Geometry-guided Dense Perspective Network for Speech-Driven Facial Animation

Jingying Liu†, Binyuan Hui†, Kun Li*, *Member, IEEE,* Yunke Liu, Yu-Kun Lai, *Member, IEEE,* Yuxiang Zhang, Yebin Liu, *Member, IEEE,* and Jingyu Yang, *Senior Member, IEEE*

Abstract—Realistic speech-driven 3D facial animation is a challenging problem due to the complex relationship between speech and face. In this paper, we propose a deep architecture, called *Geometry-guided Dense Perspective Network (GDPnet)*, to achieve speaker-independent realistic 3D facial animation. The encoder is designed with dense connections to strengthen feature propagation and encourage the re-use of audio features, and the decoder is integrated with an attention mechanism to adaptively recalibrate point-wise feature responses by explicitly modeling interdependencies between different neuron units. We also introduce a non-linear face reconstruction representation as a guidance of latent space to obtain more accurate deformation, which helps solve the geometry-related deformation and is good for generalization across subjects. Huber and HSIC (Hilbert-Schmidt Independence Criterion) constraints are adopted to promote the robustness of our model and to better exploit the non-linear and high-order correlations. Experimental results on the public dataset and real scanned dataset validate the superiority of our proposed GDPnet compared with state-of-the-art model. The code is available for research purposes at *http://cic.tju.edu.cn/faculty/likun/projects/GDPnet*.

Index Terms—Speech-driven, 3D Facial Animation, Geometry-guided, Speaker-independent

1 Introduction

The most important approach of human communication is through speaking and making corresponding facial expressions. Understanding the correlation between speech and facial motion is highly valuable for human behavior analysis. If the correlation between speech and facial motion (which is a form of low level human behavior) is correctly learned, the generated facial animation will be more reasonable and realistic. Therefore, speech-driven facial animation has drawn much attention from both academia and industry recently, and has a wide range of applications and prospects, such as gaming, live broadcasting, virtual reality, and film production [26], [28], [42]. 3D models, as a popular and effective representation for human faces, have stronger ability to show the facial motion and understand the correlation between speech and facial motion than 2D images. However, 3D models are more complicated than images, and it is more difficult to obtain realistic 3D animation results. As shown in Figure 1, our aim is to animate a 3D template model of any person according to an audio input.

Despite the great progress in speaker-specific speech-driven facial animation [3], [20], [34], speaker-independent facial animation is still a challenging problem. Some methods animate unrealistic artist-designed character rigs driven by audio [6], [43]. Others achieve more realistic animation by combining audio and video [26], [30], relying on manual processes, or focusing only on the mouth [36]. VOCA [5] achieves the first audio-driven speaker-independent 3D facial animation in any language using a realistic 3D scanned template. It could generate animation of different styles across a range of identities. But there are still three challenges to achieve realistic audio-driven 3D facial animation for an arbitrary person and language:

- The animated results are easily affected by both facial motion and geometry characteristics. Therefore, we need to consider the geometry representation of 3D models to generate more realistic animation results, in addition to relating the audio and the facial motion.
- The relation between audio and visual signals is complicated, and we need more effective neural networks to learn this non-linear and high-order relationship.
- In-the-wild speech audio usually contains noise and outliers, which challenge the robustness of the animation method.

In this paper, to address these challenges, we propose a geometry-guided dense perspective network (GDPnet), which consists of encoder and decoder modules. For the encoder, to ensure maximum information flow between layers in the network, we connect all layers (with matching feature-map sizes) directly with each other. For the decoder, we utilize attention mechanism to use global information to selectively emphasize informative features. We use an implicit dynamic geometry representation that is encoded by the network to guide the network training, and adopt two constraints from different perspectives to achieve more robust animation. Experimental results demonstrate that the non-linear geometry representation is beneficial to the speech-driven model, and our model generalizes well to arbitrary subjects unseen during training. The code is available for research purposes at *http://cic.tju.edu.cn/faculty/likun/projects/GDPnet*.

- † *Equal contribution.*
- * *Corresponding author: Kun Li (Email: lik@tju.edu.cn)*
- *Jingying Liu, Binyuan Hui, Kun Li and Yunke Liu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China.*
- *Yu-Kun Lai is with the School of Computer Science and Informatics, Cardiff University, Cardiff CF24 3AA, United Kingdom.*
- *Yuxiang Zhang and Yebin Liu are with the Department of Automation, Tsinghua University, Beijing 10084, China.*
- *Jingyu Yang is with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China.*

Figure 1: Our method is able to output reasonable and realistic 3D animated faces for any person in any language. Top: Actor from VOCASET [5]; Middle: Actor from D3DFACS [4]; Bottom: Great tribute to Mr. Albert Einstein.

Specifically, the main contributions of this work are summarized as follows:

- We propose a dense perspective network to better model the non-linear and high-order relationship between audio and visual signals and utilize an attention mechanism to use global information to selectively emphasize informative features. To the best of our knowledge, it is the first time to introduce dense connection in cross-modal tasks. By using dense connection, our network can combine the features learned in different stages and get a more informative implicit embedding from the speech, which can generate more reasonable face displacements by the decoder, leading to better performance of 3D animation.
- We adopt a non-linear face representation to guide the network training, which helps to solve the geometry-related deformation and is effective for generalization across subjects. The relation between audio and mesh output is complicated, and hence it is hard to directly get the mapping through a network. Moreover, the animated results are easily affected by both facial motion and geometry structure. By applying the non-linear face representation, our network can learn more geometry information and generate more realistic animations, in addition to relating the audio and the facial motion.
- We introduce Huber and HSIC (Hilbert-Schmidt independence criterion) constraints to promote the robustness of our model and better measure the non-linear and high-order correlations between the implicit geometry representation and the latent code without explicitly estimating the joint distribution of the random variables. As a result, both convergence speed and performance are improved.
- Our model is easy to train and fast to converge. At the same time, we achieve more accurate and realistic animation results for various persons in various languages.

2 Related Work

Despite the great progress in facial animation from images or videos [21], [38], [39], [40], less attention has been paid to speech-driven facial animation, especially animating a 3D face. However, understanding the correlation between speech and facial deformation is very important for human behavior analysis and virtual reality applications. Speech-driven 3D facial animation can be categorized into two types: speaker-dependent animation and speaker-independent animation, according to whether the method supports generalization across characters.

2.1 Speaker-dependent Animation

Speaker-dependent animation mainly uses a large amount of data to learn the animation ability in a specific situation. Cao *et al*. [3] first rely on a database of high-fidelity recorded facial motions, which includes speech-related motions, but the method relies on high-quality motion capture data. Suwajanakorn *et al*. [34] utilize a recurrent neural network trained on millions of video frames to synthesize mouth shape from audio, but this method only focuses on learning to generate videos of President Barack Obama from his voice and stock footages. Karras *et al*. [20] first propose an end-to-end network for animation. Through the input of voice and specific emotion embedding, it could output the 3D vertex positions of a fixed-topology mesh that corresponds to the center of the audio window. Besides, it could produce expressive 3D facial motion from audio in real time and with low latency. However, this kind of animation methods has limited practical applications due to its inconvenience for generalization across characters.

2.2 Speaker-independent Animation

Many works focus on the facial animation of artist-designed character rigs [6], [7], [14], [19], [33], [35], [36], [37], [43]. Liu

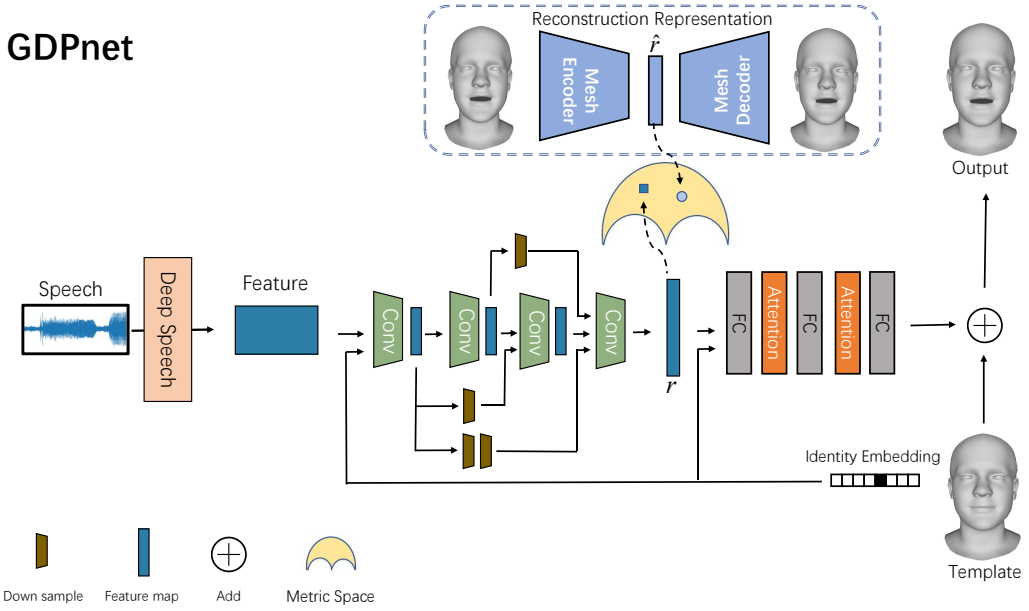


Figure 2: The architecture of our proposed geometry-guided dense perspective network.

