Emotional Speech-driven 3D Body Animation via Disentangled Latent Diffusion **Supplementary Material**

Kiran Chhatre[1] Radek Daněček[2] Nikos Athanasiou[2]
Giorgio Becherini[2] Christopher Peters[1] Michael J. Black[2] Timo Bolkart[2]*
[1]KTH Royal Institute of Technology, Sweden [2]Max Planck Institute for Intelligent Systems, Germany

APPENDIX

This supplementary material summarizes the video content in Appendix A and provides additional technical details of the speech disentangled model and the gesture generation model in Appendix B and Appendix C, respectively. We provide details about motion extractor model in Appendix D, discussions on the gesture emotion and semantics in Appendix E, details on the data preparation process in Appendix F, a review of state of the art methods in Appendix G, and additional information about the perceptual study in Appendix H.

A. Supplementary Video

The supplementary video shows the generated gestures. Specifically, it provides:

1. Gesture generations on various emotional audios,
2. Gesture emotion and style editing results,
3. Comparisons with state of the art mesh-based and skeleton-based gesture generation methods,
4. Ablation comparisons of the different components of our approach,
5. Gestures showing the diversity in the generations, and
6. Gestures generated from an in-the-wild audio sequence.

B. Speech Disentanglement Model

We explain the overall architecture in Appendix B.1 and the encoder–transformer architecture in Appendix B.2. We demonstrate the reconstruction mechanism to enforce disentanglement in Appendix B.3. Finally, we explain the training procedure and loss terms in Appendix B.4.

B.1. Architecture

We illustrate speech disentanglement model architecture in Fig. A.1. The training is conducted over audio of the same utterances spoken under different emotions or spoken by different speakers. Our model consists of three transformer encoders, a transformer fusion, and a transformer decoder. The input filterbank is simultaneously passed through content E_c, style E_s, and emotion E_e transformer encoders, producing three disentangled latents: content c, style s, and emotion e. The fusion and decoder are transformer-based layers. The transformer–fusion creates a single embedding by applying cross attention on the input triplet embeddings (c, e, s). Finally, the transformer–decoder reconstructs the original filter bank from the compressed single latent embedding produced by the transformer fusion.

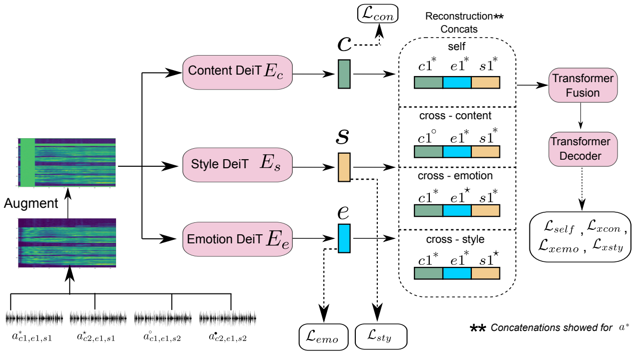


Figure A.1. **Speech disentanglement model**. An input filterbank is given to the three encoders, producing three disentangled latents, which are decoded into a reconstructed filterbank. We here show disentanglement reconstruction for one audio only, please refer Appendix B.3 for its detailed explanation.

B.2. Encoder Transformers

Similar to [6–10, 16] we employ transfer learning of vision task to our audio task by using pretrained weights of DeiT [20] (88M params) transformer that is fine-tuned on 384x384 images from ImageNet-1k [19]. We use a pretrained DeiT encoder as a component of each of the encoders, as illustrated in Fig. A.3. We linearly embed patches to features embedding of size 768 and feed them into DeiT along with trainable positional embedding of same size (768). We append class token [CLS] and distillation token

[*] Now at Google.

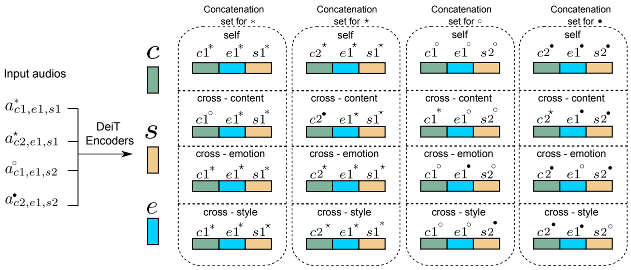

Input audios
$a^{*}_{c1,e1,s1}$
$a^{\star}_{c2,e1,s1}$
$a^{\circ}_{c1,e1,s2}$
$a^{\bullet}_{c2,e1,s2}$
DeiT Encoders
c
s
e
Concatenation set for $*$
Concatenation set for $\star$
Concatenation set for $\circ$
Concatenation set for $\bullet$
self
$c1^{*}$ $e1^{*}$ $s1^{*}$
$c2^{\star}$ $e1^{\star}$ $s1^{\star}$
$c1^{\circ}$ $e1^{\circ}$ $s2^{\circ}$
$c2^{\bullet}$ $e1^{\bullet}$ $s2^{\bullet}$
cross - content
$c1^{\circ}$ $e1^{*}$ $s1^{*}$
$c2^{\bullet}$ $e1^{\star}$ $s1^{\star}$
$c1^{*}$ $e1^{\circ}$ $s2^{\circ}$
$c2^{\star}$ $e1^{\bullet}$ $s2^{\bullet}$
cross - emotion
$c1^{*}$ $e1^{\star}$ $s1^{*}$
$c2^{\star}$ $e1^{*}$ $s1^{\star}$
$c1^{\circ}$ $e1^{\bullet}$ $s2^{\circ}$
$c2^{\bullet}$ $e1^{\circ}$ $s2^{\bullet}$
cross - style
$c1^{*}$ $e1^{*}$ $s1^{\star}$
$c2^{\star}$ $e1^{\star}$ $s1^{*}$
$c1^{\circ}$ $e1^{\circ}$ $s2^{\bullet}$
$c2^{\bullet}$ $e1^{\bullet}$ $s2^{\circ}$

Figure A.2. **Reconstruction concatenations for training forward pass**. We obtain disentangled content, emotion, and style latents from the transformer encoders. (*Self*) concatenation of triplet latent vectors is used to decode back into the original filterbank. To enforce the content disentanglement, we swap content latent vectors (*cross-content*) between given different-subjects audio pair with same utterances. Whereas to enforce style and emotion disentanglement, we swap style (*cross-style*) and emotion (*cross-emotion*) latent vectors between given same-subject audio pairs with same emotion categorical label. We repeat the procedure for quadruples of audio $\{a^{*}, a^{\star}, a^{\circ}, a^{\bullet}\}$ input in each forward pass.

