Taming Diffusion Models for Music-Driven Conducting Motion Generation

Zhuoran Zhao[1]*, Jinbin Bai[1]*, Delong Chen[2], Debang Wang[1], Yubo Pan[1]

[1] Dept. of Computer Science, School of Computing, National University of Singapore
[2] Xiaobing.AI
{zhuoran.zhao, jinbin.bai}@u.nus.edu

Abstract

Generating the motion of orchestral conductors from a given piece of symphony music is a challenging task since it requires a model to learn semantic music features and capture the underlying distribution of real conducting motion. Prior works have applied Generative Adversarial Networks (GAN) to this task, but the promising diffusion model, which recently showed its advantages in terms of both training stability and output quality, has not been exploited in this context. This paper presents `Diffusion-Conductor`, a novel DDIM-based approach for music-driven conducting motion generation, which integrates the diffusion model to a two-stage learning framework. We further propose a random masking strategy to improve the feature robustness, and use a pair of geometric loss functions to impose additional regularizations and increase motion diversity. We also design several novel metrics, including Fréchet Gesture Distance (FGD) and Beat Consistency Score (BC) for a more comprehensive evaluation of the generated motion. Experimental results demonstrate the advantages of our model. The code is released at https://github.com/viiika/Diffusion-Conductor.

Introduction

Human conductors have the remarkable ability to translate their rich comprehension of music contents into sequences of precise yet graceful conducting motion. Advancements in AIGC technologies for human motion (Mourot et al. 2022) have addressed the generation of various human motions such as speech gestures, dance movements, and instrumental motions in recent years, and researchers are now pivoting towards building AI conductors. Pioneered works of VirtualConductor (Chen et al. 2021) and M^{2}S-GAN (Liu et al. 2022) demonstrated the promising possibilities of building such systems. These works take advantage of the Generative Adversarial Network (GAN) (Goodfellow et al. 2020) to learn the probabilistic distribution of real conducting motion from a large-scale paired music-motion dataset. However, GAN-based models typically suffer from notorious issues such as mode collapse and unstable training, which impede the generation of plausible conducting motions.

Recently, diffusion models (Ho, Jain, and Abbeel 2020; Ho and Salimans 2022) have emerged as the new state-of-the-art family of deep generative models, and yield impressive performance in conditional image generation, surpassing those GAN-based methods which have dominated the field for the past few years. We hypothesize that such an advantage can be extended to the task of music-driven conducting motion generation, and in this paper, we introduce our `Diffusion-Conductor` model, which is the first diffusion-based AI conductor model.

Our learning framework comprises two consecutive stages, namely, the contrastive learning stage and the generative learning stage. The first stage builds a two-tower structure and performs music-motion contrastive pre-training to learn rich music features, those learned features are subsequently transferred to the second stage with a random masking strategy. We incorporate a DDIM-based model to learn the conditional generation of the conducting motion, and we modify the supervision signal from ϵ to x_0 for better generation performance. Furthermore, we incorporate perceptual loss to avoid over-smoothing problem and impose additional supervision on the model via two geometric regularization losses, namely velocity loss and elbow loss, to enhance the consistency and diversity of generated motions.

We use a wide array of metrics, including mean squared error (MSE), fractional shape distance (FGD), beat consistency score (BC) and diversity, to evaluate the motion produced by `Diffusion-Conductor`. Thorough comparisons demonstrated that our model outperforms the previous GAN-based method (Liu et al. 2022).

Methods

In this section, we will provide an overview of our approach, and illustrate the Training Objective applied at the various stages.

Overview of Our Approach

Our proposed architecture is illustrated in Fig. 1. In the contrastive learning stage, a contrastive pre-training network composed of a motion encoder E_{motion} and a music encoder E_{music} is used to learn music representations that are correctly aligned to their corresponding motion representations. Subsequently, a generation network G is employed during the generative learning stage to generate a motion sequence based on the music embeddings outputted by the pre-trained

*These authors contributed equally.

