DR^2: Disentangled Recurrent Representation Learning for Data-efficient Speech Video Synthesis –Supplementary Material–

Chenxu Zhang[1], Chao Wang[1], Yifan Zhao[2], Shuo Cheng[3], Linjie Luo[1], Xiaohu Guo[4]
[1]ByteDance Inc [2]Peking University
[3]Georgia Institute of Technology [4]The University of Texas at Dallas
{chenxuzhang, chao.wang, linjie.luo}@bytedance.com, zhaoyf@pku.edu.cn,
shuocheng@gatech.edu, xguo@utdallas.edu

Here we elaborate more details of the network architecture and training details in Section 1, the dataset collection and preprocessing in Section 2, limitations and social impact in Section 3.

1. Audio2gestures Network Details

1.1. Network Architecture

For the audio2gestures module, we employ two encoder sub-networks (*i.e.*, audio and pose) and three decoder sub-networks (*i.e.*, face, body, and hand).

Audio Encoder: The audio encoder network is used to extract the audio features, combined with the information of the first frame to generate the feature vectors for 128 frames. Tab. 1 shows our encoder network structure. We employ the standard 4-layer U-Net structure [7] as the backbone for audio feature extraction. Each convolution layer in our encoder network is followed by a LeakyReLU activation and batch normalization [4].

Pose Encoder: The purpose of using a pose encoder is to extract the feature of the initial pose and alleviate overfitting to the single initial pose condition. After being extracted by the pose encoder, the pose feature is used to guide the network generation for the start pose and the general appearance of the next sequence. The input of the pose encoder is $\mathbf{P}^G_{x,1}$ or $\mathbf{P}^G_{y,1} \in \mathbb{R}^{59}$. It is mapped to a 32-dimensional feature space through a multilayer perceptron (MLP) shown in Tab. 2, where each linear layer is followed by a LeakyReLU activation.

Face, Body, and Hand Decoders: In our decoder, we use three MLP modules to generate the corresponding face, body, and hands parameters, respectively. The structure of decoders is shown in Tab. 3, whose inputs are the feature vectors extracted from the encoder network, and the output is face, body, or hands parameters in 128 frames. Our body and hand decoders share a similar structure, with the only difference in the output sizes. Each convolution layer in our decoder network is followed by a LeakyReLU activation.

Table 1. Detailed network architecture of audio encoder.

Type	Kernel	Stride	Output
DeepSpeech	-	-	$1 \times 128 \times 29$
Conv 2D	4×4	2×2	$32 \times 64 \times 13$
Conv 2D	4×4	2×2	$128 \times 32 \times 5$
Conv 2D	4×4	1×1	$256 \times 32 \times 2$
Conv 2D	3×1	1×1	$256 \times 32 \times 2$
Reshape	-	-	256×64
Interpolation	Bilinear	-	256×128
U-Net	-	-	256×128
Concat **p**	-	-	288×128

Table 2. Detailed network architecture of pose encoder.

Type	Operation	Output
Initial pose		$B \times 59$
Linear	FC (59, 128)	$B \times 128$
LeakyReLU	(0.2)	$B \times 128$
Linear	FC (128, 128)	$B \times 128$
LeakyReLU	(0.2)	$B \times 128$
Linear	FC (128, 32)	$B \times 32$

1.2. Training Details

Human Body Translation: If the training subject is in sitting gestures, we only generate face, body, and hands parameters during training, as mentioned in the main manuscript. For standing gestures, additional translation

Table 3. Detailed network architecture for decoders.

Face Decoder:			
Type	Kernel	Stride	Output
Concat **s**	-	-	288×128
Conv 1D	3	1	256×128
Conv 1D	3	1	256×128
Conv 1D	3	1	10×128
Body Decoder:			
Type	Kernel	Stride	Output
Concat **s**	-	-	288×128
Conv 1D	3	1	256×128
Conv 1D	3	1	256×128
Conv 1D	3	1	35×128
Hand Decoder:			
Type	Kernel	Stride	Output
Concat **s**	-	-	288×128
Conv 1D	3	1	256×128
Conv 1D	3	1	256×128
Conv 1D	3	1	24×128

parameters are required for our body pose decoder, so our network will generate three translation values to represent the position of the current gesture. To ensure the continuity of our recurrent generation, we also add this translation information to the initial pose state.

