Audio-driven Neural Gesture Reenactment with Video Motion Graphs - Supplementary Material -

Yang Zhou[1,2] Jimei Yang[2] Dingzeyu Li[2] Jun Saito[2] Deepali Aneja[2] Evangelos Kalogerakis[1]
[1]University of Massachusetts Amherst [2]Adobe Research

Implementation and demo videos. Our implementation and demo videos can be found at our project page.

Training details. We train the entire network end-to-end with losses promoting better flow estimation and final frame reconstruction. Specifically, we first have an L1 reconstruction loss L_{rec} and a perceptual loss L_{per} between the synthesized image $\hat{I}_t$ and I_t:

$$L_{rec} = \mathcal{L}_1(I_t, \hat{I}_t) \tag{1}$$

$$L_{per} = \mathcal{L}_1(\phi(I_t), \phi(\hat{I}_t)) \tag{2}$$

where $\phi(\cdot)$ concatenates feature map activations from a pre-trained VGG19 network [6].

We then adopt another L1 reconstruction loss $\mathcal{L}^b_{rec}$ promoting better frame reconstruction directly from the warped deep features x''_i and x''_j after these pass through our generator network G. This helped predict warped deep features such that they lead to generating frames as close as possible to ground-truth in the first place. We also empirically observed faster convergence with this loss:

$$L^b_{rec} = \mathcal{L}_1(I_t, G(x''_i)) + \mathcal{L}_1(I_t, G(x''_j)) \tag{3}$$

Further, we have warping loss L^m_{warp} and L^o_{warp} by measuring the L1 reconstruction error between the target image and the source images I_i and I_j after being warped through the motion field $F^m_{t\to i}$ (Equations 2 and 3 in the main paper) and also the optical flow $F^o_{t\to i}$:

$$\begin{aligned} L^m_{warp} = \mathcal{L}_1(I_t, \mathcal{W}(I_i, F^m_{t\to i}))+ \\ \mathcal{L}_1(I_t, \mathcal{W}(I_j, F^m_{t\to j})) \end{aligned} \tag{4}$$

$$\begin{aligned} L^o_{warp} = \mathcal{L}_1(I_t, \mathcal{W}(\mathcal{W}(I_i, F^m_{t\to i}), F^o_{t\to i}))+ \\ \mathcal{L}_1(I_t, \mathcal{W}(\mathcal{W}(I_j, F^m_{t\to j}), F^o_{t\to j})) \end{aligned} \tag{5}$$

where $\mathcal{W}(I, F)$ applies backward warping flow F on image I.

Finally, we follow [2] and include a smoothness loss for both mesh flow and optical flow:

Category	Keywords
greeting	hey, hi, hello
counting	one, two, three, first, second, third
direction	east, west, north, south, back, front, away, here, around
sentiment	crazy, incredible, surprising, screaming
action	walk, drive, ride, enter, open, attach, take, move
relative	more, less, much, few
others	called

Table 1. Dictionary of common keywords.

$$L_{sm} = ||\nabla F^m_{t\to i}||_1 + ||\nabla F^m_{t\to j}||_1 + \tag{6}$$

$$||\nabla F^o_{t\to i}||_1 + ||\nabla F^o_{t\to j}||_1 \tag{7}$$

The overall loss $\mathcal{L}$ is defined as the weighted sum of all losses described above, then averaged over all training frames.

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{rec} + \lambda_p \mathcal{L}_{per} + \lambda_b \mathcal{L}^b_{rec} + \\ \lambda_m \mathcal{L}^m_{warp} + \lambda_o \mathcal{L}^o_{warp} + \lambda_s \mathcal{L}_{sm} \end{aligned} \tag{8}$$

The weights have been set empirically based on [2] as $\lambda_p = 0.01,\ \lambda_b = 0.25,\ \lambda_m = 0.25,\ \lambda_o = 0.25, \lambda_s = 0.01$.

To train the entire model, we first train the mesh flow estimator network with L^m_{warp} as a "warming" stage. Then we load a pre-trained optical flow model from [2]. Finally, we train the entire network end-to-end with the loss mentioned above. The network weights are optimized with Adam optimizer using PyTorch. The learning rate is set to 10^{-4} and weight decay to 10^{-6}. The training process is performed on 4 Nvidia GeForce 1080Ti GPUs.