et al. [26] first propose a speaker-independent method based on a Kinect sensor with video and audio input for 3D facial animation, which reconstructs 3D facial expressions and 3D mouth shapes from color and depth input with a multi-linear model and adopts a deep network to extract phoneme state posterior probabilities from the audio. However, this method relies on a lot of pre-processing and inefficient search methods. Taylor *et al*. [36] propose a simple and effective deep learning approach for speech-driven facial animation using a sliding window predictor to learn arbitrary non-linear mappings from phoneme label input sequences to mouth movements. Pham *et al*. [29] propose a regression framework based on a long short-term memory (LSTM) recurrent neural network to estimate rotation and activation parameters of a 3D blendshape face model. Based on this work, they [30] further employ convolutional neural networks to learn meaningful acoustic feature representations, but their method also needs the recurrent layer to process the information of time series. Zhou *et al*. [43] propose a three-stage network using hand-engineered audio features to regress the cartoon human. However, the animated face is not a realistic scanned face. Cudeiro *et al*. [5] first provide a self-captured multi-subject 4D face dataset and propose a generic speech-driven 3D facial animation framework that works across a range of identities. However, none of these methods take into account the influence of geometry representation on speech-driven 3D facial animation.

In this paper, we propose a speaker-independent speech-driven 3D facial animation method by designing a geometry-guided dense perspective network. The introduced non-linear geometry representation and two constraints from different perspectives are very beneficial to achieving realistic and robust animation.

3 Geometry-guided Dense Perspective Network

Figure 2 shows the architecture of our geometry-guided dense perspective network (GDPnet). First of all, we extract speech features using DeepSpeech [12] and embed the identity information to one-hot embedding. After concatenating the two kinds of information, the encoder maps it to the latent low-dimensional representation. The purpose of the decoder is to map the hidden representation to a high-dimensional space of 3D vertex displacements, and the final output mesh is obtained by adding the displacements to the template.

3.1 Problem Definition

Suppose we have three types of data $\{(\mathbf{p}, \mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{F}$. Here, the index i refers to a specific frame, and F is the total number of frames. $\mathbf{x}_i \in \mathbb{R}^{W \times D}$ is the speech feature window centered at the ith frame generated by DeepSpeech [12], where D is the number of phonemes in the alphabet plus an extra one for a blank label and W is the window size. $\mathbf{p} \in \mathbb{R}^{N \times 3}$ denotes the corresponding template mesh, reflecting the subject-specific geometry, and N is the number of vertices of the mesh. $\mathbf{y}_i \in \mathbb{R}^{N \times 3}$ denotes the ground truth for facial animation at each frame. At last, let $\hat{\mathbf{y}}_i \in \mathbb{R}^{N \times 3}$ denotes the output of our GDPnet model for the input $\mathbf{x}_i$ with template $\mathbf{p}$.

3.2 Model

Our GDPnet model consists of an encoder and a decoder, as shown in Figure 2. The input of the encoder is a DeepSpeech feature of the audio and specific identity information. In order to effectively express different subjects, we encode the identity information as one-hot embedding so as to control different speaking styles. In particular, the dimension of identity embedding is equal to the number of subjects in the training set. During inference, changing the identity embedding alters the output speaking style. We adopt dense connections to combine the features learned in different stages and get a more informative implicit embedding from the speech, and add attention layers to selectively emphasize important features. Also, we use an implicit dynamic geometry representation that is encoded by the network to guide the network training, and introduce Huber and HSIC constraints to improve the robustness of our model and better measure the non-linear and high-order correlations.

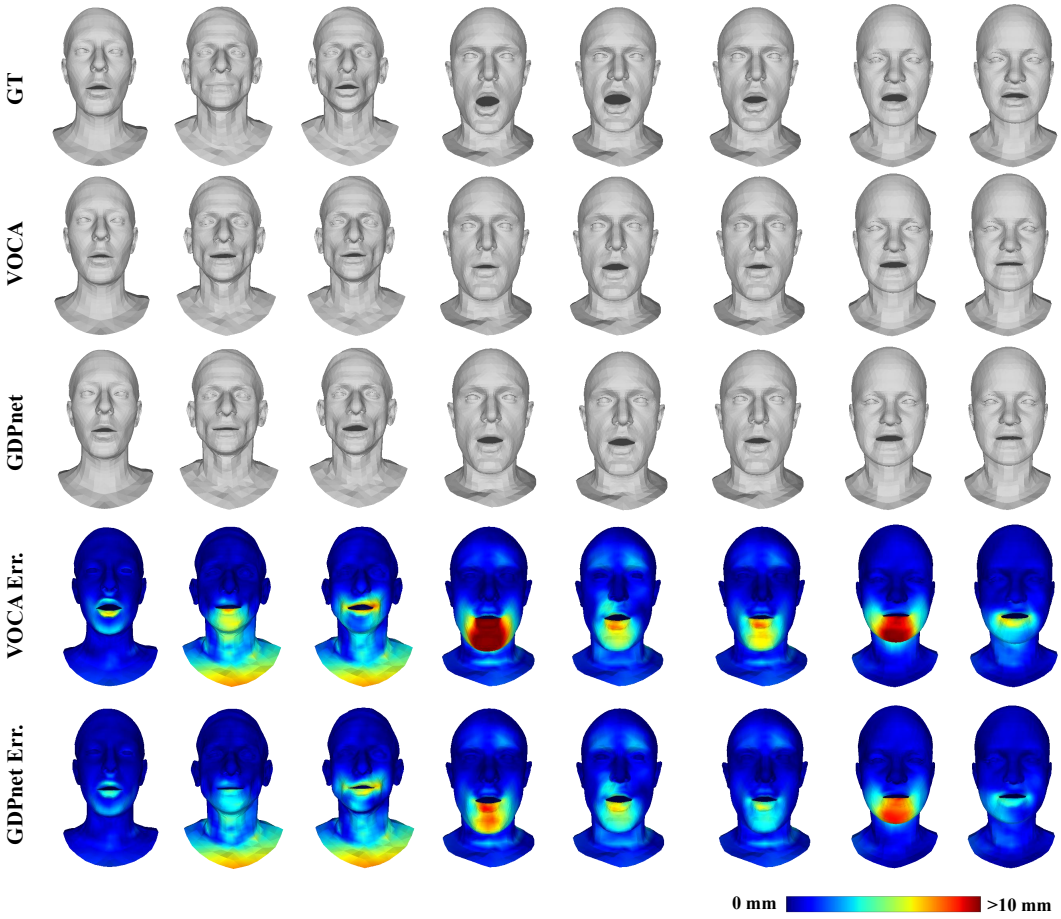


Figure 3: Qualitative evaluation results for clean audio inputs.

3.2.1 Encoder

The purpose of the encoder is to map speech features to latent representations. Similar to VOCA [5], to learn temporal features and reduce the dimensionality of the input, we stack four convolutional layers for the encoder. The problem of simply stacking the convolutional layers is that the information of the early layers can be easily lost [16]. We believe that both low-dimension and high-dimension features are important, and hence we need an effective way to combine the features at the low-dimension layer and the high-dimension layer. Dense connections are widely used in many deep learning architectures, including computer vision and natural language processing. In computer vision, the DenseNet [16] uses different numbers of dense blocks in different datasets, which can take full advantage of features and perform better compared with ResNet [13]. In natural language processing, the dense blocks are used to learn more about local dependence within the sentence, like MC-RNN [41]. Their motivation to use dense connection is to reuse shallow features and encourage the combination of features learned in different stages. Inspired from them, we use dense connections in the speech encoder of our cross-modal task. The biggest difference is that, they use the dense connections to get more features from images to generate images or get more sentence features from sentences for sentence analysis, which are single-modal tasks, while dense connections in our GDPnet help to get more features from speech to generate the 3D animation, which is cross-modal. Consequently, the ith layer receives the feature maps of all preceding layers, $\mathbf{x}_0, \ldots, \mathbf{x}_{\ell-1}$, as input:

$$\mathbf{x}_\ell = H_\ell\left([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}]\right), \tag{1}$$

where $[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}]$ refers to the concatenation of the feature maps produced in layers $0, \ldots, \ell-1$, and H_ℓ is a composite function of two operations: convolution (Conv) with 3×1 filter size and 2×1 stride, followed by a rectified linear unit (ReLU) [10].

