

MimicTalker: A Multimodal Interactive and Memory-Enhanced Framework for Real-Time Dyadic 3D Head Generation

Supplementary Material

Yinuo Wang^{1,*,\ddagger}, Yanbo Fan^{2*,\ddagger}, Xuan Wang¹, Boyao Zhou³, Yu Guo^{1,\ddagger}, Yujun Shen³, Fei Wang¹

¹State Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University

²Nanjing University ³Ant Group

In this supplementary material, we provide details of the proposed network structures (Sec. A.1), training objectives (Sec. A.2), inference details (Sec. A.3), more experimental results (Sec. B), user study (Sec. C), discussion of limitations and future works (Sec. D), and ethics (Sec. E).

A. Implementation Details

A.1. Network Structures

As shown in Fig. 2 of the main paper, the network mainly consists of Multimodal Interactive Context Extraction, Semantics-Enhanced Dynamic Interaction, and Semantic-Guided Motion Style Memory. In this section, we provide comprehensive implementation details of these modules.

Multimodal Interactive Context Extraction. The motion encoder is a two-layer MLP with ReLU activations, which maps the 56-dim motion coefficients of Speaker B into a shared latent space with 256-dim audio features. MFCC Encoder first extracts $(4N \times 80)$ -dim Mel-Frequency Cepstral Coefficients (MFCC) from Speaker B's raw audio (sampled at 16khz), where N is the number of frames. Then a 1D convolutional layer with a stride of 4 compresses the MFCC temporally into a $(N \times 256)$ -dim representation, ensuring that the audio features and motion features of Speaker B are aligned frame-wise. Audio and motion features are merged to obtain the instantaneous features using a single cross-attention layer, followed by three self-attention layers, all of which are applied with causal masks. The Contextually-Aware Memory employs a memory size of $K = 4$.

Multimodal Interactive Context Extraction. After extracting Speaker A's high-dimensional audio features $\mathbf{H}_A \in \mathbb{R}^{2N \times 384}$ using Whisper encoder [31], we use a 1D convolutional layer with a stride of 2 to obtain Speaker A's audio feature $\mathbf{Z}_A \in \mathbb{R}^{N \times 256}$ aligned with Speaker B's feature frame-wise, followed by three self-attention layers to further enhance the temporal dynamics.

The Real-Time Interaction block consists of a cross-attention layer and an adaLN layer. First, we capture the real-time interaction between Speaker A and B using cross-attention,

$$\mathbf{Z}'_B = \text{CrossAttn}(\mathbf{F}_A, \mathbf{Z}_B, \mathcal{M}). \quad (10)$$

Then, interaction feature \mathbf{Z}'_B is integrated with Speaker A's feature \mathbf{F}_A frame-wise in real-time with adaLN blocks,

$$(\gamma_B, \beta_B) = \text{MLP}(\mathbf{Z}'_B), \mathbf{F}'_{AB} = (1 + \gamma_B)\mathbf{F}_A + \beta_B. \quad (11)$$

Similarly, the Long-Term Interaction block captures the long-term interaction between Speaker A and B using cross-attention,

$$\mathbf{Z}''_B = \text{CrossAttn}(\mathbf{F}'_{AB}, \text{MLP}(\mathbf{m}^{(1:N)}), \mathcal{M}). \quad (12)$$

Then, the long-term interaction feature is integrated with adaLN blocks,

$$(\gamma_m, \beta_m) = \text{MLP}(\mathbf{Z}''_B), \mathbf{F}_{AB} = (1 + \gamma_m)\mathbf{F}'_{AB} + \beta_m. \quad (13)$$

Semantic-Guided Motion Style Memory. The style encoder consists of three Transformer encoder layers [38] followed by a self-attention pooling layer [24]. During the training of the style encoder, we use a contrastive loss where consecutive segments from the same training clip are treated as positive samples while segments from different training clips are treated as negative samples [35]. The loss function is as follows,

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{s}_i, \mathbf{s}_j)/\tau)}{\sum_{k=1}^N \mathbf{1}_{k \neq i} \exp(\text{sim}(\mathbf{s}_i, \mathbf{s}_k)/\tau)}, \quad (14)$$

where we use cosine similarity metric for the sim function, $\mathbf{1}_{k \neq i}$ is an indicator function, τ represents the temperature parameter, and \mathbf{s} denotes the style code extracted by the style encoder.