Noise Addition: In order to enhance the gestures guided by the initial state to be more general, we introduce random Gaussian noise to the initial state for loose constraints. To generate a smooth sequence without any sudden change, we need to add noise to the ground truth pose sequences at the same time. Therefore, we use a sequence-level noise addition method to ensure that 1) the start pose in the pose sequence is the same as the initial pose; and 2) the whole pose sequence is highly consistent and reasonable.

For frame t in video $\mathbf{V}_x$, we use $\hat{\mathbf{F}}_{x,t} \in \mathbb{R}^{10}$ and $\hat{\mathbf{P}}_{x,t} \in \mathbb{R}^{59}$ to represent the predicted facial expressions and pose gestures respectively. Random noise $\xi \sim \mathcal{N}(0, \sigma) \in \mathbb{R}^{59}$ is applied to both $\mathbf{P}_x^G$ and $\mathbf{P}_y^G$ to enhance the diversity of generated sequences. During training, we add noise ξ to the target sequence (*e.g.*$\mathbf{P}_x^G$) in a smooth manner:

$$\mathbf{P}_{x,t}^G \leftarrow \mathbf{P}_{x,t}^G + \max(0, 1 - t/m) \times \xi, \tag{1}$$

where the modification range m is set to 10 empirically. This method is also applied to unpaired sequences $\mathbf{P}_y^G$.

Inference Smooth: In the inference phase, since we have a recurrent scheme, the start pose $\hat{\mathbf{P}}_{x,1}$ of the current sequence $\hat{\mathbf{P}}_x$ will be very close to the end pose $\hat{\mathbf{P}}_{x-1,end}$ of the previous sequence $\hat{\mathbf{P}}_{x-1}$. However, there is no guarantee that the two adjacent poses are precisely the same.

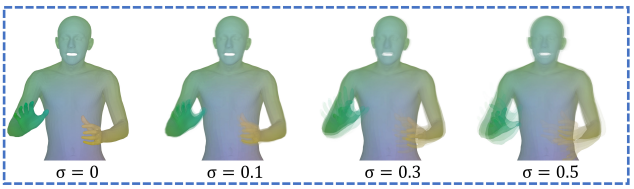

Figure 1. Visualization of different random noise hyper-parameters. Random noise is adopted for enlarging the training space of the initial pose, while larger noise values would reduce the continuity between sequences.

Therefore, we apply smooth optimization as follows:

$$\hat{\mathbf{P}}_{x,t} \leftarrow \hat{\mathbf{P}}_{x,t} + \max(0, 1 - t/k) \times \varepsilon. \tag{2}$$

where $\varepsilon = \hat{\mathbf{P}}_{x-1,end} - \hat{\mathbf{P}}_{x,1}$ and the smooth range k is set to 10 emprically.

Random noise study: Due to our extremely short training data, using the initial pose as a gesture template is prone to overfitting. Thus we use random noises $\xi \sim \mathcal{N}(0, \sigma)$ to enlarge the training space of the initial pose in Fig. 1, which further enhances the diversity of our generated results. However, it will affect the continuity between sequences. Considering the trade-off between diversity and continuity, we empirically set the range of $\sigma \in [0.1, 0.3]$.

2. Dataset

2.1. Data Preprocessing

Audio preprocessing. To convert audio signal into audio features as the network input, we use the off-the-shelf feature extractor DeepSpeech to extract the corresponding audio features. We re-sample the output audio features with linear interpolation to ensure a sampling rate of 30 FPS, corresponding to the frame rate of our videos.

3D human model. For each frame in the video, we fit a human model by the algorithm SMPLify-X [6]. However, SMPLify-X is an image-based optimization method, leading to unnatural movements for our video fitting procedure. Therefore, for human model fitting in the speech video scenario, we make the following modifications: (1) fix the body shape and global orientation for stabilization; (2) add an inter-frame motion loss to regularize the movement between two adjacent frames in a proper range; (3) apply the previous frame fitting result as the initial state for the current frame fitting to reduce fitting ambiguity.

