EMAGE: Towards Unified Holistic Co-Speech Gesture Generation via Expressive Masked Audio Gesture Modeling

Supplementary Material

This supplemental document contains seven sections:

- Evlauation Metrics (Section A).
- BEAT2 Dataset Details (Section B).
- Baselines Reproduction Details (Section C).
- Settings of EMAGE (Section D).
- Visualization Blender Addon (Section E).
- Training time (Section F).
- Importance of lower body motion (Section G).

A. Evaluation Metrics

Fréchet Gesture Distance. A lower FGD, as referenced by [67], indicates that the distribution between the ground truth and generated body gestures is closer. Similar to the perceptual loss used in image generation tasks, FGD is calculated based on latent features extracted by a pretrained network,

$$\mathrm{FGD}(\mathbf{g}, \hat{\mathbf{g}}) = \|\mu_r - \mu_g\|^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{1/2}\right), \quad (11)$$

where μ_r and Σ_r represent the first and second moments of the latent features distribution z_r of real human gestures $\mathbf{g}$, and μ_g and Σ_g represent the first and second moment of the latent features distribution z_g of generated gestures $\hat{\mathbf{g}}$. We use a Skeleton CNN (SKCNN) based encoder [1] and a Full CNN-based decoder as the autoencoder pretrained network. The network is pretrained on both BEAT2-Standard and BEAT2-Additional Data. The choice of SKCNN over a Full CNN encoder is due to its enhanced capability in capturing gesture features, as indicated by a lower reconstruction MSE loss (0.095 compared to 0.103).

L1 Diversity. A higher Diversity [33] indicates a larger variance in the given gesture clips. We calculate the average L1 distance from different N motion clips as follows:

$$\text{L1 div.} = \frac{1}{2N(N-1)} \sum_{t=1}^{N} \sum_{j=1}^{N} \left\| p_t^i - \hat{p}_t^j \right\|_1, \quad (12)$$

where p_t represents the position of joints in frame t. We calculate diversity on the entire test dataset. Additionally, to compute joint positions, the translation is set to zero, implying that L1 Diversity is focused solely on local motion.

Beat Constancy (BC). A higher BC, as defined by [34], suggests a closer alignment between the gesture's rhythm and the beat of the audio. We identify the beginning of speech as the audio beat and consider the local minima of the velocity of the upper body joints (excluding fingers) as the motion beat. The synchronization between audio and gesture is computed in the following manner:

$$\mathrm{BC} = \frac{1}{g} \sum_{b_g \in g} \exp\left(-\frac{\min_{b_a \in a} \|b_g - b_a\|^2}{2\sigma^2}\right), \quad (13)$$

where g and a represent the sets of gesture beat and audio beat, respectively.

B. BEAT2 Dataset Details

Statistics. The original BEAT dataset, as described by [39], contains 76 hours of data for 30 speakers. We exclude speakers 8, 14, 19, 23, and 29, which account for 16 hours of data, due to noise in the finger data, leaving 60 hours of data for 25 speakers (12 female and 13 male). The speech and conversation portions are categorized into BEAT2-standard and BEAT2-additional, containing 27 and 33 hours respectively. We adopt an 85%, 7.5%, and 7.5% split for BEAT2-standard, maintaining the same ratio for each speaker. BEAT2-additional is utilized for further improving the network's robustness. The results presented in this paper are based on training with BEAT2-standard speaker-2 only. The dataset includes 1762 sequences with an average length of 65.66 seconds per sequence. Each recording in a sequence is a continuous answer to a daily question. Additionally, we report a comparison between TalkShow [65] and our dataset in terms of Diversity and Beat Constancy (BC), as shown in Table 8.

Table 8. **Diversity and BC Comparisons.** The local and global diversity refers to the variance in joint positions with and without global translations, respectively.

	BC ↑	Diversity-L ↑	Diversity-G ↑
TalkShow [65]	6.104	5.273	5.273
BEAT2 (Ours)	6.896	13.074	27.541

Loss Terms of MoSh++. MoSh's optimization involves loss functions including a Data Term, Surface Distance Energy, Marker Initialization Regularization, Pose and Shape Priors, and a Velocity Constancy Term, which are described as follows:

- Data Term (E_D): Minimizing the squared distance between simulated and observed markers. In the given context, the $\tilde{M}, \beta, \Theta$, and Γ represent the latent markers, body shape, poses, and body location respectively:

$$E_D(\tilde{M}, \beta, \Theta, \Gamma) = \sum_{i,t} ||\hat{m}(\tilde{m}_i, \beta, \theta_t, \gamma_t) - m_{i,t}||^2. \quad (14)$$