[DIST] obtained from DeiT at the beginning of each filter bank sequence. We then average the 3-channel inputs of DeiT to obtain a single filterbank channel input. Finally, we use the output of the last DeiT encoder layer and project to 1D latent vector of 256 dimensions each, as our content, emotion, and style latents. We average the [CLS] and [DIST] tokens from DeiT and use it for audio emotion as well as audio style classification tasks for 8 and 30 category labels respectively.

B.3. Reconstruction Concatenations

Fig. A.2 demonstrates a detailed information of the cross-reconstruction mechanism to enforce the audio content, emotion, and style disentanglement. Each audio in the quadruple is encoded and decoded to produce the reconstructed audio filterbank. To enforce content disentanglement, we swap the content latent vectors between different-subject same-emotion audio pairs with same utterances. Similarly, we swap emotion and style latents using audio pairs from the same subject. Specifically, we swap emotion latent vector and style latent vector between same-subject same-emotion audio pairs with different utterances. The procedure is repeated for each audio in the audio quadruples.

B.4. Training and Losses

We train the speech disentanglement model on 10s-audio segments of the BEAT dataset, which provides the GT labels for emotion and subject categorical labels. We split the audio data across actors during train, validation, and test step. During training, one sample is formed by a quadruple of different audios ($a_1 = a_{c_1,e_1,s_1}, a_2 = a_{c_2,e_1,s_1}, a_3 = a_{c_1,e_1,s_2}, a_4 = a_{c_2,e_1,s_2}$), with two different contents c_1, c_2 (i.e., two different scripts), two different styles s_1, s_2 (spoken by two different subjects) and the same emotion e_1. To ensure, content, style, and emotion disentanglement, we employ a multitude of training losses. The self-reconstruction loss $\mathcal{L}_{self}$ ensures that the style, emotion, and content latents extracted from the same audio can be decoded into the original inputs:

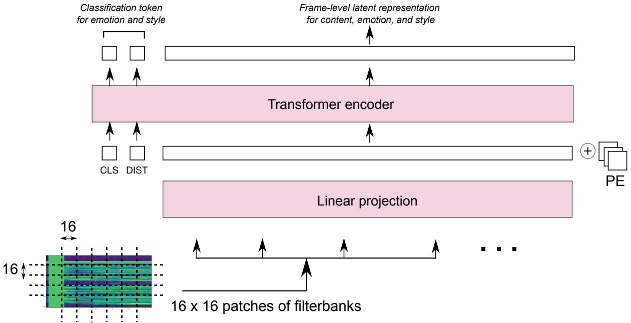

Classification token for emotion and style
Frame-level latent representation for content, emotion, and style
Transformer encoder
CLS
DIST
PE
Linear projection
16
16
16 x 16 patches of filterbanks
...

Figure A.3. **Speech encoder transformer**. We have used encoder architecture based on Touvron et al. [20]. We use this architecture as content, emotion, and style encoders in the speech disentanglement model. Following Gong et al. [6, 8], we use 10s augmented speech filterbank and split into fixed 1209 patches of 16 x 16 each, having 6 units overlap in frequency and time domain. The filterbank is passed through a linear projection layer and a learnable positional embedding (PE) is added to it.

$$\mathcal{L}_{self} = \sum_{k=1}^{4} \| D(E_c(a_k), E_s(a_k), E_e(a_k)) - a_k \|_1$$

The content loss $\mathcal{L}_{con}$ ensures that two content latents extracted from two different audios with the same content c_k but two different styles s_i, s_j match:

$$\mathcal{L}_{con} = \sum_{k=1}^{2} \| E_c(a_k) - E_c(a_{k+2}) \|_1$$

We also employ the emotion classification loss $\mathcal{L}_{emo}$ to ensure that the encoded emotion latents carry the emotion information. This is ensured by projecting them with a linear projection head into a classification vector and then computing emotion classification cross entropy loss. We use the same procedure to employ the style classification loss $\mathcal{L}_{sty}$:

$$\mathcal{L}_{emo} = -\sum_{1 \leq l_e \leq n_e} y_{l_e} \log(p_{l_e}),$$

$$\mathcal{L}_{sty} = -\sum_{1 \leq l_s \leq n_s} y_{l_s} \log(p_{l_s}),$$

with $n_e = 8$ and $n_s = 30$ denoting the number of emotion classes and training subjects respectively.

Finally, we employ the cross-reconstruction losses for emotion, style, and content. This loss ensures that we can combine any three style, content, and emotion latents and decode them into a valid reconstruction. As shown in Fig. A.1 and Fig. A.2, this is a three part cross reconstruction process. In this process, we extract content $E_c(a_*)$, emotion $E_e(a_*)$, and style $E_s(a_*)$ latents of all four different audios. Given two input audios of the different contents c_i and c_j, with the same speaker, and the same emotion, we swap the emotion latents between the audio pair, and decode the two audios back. Since the emotion class is constant within a quadruple, the emotion cross-reconstruction should be equal to the original audio. Similarly, we cross-reconstruct an input audio with two style latents of the same person, but of different sequence. Enforced by:

$$\mathcal{L}_{xemo} = \sum_{k=1}^{4} D(E_c(a_k), E_s(a_k), E_e(a_{j(k)})) - a_k,$$

$$\mathcal{L}_{xsty} = \sum_{k=1}^{4} D(E_c(a_k), E_s(a_{j(k)})), E_e(a_k)) - a_k,$$

where $j(k) = [(6-k) \mod 4] + 1$.