We follow common practice and double the number of filters (feature maps) after each convolutional layer. Applying the concatenation operation in dense connections directly would be infeasible as the sizes of feature maps are different. Therefore, we introduce 2×1 pooling layers in the feature map dimension to reduce the number of feature maps before concatenation (indicated as the Down Sample layer in Figure 2). As a direct consequence of the input concatenation, the feature maps learned by any layers can be accessed easily. Benefiting from the dense connection structure, we can reuse features effectively, which makes the encoder learn more specific and richer latent representations.

3.2.2 Decoder

The decoder maps the latent representation to a high-dimensional space of 3D vertex displacements, and the final output mesh is obtained by adding the displacements to the template vertex positions. To achieve this, we stack two fully connected layers with $\tanh$ activation function.

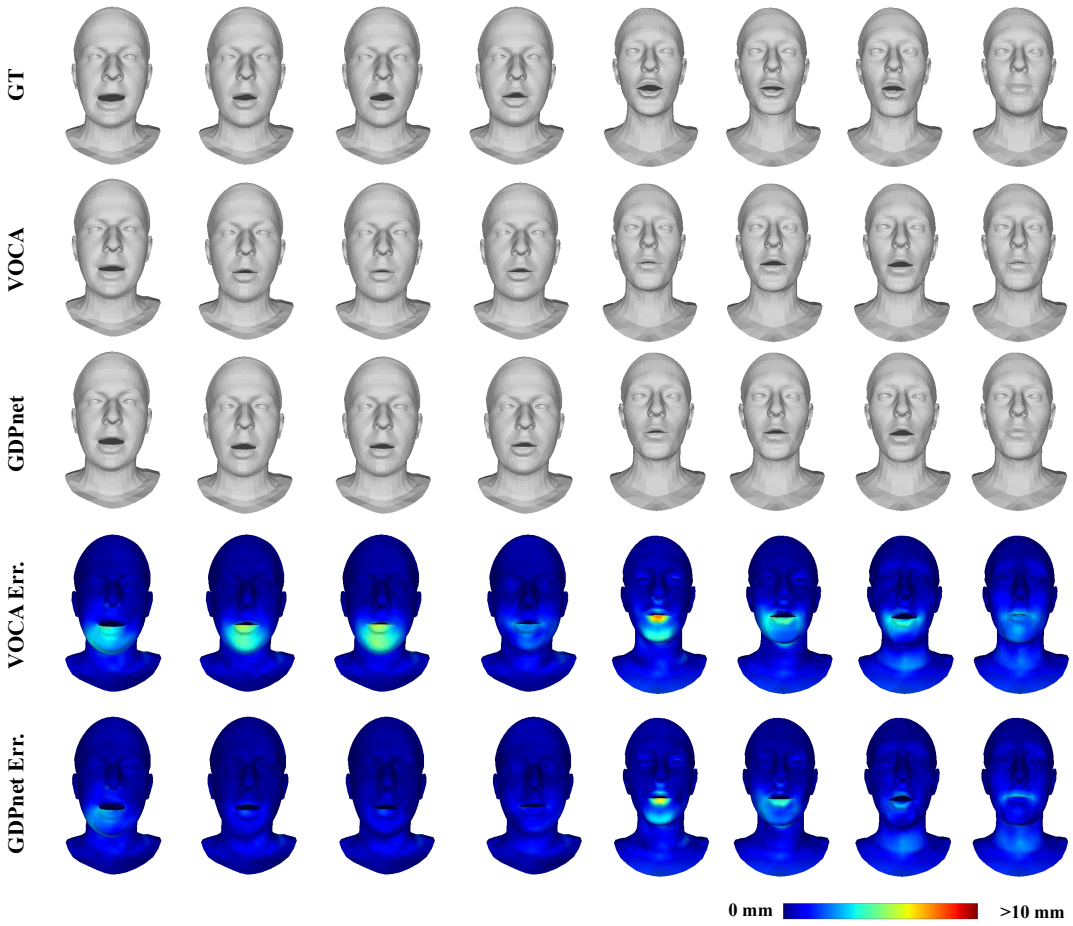


Figure 4: Qualitative evaluation results for noisy audio inputs.

Inspired by the attention mechanism in image classification [15], we add attention mechanism to perform feature recalibration. In this way, the network learns to use global information to selectively emphasize informative features and suppress less useful ones. Let $\mathbf{x}_\ell \in \mathbb{R}^{C\times 1}$ denote the input of the attention layer, where C is the number of feature maps, and the attention value $\mathbf{a}_\ell$ can be calculated by

$$\mathbf{a}_\ell = \sigma\left(\mathbf{W}_2\delta\left(\mathbf{W}_1\mathbf{x}_\ell\right)\right), \tag{2}$$

where σ refers to the ReLU function and δ refers to the sigmoid function. $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{2}\times C}$ and $\mathbf{W}_2 \in \mathbb{R}^{C\times\frac{C}{2}}$ denote the learnable parameter weights for the attention block. The final output of the attention block is obtained by

$$\widetilde{\mathbf{x}}_l = \mathbf{x}_l \otimes \mathbf{a}_\ell. \tag{3}$$

Here, $\otimes$ is element-wise multiplication. Through the attention block, the model can adaptively select important features for the current input samples, and different inputs can generate different attention responses.

The final output layer is a fully connected layer with linear activation function, which produces $N \times 3$ output, corresponding to the 3-dimensional displacement vectors of N vertices. $N = 5023$ is used in our experiments. The final mesh can be generated by adding this output to the identity template. In order to make the training more stable, the weight of this layer is initialized by 50 PCA components which are calculated from the vertex displacements of the training data and scaled by the PCA standard deviation [5]. The deviation of this layer is initialized by zero.

3.3 Geometry-guided Training

The encoder-decoder structure described above can be regarded as a cross-modal process. The encoder maps the speech mode to the latent representation space, while the decoder maps the latent representation space to the mesh mode. We refer to the latent representation as a cross-modal representation, which should express the expression and deformed geometry of a certain identity. It can be exactly related to the reconstructed expression in the 3D face representation and reconstruction using autoencoders [18], [24], [31]. Specifically, r refers to the latent representation generated from the speech input network and $\hat{r}$ refers to that generated from MGCN (Multi-column Graph Convolutional Network) [24]. Figure 2 clearly shows the definition. Since r is generated only from speech, to get better geometries during the training, we use MGCN [24] to extract geometry representation for each training mesh due to its ability to extract non-local multi-scale features. The MGCN is an encoder-decoder architecture with multi-column graph convolutional networks to capture features of different scales and learn a better latent space representation. In this way, we can have a geometry representation corresponding to each frame in the speech rather than just using the identity information of the template for the speech during the training. Using this 3D geometry representation $\hat{r}$ can effectively constrain our cross-modal representation. We assume that our implicit dynamic geometry

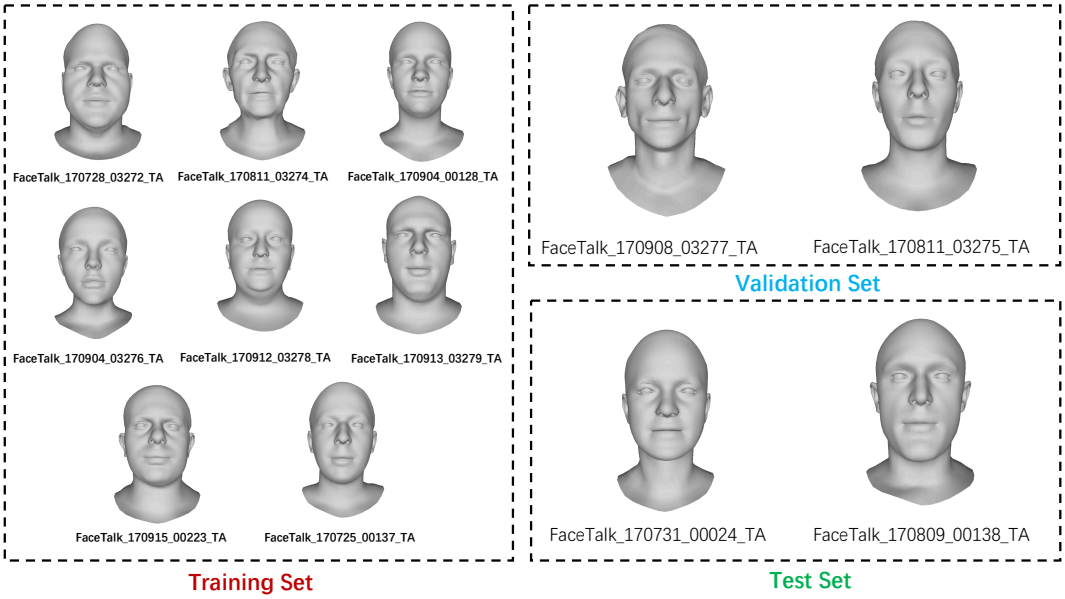


Figure 5: Specific subject names for training, validation and test.

representation that is encoded by the network contains the expression and deformed geometry of a certain identity, and the nonlinear representation $\hat{r}$ from MGCN should be highly correlated with the representation r from the speech input. Here we introduce two approaches of measurement: Huber [17] constraint and Hilbert-Schmidt independence criterion (HSIC) constraint.

