Appendices

A. Dataset

A.1. Dataset Description

Our dataset is built from the in-the-wild talking videos of four persons with various poses. The dataset contains high-quality 3D holistic body mesh annotations that are reconstructed from video clips of 26.9 hours in total. Each clip is less than 10 seconds. Fig. 7 illustrates the distributions of video durations from different characters.

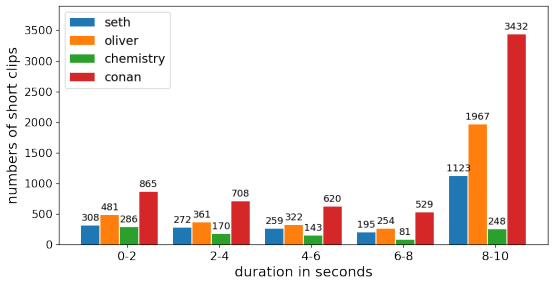

Figure 7. The distribution of the number of short clips for each character (0-10 seconds) of different speakers.

A.2. Good Practices for Improving p-GT

Preliminary. The 3D holistic body meshes consist of face, hands, and body, which is achieved by adopting SMPL-X [47]. It uses standard vertex-based linear blend skinning with learned corrective blend shapes and has $N = 10475$ vertices and $K = 67$ joints. Let W be the linear blend skinning function, the predicted mesh vertices can be represented as $v = W(\theta, \psi, \beta) \in \mathbb{R}^{N\times 3}$. Let $V = \{v_t | v_t \in \mathbb{R}^{N\times 3}\}_{t=1}^{T}$ and $J = \{j_t | j_t \in \mathbb{R}^{67}\}_{t=1}^{T}$ be the temporal sequence of mesh vertices and its 3D joint locations regressed from a linear regressor. We also denote $P^b = \{p_t^b | p_t^b \in \mathbb{R}^{32}\}_{t=1}^{T}$ and $P^h = \{p_t^h | p_t^h \in \mathbb{R}^{24}\}_{t=1}^{T}$ as the temporal sequence of the coefficients of the latent space of VPoser and low dimensional pose space after principal component analysis (PCA) for the body and hands respectively. For time interval $[1:t]$, $V_{1:t} = (v_1, ..., v_t)$, $J_{1:t} = (j_1, ..., j_t)$, $P_{1:t}^b = (p_1^b, ..., p_t^b)$ and $P_{1:t}^h = (p_1^h, ..., p_t^h)$ represent segments of mesh vertices, 3D joints, body pose, and hand pose, respectively. Note that we use a fixed pose (sitting or standing) for the invisible lower body. And in a temporal sequence of the p-GT holistic motions m_i, at each time step t, the facial representation $m_t^f = [\theta_t^f, \psi_t] \in \mathbb{R}^{103}$ is a concatenation of jaw orientation and expression, and the body and hand motions are respectively represented by their poses $m_t^b = \theta_t^b \in \mathbb{R}^{63}$ and $m_t^h = \theta_t^h \in \mathbb{R}^{90}$.

Initialization. Since optimization-based methods are often slow and sensitive to the initialization. In contrast, regression-based methods tend to give a reasonable, but not well pixel-aligned results. Therefore, we use the results from PIXIE [22] and PyMAF-X [69] to initialize the parameters of body and hand pose, respectively. Results from DECA [23] are used to initialize the parameters of jaw pose and facial expression.

Data terms. We extend the data term by incorporating body silhouettes, facial landmarks, facial shapes, and facial details.

Firstly, to deal with the imperfect 2D landmarks by Openpose [10], we introduce the silhouette constraint to encourage the rendered SMPL-X body to be inside the human body mask. Ground-truth person segmentations are expensive to obtain for in-the-wild datasets. Hence, we employ an off-the-shelf segmentation model, Deeplab V3 [12] to generate p-GT person semantics mask $M_{sil} \in \mathbb{R}^{T\times h\times w}$, where H and W are the height and width of the input image. Pytorch3D is used as the differential renderer to process the rendered pixels of all mesh triangles, leading to the predicted semantics mask $\widehat{M}_{sil} \in \mathbb{R}^{T\times h\times w}$. The silhouette loss term is given by:

$$\mathcal{L}_{sil} = \sum ||d(\widehat{M}_{sil}) \odot d_{edt}(g(M_{sil}))||_2, \quad (4)$$

where $g(x) = MaxPool(x) - x$ is a function for detecting the edge of the binary mask. d_{edt} is a distance function to calculate the smallest Euclidean distance from the background point to the silhouette boundary.