Given two input audios of the same contents, different speakers s_i and s_j, and same emotion, we swap the content latents between the audio pair, and decode the two audios back. Since the utterances being spoken are the same and we keep the original style and emotion constant, the cross reconstruction for the swapped content should be equal to original audio. This is enforced by:

$$\mathcal{L}_{xcon} = \sum_{k=1}^{4} D(E_c(a_{j(k)})), E_s(a_k), E_e(a_k)) - a_k$$

where $j(k) = [(1+k) \mod 4] + 1$.

The combined audio loss is given as:

$$\begin{aligned}\mathcal{L}_{dis} = &\mathcal{L}_{xcon} + \mathcal{L}_{xemo} + \mathcal{L}_{xsty} \\ &+ \mathcal{L}_{self} + \mathcal{L}_{emo} + \mathcal{L}_{con} + \mathcal{L}_{sty}\end{aligned}$$

Once trained, the speech disentanglement model produces three disentangled latents for content, style and emotion. These latents serve as the input to our diffusion model.

B.5. Implementation Details

The encoder transformer DeiT (88M parameters) that is finetuned on 384x384 images from ImageNet-1k is obtained from PyTorch image models (timm) [21]. The content, emotion, and style latent vectors are of 256 dimension. The transformer–fusion includes 2 layers and 4 heads. The transformer–decoder includes 4 heads and 4 layers. The input dimension of fusion block is 768 to accommodate three content, emotion, and style latent codes. Each 2D filterbank is of 1024 x 128, where 128 represents the number of mel-frequency bins.

B.6. Ablation Experiments

We conduct two ablation studies with the speech disentanglement model. One to only disentangle emotion from content (dropping $\mathcal{L}_{sty}$, $\mathcal{L}_{xsty}$). The other to only disentangle only style from content (dropping $\mathcal{L}_{emo}$, $\mathcal{L}_{xemo}$). Tab. A.1 shows the accuracy and F1 scores for emotion and style latent vectors. The Emotion Accuracy (EA), Style Accuracy (SA), Emotion F1 Score (EF1), and Style F1 Score (SF1) in our speech disentanglement model exhibit only marginal differences compared to the results obtained in the ablation experiments. We report the test set self- and cross-reconstruction errors in Tab. A.2. The cross-reconstruction errors are comparable to self-reconstruction errors which indicates that the individual latents from different audios can be combined to produce valid outputs. This holds for the main model and also the ablated models. However, the ablated models are not able to factor the audio into all three components due to the dropped loss terms. We observe the robust performance of our audio model, by accounting for the complex interplay between emotion and style. By concurrently disentangling three latent vectors, our approach effectively captures the intricate relationships in the audio

Table A.1. **Audio emotion and style disentanglement ablation.** We show scores for Emotion Accuracy (EA), Style Accuracy (SA), Emotion F1 Score (EF1), and Style F1 Score (SF1) in our speech disentanglement model and ablation experiments. Although there are slight differences, our model effectively captures the complex relationships between emotion and style by disentangling three latent vectors simultaneously. The best scores are highlighted in green and second best in blue.

Method	EA (%) ↑	EF1↑	SA (%) ↑	SF1↑
Ours	91.531	0.914	96.060	0.960
Emo-disentangle	91.966	0.918	—	—
Sty-disentangle	—	—	96.095	0.961

Table A.2. **Audio latent component factorization ablation.** Self and cross-reconstruction errors show comparable performance, suggesting that individual latents from different audio sources can be effectively combined to yield valid outputs. The best scores are highlighted in green and second best in blue.

Method	Self↓	XCon↓	XEmo↓	XSty↓
Ours	.3739	.3740	.3816	.3815
Emo-disentangle	.3793	.3792	.3905	—
Sty-disentangle	.3769	.3770	—	.3887

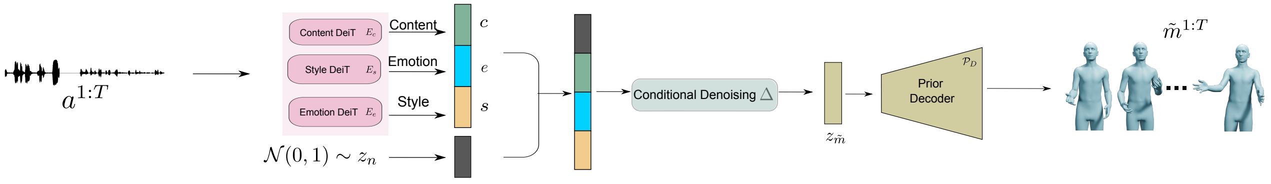


Figure A.4. **Inference.** We sample z_n and employ the three conditioning latents from a test-time audio c, e, s. We iteratively apply Δ to generate the fully denoised $z_{\tilde{m}}$ which is decoded by $\mathcal{P}_D$ into the final motion $\tilde{m}^{1:T}$.

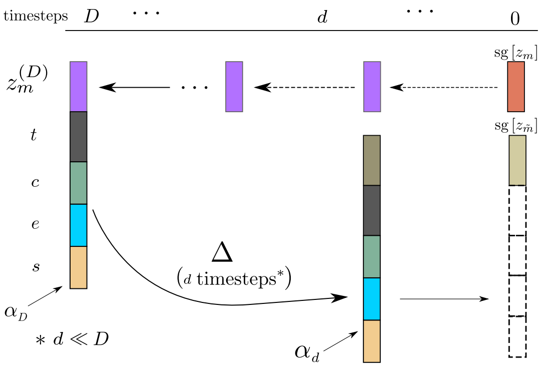


Figure A.5. **Conditional latent diffusion.** In the diffusion process (right to left) we obtain a noisy motion latent, whereas in the denoising process (left to right) we obtain a conditioned denoised motion latent.

data, allowing to jointly model and distinguish both emotion and style factors.