3.3.1 Huber Constraint

Most work uses the ℓ_2 loss to measure the distance between two vectors, but this measurement is more easily affected by noise and outliers. ℓ_1 loss is a better choice for robustness, but it is discontinuous and non-differentiable at position 0, leading to the difficulty for optimization. Huber loss adopts a piece-wise method to integrate the advantages of ℓ_1 loss and ℓ_2 loss and has been widely used in a variety of tasks.

Definition 1. *Assuming that there are two vectors r and $\hat{r}$, the Huber constraint L_ξ is defined as*

$$L_\xi : \mathbb{R} \to [0, +\infty),$$
$$L_\xi(r, \hat{r}) = \begin{cases} \frac{r-\hat{r}^2}{2} & \text{if } |r - \hat{r}| \leq \xi \\ \xi|r - \hat{r}| - \frac{\xi^2}{2} & \text{otherwise,} \end{cases} \tag{4}$$

where $\xi > 0$ is the parameter that balances bias and robustness, and is set to 1.0 as default setting.

The parameter ξ controls the blending of ℓ_1 and ℓ_2 losses which can be regarded as two extremes of the Huber loss with $\xi \to \infty$ and $\xi \to 0$, respectively. For smaller values of $|r - \hat{r}|$, the loss function L_ξ is ℓ_2 loss, and the loss function becomes ℓ_1 loss when the magnitude of $|r - \hat{r}|$ exceeds ξ.

3.3.2 HSIC Constraint

In addition to the distance between the two expressions, we also constrain from the perspective of correlations. The relation between the audio and the animated meshes is complicated, and hence it is hard to directly get the mapping through a network. Moreover, the animated results are easily affected by both facial motion and geometry characteristics. Therefore, we adopt a nonlinear face representation to guide the network training. The HSIC measures the nonlinear and high-order correlations between the implicit geometry representation and the latent code without explicitly estimating the joint distribution of the random variables. It has been successfully used in multi-view learning [2], [27]. As a result, both convergence speed and performance are improved.

Assuming that there are two variables $R = [r_1, \ldots, r_i, \ldots, r_M]$ and $\hat{R} = [\hat{r_1}, \ldots, \hat{r_i}, \ldots, \hat{r}_M]$, M is the batch size. We define a mapping $\phi(r)$ to kernel space $\mathfrak{H}$, where the inner product of two vectors is defined as $k(r_i, r_j) = \langle \phi(r_i), \phi(r_j) \rangle$. Then, $\phi(\hat{r})$ is defined to map $\hat{r}$ to kernel space $\mathfrak{G}$. Similarly, the inner product of two vectors is defined as $k(\hat{r}_i, \hat{r}_j) = \langle \phi(\hat{r}_i), \phi(\hat{r}_j) \rangle$.

Definition 2. *HSIC is formulated as*

$$\begin{aligned} \mathrm{HSIC}(\mathbb{P}_{R\hat{R}}, \mathfrak{H}, \mathfrak{G}) &= \|C_{R\hat{R}}\|^2 \\ &= \mathbb{E}_{R\hat{R}R'\hat{R}'}\left[k_R(R, R')\, k_{\hat{R}'}\left(\hat{R}, \hat{R}'\right)\right] \\ &+ \mathbb{E}_{RR'}\left[k_R(R, R')\right]\mathbb{E}_{\hat{R}'}\left[k_{\hat{R}}\left(\hat{R}, \hat{R}'\right)\right] \\ &- 2\mathbb{E}_{R\hat{R}}\left[\mathbb{E}_{R'}\left[k_R(R, R')\right]\mathbb{E}_{\hat{R}'}\left[k_{\hat{R}}\left(\hat{R}, \hat{R}'\right)\right]\right], \end{aligned} \tag{5}$$

where k_R and $k_{\hat{R}}$ are kernel functions, $\mathfrak{H}$ and $\mathfrak{G}$ are the Hilbert spaces, and $E_{R\hat{R}}$ is the expectation over R and $\hat{R}$. Let $\mathcal{D} := \{(r_1, \hat{r}_1), \cdots, (r_m, \hat{r}_m)\}$ drawn from $\mathbb{P}_{R\hat{R}}$. The empirical version of HSIC is induced as:

$$\mathrm{HSIC}(\mathcal{D}, \mathfrak{H}, \mathfrak{G}) = (N-1)^{-2}\,\mathrm{tr}\left(K_1 H K_2 H\right), \tag{6}$$

where $tr(\ldots)$ is the trace of a square matrix. K_1 and K_2 are the Gram matrices with $k_{1,ij} = k_1(r_i, r_j)$ and $k_{2,ij} = k_2(\hat{r}_i, \hat{r}_j)$. H centers the Gram matrix which has zero mean in the feature space:

$$H = I_m - \frac{1}{m} 1_m 1_m^T. \tag{7}$$

Please refer to [11] for more detailed proof of HSIC.

Table 1: Performance (mm) and training time (s) of different GDPnet variants.

Variant	HSIC	Huber	Dense	Attention	Validation	Test	Training Time
(a)					5.861	7.701	98m58s
(b)	✓				5.842	7.655	49m24s
(c)		✓			5.867	7.665	**46m14s**
(d)	✓		✓		5.858	7.628	49m50s
(e)	✓			✓	5.783	7.576	50m11s
(f)	✓		✓	✓	**5.775**	**7.520**	52m35s

Table 2: Quantitative results on VOCASET dataset (mm).

	Validation			Test			Noise
	$Speaker_1^{val}$	$Speaker_2^{val}$	Mean	$Speaker_1^{test}$	$Speaker_2^{test}$	Mean	
VOCA [5]	**4.073**	7.649	5.861	9.657	5.844	7.701	7.890
GDPnet	4.084	**7.467**	**5.775**	**9.377**	**5.663**	**7.520**	**7.721**

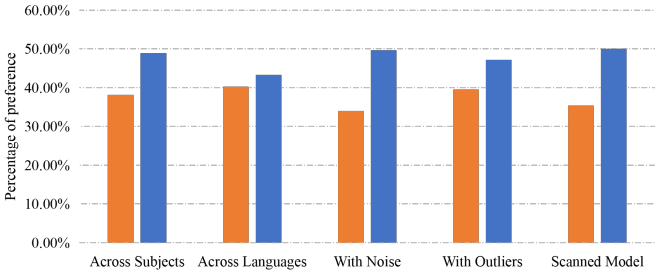


Figure 6: User study result. The blue bars show the percentage of people who give higher scores for our GDPnet than VOCA [5], and the orange bars are the opposite.

Table 3: Results of Wilcoxon signed rank test in five cases of the user study.

Cases	VOCA [5]	GDPnet	z	p
	Mean	Mean		
Across Subjects	4.98	5.55	7.959	< 0.01
Across Languages	5.00	5.43	7.645	< 0.01
With Noise	4.89	5.41	7.332	< 0.01
With Outliers	4.90	5.38	4.863	< 0.01
Scanned Models	4.89	5.42	7.664	< 0.01

3.4 Loss Function

The loss of the proposed GDPnet consists of three parts, *i.e.*, reconstruction loss, constraint loss and velocity loss:

$$L = L_r + \lambda_1 L_c + \lambda_2 L_v, \tag{8}$$

where λ_1 and λ_2 are positive constants to balance loss terms. The reconstruction loss L_r computes the distance between the predicted output and the ground truth:

$$L_r = \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_F^2 . \tag{9}$$

During the training stage, the reconstruction representation constraint L_c could use Huber or HSIC as we discuss in Section 3.3. The choice of these two constraints is a trade-off, as Huber constraint has faster convergence and HSIC constraint has better performance, which will be discussed in Section 4.2. Besides, we have the velocity loss

$$L_v = \|(\mathbf{y}_i - \mathbf{y}_{i-1}) - (\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_{i-1})\|_F^2 , \tag{10}$$

to induce temporal stability, which considers the smoothness of prediction and ground truth in the sequence context.

