GEN-SER: WHEN THE GENERATIVE MODEL MEETS SPEECH EMOTION RECOGNITION

Taihui Wang[12], *Jinzheng Zhao*[124], *Rilin Chen*[12], *Tong Lei*[2], *Wenwu Wang*[4], *Dong Yu*[3]

[1]Tencent Multimodal Models Department, Beijing, China
[2]Tencent AI Lab, Beijing, China
[3]Tencent AI Lab, Bellevue, WA, USA
[4]Centre for Vision, Speech and Signal Processing, University of Surrey, UK

ABSTRACT

Speech emotion recognition (SER) is crucial in speech understanding and generation. Most approaches are based on either classification models or large language models. Different from previous methods, we propose Gen-SER, a novel approach that reformulates SER as a distribution shift problem via generative models. We propose to project discrete class labels into a continuous space, and obtain the terminal distribution via sinusoidal taxonomy encoding. The target-matching-based generative model is adopted to transform the initial distribution into the terminal distribution efficiently. The classification is achieved by calculating the similarity of the generated terminal distribution and ground truth terminal distribution. The experimental results confirm the efficacy of the proposed method, demonstrating its extensibility to various speech-understanding tasks and suggesting its potential applicability to a broader range of classification tasks.

***Index Terms*—** Speech emotion recognition, distribution transport, generative model, target matching

1. INTRODUCTION

Speech emotion recognition (SER) is an important task in speech understanding and is beneficial for human-computer interaction [1]. It has wide applications in quality assessment [2] and monitoring [3], and can also be used as a reward model for the text-to-speech system [4]. SER can also be included as an auxiliary task for the speech tokenizer to facilitate learning of semantic information [5].

Previous methods typically employ a classification pipeline consisting of a speech encoder followed by a classifier. Early work used processed representations such as the short-time Fourier transform [6] and Mel spectrogram [1] as input features. Neural modules like convolutional neural network (CNN) [6] or long short term memory (LSTM) [1] were then applied to process these features, with a fully connected layer serving as the classifier. More recently, pretrained models have been adopted for feature extraction: for example, emotion2vec [7] leverages online knowledge distillation combined with supervised fine-tuning for emotion recognition. With advances in large language models (LLMs), several studies [8, 9] have investigated LLM-based approaches for SER. In such pipelines, a speech encoder (e.g., WavLM [10]) extracts speech representations that are aligned to the LLM input via an adapter, and the LLM is prompted (e.g., "Describe the emotion of the speaker") to produce the emotion label.

Li et al. [11] propose a diffusion-based classifier that treats classification as a conditional density-estimation problem, where class labels serve as conditional inputs, and classification is achieved by selecting the class that minimizes the noise-prediction error. This approach bridges the gap between generative and discriminative paradigms. However, it suffers from computational inefficiency, as inference latency scales linearly with the number of classes due to per-class error estimation requirements. Another related work applies generative modeling to voice activity detection (VAD) by projecting binary VAD labels into a latent space via an autoencoder [12]. Compared with the per-class evaluation required by the diffusion-based classifier [11], latent-space projection can improve computational efficiency by avoiding separate processing for each label. While latent-space projection enables efficient flow matching, its reliance on reconstruction fidelity hinders scalable deployment as training autoencoders for multi-class tasks is challenging and inevitably causes reconstruction errors during label space transformation.

In this paper, we propose Gen-SER, a generative approach that reframes SER as a distribution-transport problem. Our contributions are threefold. First, we are the first to reformulate SER as a distribution transport problem via generative models, to our knowledge. Second, we propose the sinusoidal taxonomy encoding method and map discrete labels to continuous space, avoiding multi-class autoencoder training hurdles. Third, our proposed target-matching generative model with logistic mean and bridge variance schedules enables efficient distribution transport. Experimental results demonstrate the efficacy of the proposed method in SER. Furthermore, its performance on the gender classification task highlights the model's extensibility, indicating considerable potential for adaptation to a broader range of classification tasks.

2. THE PROPOSED METHOD

2.1. Methodology

We assume: 1) speech signals expressing different emotions follow distinct initial distributions; 2) each class label corresponds to a pre-defined terminal distribution; and 3) distributions can be transported from the initial to the terminal. Based on these assumptions, the SER task can be formulated as a distribution transport problem. Given an observed data sample $\mathbf{x}_1$ drawn from an emotion-specific distribution, we transport it to $\mathbf{x}_0$ that follows the target distribution of a class via a generative model. Finally, we compute the similarity between the generated $\mathbf{x}_0$ and each class's target distribution, assigning the sample to the class with maximal similarity.