C. Gesture Generation Model

C.1. Motion Prior And Latent Denoiser

In this section we include detailed illustrations of the motion prior and latent denoiser. Fig. A.4 illustrates the inference process employed by our model. Fig. A.5 illustrates the forward diffusion and the reverse audio-conditioned denoising process, operating at the latent space. Finally, A.7 shows the diagram of the architecture of the motion prior.

C.2. Methods Trained on Coarse Skeletal Data

We compare AMUSE with methods trained on coarse skeletal data. We choose DSG [22], CaMN [13], Zhu et al. [23] and MoGlow [1] as recent gesture generation models using audio input. AMUSE produces more synchronized gestures and better represents the underlying audio emotion compared to the state of the art methods trained on skeletal data, as shown in our supplementary video. Additionally, these methods are not trained to output 3D meshes. We observe uncanny poses and self-penetrations as shown in Fig. A.8. In our video, we provide additional comparison with these skeleton based methods in both formats, the original predictions of those models and 3D meshes which are created via Inverse Kinematics (IK). We exclude the conversion to 3D mesh for Zhu et al. [23] because the output skeleton format is incompatible with SMPL-X topology.

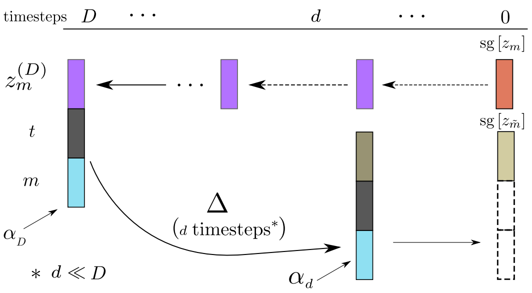


Figure A.6. **Ablation conditional latent diffusion.** In the denoising process (left to right) of the ablation model, we obtain a denoised motion latent that is conditioned on a compressed non-disentangled latent vector m instead of three disentangled latents that are used in the final model.

C.3. Ablation Experiments

Without speech disentanglement model. This subsection illustrates the difference between the final AMUSE model and the ablation of AMUSE w/o audio disentanglement. AMUSE w/o audio disentanglement uses 8 linear-layered auto-encoder that operates directly on raw audio MFCC features to produce single latent vector m. Since AMUSE w/o audio disentanglement does not operate over the three disentangled latents of content, emotion, and style but instead only one non-disentangled latent m, the latent diffusion process also only takes one latent on the input m as shown in Fig. A.6. By design, this model lacks the gesture editing capabilities.

Without motion prior. We employ our latent denoiser only in this ablation model. We completely removed the motion prior component and replace it with a linear projection head. The ablation model without motion prior is not able to converge and produces mostly static motions (refer to the supplementary video). This signifies the importance of having a motion prior component in our AMUSE architecture.

Quantitative evaluation of ablation experiments. Following the procedure described in the Sec. 5.1 , we report quantitative evaluation scores in Tab. A.3, comparing

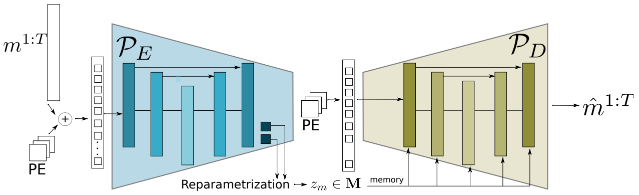


Figure A.7. **Motion prior network**. The motion prior is VAE encoder decoder architecture inspired from Chen et al. [4]. Both encoder and decoder follow a U-Net like structure with skip connections between transformer blocks. The learnable positional embeddings (PE) are injected into each multi-head attention layer.

AMUSE with the ablation models and GT. The version w/o speech disentanglement model produces lower-quality gestures and lacks editing capabilities compared to the complete model. This is because it lacks a component for separating emotion, content and style in the audio. The scores for the ablation models without motion prior are the lowest, indicating that this model did not converge successfully. Additionally, in Tab. A.4 we report improved FGD and Div scores when the motion prior and diffusion model are trained jointly compared to when trained separately, indicating that joint training yields superior results. Furthermore, we conduct additional ablation experiments with and without alignment losses ($\mathcal{L}_{align}$, $\mathcal{L}_{Valign}$). Including alignment losses results in a GA of 46.79%, whereas without them, the GA drops to 30.89%, demonstrating the alignment losses effectiveness. Moreover, we compute the average jerk of the left and right hands for motion sequences belonging to the same audio of [22], ours, and GT, reporting it in m/s^3 as 1.18, 1.10, and 0.065, respectively. This signifies that the GT motion is the most steady, whereas ours is slightly smoother over time compared to [22].

Table A.3. **Ablation of AMUSE components.** The model without audio disentanglement produces lower-quality gestures and lacks editing capabilities. The model without motion prior perform poorly due to convergence issues. Among the methods being compared, we highlight the best scores in green and second best in blue.

Method	SRGR↑	BA↑	FGD↓	Div→	GA[a]↑
GT	—	0.83	—	27.83	64.04
Ours	0.36	0.81	388.63	25.06	46.76
Ours-No-Prior	0.25	0.20	987.90	13.41	15.42
Ours-No-Audio-Model	0.31	0.78	633.27	21.08	26.88

[a] GA is average of all 8 emotions.

Table A.4. **Ablation of AMUSE training.** We observe improved FGD and Div scores when the motion prior and diffusion model are jointly trained, highlighting superior performance compared to separate training methods. We highlight the best scores in green and second best in blue.