3.5 Implementation Details

Our GDPnet is implemented using Tensorflow [1] and trained with the Adam optimizer [22] on an NVIDIA GeForce GTX 1080 Ti GPU. We train our model for 50 epochs with a learning rate of $1e-4$ without learning rate decay. We use Adam with a momentum of 0.9, which optimizes the loss function between the output mesh and the ground-truth mesh. The balancing weights for loss terms are set to $\lambda_1 = 0.1$ and $\lambda_2 = 10.0$, respectively. For network architecture, we use a windows size of $W = 16$ with $D = 29$ speech features, and set the dimension of latent representation as 64.

4 Experiments

In this section, we first introduce the experimental setup including the dataset, training setup and the metric. Then, we evaluate the performance of our GDPnet quantitatively and qualitatively compared with the state-of-the-art method. We also conduct a blind user study. Finally, we perform an ablation study to analyze the effects of different components of our approach.

4.1 Experimental Setup

4.1.1 Dataset

VOCASET [5] provides high-quality 3D scans with about 29 minutes of 4D scans captured at 60 fps as well as alignments of the entire head including the neck. The raw 3D head scans are registered with a sequential alignment method using the publicly available generic FLAME model [25]. Each registered mesh has 5023 vertices with 3D coordinates. In addition to high-quality face models, VOCASET also provides the corresponding voice data, which is very useful to train and evaluate speech-driven 3D facial animation. In total, it has 12 subjects and 480 sequences each

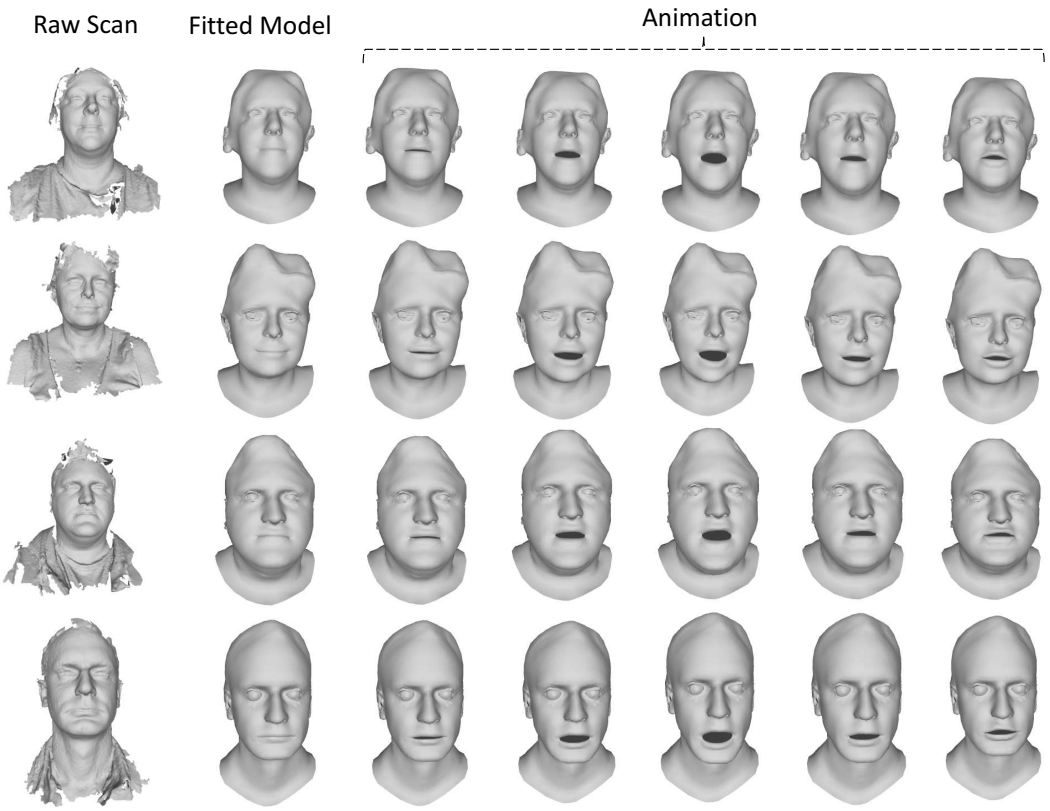


Figure 7: Our method generalizes across various scanned models from 3dMD dataset [9].

containing a sentence spoken in English with a duration of 3-5 seconds. The sentences are taken from a diverse corpus similar to [8]. As we know, the posture, head rotation and other subjective information of the speaker cannot be completely judged only by voice. In order to eliminate the influence of pose and distortion on the model, we only use the unposed data for training, so that we can effectively make use of the template information to obtain more realistic animation results using the unknown voices.

4.1.2 Training Setup

In order to train and test effectively, we split 12 subjects into a training set, a validation set and a test set, as VOCA [5] did. Furthermore, we split the remaining subjects as 2 for validation and 2 for testing. The training set consists of all sentences of eight subjects. For the validation and test sets, 20 unique sentences are selected so that they are not shared with any other subject. The specific data division is shown in Figure 5. Note that there is no overlap between training, validation and test sets for subjects or sentences.

4.1.3 Metric

To quantitatively evaluate the performances of the proposed method and the compared method, we adopt mean squared error (MSE), *i.e.*, the average squared difference between the estimated value and the ground-truth value. Specifically, the MSE between the generated mesh $\hat{y}$ and the ground-truth mesh y is defined as:

$$mse\left(\hat{y}, y\right) = \frac{1}{N}\sum_{i=1}^{N}\left\|v_i - \hat{v}_i\right\|_2, \tag{11}$$

where v is a vertex of the mesh, and N is the number of vertices.

4.2 Ablation Study

Furthermore, we study the impact of different components in our GDPnet. Specifically, we analyze four key components: HSIC constraint, Huber constraint, dense connection structure in the encoder and attention mechanism. By taking one or several components into account, we obtain six variants as follows:
(a) without any of the components;
(b) with HSIC constraint loss to leverage geometry-guided training strategy;
(c) with Huber constraint loss to leverage geometry-guided training strategy;
(d) with HSIC constraint and dense connection structure in the encoder;
(e) with HSIC constraint and attention mechanism in the decoder;
(f) with HSIC constraint, dense connect structure and attention mechanism.

In Table 1, we compare the mean squared errors of different variants on the validation set and the test set, together with the training time. The training convergence time is the time when the best validation accuracy was achieved. With HSIC or Huber constraint, the adoption of our geometry-guided training strategy will speed up the training convergence of the network (less than half of the time). Besides, the accuracy is improved from 7.701 to 7.655 by using the HSIC/Huber regularization. The convergence speed of using Huber constraint is the fastest, because the calculation time of Huber loss is less than that of HSIC loss. However, the performance of using Huber constraint is slightly worse than that of using HSIC constraint, since the correlation measurement of HSIC is more consistent with this task. Therefore, we use the HSIC constraint in the following comparison experi-