We show the detailed training procedure for our numerical evaluation. For the Personal story dataset, we train the model on each individual speaker video and report the evaluation numbers accordingly. The compared methods are trained on each speaker video for a fair comparison. For the TED-talks dataset, we train a single model on the entire

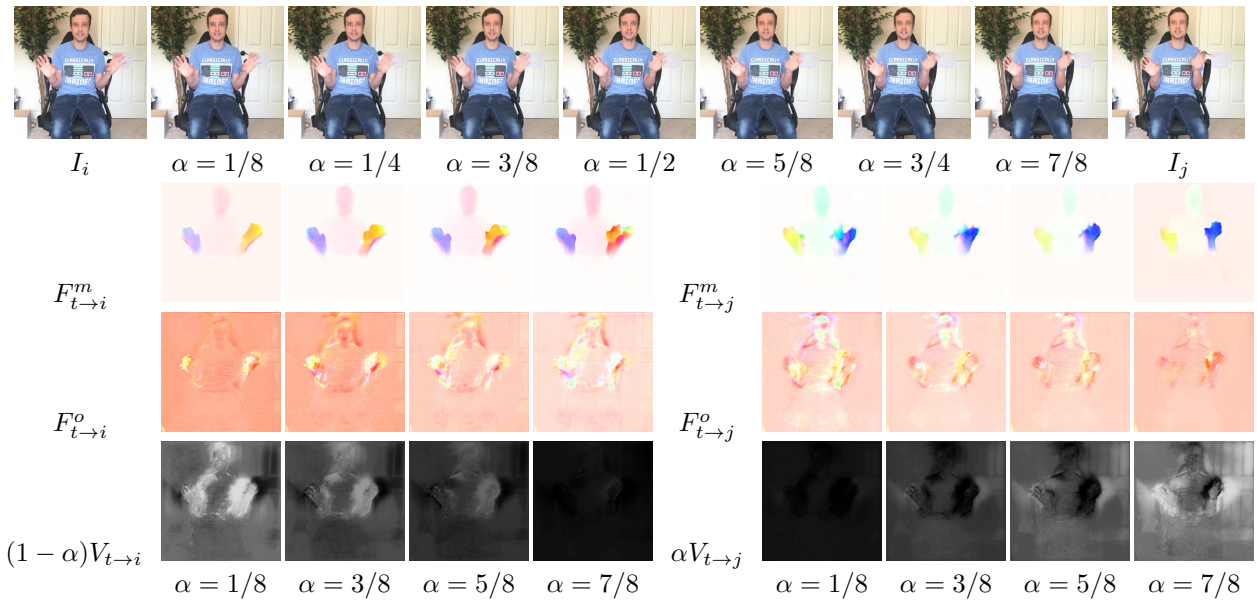

Figure 1. Pose-aware video blending results for target blending weights $\alpha \in (0, 1)$. **Top row**: synthesized in-between frames with blended human gestures for different blending weights. **Bottom rows**: intermediate mesh flows, optical flows and visibility maps results for corresponding blending weights.

training split of the dataset. We evaluate our model generalization on the testing split which contains unseen speakers. The comparison methods are also trained on multiple speakers on this dataset for a fair comparison.

The TED-Talks dataset proposed by [5] contains a list of Youtube video URL links, corresponding frame indices, and cropped areas with auto-detected upper bodies of speakers inside. They are not directly helpful to create the audio-driven reenacted video results. This is because 1) the original dataset only contains very short video clips, e.g., with a duration of a few seconds, which are not sufficient to create rich video motion graphs; 2) the frames in the original dataset are processed to 384×384 resolution by cropping and scaling the upper body of the speaker from a zoomed-out full body frame. As a result, the frames do not have high resolution and high quality. In this case, we use such dataset for numerical evaluation and easier reproduction purpose. To achieve high resolution and high quality audio-driven reenacted TED-talks videos shown on our project page, we use the original full Youtube videos. We manually select the frames with the zoomed-in camera where the upper body of the speaker appears at high resolution and high quality (see examples in our HTML files). The selected frames have sufficient length to create reasonable video motion graphs. Finally we fine-tune the model on these frames from each specific speaker and generate reenacted video results given test audios.

Pose-aware video blending network results. Fig. 1 shows output images from the video blending network for different blending weights, along with results from our intermediate stages.

Dictionary of common keywords. Referential gestures, especially iconic and metaphoric gestures, have strong correlations with the transcript [3, 7]. They usually appear together with certain keywords, such as action verbs, concrete objects, abstract concepts, and relative quantities to co-express the speech content [1]. We gather a few frequently used such keywords co-occurring with referential gestures in our speaker videos, as shown in Table. 1.

Network architecture details. The spatial encoder network E_s takes as input the RGB image I_i, the foreground mask I_{mask}, and an image containing the rendered skeleton I_{skel} representing the SMPL pose parameters. Fig. 2 shows an example of these input images.