2.2. 3D Human Model Fitting Details

As mentioned in the main manuscript, we use SMPL-X model [3, 5, 6] to represent the speech state of each person. Our 3D human model-fitting method is the modified SMPLify-X algorithm [6]. Since our task is human model

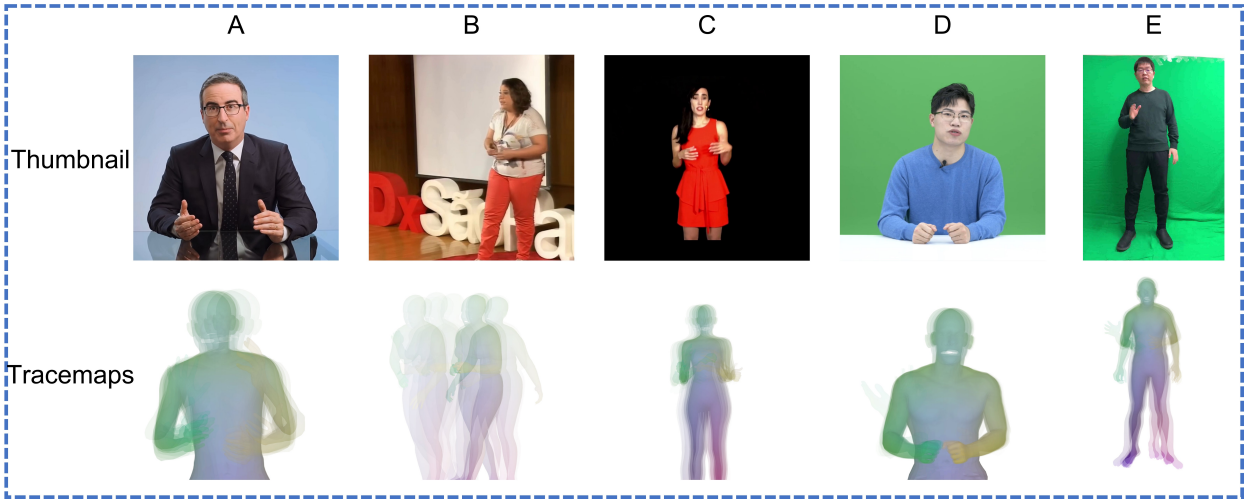

Figure 2. Illustration of the test videos in our experiments. Here we exhibit a representative video frame for each speaker, where sequences A, B, and C are videos collected online, and sequences D and E are our self-captured videos, both containing rich gestures such as sitting and standing. Below each person are trace maps of the human model that are tracked from different keyframes, which indicate the motion range of each person and their common gestures.

fitting in the speech video scenario, we make the following modifications: (1) fix the body shape and global orientation for stabilization; (2) add an inter-frame motion loss to regularize the movement between two adjacent frames in a proper range; (3) apply the previous frame fitting result as the initial state for the current frame fitting to reduce possible ambiguities.

- We first fit the initial frame following the SMPLify-X algorithm [6]. Then the body shape and global orientation are fixed during the fitting of other frames. Specifically, we set the initial values of these parameters and cut off the gradient during the optimization process.

- The movements of human bodies and hands in our video are continuous. However, the OpenPose [1] results are usually inaccurate or invalid. And the original SMPLify-X algorithm will produce severe jitter, which is not in line with real body movements. Therefore, in addition to the original fitting loss $\mathcal{L}_{\text{SMPLify-X}}$, we propose to introduce body pose motion loss $\mathcal{L}_{\text{body}}$, hand pose motion losses $\mathcal{L}_{\text{handl}}$ and $\mathcal{L}_{\text{handr}}$ to regularize the movements between two adjacent frames in a proper range. The motion loss function $\mathcal{L}_{\text{Fit}}$ is defined as:

$$\begin{aligned}
\mathcal{L}_{\text{Fit}} &= \mathcal{L}_{\text{SMPLify-X}} + \omega\mathcal{L}_{\text{body}} + \omega\mathcal{L}_{\text{handl}} + \omega\mathcal{L}_{\text{handr}}, \\
\mathcal{L}_{\text{body}} &= ||b - \hat{b}||_2^2, \\
\mathcal{L}_{\text{handl}} &= ||h_l - \hat{h}_l||_2^2, \\
\mathcal{L}_{\text{handr}} &= ||h_r - \hat{h}_r||_2^2,
\end{aligned} \tag{3}$$

  where b, h_l, and h_r indicate the parameters of body pose, left-hand pose, and right-hand pose in the current frame, respectively. $\hat{b}$, $\hat{h}_l$, and $\hat{h}_r$ indicate the parameters of body pose, left-hand pose, and right-hand pose in the previous frame, respectively. ω is set to $1e2$ empirically.

