \|\mathbf{I}_\text{d} - \hat{\mathbf{I}}\|_1 \tag{1}$$

- VGG16-based LPIPS Loss [4]. To enhance perceptual realism and sharpness, we introduce:

$$\mathcal{L}_{lpips} = \sum_{i=1}^{N} \|VGG^i(\mathbf{I}_\text{d}) - VGG^i(\hat{\mathbf{I}})\|^2 \tag{2}$$

where N denotes the number of feature layers in each respective pre-trained VGG model.



(a) Motion Trainer Decoders Architecture

(b) Portrait Animator Details

Figure S1. Detailed architecture of DeX-Portrait.

- Component LPIPS Loss for Facial Regions. A face-aware loss focusing on facial details:

$$\mathcal{L}_{clpips} = \sum_{i=1}^{N} \|M \odot (VGG^i(\mathbf{I}_\mathrm{d}) - VGG^i(\hat{\mathbf{I}}))\|^2 \quad (3)$$

where $\boldsymbol{M}$ is the facial segmentation mask.

- StyleGAN2-based Discriminator Loss. We employ a co-trained StyleGAN2 discriminator D to calculate adversarial loss:

$$\mathcal{L}_{adv} = Softplus(-D(\hat{\mathbf{I}})) \quad (4)$$

Finally, the total loss is combined as:

$$\mathcal{L}_{total} = \lambda_r \mathcal{L}_1 + \lambda_{lpips}\mathcal{L}_{lpips} + \lambda_{clpips}\mathcal{L}_{clpips} + \mathcal{L}_{adv} \quad (5)$$

where $\lambda_r = 10$, $\lambda_{lpips} = 1$, $\lambda_{clpips} = 100$.

B.2. Dataset Utilization

During the Motion Training and the Diffusion Training stages, beyond the previously mentioned pose and expression augmentation strategies, we further employ cross-view training for the multi-view datasets NeRSemble and ava-256. Specifically, within each training batch, one source frame is paired with four driving frames, which may collected from different camera viewpoints of the same subject. This training paradigm artificially simulates extreme head pose variations, thereby expanding the distribution range of our training data. For temporal module training, each batch consists of one source frame and 24 consecutive driving frames, with video sequences containing significant background motion being dynamically filtered out through real-time computation.

C. More Results

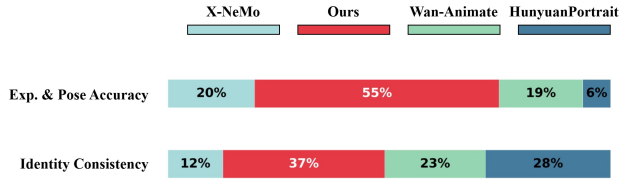


Figure S2. We collect 632 user study responses, and compare the performance of our method against four SOTA methods in terms of identity consistency, expression and pose accuracy. This figure reports the percentage of users who ranked each model as the best across the two tests.

For the cross-reenactment scenario, we additionally present comprehensive comparisons against state-of-the-art (SOTA) methods on benchmark datasets to demonstrate the superiority of our framework in identity preservation and precision control of head pose and expression (Fig. S3 and Fig. S2).

We also present visual demonstrations of two key applications enabled by our framework: expression-only editing and pose-only editing. As illustrated in Fig. S5, expression-only editing selectively modifies pixel values within local facial regions surrounding the target face. Owing to the high precision of our framework, the edited content can be seamlessly paste back into the original facial area without visible artifacts. Fig. S4 demonstrates pose-only editing capabilities, which enable users to reorient head poses in portrait images while strictly preserving the original facial expres-