A.2. Training Objectives

The training objective incorporates both motion accuracy and motion naturalness. For motion accuracy, we use Mean

*Equal contribution

^{\ddagger}Corresponding authors

^{\ddagger}This work was done during Yinuo Wang's internship at Ant Group

Extractor	FD↓			P-FD↓			MSE↓			SID↑			rPCC↓		
	EXP	JAW ×10 ³	POSE ×10 ²	EXP	JAW ×10 ³	POSE ×10 ²	EXP ×10 ¹	JAW ×10 ³	POSE ×10 ²	EXP	JAW	POSE	EXP ×10 ²	JAW ×10 ¹	POSE ×10 ¹
GPT-4o	7.12	1.32	2.71	8.20	1.41	2.93	3.09	0.93	1.67	3.63	2.40	1.90	4.16	1.18	2.03
Phi-3.5-mini	7.16	1.32	2.77	8.24	1.41	3.00	3.09	0.93	1.69	3.61	2.40	1.88	4.25	1.17	2.05
GPT-4o	15.89	2.27	4.15	17.25	2.38	4.42	5.28	1.36	2.29	3.04	2.11	1.57	6.65	1.48	2.59
Phi-3.5-mini	15.91	2.29	4.22	17.27	2.39	4.49	5.27	1.36	2.31	3.03	2.11	1.57	6.70	1.48	2.64

Table 4. Results using different LLMS on DualTalk dataset. The top half is the result on the test set, and the bottom half is the result on the OOD set.

Squared Error (MSE) loss between the predicted head motion $\hat{\mathbf{M}}_A$ and ground truth head motion \mathbf{M}_A as follows,

$$\mathcal{L}_{mot} = \text{MSE}(\hat{\mathbf{M}}_A, \mathbf{M}_A) = \frac{1}{N} \sum_{i=1}^N (\hat{\mathbf{M}}_A^{(i)} - \mathbf{M}_A^{(i)})^2. \quad (15)$$

where N is the sequence length.

To ensure motion naturalness, we compute loss between the predicted and ground truth head motion velocities using MSE. The head motion velocity \mathbf{V} is calculated as the motion difference of consecutive frames,

$$\mathbf{V}^{(i)} = \mathbf{M}^{(i+1)} - \mathbf{M}^{(i)}, i = 0, 1, \dots, N-1. \quad (16)$$

Then, the velocity loss is computed as follows,

$$\mathcal{L}_{vel} = \text{MSE}(\hat{\mathbf{V}}_A, \mathbf{V}_A) = \frac{1}{N-1} \sum_{i=1}^{N-1} (\hat{\mathbf{V}}_A^{(i)} - \mathbf{V}_A^{(i)})^2. \quad (17)$$

Finally, the total loss is the sum of these two components,

$$\mathcal{L} = \mathcal{L}_{mot} + \mathcal{L}_{vel}. \quad (18)$$

A.3. Inference Details

During inference, MimicTalker generates head motions in real-time conversations. Specifically, at the start of a conversation when data is insufficient, the semantic information and style code are set to default values (e.g. *the intention / topic is currently undetermined* for the intentions and topic, and an all-zero value for the style encoder). Then, for every k rounds of conversation, we feed the conversation transcript from these k rounds as well as the previous k rounds to the LLM, and let it decide whether the intention or topic needs to be updated. In the meantime, for every w seconds, we extract the motion style of the generated motion within this w -seconds interval using the style encoder and store it in the MSM, with the intention at that period as the key. When generating the motion style of the next frame, the current intention of the interactive head is used as a query to retrieve the motion style for guidance, as described in Sec. 3.4 of the main paper.

B. Additional Experimental Results

B.1. Additional Visual Results

We provide additional visual results of MimicTalker compared with other methods, including FaceFormer [11], CodeTalker [40], SelfTalk [28], L2L [25], EmoTalk [29], and DualTalk [30]. As shown in Fig. 4, our method outperforms existing methods in terms of motion accuracy, style consistency, and emotion responsiveness. From Fig. 4 (a) to (c), it can be seen that our method generates appropriate reactions to the interlocutor. For instance, it produces a serious face in response to a heavy topic in (a), a smiling face that mirrors the interlocutor’s laughter in (b), and a sad face that conveys compassion in (c). From Fig. 4 (d) to (f), it can be seen that our method generates precise lip motions across various emotional expressions, such as speaking while laughing in (d), speaking while recalling a memory in (e), and speaking with a sad expression in (f).

B.2. Using A Lightweight Local LLM

To examine whether our method depends on a specific LLM, we replace GPT-4o with Phi-3.5-mini, a lightweight locally deployed model that requires only around 8 GB of VRAM. As shown in Tab. 4, the resulting performance remains comparable. This demonstrates that MimicTalker is not tied to any specific LLMs.

C. User Study

We conduct a user study to further evaluate the performance of MimicTalker. We randomly select 20 interlocutor videos from the DualTalk dataset, and generate their corresponding interactive head videos. 15 volunteers are asked to rate (the score ranges from 1 to 5, with 5 being the best) every generated video by each evaluated method from three perspectives, i.e. 1) naturalness to evaluate whether the generated motions are natural and smooth, 2) interactivity to evaluate whether the generated motions react to the interlocutor vividly and expressively, 3) similarity to GT to evaluate whether the generated motions match the motion pattern of GT and its reactions of the interlocutor. Tab. 5 presents the results of the user study. The results demonstrate that our method significantly outperforms other methods in every aspect evaluated.