We show our *Spatial Encoder* network structure for generating the mesh flow warping field in Table 2. In this table, the left column indicates the spatial resolution of the feature map output. The *ResBlock down* block is a 2-strided convolutional layer with a 3×3 kernel followed by two residual blocks. The *ResBlock up* block is a nearest-neighbor upsampling with a scale of 2, followed by a a 3×3 convolutional layer and then two residual blocks. The term *Skip* means skip connection that concatenates the feature maps of an encoding layer and decoding layer with the same spatial resolution. For Personal story dataset, the input and gener-

ated images are in 512×512 resolution, while for TED-talks dataset, the image resolution is 384×384.

The *Mesh Flow Estimator* and *Image Generator* network follows the structure of the *Spatial Encoder* network (see Table 2), but the input and output number of channels are different. For the *Mesh Flow Estimator* network, the number of input feature channel is 13 and output feature channel is 2. For the *Image Generator* network, the number of input feature channel is 19 and output feature channel is 3. Besides, the *Image Generator* network uses in the end a $tanh(\cdot)$ activation to regularize the image values between $[0, 1]$.

Audio-driven Beam search details. We initialize a beam search [4] procedure in the video motion graph to find K plausible paths matching the target speech audio segments. We set K to 20. The beam search initializes K paths starting with K random nodes as the first frame a_0 for the target audio, then expands in a breadth-first-search manner to find paths ending at a *target graph node* whose audio feature matches the target audio feature at the endpoint of the first segment a_1, associated with either an activated audio onset or the same non-empty keyword feature. Note that there can be multiple target graph nodes sharing the same audio feature with a_1.

During the beam search, all the explored paths are sorted based on a *path transition cost*, plus a *path duration cost*. The path transition cost is defined as the sum of node distances between all consecutive nodes m, n along the path, i.e. $\sum_{m,n}(d_{feat}(m, n) + d_{img}(m, n))$. The cost of synthetic transitions are always higher than natural ones. Thus, the path cost prevents using implausible paths with too many synthetic transitions.

When a path reaches a *target* graph node, we check its duration. Due to the sparsity of the graph, there may not be any path matching exactly the target audio segment length L_i. Still, the path length should be similar to L_i, otherwise one would need to accelerate or decelerate the path too much to adjust it to the exact length, leading to unnaturally fast or slow gestures. We only accept paths with duration $L'_s \in [0.9L_s, 1.1L_s]$ since these can be slightly adjusted, e.g. re-sampled, to match the target segment duration. For the above range, we observed that the motion still looks natural. Nevertheless, we also add a path duration cost $|1-L'_s/L_s|$ to favor paths during beam search with duration closer to the target duration.

When the speech audio is silent, the searched motion graph paths go through nodes without audio onset features, which are often the frames with rest poses.

After processing the first segment $a_0 \rightarrow a_1$, we start another beam search for the next segment $a_1 \rightarrow a_2$. Here, the path expansion starts with the last node of the K paths discovered from previous iteration. The expansion continues with the same search procedure as above. In order, the searches run iteratively for all the rest segments $a_s \rightarrow a_{s+1}$, $s \in [1, S]$ while always keeping the most plausible K paths. All searched K paths can be used to generate various plausible results for the same target speech audio (see demo videos on our project page). The best path is picked in our experiments.

(a) (b) (c)

Figure 2. An example of inputs to our spatial encoder network (a) input image frame, (b) corresponding foreground human mask and (c) rendered skeleton image.

User study details. We provide here more details about the user study.

We have a pool of 381 queries (127 videos from each method $\times$ 3 comparison pairs). For each query, we show two videos in parallel randomly placed at left/right positions. The participants are asked which speaker's gestures are more consistent with the speech audio and vote for one of the two choices: "left animation", "right animation". Fig. 3 shows the webpage layout used in our questionnaires. The layout shows two video results to the participants, a question on the bottom and two choices ("left"/"right"). To enable the selection of either choice, the users must watch both videos until the end. We also explicitly instruct them to focus on the speakers' hand gestures and ignore the masked facial area.

Our questionnaires also include a similar page layout showing tutorial examples in the beginning. The tutorial shows a pair of videos with clear differences: one video is from ground-truth in which the speaker's gestures are naturally consistent with the audio; the other video is a failure case, which shows gestures that are inconsistent with audio at some places. For these tutorial cases only, we let the participants pick an answer first and then let them know whether their answer is correct or wrong and explain why.