2.2. Generate the data sample $\mathbf{x}_1$ using the speech signal

Unlike conventional generative models that sample from a Gaussian prior, we use the pre-trained HuBERT model [13] to extract meaningful features from the speech signal. We use output from the final HuBERT layer as $\mathbf{x}_1$ for the input and use output from the preceding layers as conditioning $\mathbf{X}_c$.

$$\mathbf{X}_c, \mathbf{x}_1 = Average(HuBERT(\mathbf{s})), \tag{1}$$

where $\mathbf{s}$ is the time-series speech signal, $HuBERT(\cdot)$ denotes processing the speech signal by the pre-trained HuBERT model, and $Average(\cdot)$ means averaging along the temporal axis.

2.3. Generate the data sample $\mathbf{x}_0$ using the class label

Instead of training an auto-encoder to map class indices into embedding vectors [12], we first map the distinct categories to their corresponding ordinal indices, and then generate embedding vectors in the continuous space based on these indices using sinusoidal taxonomy encoding method

$$\mathbf{x}_0(b) = sin\left(\frac{2\pi \boldsymbol{l}}{L} * (i_b + 1)\right), \tag{2}$$

where b denotes the class label, i_b denotes the index of class b, L is the length of the embedding vector, and $\boldsymbol{l}$ denotes a vector consisting of integers ranging from 0 to $L-1$. This encoding method has two advantages. Firstly, the generated embedding vectors are continuous, which facilitates learning. Secondly, the embedding vectors of distinct categories are mutually orthogonal. In the following discussion, the class label b in $\mathbf{x}_0(b)$ is omitted for conciseness unless stated otherwise.

2.4. Generative model for distribution transport

The goal of the generative model is to generate $\mathbf{x}_0$ given $\mathbf{x}_1$ step by step. This distribution transport problem is modeled by an ordinary differential equation (ODE) [14]

$$\frac{d\mathbf{x}_t}{dt} = \boldsymbol{u}\left(\mathbf{x}_0, t, \mathbf{x}_1\right), \tag{3}$$

where the vector field $\boldsymbol{u}\left(\mathbf{x}_0, t, \mathbf{x}_1\right)$ determines the evolution of the perturbed signal $\mathbf{x}_t$, guiding the flow along the probability path. Considering a special case where $\mathbf{x}_t$ is Gaussian

$$p(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t(\mathbf{x}_0, \mathbf{x}_1), \sigma^2(t)\mathbf{I}), \tag{4}$$

where $\mathbf{I}$ denotes the identity matrix, the vector fields can be obtained according to [14] as

$$\boldsymbol{u}\left(\mathbf{x}_0, t, \mathbf{x}_1\right) = \frac{\sigma_t'}{\sigma_t}\left(\mathbf{x}_t - \boldsymbol{\mu}_t\left(\mathbf{x}_0, \mathbf{x}_1\right)\right) + \boldsymbol{\mu}_t'\left(\mathbf{x}_0, \mathbf{x}_1\right), \tag{5}$$

where $\sigma_t' = \frac{d}{dt}\sigma_t$ and $\boldsymbol{\mu}_t'\left(\mathbf{x}_0, \mathbf{x}_1\right) = \frac{d}{dt}\boldsymbol{\mu}_t\left(\mathbf{x}_0, \mathbf{x}_1\right)$. The $\boldsymbol{\mu}_t(\mathbf{x}_0, \mathbf{x}_1)$ is designed to follow the logistic schedule as

$$\boldsymbol{\mu}_t(\mathbf{x}_0, \mathbf{x}_1) = \mathbf{x}_0 + \frac{\mathbf{x}_1 - \mathbf{x}_0}{e^{k/2} - 1}\left(\frac{1 + e^{k/2}}{1 + e^{-k(t-0.5)}} - 1\right), \tag{6}$$

where k determines how steeply the transition occurs. Increasing k makes the switch from $\mathbf{x}_0$ to $\mathbf{x}_1$ more quickly around $t = 0.5$. The logistic mean schedule has been shown to be able to perturb the signal more effectively [15] than the widely used variance exploding, variance preserving, and linear mean schedule. For the variance, we employ the bridge schedule,

$$\sigma_t = \sigma\sqrt{t(1-t)}, \tag{7}$$

where σ is a hyperparameter determining the maximal Gaussian distribution. Note that σ_t remains non-zero and the timestep t is limited between 0 and 1.

2.5. Target estimator and training objective

Instead of estimating the whole vector field u in flow-matching, we adopt the target-matching based generative model that directly predicts the target embedding vector $\mathbf{x}_0$. Concretely, a neural network parameterized by θ mapping $\mathbf{x}_t$, the conditioning $\mathbf{X}_c$, and timestep t to an output $\mathbf{x}_\theta\left(\mathbf{x}_t, \mathbf{X}_c, t\right)$ that approximates $\mathbf{x}_0$. Compared to the widely used score-mathcing and flow-matching based generative models, the target-matching model has been proven to be more stable and efficient [15].