- Surface Distance Energy (E_S): Ensuring markers main-

tain prescribed distances from the body surface:

$$E_S(\beta, \tilde{M}) = \sum_i ||r(\tilde{m}_i, S(\beta, \theta_0, \gamma_0)) - d_i||^2. \quad (15)$$

- Marker Initialization Regularization (E_I): Penalizing deviations of estimated markers from initial positions:

$$E_I(\beta, \tilde{M}) = \sum_i ||\tilde{m}_i - v_i(\beta)||^2. \quad (16)$$

- Pose and Shape Priors: Penalizing deviations from mean shape and pose:

$$E_\beta(\beta) = (\beta - \mu_\beta)^T \Sigma_\beta^{-1} (\beta - \mu_\beta), \quad (17)$$

$$E_\theta(\Theta) = \sum_t (\theta_t - \mu_\theta)^T \Sigma_\theta^{-1} (\theta_t - \mu_\theta). \quad (18)$$

- Velocity Constancy Term (E_u): Reducing marker noise and ensuring movement consistency:

$$E_u(\Theta) = \sum_{t=2}^{n} ||\theta_t - 2\theta_{t-1} + \theta_{t-2}||^2. \quad (19)$$

The overall objective function is the weighted sum of these terms, balancing accuracy and plausibility:

$$E(\tilde{M}, \beta, \Theta, \Gamma) = \sum_{\omega \in \{D,S,\theta,\beta,I,u\}} \lambda_\omega E_\omega(\cdot). \quad (20)$$

More details and pseudo code of the head and neck shape optimization are available in the code release.

Details of FLAME Parameter Optimization. To animate a face using the SMPL-X model with ARKit parameters from the BEAT dataset, we estimate FLAME expression parameters by minimizing the geometric error between an animated ARKit and FLAME model. Addressing the optimization challenges posed by differing mesh structures, we construct an ARKit-compatible FLAME model utilizing Faceit, a Blender add-on tailored for crafting ARKit blendshapes. By driving the ARKit-aligned FLAME model with each set of ARKit parameters from the BEAT dataset, we obtain original FLAME expression parameters by minimizing the L2 distance loss between equivalent vertices. Finally, the optimized FLAME expression parameters can be directly applied to SMPL-X. For facial identity parameters, we preserve the same identity parameters on SMPL-X after body fitting with MoSh++ [46].

C. Baselines Reproduction Details

Number of Joints. All baseline methods output full-body joint rotations represented by $\mathbf{g} \in \mathbb{R}^{T\times(55\times6)}$ and, in addition to rotations, they decode global translations $\in \mathbb{R}^{T\times3}$. To provide a thorough comparison, we present subjective results for both the upper body (excluding global motion) and the full body.

Autoregressive Training. We observe that autoregressive training/inference-based models, such as FaceFormer and CodeTalker [19, 61], perform worse than non-autoregressive methods. In non-autoregressive settings, only positional embedding is used as input for cross-attention to audio features, particularly when training with Rot6D and axis-angle representations. The network architecture of FaceFormer and CodeTalker is based on transformers and was initially proposed for training with the representation of vertex offsets. As shown in Table 9, we find that non-autoregressive training improves performance with FLAME's parameters. The results in this paper are obtained using a non-autoregressive training approach. Non-autoregressive training techniques have also been employed in the training of EMAGE.

Table 9. **Vertex Errors (MSE) with Different Training Methods.** 'FF' and 'CT' refer to FaceFormer [19] and CodeTalker [61], respectively. 'TF', 'AR', and 'NonAR' represent Teacher-Force, AutoRegressive, and Non-AutoRegressive training, respectively. We train on the VOCA dataset with a vertex loss, and BEAT2 with a FLAME parameter loss combined with a vertex loss. Results indicate that the same method performs differently when using the two representations; in BEAT2, non-autoregressive training demonstrates superior performance. The average MSE is calculated on 5023 and 10475 vertices for VOCA and BEAT2, respectively:

	FF-TF	FF-AR	FF-NonAR	CT-TF	CT-AR	CT-NonAR
VOCA (x10-7)	6.636	6.023	6.138	7.914	7.637	7.541
BEAT2 (x10-7)	2.167	3.704	1.195	2.079	4.120	1.243

Adversarial Training. We omit the adversarial training in Speech2Gesture [25], CaMN [39], and Habibie *et al* [28], because their outputs with adversarial training show noticeable jitter, even when we increase the weight for the velocity loss. Similar effects are also observed in training with 3D data for Speech2Gesture [25], as reported in the study by [33].