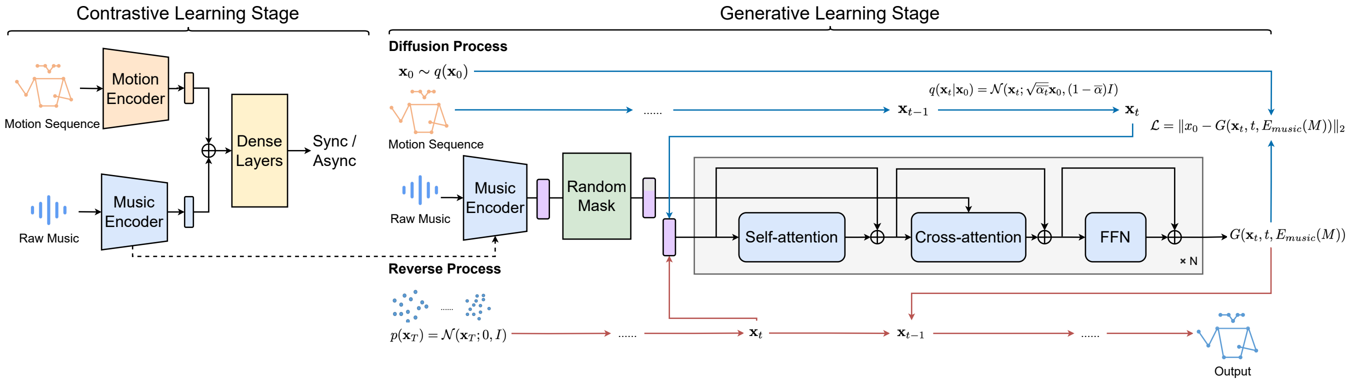
Contrastive Learning Stage
Generative Learning Stage
Motion Sequence
Motion Encoder
Dense Layers
Sync / Async
Raw Music
Music Encoder
Diffusion Process
$\mathbf{x}_0 \sim q(\mathbf{x}_0)$
$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\overline{\alpha}_t}\mathbf{x}_0, (1-\overline{\alpha})I)$
Motion Sequence
......
$\mathbf{x}_{t-1}$
$\mathbf{x}_t$
$\mathcal{L} = \|x_0 - G(\mathbf{x}_t, t, E_{music}(M))\|_2$
Raw Music
Music Encoder
Random Mask
Self-attention
Cross-attention
FFN
× N
$G(\mathbf{x}_t, t, E_{music}(M))$
Reverse Process
......
$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; 0, I)$
......
$\mathbf{x}_t$
$\mathbf{x}_{t-1}$
......
Output

Figure 1: Overview of the proposed framework. The colors of the arrows in Generative Learning Stage represent different stages: blue for training, red for inference, and black for both training and inference.

E_{music}. To further facilitate motion generation while undergoing the denoising process, we make use of the denoising diffusion implicit model (DDIM) (Song, Meng, and Ermon 2020) and introduce a Cross-Modality Linear Transformer. During inference, a Gaussian distribution noise is sampled according to the given random seed and fed into the denoising process with cross-attention between the music features. Finally, music-driven conducting motions will be generated. Detailed descriptions of our methods are presented in the following sections.

Contrastive Pre-training The contrastive pre-training network comprises three components: a motion encoder E_{motion}, a music encoder E_{music}, and a set of dense layers f. The motion embeddings and music embeddings generated by E_{motion} and E_{music} are concatenated and then passed to f, after which a binary cross-entropy loss is applied to assess whether music and motion are appropriately paired. Specifically, the music encoder E_{music} is used to generate music features from raw music and consists of three groups of layers, with each layer comprised of three residual layers and a max-pooling layer. Meanwhile, the motion encoder E_{motion} is employed to generate motion features for the conducting motion sequence. To analyze the conducting motion both spatially and temporally, we make use of the Spatial-Temporal Graph Convolutional Network (ST-GCN) (Yan, Xiong, and Lin 2018), which has been used extensively in human pose estimation tasks.

Diffusion Model for Motion Generation Diffusion models involve a diffusion process and a reverse process. The diffusion process adds Gaussian noise to the motion sequence data in accordance with the Markov chain rule to approximate the posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$.

Most prior works (Ho, Jain, and Abbeel 2020; Nichol et al. 2021; Zhu et al. 2023) train the model to predict noise $\epsilon_\theta(\mathbf{x}_t, t, E_{music}(M))$ and then calculate the mean square error between ϵ and $\epsilon_\theta(\mathbf{x}_t, t, E_{music}(M))$ to optimize the model.

Here, we instead follow (Ramesh et al. 2022; Tevet et al. 2022) by directly predicting the motion x_0 and using the mean square error on this prediction which yields better generation performance. Subsequently, the reverse process can be employed to denoise the motion sequence step by step and generate a clean motion sequence conditioned on the given music embeddings.

Cross-Modality Linear Transformer To serve as a denoising model, we use a transformer (Vaswani et al. 2017). We initially utilize a music encoder to extract the music embeddings, the pre-training of which during the contrastive learning stage can facilitate the generation process. Subsequently, a self-attention module is employed to enable motion features from different times to interact with each other. Additionally, a cross-attention module is used to fuse music embeddings and motion sequence together while a feed-forward network is used to generate motion as $G(\mathbf{x}_t, t, E_{music}(M))$.