Method	FGD↓	Div→
Ours	388.63	25.06
Ours-Disjoint	362.33	24.49

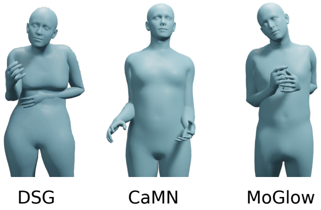


Figure A.8. **Coarse skeleton-based methods.** Here we compare DSG [22], CaMN [13], and MoGlow [1]. Unlike SMPL-X-based models, these are trained using different skeletal hierarchies without volumetric 3D shapes. Retargeting them onto the SMPL-X skeleton with IK causes uncanny poses and self-penetration.

D. Motion Feature Extractor Model

We employ the motion extractor model M for computing all quantitative evaluation metrics. Our motion extractor encoder model design is inspired by Petrovich et al. [18], in an autoencoder setting (i.e., without a probabilistic variational component). We append a CLS token at the beginning of the motion sequence and supervise with a cross-entropy emotion classification objective $\mathcal{L}_{Memo}$ applied to the output CLS token. We train the motion extractor model on the BEAT training data. Once trained, we use the latent space features to compute evaluation metrics as described in the Sec. 5.1.

$$\mathcal{L}_{Memo} = -\sum_{1 \leq l_e \leq n_e} y_{l_e} \log(p_{l_e})$$

with $n_e = 8$ denotes the number of emotion classes.

E. Gesture Emotions And Semantics

We quantitatively evaluate our method using metrics SRGR, beat align, FGD, diversity, and gesture emotion accuracy. Leveraging the latent space features from the motion extractor model M, we compute SRGR and gesture emotion accuracy. Additionally, we directly utilize the generated motion sequence to calculate the beat align score.

Semantic-Relevant Gesture Recall (SRGR). In the SRGR metric score, similar to Liu et al. [13], we use ground truth semantic score as weight for the Probability of Correct Keypoint (PCK) between the generated gestures and ground truth gestures, where PCK is the number of joints successfully recalled for a given threshold δ. Following the approach suggested by BEAT authors:

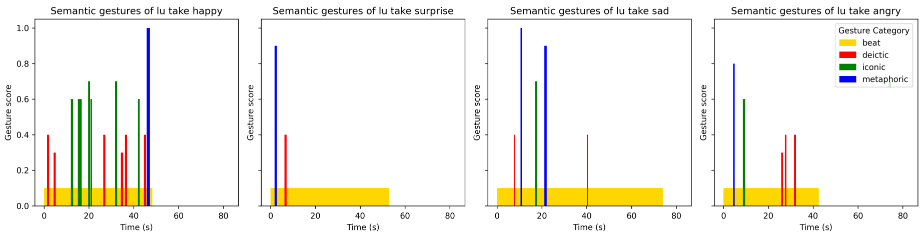


Figure A.9. **Emotional gesture variation.** Semantic scores for various emotions within the same subject shows how the subject expresses gestures differently for each emotion. This reflects the subject's interpersonal style specific to each emotion.

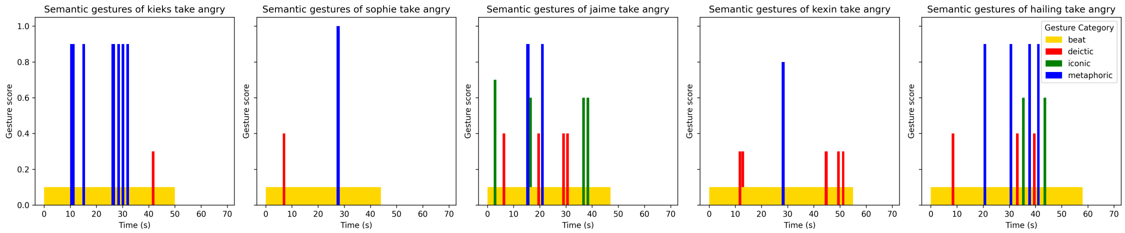


Figure A.10. **Emotional gesture individuality.** Semantic scores across various subjects for the same emotion reveal how different subjects express gestures uniquely for identical utterances within same emotion. There is variability in expressiveness, with some subjects being more expressive (eg. Jamie, Hailing) than the others (eg. Sophie, Kexin).

$$\text{SRGR } = \lambda \sum \frac{1}{\text{T x J}} \sum_{t=1}^{T} \sum_{j=1}^{J} \mathbf{1}\left[\left\|p_t^j - \hat{p}_t^j\right\|_2 < \delta\right]$$

where $\mathbf{1}$ is the indicator function, T, J are the set of frames and number of joints, we use SRGR to measure how well our model recalls gestures in the relevant clip. This metric reflects human perception of valid gesture diversity. The metric is computed based on the scores assigned by 118 annotators from Amazon Mechanical Turk (AMT), who evaluated the semantic relevance on a continuous scale of 0-1. The scores are provided for four gesture types: beat (*rhythmic movements*), iconic (*representative movements*), deictic (*indicative or pointing movements*), and metaphoric (*symbolic or figurative movements*). SRGR metric needs GT semantic scores for computation.

Ground-truth semantic scores. We obtain the ground-truth semantic score, provided by the BEAT authors, for computing the SRGR. In Fig. A.9, we present semantic scores for the same subject across various emotions, while Fig. A.10 illustrates semantic scores for all subjects expressing the same emotion. This allows us to observe how subjects gesture differently with different emotions and how different subjects gesture for the same emotion. While we acknowledge the high-quality dataset introduced by the BEAT authors, our model has the potential to deliver even better results and improved expressivity with an enhanced dataset quality.

Beat alignment. Following Li et al. [12], we compute the beat align score. To compute the beat alignment score, we use six joints: left wrist, left elbow, left shoulder, right wrist, right elbow, and right shoulder, similar to Liu et al. [13]. We measure the synchronization between the generated 3D motion and the input speech by calculating the beat align score. This score gauges the average distance between each kinematic beat and its nearest speech audio beat, following a unidirectional approach – recognizing that gesture motion may not align with every speech audio beat. AMUSE achieves the highest beat align score in correlating speech audio and gestures compared to the other methods.