- SMPLify-X algorithm is adopted to fit the 3D human model from 2D key points. However, this fitting process is an ill-posed problem because the same 2D key points may correspond to different 3D gestures. Therefore, we can adapt the parameter values of the previous frame, which not only enhances the smoothness between video frames but also provides prior knowledge when fitting the current frame. At the beginning of fitting each frame (except the first frame), we use the parameter values of the previous frame as the initial value of the current fitting. In this way, not only is the fitting time shortened but the fitting errors can also be reduced.

Experimental results show that our fitting method outperforms the SMPLify-X algorithm. For example, when the detection result of OpenPose is invalid (*e.g.*, Fig. 3 (a)), our method can also generate reasonable fitting results (Fig. 3 (c)) based on the information constraints from the previous frame.

2.3. Dataset Subjects

For dataset construction, we collect videos from 5 subjects, spanning over various gestures, including sitting,

Figure 3. Comparison of fitting results between SMPLify-X and our method. While SMPLify-X occasionally fails when encountering fast motion, our proposed modifications can still achieve stable results.

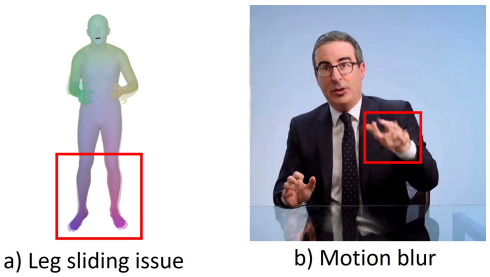

Figure 4. Failure cases with leg sliding issue and motion blur. (a) When the individual remains stationary, subtle translations are observed in the foot and leg regions. (b) Motion-induced blurriness is evident in the moving hand.

standing, and moving. Here we visualize representative cases used in our experimental comparisons (shown in Fig. 2). Below each person, we also exhibit the randomly sampled motion traces of the corresponding person, from which we can see the motion range of each person and their common gestures.

3. Limitation and Social Impact

3.1. Limitations

Leg sliding issue. Directly constraining the SMPLX parameters has led to the problem of leg sliding in Fig. 4(a). Such constraints fail to ensure a robust connection between the legs and the ground, resulting in observable leg translational movements. A potential remedy is to enhance our model by incorporating constraints based on foot keypoints. Nonetheless, given our scant training dataset, this adaptation is notably challenging.

Motion-induced blur. As depicted in Fig. 4(b), specific regions (highlighted in red) demonstrate pronounced motion blur, even when the rest of the body exhibits minimal movement and maintains clarity. This blurring can be attributed to two primary reasons: a) the paucity of our training data, and b) the inherent limitations of the vid2vid [2] rendering technique in managing localized blurring. A prospective solution could involve embedding a deblur module during the rendering phase, utilizing poses to pinpoint and mitigate regions prone to motion blur.

3.2. Social Impact

Our method could generate realistic videos with diverse gestures. This work can positively inspire future computer vision and deep learning research in many application areas, including virtual human creation, cross-modal animation, and co-speech gesture generation. However, we should also consider the misuse of this work since the source code will be publicly available. Therefore, we require all videos generated using our method can only be used for academic research and be marked as generated. The proper use of this technology will foster positive social development. In addition, we hope that the videos generated by our method can serve as training data to help improve the development of full-body fake video detection.

References

[1] 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*, 2019. 3

[2] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *ICCV*, 2019. 4

[3] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Monocular expressive body regression through body-driven attention. In *ECCV*, 2020. 2

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 1

[5] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM TOG*, 2015. 2

[6] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, 2019. 2, 3

[7] 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*, 2015. 1