We also adapt a user validation check to filter out unreliable MTurkers. Specifically, after the tutorial, our questionnaires showed 10 queries in a random order. 3 of the queries were repeated twice (i.e., we had 7 unique queries per questionnaire). We randomly flipped the two videos each time to detect unreliable participants giving inconsistent answers. We filter out unreliable MTurk participants who give different answers to two (or more) of the repeated queries in the questionnaire or took less than 5 minutes to complete it. Each participant was allowed to answer one questionnaire

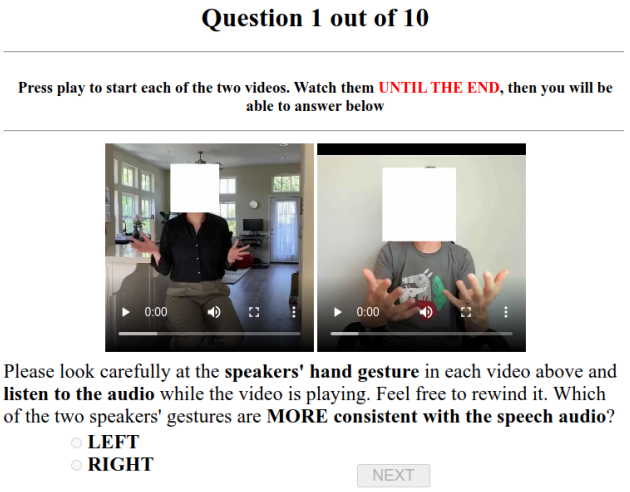

Figure 3. User study questionnaire page.

512×512	Input RGB image, foreground mask image, and rendered skeleton image
256×256	ResBlock down $(16 + 2) \rightarrow 32$
128×128	ResBlock down $32 \rightarrow 64$
64×64	ResBlock down $64 \rightarrow 128$
32×32	ResBlock down $128 \rightarrow 256$
16×16	ResBlock down $256 \rightarrow 512$
8×8	ResBlock down $512 \rightarrow 512$
8×8	ResBlock up $512 \rightarrow 512$
16×16	Skip + ResBlock up $(512 + 512) \rightarrow 512$
32×32	Skip + ResBlock up $(512 + 512) \rightarrow 256$
64×64	Skip + ResBlock up $(256 + 256) \rightarrow 128$
128×128	Skip + ResBlock up $(128 + 128) \rightarrow 64$
256×256	Skip + ResBlock up $(64 + 64) \rightarrow 32$
512×512	Skip + ResBlock up $(32 + 32) \rightarrow 16$

Table 2. Spatial Encoder network structure.

maximum to ensure participant diversity. We collected answers from 113 reliable participants for our user study. We paid \$1 per questionnaire. All comparison outcomes are statistically significant using a z-test ($p<.05$).

Importance of the reference video. The key idea of using reference video is that it provides personalized gestures. Directly animating a single portrait is hard since it is not clear what are the '*correct*' gestures. There are many applications of our setup. For example, in video production, there is a need to add or remove sentences from existing clips. In online education, different video lessons can be created based on a reference video.

Runtime speed Generating a video from a 15 second input audio and a 2 minute reference video takes about 43 seconds in total. Here is the breakdown: (a) 8 seconds are used for audio-driven search to find graph paths in the video motion graph, (b) 35 seconds are used for synthesizing all transitions. Specifically, for a 15 second input audio, there are maximum of 4 synthetic transitions in our examples, with 8 blended frames created per transition. For blending, obtaining the initial mesh flow from human fitting takes 1 second, then synthesizing each blended frame takes 0.1 seconds measured on a single Tesla V100 GPU.

Personal data / human subjects. The Personal story dataset contains 7 videos with 6 different speakers (5 male, 1 female). The number of frames ranges from 4465 to 19176 (148 to 639 seconds). We collected it under the permission from each speaker to include frames, clips and full video in the paper submission. We also used the TED-talks dataset from the previous work [5]. The perceptual user study is collected with the approval of IRB.

References

[1] Chien-Ming Huang and Bilge Mutlu. Modeling and evaluating narrative gestures for humanlike robots. In *Robotics: Science and Systems*, 2013. 2

[2] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proc. CVPR*, 2018. 1

[3] David McNeill. *Hand and mind: What gestures reveal about thought*. University of Chicago press, 1992. 2

[4] Steven M Rubin and Raj Reddy. The locus model of search and its use in image interpretation. In *IJCAI*, 1977. 3

[5] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *Proc. CVPR*, 2021. 2, 4

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 1

[7] Youngwoo Yoon, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. Speech gesture generation from the trimodal context of text, audio, and speaker identity. *ACM Trans. on Graphics (TOG)*, 2020. 2