Gesture emotion accuracy. Gestural emotions are complex, influenced by internal states of subject, social signals, and their perception vary significantly across individuals with diverse cultural backgrounds. AMUSE is designed to capture perceived gestural emotions. While we demonstrate AMUSE with a gesture emotion recognition accuracy of 46.76% and AMUSE-Edit with 34.18%, outperforming other state of the art methods, it is important to note that recognizing emotion from gestures remains a challenging task in computer vision. We observe the gesture emotion accuracy for the GT sequence is 64.04%. There is still ample room for improvement in addressing this complex problem. Additionally, in Fig. A.11, we present the confusion matrix for ground truth (GT) emotion predictions on the left and reconstructions (gestures generated using the original style, emotion, and content latents of a given audio) on the right. We observe a robust correlation between the predictions on GT and the reconstructions for all eight emotions. Additionally, we conducted experiments on gesture edits by

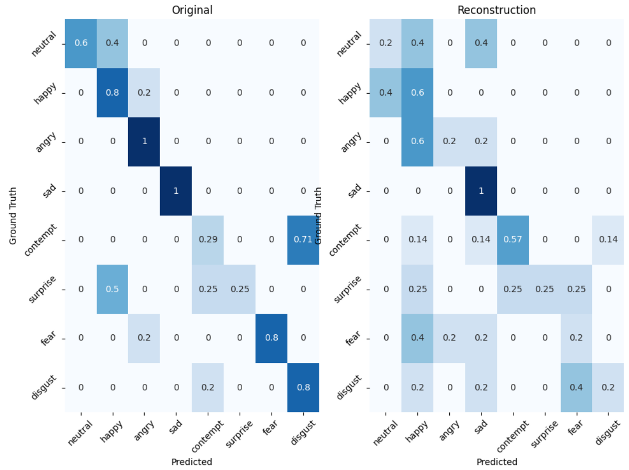


Figure A.11. **Confusion matrix comparing gestures from the ground truth (GT) and regenerated emotion predictions**.

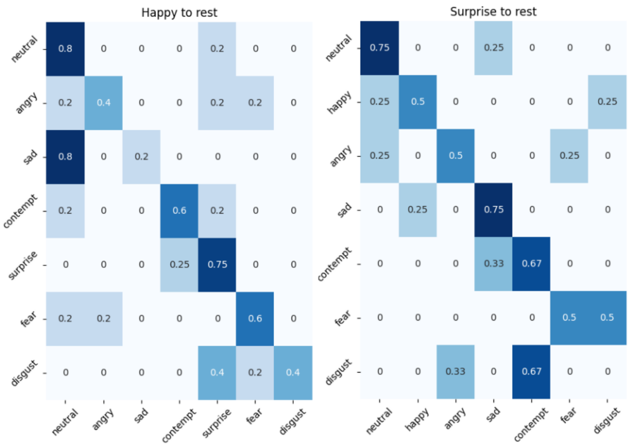


Figure A.12. **Emotion edit confusion matrix, displaying transitions from Happy (left) and Surprise (right) emotions to others**. X-axis is for predictions and Y-axis is for ground truth.

swapping emotion latents from one audio with those from another audio of the same subject but with a different utterance. In Fig. A.12, we showcase two exemplars, transforming from *Happy to rest* and *Surprise to rest*. Given the diversity of eight emotions, gestural edits offer numerous possibilities, rendering this a broad and challenging problem. Although Fig. A.12 displays promising results for emotion label predictions with clear diagonal pattern of the confusion matrix, we acknowledge the inherent difficulty in solving this problem.

F. Data Preparation

In this section, we describe the processing and alignment of different modalities, and the BEAT [13] data subsets employed to train the different models of our framework. We do not use the entirety of the BEAT dataset to train AMUSE. BEAT contains 30 speakers. We filter out subjects with little expressivity in their motion through visual inspection of GT, leaving us with 22 subjects. Furthermore, BEAT has a subset that all the subjects speak the same sentences in the same emotions. The rest of the dataset contains unique sentences which are spoken only by one speaker and not the others. We filter out all of these unique sentences. What remains is a subset of 16 sentences (2 per emotion, for 8 emotions), spoken by every subject. This is critical since the training of the speech disentanglement module requires perfect temporal correspondence between the audios of the same sentences. Except where explicitly stated otherwise, we have used this subset and split it into train, validation, and test sets. This subset is 5.71 hours long. We use the same data to train our speech disentanglement model. Further, we train our motion prior network $(\mathcal{P}_E, \mathcal{P}_D)$ with the extracted SMPL-X motions of the same subsets. Finally, the denoiser, Δ, and feature extractor used for evaluation, M, are trained on the same subset and splits.

G. Review of State of the Art Methods

Data selection and input formats. To train AMUSE effectively, we require data in the form of 3D point clouds rather than coarse BVH skeletons. Additionally, training requires common utterances from multiple subjects expressing various emotions for audio disentanglement. Many available gesture datasets, including [5, 17], come in various motion capture skeleton formats with different underlying kinematic hierarchies that are incompatible with our conversion procedure to obtain SMPL-X meshes and do not meet the requirement of speech common utterances. In contrast, for the BEAT dataset, we obtained the initial data in the form of 3D point clouds from the dataset authors. We use Mosh++ [14, 15] to extract SMPL-X pose and shape parameters, along with global translation and orientation, from the 3D point cloud. This data was then used to train AMUSE.

SOTA methods and modifications. Given our primary objective is to generate 3D emotional gestures from au-

Table A.5. **Perceptual study.** We demonstrate aggregate scores of our perceptual study. and we disregard indifferent scores. The *ours* and *others* are sum of % preference for (strongly ours and weakly ours) and (strongly other and weakly other) respectively. Only the best scores are highlighted in green.