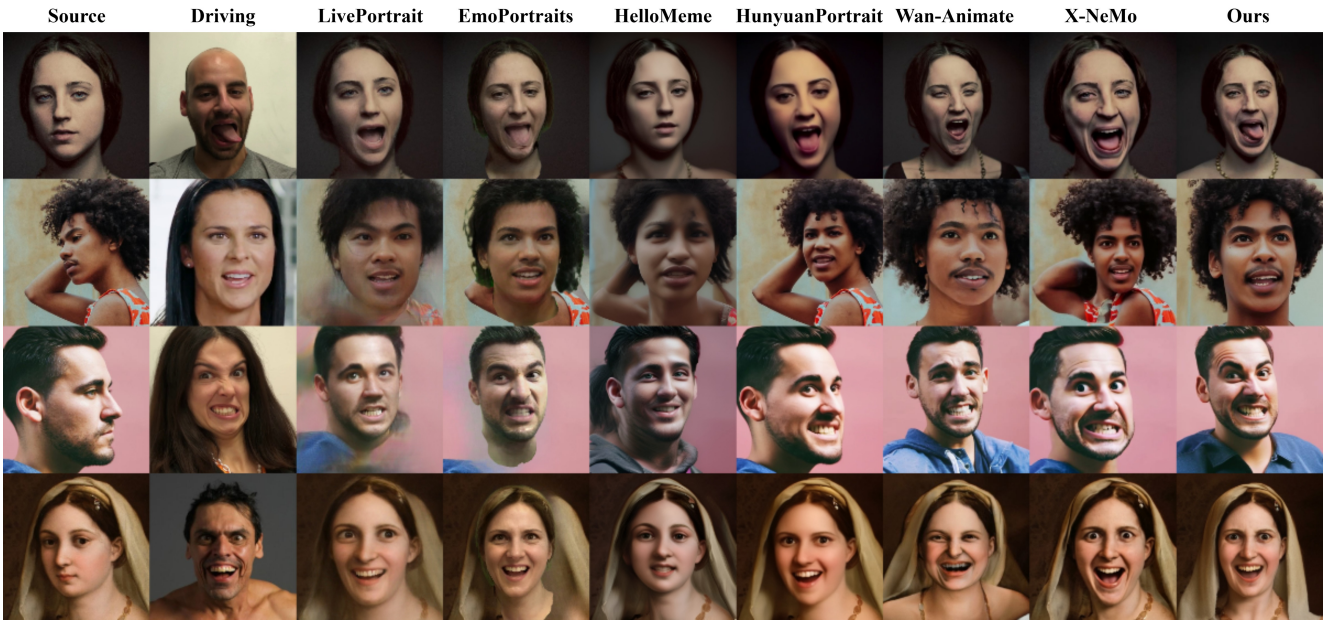


Figure S3. Qualitative comparison on cross-reenactment.

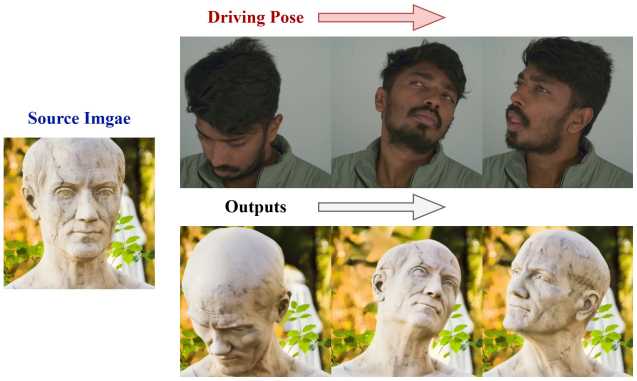


Figure S4. Visual results of pose-only editing.

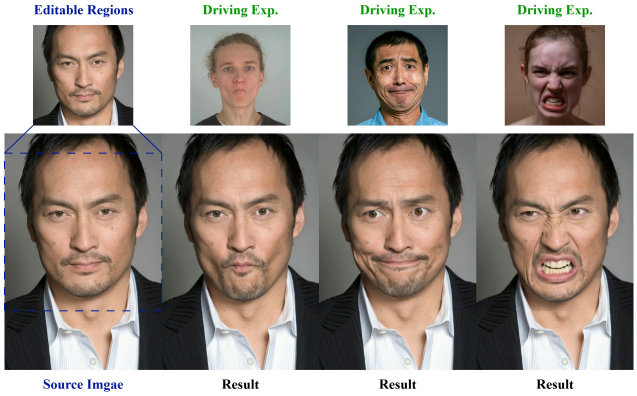


Figure S5. Visual results of expression-only editing.

sion. This disentangled control over pose and expression spaces validates the effectiveness of our architecture.

References

[1] Jianzhu Guo, Dingyun Zhang, Xiaoqiang Liu, Zhizhou Zhong, Yuan Zhang, Pengfei Wan, and Di Zhang. Liveportrait: Efficient portrait animation with stitching and retargeting control. *arXiv preprint arXiv:2407.03168*, 2024. 1

[2] Li Hu. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8153–8163, 2024. 1

[3] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 1

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[5] Xiaochen Zhao, Hongyi Xu, Guoxian Song, You Xie, Chenxu Zhang, Xiu Li, Linjie Luo, Jinli Suo, and Yebin Liu. X-nemo: Expressive neural motion reenactment via disentangled latent attention. *arXiv preprint arXiv:2507.23143*, 2025. 1