Lower Body VQ-VAE for TalkShow. We introduce an additional VQ-VAE for TalkShow, utilizing their autoregressive (AR) model to jointly predict the class index of the upper body, hands, and lower body. The global translations are encoded in conjunction with lower body joints.

D. Settings of EMAGE

Training. We train our method for 400 epochs, gradually increasing the ratio of masked joints from 0 to 95% linearly according to the training epoch. This approach proves more

effective than a fixed masked ratio, such as 25%, based on our experiments. The learning rate is 2.5e-4, and we use the Adam optimizer with a gradient norm clipped at 0.99 to ensure stable training.

Structure of VQ-VAE. We employ the same CNN-based VQ-VAE [27] for all four body segments. The downsample rate is set to 1 to achieve the best reconstruction quality. We utilize a feature length of 512 for the codebook entries and set the codebook size to 256. The total decoding space for body gestures is represented as $\in \mathbb{R}^{T\times 256^3}$. The VQ-VAE is trained for 200 epochs, with a learning rate of 2.5e-4 for the initial 195 epochs, which is then decreased to 2.5e-4 for the last 5 epochs.

Global Motion Predictor. We train the Global Motion Predictor using an architecture that mirrors the CNN-based structure of our VQ-VAE's encoder and decoder The input consists of local motions and predicted foot contact labels $\in \mathbb{R}^{T\times 334}$, and it outputs global translations $\in \mathbb{R}^{T\times 3}$.

E. Visualization Blender Add-on

For straightforward visualization of our BEAT2 dataset, we utilize the SMPL-X Blender add-on [50]. As the latest SMPL-X add-on does not support the full range of facial expressions for SMPL-X, we extract 300 expression meshes from the original SMPL-X model and added them as individual blendshape targets into the SMPL-X model within the Blender add-on.

F. Training time

We report the training time on a single L4, V100 and 4090 with a batch size (BS) of 64 for the best performance.

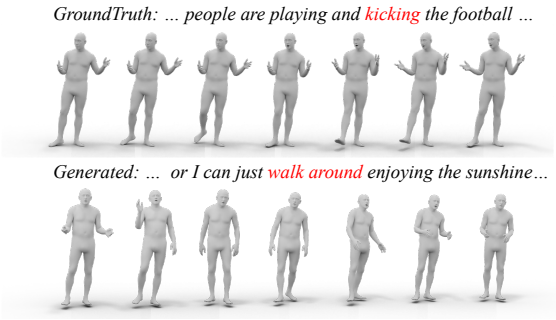
	1-speaker		25-speaker			
EMAGE	1-epoch	400-epoch	1-epoch	100-epoch	Mem.	BS
L4 (24G)	239s	26.5h	3197s	89.6h	20.1G	64
V100 (32G)	155s	17.2h	2073s	58.1h	20.1G	64
4090 (24G)	72s	8.0h	963s	27.1h	20.1G	64

Addtionally, pretraining of the 5 $\times$ VQVAEs for face, hands, upper body, lower body, and global motion would take 22.4 hours on 5 $\times$ 4090 GPUs.

	1-speaker		25-speaker			
VQVAEs $\times$ 1	1-epoch	700-epoch	1-epoch	100-epoch	Mem.	BS
L4 (24G)	200s	39.5h	2760s	74.4h	13.8G	64
V100 (32G)	131s	25.5h	1727s	48.0h	13.8G	64
4090 (24G)	61s	11.9h	802s	22.4h	13.8G	64

G. Importance of lower body motion

Lower body motion allows gestures **semantically aligned** with the content of the audio to achieve more vivid and impressive results, *e.g.*, "hiking in nature" with a walking gesture, "playing football" with a kicking motion; see figure below. Compared with the upper body, it is more weakly related to the audio, but it still has connections in the above cases.



In the implementation of EMAGE, we first obtain the latents of different body components with separate MLPs. Then, the lower body motion decoder **leverages all latents of "audio", "upper body", and "hands"** for cross-attention based lower-body motion decoding. We have also observed that decoding directly from audio increases diversity but reduces the coherence of the results on BEATv1.3.

	FGD ↓	BC˜↑	Diversity ↑	MSE↓	LVD ↓
audio only	6.209	6.683	**13.714**	1.183	8.788
audio + upper + hands	**5.423**	**6.794**	13.075	**1.180**	**8.715**