Criteria →	Emotion		Synchronization	
Method ↓	Ours	Others	Ours	Others
GT	38	51	35	52
TalkSHOW-BEAT	46	39	62	27
TalkSHOW	54	34	65	28
Habibie et al.	48	42	66	28

dio input, we mainly compare state-of-the-art methods that use audio input alone and output a 3D mesh. We exclude methods that incorporate additional inputs, such as arbitrary lengths of target motion style, as they deviate from our main objective, for example, Ghorbani et al. [5]. Other recent works [2, 3] have proposed methods for generating gestures from speech. However, making direct comparisons is difficult as the code for their approaches is not publicly available. We retrained Henter et al. [11] using publicly available code and instructions, due to the unavailability of a pretrained model. In our comparison, we used publicly available DSG [22] model that was trained on the BEAT dataset of coarse skeletal format. We also made modifications to the TalkSHOW code, incorporating emotion labels as input, and retrained it on the same data used for training our model. The emotion categorical labels were injected inline with existing subject labels using one-hot vectors. AMUSE outperforms both DSG and TalkSHOW-BEAT as well as other SOTA methods in all comparisons.

H. Additional Perceptual Study Details

Here we describe additional details of the AMT study reported in the Sec. 5.3. We show aggregate preference scores in Tab. A.5. AMUSE outperforms all methods compared against in both criteria - synchronization with the speech and the appropriateness with respect to the specified emotion. In contrast, Ground Truth (GT) consistently outperforms AMUSE in both tasks. This outcome emphasizes the complexity of the problem, where achieving synchrony with speech and meeting specified emotional appropriateness remain challenging objectives.

Data. We randomly select three videos per emotion from the BEAT dataset for our perceptual study. We only use sequence that were not part of training or validation set. Due to high number of subjects, we limit the input audios data to only two subjects.

The template layout. Fig. A.13 depicts the design template that the participants were shown. The left–right position of our method and the competing methods was randomized to factor out any biases that participants may have for one side or the other.

Catch trials. Each participant was also shown three catch trials, where a GT video was shown alongside a broken motion filled with artifacts. Participants that did not select weak or strong preference for the GT video in any of the catch trials were labeled as uncooperative or inattentive and were not considered in the analysis. We selected 22, 20, 23, and 25 participants for TalkSHOW-BEAT, TalkSHOW, Habibie et al., and GT, respectively, from a total of 25 Amazon Mechanical Turk workers.

References

[1] Simon Alexanderson, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow. Style-controllable speech-driven gesture synthesis using normalising flows. *Computer Graphics Forum (CGF)*, 39(2):487–496, 2020. 4, 5

[2] Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, Denoise, Action! Audio-driven motion synthesis with diffusion models. *Transactions on Graphics (TOG)*, 42(4):1–20, 2023. 8

[3] Tenglong Ao, Zeyi Zhang, and Libin Liu. GestureDiffuCLIP: Gesture diffusion model with CLIP latents. *Transactions on Graphics (TOG)*, 2023. 8

[4] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Computer Vision and Pattern Recognition (CVPR)*, pages 18000–18010, 2023. 5

[5] Saeed Ghorbani, Ylva Ferstl, Daniel Holden, Nikolaus F. Troje, and Marc-André Carbonneau. Zeroeggs: Zero-shot example-based gesture generation from speech. *Computer Graphics Forum (CGF)*, 42(1):206–216, 2023. 7, 8

[6] Yuan Gong, Yu-An Chung, and James R. Glass. AST: audio spectrogram transformer. In *Interspeech 2021*, pages 571–575, 2021. 1, 2

[7] Yuan Gong, Yu-An Chung, and James Glass. PSLA: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3292–3306, 2021.

[8] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James R. Glass. SSAST: self-supervised audio spectrogram transformer. In *AAAI Conference on Artificial Intelligence*, pages 10699–10709, 2022. 2

[9] Daniel Grzywczak and Grzegorz Gwardys. Deep image features in music information retrieval. *International Journal of Electronics and Telecommunications*, 60:187–199, 2014.