Secondly, to get a better facial geometry in SMPL-X, we minimize the difference between the facial shape in SMPL-X and the reconstructed facial shape from MICA [73]. We term this as a facial shape objective $\mathcal{L}_{FS}$ given by:

$$\mathcal{L}_{FS} = ||M_{g1}(V_{SMPL-X}) - M_{g2}(V_{MICA} + t_{FS})||_2, \quad (5)$$

where $V_{SMPL-X} \in \mathbb{R}^{N\times 3}$ is the SMPL-X vertices at neutral pose (i.e. $\theta = 0, \psi = 0$). $V_{MICA} \in \mathbb{R}^{5023\times 3}$ is the MICA shape, and $t_{FS} \in \mathbb{R}^3$ is the offset of V_{MICA} from V_{SMPL-X}. M_{g1} and M_{g2} are functions that maps the original mesh vertices of V_{SMPL-X} and V_{MICA} to the corresponding 1787 vertices of frontal face part, respectively.

Thirdly, to get better facial expression, we use MediaPipe [32] to extract 105 of 468 dense 2D facial landmarks for each image. The loss term $\mathcal{L}_{FE}$ is calculated as:

$$\mathcal{L}_{FE} = \sum_t ||U_{1:t} - \widehat{U}_{1:t}||_2, \quad (6)$$

where $U_{1:t}$ and $\widehat{U}_i$ are temporal segments of landmarks from MediaPipe [32] and the 2D projection of the corresponding 3D joints $J_{1:t}$, respectively.

Lastly, to obtain high-frequency resolution facial details, we employ face expression tracking to monocular RGB

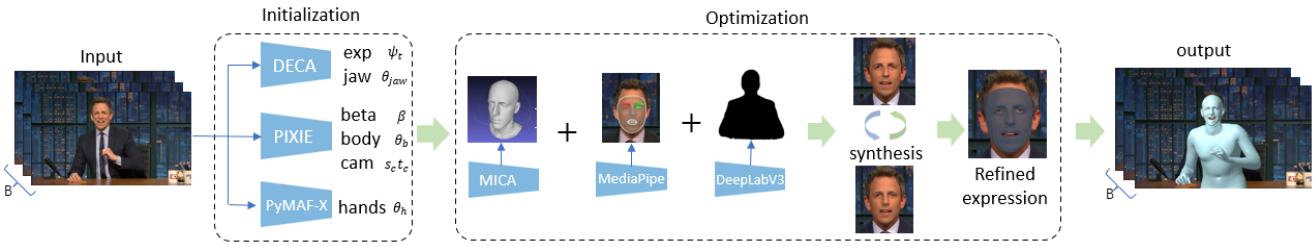

Figure 8. The architecture of SHOW. It consists of initialization and optimization modules. Specifically, given an input the image sequence, firstly, PIXIE [22], DECA [17] and PyMAF-X [69] are used to initialize the parameters of SMPL-X. Secondly, the optimization routine incorporates body silhouettes from DeepLab V3 [12], facial landmarks from MediaPipe [32], and facial shapes from MICA [73]. Then, it uses a photometric loss between the rendered faces and the input image to better capture facial details. Lastly, SHOW outputs the final results.

images in a self-supervised fashion. Specially, we follow [23, 73] to reconstruct the face jointly with an illumination model based on spherical harmonics and a Lambertian material assumption:

$$\mathcal{L}_{FR} = \sum_t ||I_r(\mathcal{M}_{S2F}(V_{1:t})) - I^{head}_{1:t}||_2, \tag{7}$$

where $\mathcal{M}_{S2F}$ is a function that selects the head part of $V_{1:t}$. I_r is the forward pass of differential rendering. $I^{head}_{1:t}$ is the cropped head image from input image. Note that we choose different scales (e.g. $256, 512, 1024$) for different stages in the optimization procedure.