Random Mask Inspired by masked language modeling and masked image modeling, we incorporated a random mask (Zhong et al. 2020; Tevet et al. 2022) block after the music encoder to train the diffusion model with both music-conditional and unconditional elements. This can potentially allow us to trade off between diversity and quality for improved generalization performance.

Training Objective

Contrastive Learning Stage. At the contrastive learning stage, we adopt a binary cross-entropy loss to learn the representation of music under the supervision of motion, which can be formulated as:

$$\mathcal{L}_{bce} = \sum_{i,j=1}^{N} (c_{ij} log_2(f[E_{music}(M_i) \oplus E_{motion}(X_j)]) + (1 - c_{ij}) log_2(1 - f[E_{music}(M_i) \oplus E_{motion}(X_j)])) \tag{1}$$

where c_{ij} is defined by

$$c_{ij} = \begin{cases} 1, & \text{i = j} \\ 0, & \text{otherwise} \end{cases}$$

M_i and X_j represent the i-th music data and the j-th motion data respectively, where $\oplus$ denotes the feature concatenation operation. Both E_{music} and E_{motion} denote the music and motion encoders, respectively, and f represents the dense layers.

Generative Learning Stage. The overall training loss for generative learning stage consists of three parts including diffusion loss $\mathcal{L}_{ddim}$, perceptual loss $\mathcal{L}_{perc}$ and geometric loss $\mathcal{L}_{geo}$:

$$\mathcal{L} = \lambda_{ddim}\mathcal{L}_{ddim} + \lambda_{perc}\mathcal{L}_{perc} + \lambda_{geo}\mathcal{L}_{geo} \tag{2}$$

where λ_{ddim}, λ_{perc} and λ_{geo} are weighting factors for each loss term.

Diffusion Loss. We follow (Ramesh et al. 2022; Tevet et al. 2022) to directly predict the motion x_0 rather than predicting the noise ϵ as formulated by (Ho, Jain, and Abbeel 2020), for plausible and improved generation performance. The diffusion loss can be demonstrated as follows:

$$\mathcal{L}_{ddim} = ||x_0 - G(\mathbf{x}_t, t, E_{music}(M))||_2^2 \tag{3}$$

where x_0 is the original motion sequence and $G(\mathbf{x}_t, t, E_{music}(M))$ denotes the final step of motion sequence generated by the diffusion model.

Perceptual Loss. Moreover, we employ a perceptual loss to minimize distance between the extracted feature from generated motion and ground-truth motion.

$$\mathcal{L}_{perc} = |E_{motion}(x_0) - E_{motion}(\hat{x_0})| \tag{4}$$

where E_{motion} is the motion encoder pre-trained in the contrastive learning stage and $\hat{x_0}$ equals $G(\mathbf{x}_t, t, E_{music}(M))$.

Geometric Loss. A geometric loss is employed to regularize the generative model, enforcing physical properties and preventing artifacts in order to generate natural and coherent motion. This consists of a velocity loss (Tevet et al. 2022) and an elbow loss; the former ensures that the velocity of the generated motion coincides with the ground-truth motion and the latter encourages more intensive arm swing for more vivid motion. The geometric loss is demonstrated as follows:

$$\mathcal{L}_{geo} = \lambda_{vel}\mathcal{L}_{vel} + \lambda_{elbow}\mathcal{L}_{elbow} \tag{5}$$

$$\mathcal{L}_{vel} = \frac{1}{N-1}\sum_{i=1}^{N-1} ||({x_0}^{i+1} - {x_0}^{i}) - ({\hat{x_0}}^{i+1} - {\hat{x_0}}^{i})||_2^2 \tag{6}$$

$$\mathcal{L}_{elbow} = -\frac{1}{N-1}\sum_{i=1}^{N-1} ||\hat{x_0}_{elbow}^{i+1} - \hat{x_0}_{elbow}^{i}||_2^2 \tag{7}$$

where λ_{vel} and λ_{elbow} are weighting factors for each term.

Experiment

Datasets

We leverage the ConductorMotion100 dataset (Chen et al. 2021) for training purposes.

Evaluation Metrics

We use four metrics that are commonly utilized in motion generation and relative fields to evaluate our method.

Mean Squared Error (MSE). Mean squared error (MSE) is the most direct way to measure how closely the generated motion corresponds to the ground truth motion and has been widely used as an evaluation metric in music-to-motion tasks (Kao and Su 2020; Tang, Jia, and Mao 2018). The representation of MSE is defined as follows:

$$MSE(X, \hat{X}) = \|X - \hat{X}\|_2^2$$

where X denotes the ground-truth motion and $\hat{X}$ denotes the generated motion.