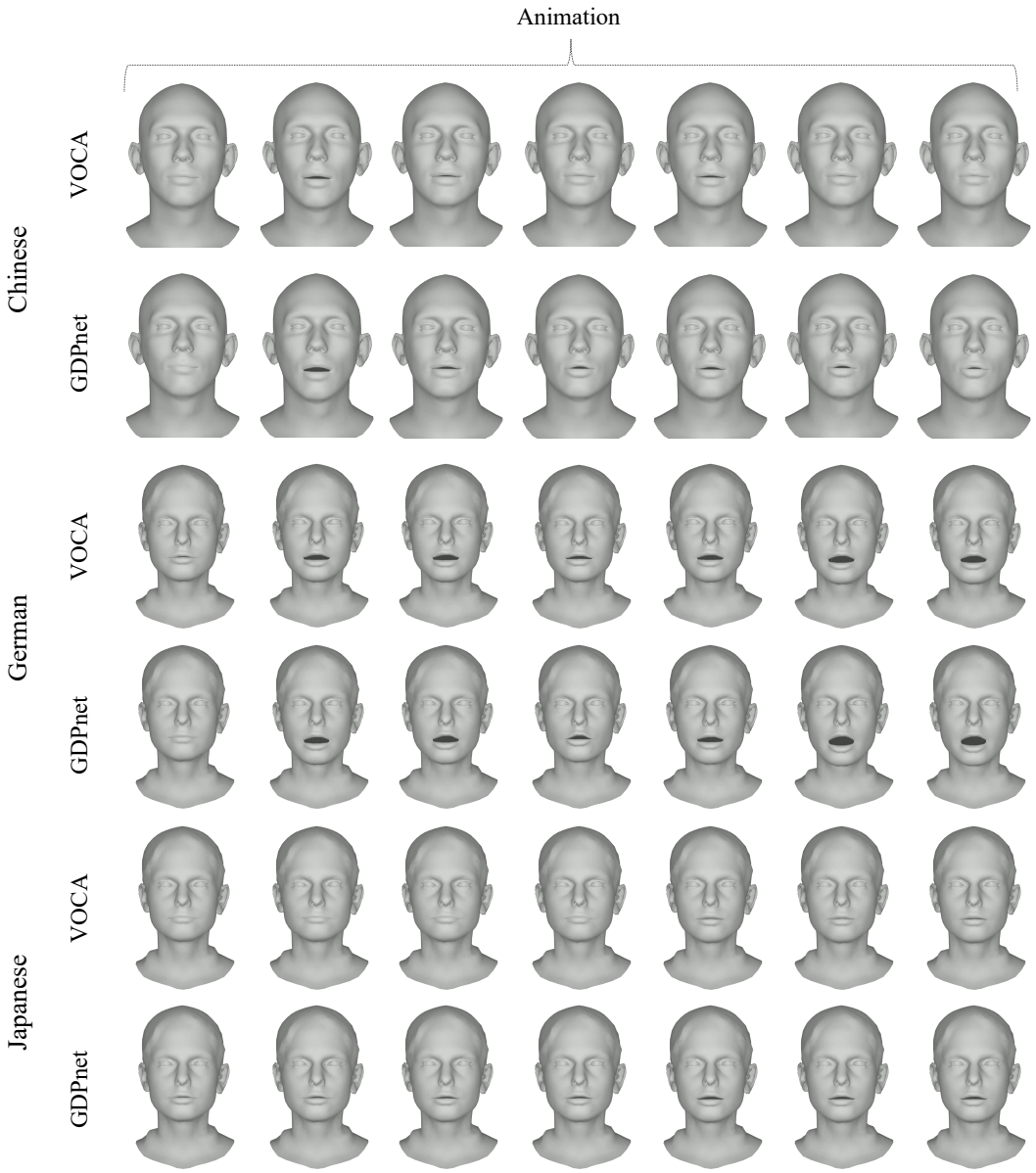


Figure 8: Our method generalizes natural and realistic animations across languages, compared with VOCA [5].

ments. In summery, each module in our GDPnet can improve the performance of animation effectively, especially when using both dense connection structure and attention mechanism.

4.3 Comparison

In this section, we compare our method with a state-of-the-art method, VOCA [5], quantitatively and qualitatively with a user study. VOCA [5] is the only state-of-the-art method that achieves the same goal with our work: generating realistic 3D facial animation given an audio in any language and any 3D face model.

4.3.1 Quantitative Evaluation

We first evaluate the quantitative results of our GDPnet method and VOCA [5] on the VOCASET dataset. For fair comparison, we use the same dataset split as VOCA [5]. As presented in Table 2, we calculate the mean squared error for each subject in the validation set and the test set. It can be seen that the overall performance of our model is better than VOCA, demonstrating the better generalization ability of our model. In order to more clearly formulate the different speakers in the validate and test sets, we denote the ith subject in the validate set as $Speaker_i^{val}$ similar to the test set. Our GDPnet improves accuracy by $0.182mm$ on $Speaker_2^{val}$ and achieves competitive performance on $Speaker_1^{val}$ in the validation set. It is worth noting that, in the test set, our method reduces $0.280mm$ error for $Speaker_1^{test}$ and error by $0.181mm$ for $Speaker_2^{test}$. This proves that GDPnet is more generalized than VOCA. Some visual results are shown in Figure 3. The per-vertex errors are color-coded on the reconstructed mesh for visual inspection. Our method obtains more accurate results which are closer to the ground truths.

In order to evaluate the robustness of our method, we combine a speech signal with Gaussian noise, natural noise[1] or outliers, and use the polluted signal as the input. The averaged errors over

1. From http://soundbible.com

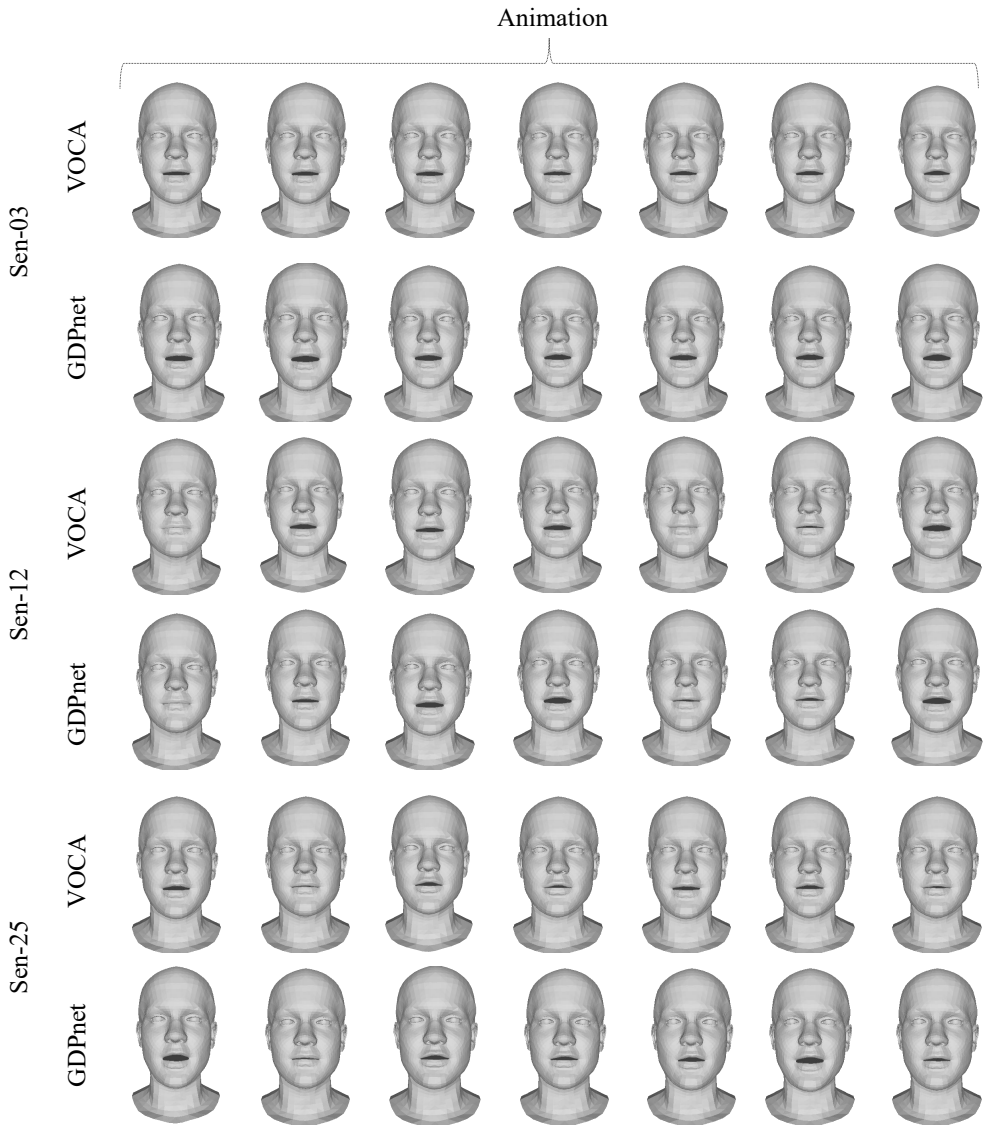


Figure 9: Our method generalizes natural and realistic animations across sentences, compared with VOCA [5].

the noisy cases are given in Table 2. Our method also obtains more accurate results for the noisy cases. Some visual results with Gaussian noisy inputs are shown in Figure 4. The per-vertex errors are color-coded on the reconstructed mesh for visual inspection. More results with different noises are shown in Figure 10. These results demonstrate that our GDPnet is more robust to noise and outliers.

4.3.2 Qualitative Evaluation and User Study

To evaluate the generalizability of our method, we perform qualitative evaluation and perceptual evaluation with a user study, compared with the state-of-the-art method.