Regularization. Different regularization terms in SMPLify-X prevent the reconstruction of unrealistic bodies. To derive more reasonable regularization terms, we explicitly take the video prior into account.

To reduce the jittery results caused by the noisy 2D detected keypoints, we introduce a smooth term for body and motion poses (P^b and P^h). They are defined as:

$$\mathcal{M}_b = \sum_t ||P^b_{2:t} - P^b_{1:t-1}||_2, \tag{8}$$

$$\mathcal{M}_h = \sum_t ||P^h_{2:t} - P^h_{1:t-1}||_2. \tag{9}$$

We also add constant-velocity smooth term $\mathcal{M}_j$ on J:

$$\mathcal{M}_j = \sum_t ||J_{3:t} + J_{1:t-3} - 2 \times J_{2:t-2}||_2, \tag{10}$$

Furthermore, to prevent the inter-penetration of two hands, we use Collision Penalizer [47] and denote this loss term as L_{pen}.

Training Losses. The final objective function is given by:

$$E(\beta, \{\theta\}^T_{t=1}, \{\psi\}^T_{t=1}, \psi_{light}, \psi_{lbs}, t_{FS}) = \sum_{t=1}^{T}(E_{SMPLify-X}(t)) + \lambda_{FE}\mathcal{L}_{FE} + \lambda_{FS}\mathcal{L}_{FS} + \lambda_{FR}\mathcal{L}_{FR} + \lambda_{mb}\mathcal{M}_b + \lambda_{mh}\mathcal{M}_h + \lambda_{mj}\mathcal{M}_j + \lambda_{sil}\mathcal{L}_{sil} + \lambda_{pen}\mathcal{L}_{pen}, \tag{11}$$

where $\psi_{light} \in R^3$ is the spherical harmonic coefficients representing the environmental illumination. $\psi_{lbs} \in R^{128}$ is the linear blend skinning parameters of albedo model. $E_{SMPLify-X}(t)$ is the basic prior on single image as describe in [47]. Weights λ steer the influence of each term.

Optimization. Following [47], we adopt the Limited-memory BFGS [46] with strong wolfe line search for optimization. An iterative fitting routine is used for better fitting. With proper initialization, we minimize the objective function using a five-stage fitting procedure to avoid the local minima trap and reduce the optimization time. The learning rate is set to 1. As the required GPU memory increases dramatically with the image batch size for neural rendering, we use a mini-batch of 50 on NVIDIA Tesla V100.

B. Network Architecture Details

B.1. Face Generator

The raw audio input is normalized to zero mean and unit variance, and then is fed to encoder, which consists of an audio feature extractor, a transformer encoder, and a full-connected layer. The audio feature extractor is followed by an interpolation operation, in which the audio feature is re-sampled into target frames. For the decoder, it comprises six temporal convolution layers (with a kernel size, stride and padding of 3, 1 and 1 respectively) and a full-connected layer. Each temporal convolution layer is followed by layer normalization [6] and a Leaky RELU activation function [43]. We adopt SGD with momentum and a learning rate of