Fréchet Gesture Distance (FGD). FGD is frequently used to measure the distance between the synthesized gesture distribution and the real data distribution (Zhu et al. 2023). Since gesture motion and conducting motion are closely related, both being represented as keypoints, we employ FGD to evaluate the distance of the generated conducting motion distribution and the ground-truth conducting motion distribution. FGD is demonstrated as follows:

$$FGD(Y, \hat{Y}) = \|\mu_{gt} - \mu_{gen}\|_2^2 + \mathrm{Tr}(\Sigma_{gt} + \Sigma_{gen} - 2\sqrt{\Sigma_{gt}\Sigma_{gen}})$$

where μ_{gt} and Σ_{gt} stand for the mean and variance of the latent feature distribution of the ground-truth motion X, while μ_{gen} and Σ_{gen} are the mean and variance of the latent feature distribution of the generated motion $\hat{X}$.

Beat Consistency Score (BC). Beat Consistency Score is a metric to evaluate motion-music correlation in terms of the similarity between the motion beats and music beats. We follow (Li et al. 2021) to define motion beats as the local minima of kinetic velocity and use librosa (McFee et al. 2015) to extract music beats. Beat Consistency Score computes the average distance between every music beat and its nearest motion beat:

$$BC = \frac{1}{|\mathcal{B}^x|}\sum_{i=1}^{|\mathcal{B}^x|} \exp\Big(-\frac{\min_{\forall t_j^x \in \mathcal{B}^x} \|t_j^x - t_i^y\|_2^2}{2\sigma^2}\Big)$$

where $\mathcal{B}^x = \{t_j^x\}$ represent motion beats and $\mathcal{B}^y = \{t_i^y\}$ represent music beats, and σ is the parameter to normalize sequences, which is set to 3 empirically.

Diversity. Similar to prior works (Zhu et al. 2023; Li et al. 2021), we evaluate our model's ability to generate diverse conducting motions given various input music. Like (Zhu et al. 2023), we choose 500 generated samples randomly and calculate the mean absolute error between the generated latent motion features and the shuffled features.

Implementation Details

For the diffusion model, we set the diffusion steps to 1000 and use Adam (Kingma and Ba 2014) for optimization with a learning rate of 2e-4 and batch size of 48. We train the diffusion model over 500 epochs, setting the unconditional rate of random mask to 0.1. For the weighting factors in the training objective, we set $\lambda_{ddim} = 1$, $\lambda_{perc} = 0.000001$, $\lambda_{geo} = 1$, $\lambda_{vel} = 0.1$, $\lambda_{elbow} = 0.1$. Experiments are conducted on two NVIDIA TESLA V100 GPUs.

Methods	MSE ↓	FGD ↓	BC ↑	Diversity ↑
VirtualConductor	0.0054	1051.97	0.109	1012.06
Diffusion-Conductor	**0.0042**	**812.01**	**0.119**	**1152.06**

Table 1: Main results on ConductorMotion100 test set

Main Results

As shown in Table 1, we report four metrics compared with VirtualCondutor (Chen et al. 2021) on ConductorMotion100 test set. It is shown that our mothod outperforms VirtualConductor on all the four metrics.

We further visualize the beat consistency between the music and generated conducting motion, making a comparison with VirtualConductor. As illustrated in Fig.2, our generated motion beats are better able to match the given music beats.

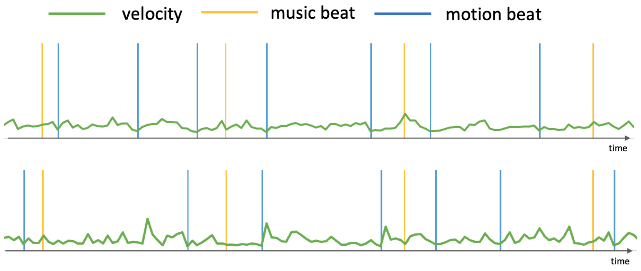

Figure 2: Qualitative comparison of beat consistency between VirtualConductor (top) and ours (bottom).

In addition, we provide visualizations of motion generation conditioned on music which were not included in the training or test sets. We randomly select the following symphonies: Tchaikovsky Piano Concerto No.1, Beethoven's Symphony No.7, The Marriage of Figaro Overture, and Vivaldi Four Seasons (Spring) (see Fig. 3).