In the study, we show the video results of VOCA [5] (Method A) and our method (Method B) for the same sentence in five cases with two examples per case (12 questions in total including question related to gender and age of the participant): generalization across unseen subjects, generalization across languages, robustness to noise, robustness to outliers, and application to scanned models, which is a side by side comparison [23]. The users are required to score the results of VOCA [5] (Method A) and our method (Method B) separately from 1 to 7 where 1 means extremely poor and 7 means excellent. In the user study, we have collected 144 answers, including 76 females and 68 males with different ages (2 users under 18, 128 users between 18 and 40, 13 users between 40 and 60, and 1 user above 60). Table 3 gives the results of Wilcoxon signed rank test [32] in the five cases. The p values are less than 0.01, which means there is great significant difference between the results of the two methods. The mean values of our model are better than VOCA [5]. This demonstrates that our model has a better performance than VOCA [5]. Figure 6 shows the results of user preference for VOCA [5] and our method in the five cases. The blue bars show the percentage of people who give higher scores for our method than VOCA [5], and the orange bars are the opposite. As shown in the figure, our method has higher approval ratings.

- **Generalization across unseen subjects and real scanned subjects:** Our method can animate any model that has the consistent topology with the FLAME. To demonstrate the generalization capability of our method, we non-rigidly register the FLAME model against several

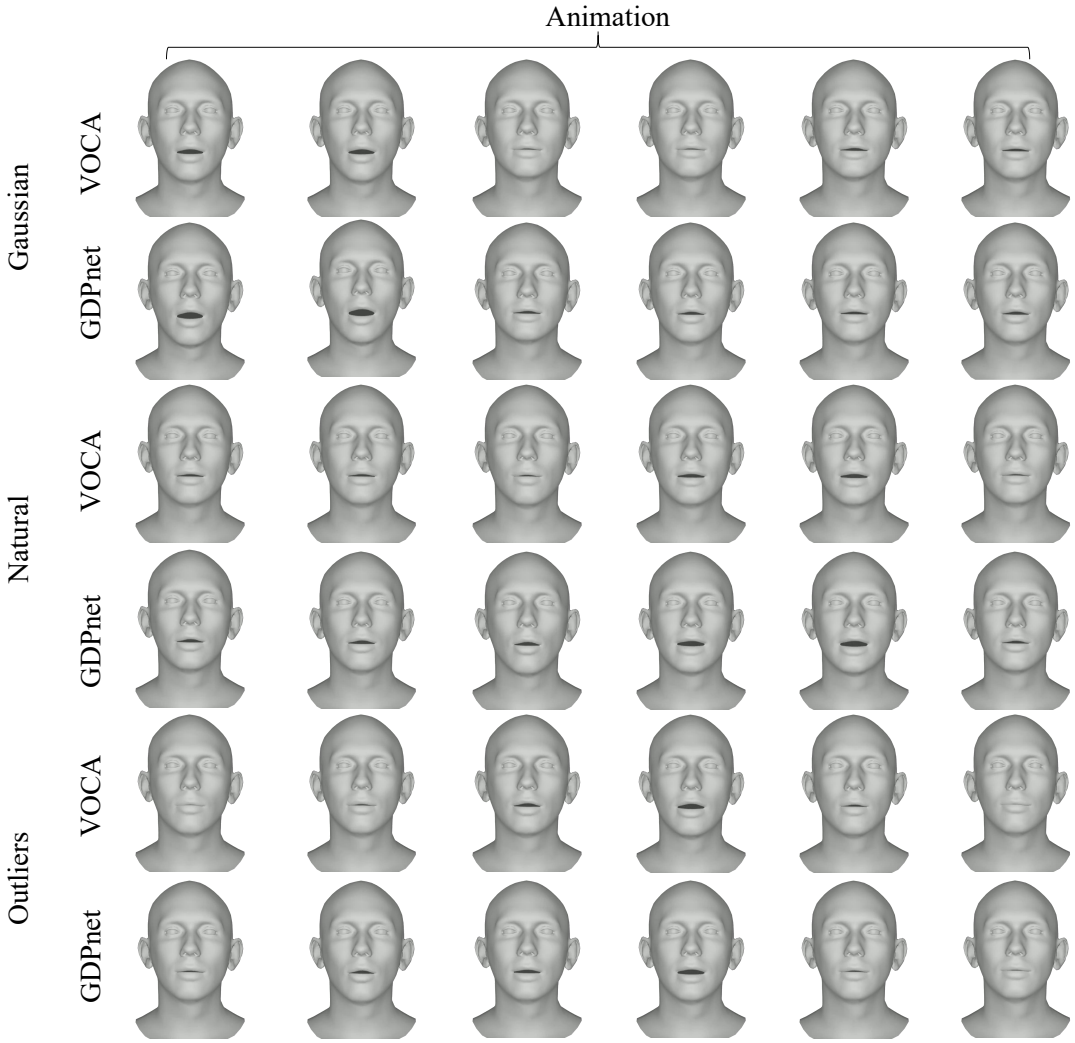


Figure 10: Our method is robust to various noise and outliers in the input audio, compared with VOCA [5].

scanned models from 3dMD dataset [9], a self-scanned model and a model of Albert Einstein downloaded from TurboSquid [2]. Specifically, we first manually define some 3D landmarks and fit the FLAME model to these 3D landmarks. Then, we adopt ED graph-based non-rigid deformation and per-vertex refinement to obtain a fitted mesh with geometry details. Figure 1 shows some animation results on unseen subjects in VOCASET [5], D3DFACS [4] and our fitted dataset, driven by the same audio sequence. Figure 7 gives more results on our fitted dataset. Video results compared with VOCA [5] are shown on the project website[3]. Our method achieves more reasonable and realistic 3D facial animation results.

- **Generalization across languages and sentences:** Although trained with speech signals in English, our model can generate animation results in any language. Because our 3D face animation results are generated according to the signal characteristics of the speech instead of the word or language, our model generalizes well across languages and sentences. The advantages of our method are fast convergence, reasonable output, robustness to noise and outliers, and good generalization across unseen subjects, languages and sentences. Figure 8 shows some examples of our generalization across languages, compared with VOCA [5], and Figure 9 shows some examples of our generalization across sentences. The video on the project website[3] gives the detailed results. The results demonstrate that our method is able to achieve more obvious facial motion for these examples with different languages and sentences in these examples.

- **Robustness to noise and outliers:** To demonstrate our robustness to noise and outliers, we combine a speech signal with Gaussian noise, natural noise or outliers, and use the polluted signal as the input. Figure 10 shows a comparison between VOCA [5] and our model. Benefiting from the geometry-guided training strategy, our model not only has a faster training convergence time, but also has better robustness. Also, the video on the project website[3] shows more visual results.

4.4 Failure Case and Discussion

In terms of the whole sequence, there are no obviously wrong cases for our results. By carefully comparing the differences

2. https://www.turbosquid.com
3. http://cic.tju.edu.cn/faculty/likun/projects/GDPnet

between the predicted result and the ground-truth for each frame, some small differences, *e.g.*, the subtle change of expression and the range of mouth opening and closing, can be occasionally found. Figure 11 gives an example of this failure case, which is wrong (or inaccurate) only at a certain time instance. The ground-truth is closing the mouth, while our estimated model opens the mouth a little by taking an action for the sound signal. That is to say, for the speech corresponding to the mouth fully closed, our method cannot judge the expression of the speaker simply from the voice. In addition to the fundamental shortcoming of data-driven methods that struggle to generate extreme cases, the limited speaking styles are also a main reason. The current model only uses a separate identity coding for the style control and the style attributes are not explicitly modeled, which results in the poor diversity of style for the animation. Only using audio features cannot achieve perfect 3D facial animation. In the future work, we will use a GAN-based approach to achieve richer style changes.

5 Conclusion

In this paper, we propose a geometry-guided dense perspective network (GDPnet) to animate a 3D template model of any person speaking the sentences in any language. We design an encoder with dense connection to strengthen feature propagation and encourage the re-usage of audio features, and a decoder with attention mechanism to better regress the final 3D facial mesh. We also propose a geometry-guided training strategy with two constrains from different perspectives to achieve more robust animation. Experimental results demonstrate that our method achieves more accurate and reasonable animation results and generalizes well to unseen subjects.

Figure 11: One failure case using our method.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (62171317, 62122058, 61771339). We are grateful to the Associate Editor and anonymous reviews for their help in improving this paper.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang. Diversity-induced multi-view subspace clustering. In *Proc. CVPR*, 2015.

[3] Y. Cao, W. C. Tien, P. Faloutsos, and F. H. Pighin. Expressive speech-driven facial animation. *ACM Trans. on Graphics*, 24:1283–1302, 2005.

[4] D. Cosker, E. Krumhuber, and A. Hilton. A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling. In *Proc. ICCV*, pages 2296–2303. IEEE, 2011.