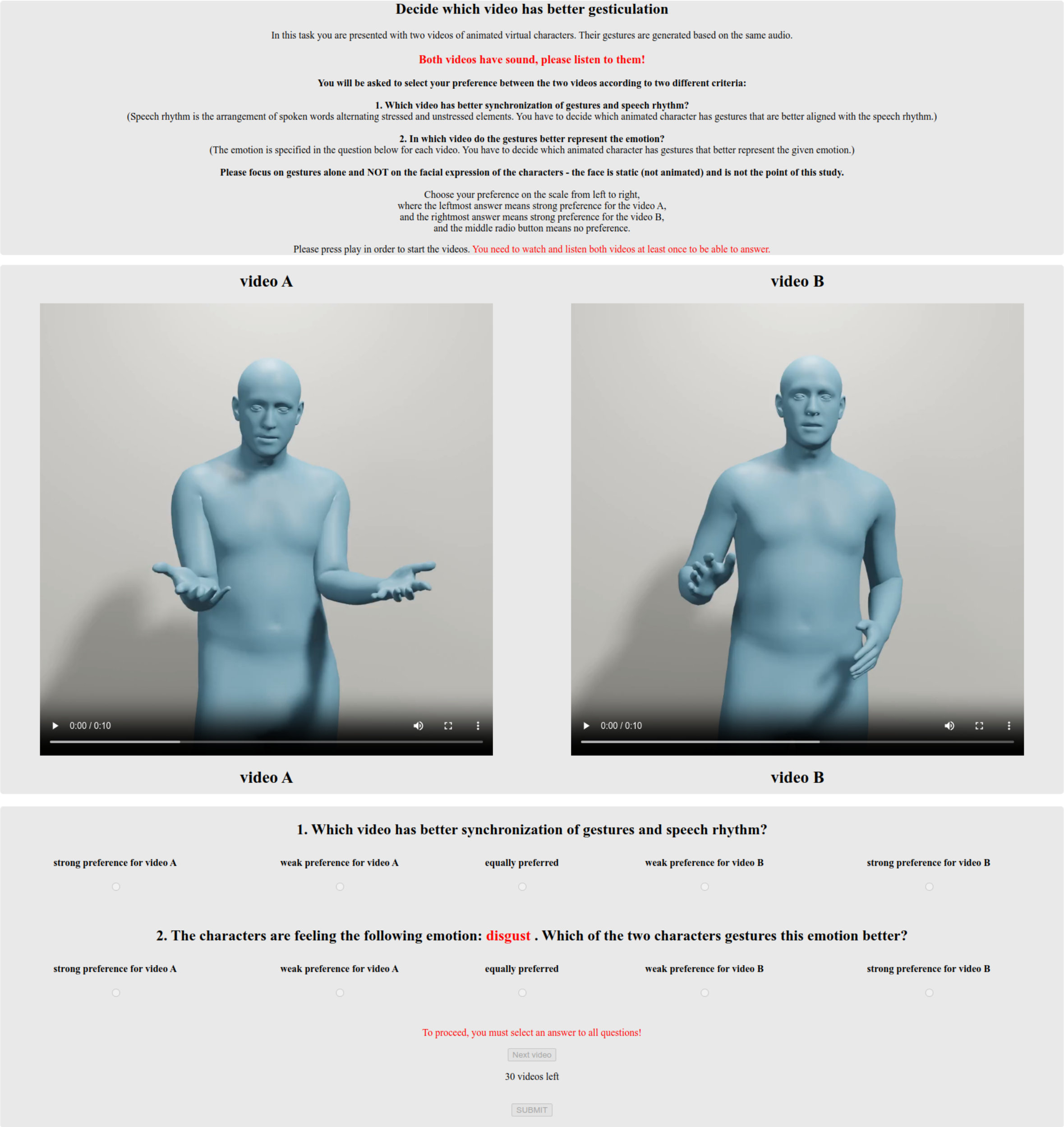
[10] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Esresnet: Environmental sound classification based on visual domain models. In *International Conference on Pattern Recognition (ICPR)*, pages 4933–4940. IEEE, 2020. 1

[11] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. MoGlow: Probabilistic and controllable motion synthesis using normalising flows. *Transactions on Graphics (TOG)*, 39 (4):236:1–236:14, 2020. 8

[12] Ruilong Li, Shan Yang, D. A. Ross, and Angjoo Kanazawa. AI choreographer: Music conditioned 3D dance generation with AIST++. In *International Conference on Computer Vision (ICCV)*, 2021. 6

[13] Haiyang Liu, Zihao Zhu, Naoya Iwamoto, Yichen Peng, Zhengqing Li, You Zhou, Elif Bozkurt, and Bo Zheng. BEAT: A large-scale semantic and emotional multi-modal dataset for conversational gestures synthesis. *European Conference on Computer Vision (ECCV)*, 2022. 4, 5, 6, 7

[14] Matthew M. Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and shape capture from sparse markers. *Transactions on Graphics (TOG)*, 33(6):220:1–220:13, 2014. 7



Decide which video has better gesticulation

In this task you are presented with two videos of animated virtual characters. Their gestures are generated based on the same audio.

Both videos have sound, please listen to them!

You will be asked to select your preference between the two videos according to two different criteria:

1. Which video has better synchronization of gestures and speech rhythm?
(Speech rhythm is the arrangement of spoken words alternating stressed and unstressed elements. You have to decide which animated character has gestures that are better aligned with the speech rhythm.)

2. In which video do the gestures better represent the emotion?
(The emotion is specified in the question below for each video. You have to decide which animated character has gestures that better represent the given emotion.)

Please focus on gestures alone and NOT on the facial expression of the characters - the face is static (not animated) and is not the point of this study.

Choose your preference on the scale from left to right,
where the leftmost answer means strong preference for the video A,
and the rightmost answer means strong preference for the video B,
and the middle radio button means no preference.

Please press play in order to start the videos. You need to watch and listen both videos at least once to be able to answer.

video A **video B**

video A **video B**

1. Which video has better synchronization of gestures and speech rhythm?

strong preference for video A	weak preference for video A	equally preferred	weak preference for video B	strong preference for video B
○	○	○	○	○

2. The characters are feeling the following emotion: disgust . Which of the two characters gestures this emotion better?

strong preference for video A	weak preference for video A	equally preferred	weak preference for video B	strong preference for video B
○	○	○	○	○

To proceed, you must select an answer to all questions!

30 videos left

Figure A.13. **The layout of the perceptual study.** The participant is shown two videos and asked to enter their preference according to two criteria - synchronization with the speech and the appropriateness with respect to the specified emotion.

[15] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *International Conference on Computer Vision (ICCV)*, 2019. 7

[16] Kamalesh Palanisamy, Dipika Singhania, and Angela Yao. Rethinking cnn models for audio classification. *arXiv:2007.11154*, 2020. 1

[17] Kunkun Pang, Dafei Qin, Yingruo Fan, Julian Habekost, Takaaki Shiratori, Junichi Yamagishi, and Taku Komura. BodyFormer: Semantics-guided 3d body gesture synthesis with transformer. *Transactions on Graphics (TOG)*, 42(4), 2023. 7

[18] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)*, 2021. 5

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy,

Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. 1

[20] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, pages 10347–10357, 2021. 1, 2

[21] Ross Wightman. PyTorch image models (timm). https://timm.fast.ai/. Accessed: 2023-11-25. 3

[22] Sicheng Yang, Zhiyong Wu, Minglei Li, Zhensong Zhang, Lei Hao, Weihong Bao, Ming Cheng, and Long Xiao. Diffusestylegesture: Stylized audio-driven co-speech gesture generation with diffusion models. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023. 4, 5, 8

[23] Lingting Zhu, Xian Liu, Xuanyu Liu, Rui Qian, Ziwei Liu, and Lequan Yu. Taming diffusion models for audio-driven co-speech gesture generation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 10544–10553, 2023. 4