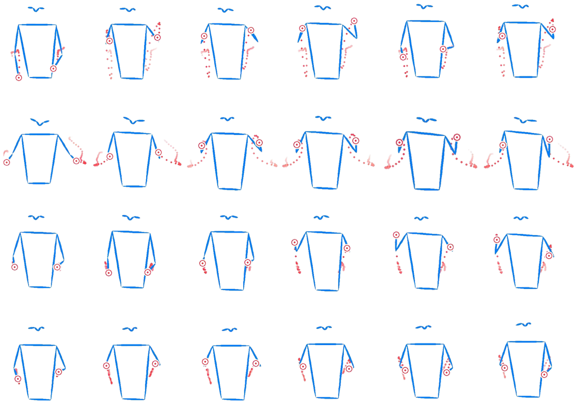

Figure 3: Visualization of the four symphonies.

Ablation Study

Comparison of predicting ϵ and x_0. We further investigate the effect of predicting the noise ϵ versus the motion x_0 via an additional study. The results as indicated in Fig. 4 show that the model trained by minimizing the loss between the noise ϵ performs much worse than one trained by minimizing the loss between motion x_0, which fails to generate plausible motion sequences in longer frames, whereas predicting x_0 successfully produces stable and plausible motion sequences. These results demonstrate the effectiveness of our design-choice to predict the motion rather than noise for each diffusion step.

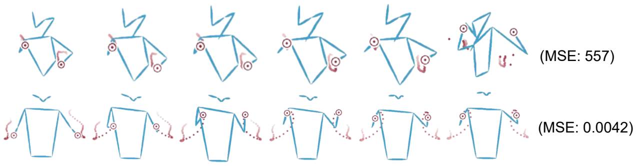

Figure 4: Qualitative comparison of generated motion of predicting ϵ (top) and x_0 (bottom) on ConductorMotion100 test set.

Effect of geometric loss. We examine the effect of incorporating a geometric loss in the training objective and compare it with one trained without its use. The results indicated in Table 2 show that the model trained with geometric loss can achieve better performance than the model trained without it on the test set. Furthermore, as visualized in Fig. 5, the model trained with geometric loss is able to produce motion with more vivid arm swings and plausible poses, which confirms its effectiveness in producing high quality motion.

Method	MSE ↓	FGD ↓	BC ↑	Diversity ↑
w/o geometric loss	0.0045	822.07	0.116	1127.90
w geometric loss	**0.0042**	**812.01**	**0.119**	**1152.06**

Table 2: Comparison of four metrics on ConductorMotion100 test set with and without geometric loss

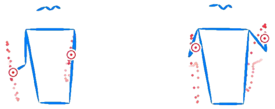

Figure 5: Qualitative comparison of generated motion of w/o (left) and w (right) geometric loss.

Conclusion

In this paper, we present `Diffusion-Conductor`, a novel DDIM-based approach for music-driven conducting motion generation, which integrates the diffusion model to a two-stage learning framework. And extensive experiments on several metrics, including Fréchet Gesture Distance (FGD) and Beat Consistency Score (BC) demonstrated the superiority of our approach.

References

Chen, D.; Liu, F.; Li, Z.; and Xu, F. 2021. VirtualConductor: Music-driven Conducting Video Generation System. *CoRR*, abs/2108.04350.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Kao, H.-K.; and Su, L. 2020. Temporally guided music-to-body-movement generation. In *Proceedings of the 28th ACM International Conference on Multimedia*, 147–155.

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

Li, R.; Yang, S.; Ross, D. A.; and Kanazawa, A. 2021. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13401–13412.

Liu, F.; Chen, D.-L.; Zhou, R.-Z.; Yang, S.; and Xu, F. 2022. Self-supervised music motion synchronization learning for music-driven conducting motion generation. *Journal of Computer Science and Technology*, 37(3): 539–558.

McFee, B.; Raffel, C.; Liang, D.; Ellis, D. P.; McVicar, M.; Battenberg, E.; and Nieto, O. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 18–25.

Mourot, L.; Hoyet, L.; Le Clerc, F.; Schnitzler, F.; and Hellier, P. 2022. A Survey on Deep Learning for Skeleton-Based Human Animation. In *Computer Graphics Forum*, volume 41, 122–157. Wiley Online Library.

Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Tang, T.; Jia, J.; and Mao, H. 2018. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *Proceedings of the 26th ACM international conference on Multimedia*, 1598–1606.

Tevet, G.; Raab, S.; Gordon, B.; Shafir, Y.; Cohen-Or, D.; and Bermano, A. H. 2022. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2020. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 13001–13008.

Zhu, L.; Liu, X.; Liu, X.; Qian, R.; Liu, Z.; and Yu, L. 2023. Taming Diffusion Models for Audio-Driven Co-Speech Gesture Generation. *arXiv preprint arXiv:2303.09119*.