[5] D. Cudeiro, T. Bolkart, C. Laidlaw, A. Ranjan, and M. J. Black. Capture, learning, and synthesis of 3D speaking styles. In *Proc. CVPR*, June 2019.

[6] C. Ding, L. Xie, and P. Zhu. Head motion synthesis from speech using deep neural networks. *Multimedia Tools and Applications*, 74(22):9871–9888, 2015.

[7] P. Edwards, C. Landreth, E. Fiume, and K. Singh. JALI: an animator-centric viseme model for expressive lip synchronization. *ACM Trans. on Graphics*, 35(4):1–11, 2016.

[8] W. M. Fisher. The DARPA speech recognition research database: specifications and status. In *Proc. DARPA Workshop on Speech Recognize*, pages 93–99, 1986.

[9] A. Ghosh, G. Fyffe, B. Tunwattanapong, J. Busch, X. Yu, and P. Debevec. Multiview face capture using polarized spherical gradient illumination. *ACM Trans. on Graphics*, 30(6):129, 2011.

[10] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proc. AISTATS*, 2011.

[11] A. Gretton, O. Bousquet, A. J. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Proc. ALT*, 2005.

[12] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng. Deep speech: Scaling up end-to-end speech recognition. *Computer Science*, abs/1412.5567, 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016.

[14] P. Hong, Z. Wen, and T. S. Huang. Real-time speech-driven face animation with expressions using neural networks. *IEEE Trans. on Neural Networks*, 13(4):916–927, 2002.

[15] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2017.

[16] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. In *Proc. CVPR*, pages 2261–2269, 2016.

[17] P. J. Huber. Robust estimation of a location parameter. In *Annals of Mathematical Statistics*, 1964.

[18] Z.-H. Jiang, Q. Wu, K. Chen, and J. Zhang. Disentangled representation learning for 3D face shape. In *Proc. CVPR*, pages 11957–11966, 2019.

[19] P. Kakumanu, R. Gutierrez-Osuna, A. Esposito, R. Bryll, A. Goshtasby, and O. Garcia. Speech driven facial animation. In *Proc. of the 2001 Workshop on PUI*, pages 1–5, 2001.

[20] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Trans. on Graphics*, 36:94:1–94:12, 2017.

[21] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Nießner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt. Deep video portraits. *ACM Trans. on Graphics*, 37(4):1–14, 2018.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] T. Kucherenko, P. Jonell, S. van Waveren, G. E. Henter, S. Alexandersson, I. Leite, and H. Kjellström. Gesticulator: A framework for semantically-aware speech-driven gesture generation. In *Proc. ICMI*, pages 242–250, 2020.

[24] K. Li, J. Liu, Y.-K. Lai, and J. Yang. Generating 3D faces using multi-column graph convolutional networks. In *Proc. CGF*, 2019.

[25] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4D scans. *ACM Trans. on Graphics*, 36(6), 2017.

[26] Y. Liu, F. Xu, J. Chai, X. Tong, L. Wang, and Q. Huo. Video-audio driven real-time facial animation. *ACM Trans. on Graphics*, 34:182:1–182:10, 2015.

[27] K. W.-D. Ma, J. P. Lewis, and W. B. Kleijn. The hsic bottleneck: Deep learning without back-propagation. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, abs/1908.01580, 2019.

[28] Y. Pei and H. Zha. Transferring of speech movements from video to 3D face space. *IEEE Trans. on Visualization and Computer Graphics*, 13(1):58–69, 2007.

[29] H. X. Pham, S. Cheung, and V. Pavlovic. Speech-driven 3D facial animation with implicit emotional awareness: A deep learning approach. In *Proc. CVPRW*, pages 2328–2336, 2017.

[30] H. X. Pham, Y. Wang, and V. Pavlovic. End-to-end learning for 3D facial animation from speech. In *Proc. ICMI*, 2018.

[31] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3D faces using convolutional mesh autoencoders. In *Proc. ECCV*, 2018.

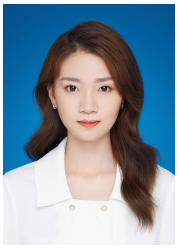
[32] D. Rey and M. Neuhäuser. Wilcoxon-signed-rank test. In *International Encyclopedia of Statistical Science*, pages 1658–1659. Springer, Berlin, Heidelberg, 2011.

[33] G. Salvi, J. Beskow, S. Al Moubayed, and B. Granström. SynFace-speech-driven facial animation for virtual speech-reading support. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009(1):191940, 2009.

[34] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Trans. on Graphics*, 36:95:1–95:13, 2017.

[35] S. Taylor, A. Kato, I. Matthews, and B. Milner. Audio-to-visual speech conversion using deep neural networks. In *Proc. INTERSPEECH*, 2016.

[36] S. L. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. K. Hodgins, and I. A. Matthews. A deep learning approach for generalized speech animation. *ACM Trans. on Graphics*, 36:93:1–93:11, 2017.
[37] S. L. Taylor, M. Mahler, B.-J. Theobald, and I. Matthews. Dynamic units of visual speech. In *Proc. EUROSCA*, pages 275–284, 2012.
[38] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of RGB videos. In *Proc. CVPR*, pages 2387–2395, 2016.
[39] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Trans.on Graphics*, 30(4):1–10, 2011.
[40] Q. Wu, J. Zhang, Y.-K. Lai, J. Zheng, and J. Cai. Alive caricature from 2D to 3D. In *Proc. CVPR*, pages 7336–7345, 2018.
[41] C. Xu, W. Huang, H. Wang, G. Wang, and T.-Y. Liu. Modeling local dependence in natural language with multi-channel recurrent neural networks. In *Proc. AAAI*, volume 33, pages 5525–5532, 2019.
[42] Zhigang Deng, U. Neumann, J. P. Lewis, Tae-Yong Kim, M. Bulut, and S. Narayanan. Expressive facial animation synthesis by learning speech coarticulation and expression spaces. *IEEE Trans. on Visualization and Computer Graphics*, 12(6):1523–1534, 2006.
[43] Y. Zhou, S. Xu, C. Landreth, E. Kalogerakis, S. Maji, and K. Singh. Visemenet: audio-driven animator-centric speech animation. *ACM Trans. on Graphics*, 37:161:1–161:10, 2018.



Jingying Liu received the B.E. degree from the Harbin engineering University, Harbin, China, in 2018. She is currently pursing the M.S. degree with the College of Intelligence and Computing, Tianjin University, Tianjin, China. Her research interests include 3D reconstruction and deep learning.

Binyuan Hui received the B.E. degree from the Northeastern University, Shenyang, China, in 2018. He is currently pursing the M.S. degree with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interest is deep learning.

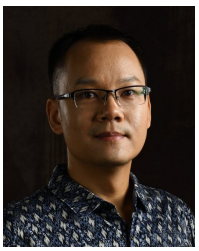
Kun Li received the B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2006, and the master and Ph.D. degrees from Tsinghua University, Beijing, in 2011. She visited École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2012 and from 2014 to 2015. She is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. Her research interests include dynamic scene 3D reconstruction and image/video processing.

Yunke Liu received the B.E. degree from the School of Computer, Northeastern University at Qinhuangdao, China, in 2019. She is currently pursuing the M.E. degree with College of Intelligence and Computing, Tianjin University, Tianjin, China. Her interests include 3D reconstruction and computer vision.

Yu-Kun Lai received his bachelor and Ph.D. degrees in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a Professor in the School of Computer Science & Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing and computer vision. He is on the editorial boards of Computer Graphics Forum and The Visual Computer.

Yuxiang Zhang is currently pursuing the Ph.D. degree with Department of Automation, Tsinghua University, Beijing, China. His research interests include computer vision and graphics.

Yebin Liu received the BE degree from Beijing University of Posts and Telecommunications, China, in 2002, and the PhD degree from the Department of Automation, Tsinghua University, Beijing, China, in 2009. He has been working as a research fellow at the computer graphics group of the Max Planck Institute for Informatik, Germany, in 2010. He is currently an associate professor in the Department of Automation, Tsinghua University. His research areas include computer vision, computer graphics and computational photography.



Jingyu Yang (M'10-SM'17) received the B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2003, and Ph.D. (Hons.) degree from Tsinghua University, Beijing, in 2009. He has been a Faculty Member with Tianjin University, China, since 2009, where he is currently a Professor with the School of Electrical and Information Engineering. He was with Microsoft Research Asia (MSRA) in 2011, and the Signal Processing Laboratory, EPFL, Lausanne, Switzerland, in 2012, and from 2014 to 2015. His research interests include image/video processing, 3D imaging, and computer vision.