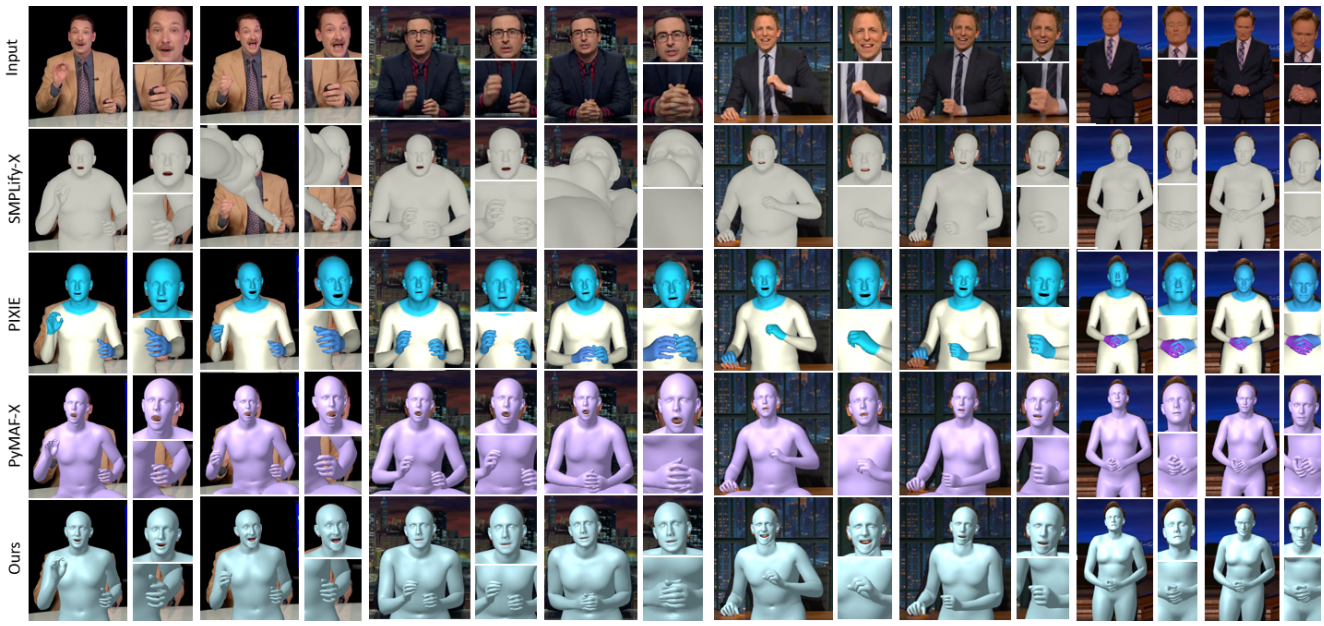

Figure 9. The 3D holistic body reconstructions of four subjects from SMPLify-X, PIXIE, PyMAF-X, and ours. Compared to other methods, ours produces more accurate and stable results with details.

0.001 as the optimizer. The face generator is trained with batchsize of 1 for 100 epochs, in which each batch contains a full-length audio and corresponding facial motions.

B.2. Body and Hand Generator

VQ-VAEs Details. The VQ-VAE takes body or hand motions as input. The encoder of each VQ-VAE is composed of three residual layers, which includes three temporal convolution layers (with a kernel size, stride and padding of 3, 1 and 1 respectively) followed by batch normalization [28] and a Leaky RELU activation function [43]. The encoder is interleaved with a temporal convolution layer with a kernel size, stride and padding of 4, 2 and 1 respectively after every residual layer except the last so that the temporal window size w is equal to 4. On the top of the encoder, a full-connected layer is added to reduce the dimension before quantization. The decoder is symmetric with the encoder. We adopt Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a learning rate of 0.0001 as the optimizer. The commitment loss weight β is set to 0.25. The VQ-VAEs are trained with a batchsize of 128 and a sequence length of 88 frames for 100 epochs.

Autoregressive Model Details. The autoregressive model consists of an audio encoder and a Gated PixelCNN [57]. The audio encoder, which has the same structure as the VQ-VAE encoder, takes MFCC feature as input. Then we concatenate the output of the audio encoder and VQ-VAEs encoders and feed it to the Gated PixelCNN. The Gated PixelCNN has 15 gated convolution layers conditioned on identity, in which the convolution kernel is masked to make sure the model cannot read future information. We adopt Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a learning rate of 0.0001 as the optimizer. The autoregressive model is trained with a batchsize of 128 and a sequence length of 88 frames for 100 epochs.

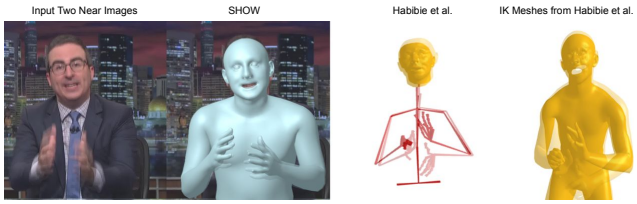

Figure 10. Holistic body reconstruction compared to Habibie et al.