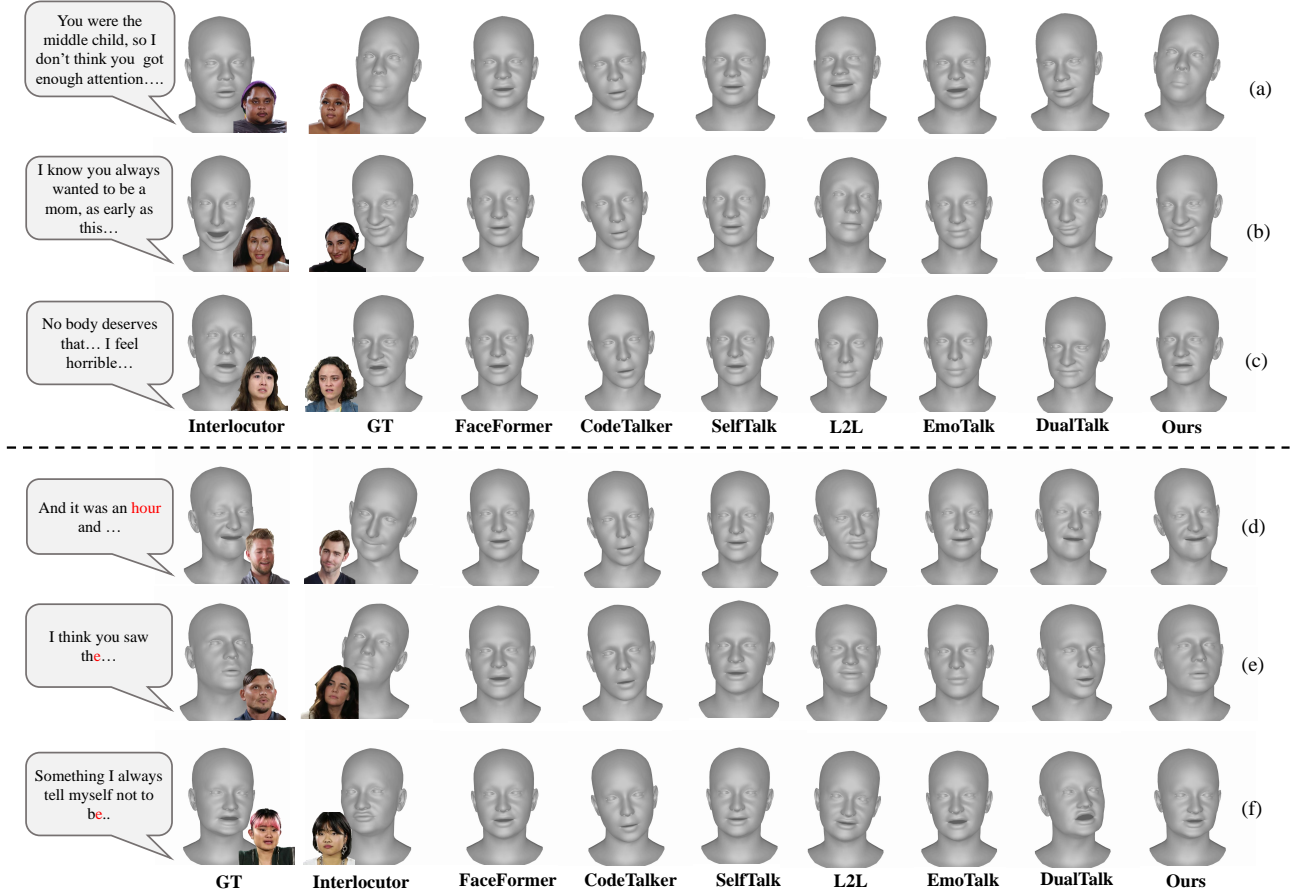


Figure 4. Visualization of the compared methods. Our method exhibits accurate motion, consistent style, vivid facial expressions, and coherent reactions. The top half is the result of the interlocutor leading the conversation. The bottom half is the result of the interactive head leading the conversation, where the phonemes corresponding to the displayed frames are marked in red.

Method	Natural- ness \uparrow	Interact- ivity \uparrow	Similarity to GT \uparrow
EmoTalk [29]	1.71	1.71	1.58
DualTalk [30]	3.55	3.48	3.37
MimicTalker	4.24	4.23	4.17

Table 5. User study results. The best results are in bold.

D. Limitations and Future Works

The main limitation of MimicTalker is the need for an additional semantic inference module for in-depth conversation analysis, which requires users to either deploy a local LLM or rely on online LLM API services.

While MimicTalker focuses on dyadic interactive head generation, real-world human-agent interaction scenarios may involve multiple users participating in a conversation simultaneously. In future work, we will investigate group interaction settings, enabling the agent to process multiple

interlocutors and generate appropriate responses for them in real time.

E. Ethics

MimicTalker may raise ethical concerns, particularly regarding privacy and potential misuse. MimicTalker is designed to synthesize head videos in real-time, face-to-face conversation scenarios, and could potentially be misused to impersonate individuals in online meetings or fabricated videos. In case of misuse and privacy violation, access control policies and watermarking can be implemented. Additionally, open-sourcing MimicTalker will be accompanied by clear guidelines to discourage unethical use.