C. More Comparison

Habibie el al. [27] v.s. SHOW. Habibie et al. [27] represent body, hands, and face separately. The lack of connection between body and face/hands results in unnatural poses of the face/hands w.r.t. the body. Fig. 10 a) shows that the hand and head poses of the body mesh, reconstructed from their estimated 3D skeleton, are less accurate than ours. Generated video results are further jittery. In contrast, SHOW generates more stable and accurate holistic body meshes.

Experimental Results. We compare our method with more other approaches and more metrics in Tab. 5. Specifically, We add Frechet Gesture Distance (FGD) [68] to measure the motion realism and beat consistency (BC) [42] to measure

Method	Habibie	Audio VAE	Audio+Motion VAE	Audio2Gesture[40]	Ours w/o c-c	Ours w/ c-c
FGD ↓	239.32	121.01	166.65	203.99	147.81	**74.88**
Variance ↑	0	0.044	0.176	0.240	**0.922**	0.821
BC (GT 0.868)	0.948	0.746	0.822	0.943	0.851	**0.872**

Table 5. More experimental results.

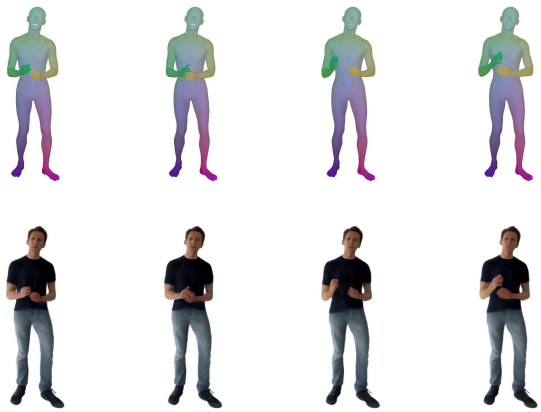

Figure 11. The application with SMPLpix to create photo-realistic neural avatars. Top row (input): the mesh vertices provided by TalkSHOW and their colors projected onto the image plane, bottom row: rendered output.

the alignment between the generated body motion and input audio, and compare with another audio-to-body motion baseline [40]. Our method outperforms the baselines in all these metrics and generates more diverse body motions, which are better aligned with the input audio.

D. Application

One application of our speech-to-motion generation is to create the photo-realistic neural avatars through neural renderers such as SMPLpix [49]. Given the mesh vertices provided by TalkSHOW and their colors, we first project them onto the image plane. Then, with the projected mesh vertices, SMPLpix allows us to efficiently synthesise photo-realistic images of humans. As TalkSHOW can produce continuous yet diverse motions, integrating SMPLpix with our motion generation framework enables us generate human avatars under different poses (see Fig. 11), leading to end-to-end photo-realistic video generation.

E. Discussions

Reconstruction. SHOW is based on SMPLify-X whose supervision signal is obtained from 2D keypoint reprojection. Thus, it is sensitive to severe hand shape deformation and heavy occlusion. A future direction would be to leverage advanced hand model with rich shape and pose space. Besides, SHOW can only handle static camera cases currently. In the future, we plan to extend it to moving cameras.

Audio2motion. While we have demonstrated that TalkSHOW can generate realistic, coherent, and diverse holistic body motion with facial expression, body, and hand motions, it is subject to a limitation that can be addressed in the future. For the face generator, we mainly focus on facial motion (e.g. lip motion) and might not handle the very complex facial movements caused by emotions. In the future, we plan to extend to model this sort of part.

F. Risks and Potential Misuse

This work is intended for studying the translation from human speech to holistic body motion, helping building virtual agents to behave realistically and interact with listeners meaningfully. Since our techniques can generate a realistic and diverse 3D talking humans from audio, there is a risk that such technique could be potentially misused for fake video generation. For instance, a fake speech could be used to construct highly realistic 3D holistic body motion while it never happened. Thus, we should use such technology responsibly and carefully. We hope to raise the public's awareness about a safe use of such technology.