The neural network $\mathbf{x}_\theta$ comprises three stages. In the first stage, the conditions $\mathbf{X}_c$ are weighted by learnable parameters and summed into a fused condition $\mathbf{x}_c$. In the second stage, the embedding vectors $\mathbf{x}_t$ and $\mathbf{x}_c$ are concatenated to form an embedding vector with a length of $2L$. A linear layer is then applied to project this concatenated vector into an embedding space with a dimension of L. This step facilitates the fusion of information between the perturbation $\mathbf{x}_t$ and the condition $\mathbf{x}_c$. In the third stage, a stacked transformer architecture is used for target estimation. Within this transformer, the timestep t is explicitly injected into the network via adaptive RMS-norm [16]. We use the target matching loss to train $\mathbf{x}_\theta$ as

$$\mathcal{L}_{tm}(\theta) = \mathbb{E}_{\mathbf{x}_t|\mathbf{x}_1, \mathbf{X}_c, t, \mathbf{x}_0}\left[\left\|\mathbf{x}_\theta\left(\mathbf{x}_t, \mathbf{X}_c, t\right) - \mathbf{x}_0\right\|^2\right], \tag{8}$$

Algorithm 1 Training

```
Input: Data pairs $(\mathbf{s}, b)$, pre-trained HuBERT
For each epoch do:
    Obtain $\mathbf{x}_1$ and $\mathbf{X}_c$ according to Eq. (1)
    Obtain $\mathbf{x}_0$ according to Eq. (2)
    Sample timestep $t$ from $\mathcal{U}([t_{eps}, T])$
    Obtain $\boldsymbol{\mu}_t(\mathbf{x}_0, \mathbf{x}_1)$ and $\sigma_t$ by Eqs. (6) and (7)
    Sample $\mathbf{z}$ independently from $\mathcal{N}(0, 1)$
    Obtain perturbed signal: $\mathbf{x}_t = \boldsymbol{\mu}_t(\mathbf{x}_0, \mathbf{x}_1) + \sigma_t \mathbf{z}$
    Estimate $\hat{\mathbf{x}}_0 = \mathbf{x}_\theta(\mathbf{x}_t, \mathbf{X}_c, t)$
    Compute loss according to Eq. (8)
    Backpropagate to update $\mathbf{x}_\theta$
Output: Optimized target predictor $\mathbf{x}_\theta(\mathbf{x}_t, \mathbf{X}_c, t)$
```

Algorithm 2 Inference

```
Input: Speech signal $\mathbf{s}$, target predictor $\mathbf{x}_\theta(\mathbf{x}_t, \mathbf{X}_c, t)$,
       number of sampling steps $N$
       pre-trained HuBERT
Initialization: Obtain $\mathbf{x}_1$ and $\mathbf{X}_c$ according to Eq. (1),
       $\mathbf{x}_t = \mathbf{x}_1, t = T, n = 1$
while $n \leq N$ do:
    Compute the vector field $\boldsymbol{u}_\theta(\mathbf{x}_\theta, t, \mathbf{x}_1)$ by Eq. (9)
    Update $\hat{\mathbf{x}}_t$ according to Eq. (10)
    Estimate $\mathbf{x}_\theta(\mathbf{x}_t, \mathbf{X}_c, t)$
    Update timestep: $t = T - n/N, n = n + 1$
Classification: Predict the class by Eq. (12)
Output: The predicted class index $\hat{i}_b$
```

where $\| \cdot \|$ denotes the Euclidean norm.

We summarize the training process in Algorithm 1, in which the target estimator $\mathbf{x}_\theta(\mathbf{x}_t, \mathbf{X}_c, t)$ is optimized. Given a speech signal $\mathbf{s}$ and its class label b, we extract the initial data sample $\mathbf{x}_1$ and the conditioning variable $\mathbf{X}_c$ using Eq. (1). Then we derive the target terminal sample $\mathbf{x}_0$ via Eq. (2). Subsequently, t is drawn from a uniform distribution over $[\epsilon, T]$, with ϵ ensuring the numerical stability, and T being the maximum diffusion time. In the next step, the perturbed signal $\mathbf{x}_t$ is given by $\mathbf{x}_t = \boldsymbol{\mu}_t(\mathbf{x}_0, \mathbf{x}_1) + \sigma_t \mathbf{z}$, where $\mathbf{z}$ is independently sampled from $\mathcal{N}(0, 1)$. $\boldsymbol{\mu}_t(\mathbf{x}_0, \mathbf{x}_1)$ and σ_t are calculated according to Eq. (6) and Eq. (7), respectively. Finally, the loss described in Eq. (8) is computed and backpropagated to update the parameters of $\mathbf{x}_\theta(\mathbf{x}_t, \mathbf{X}_c, t)$

2.6. Sampling method

Following the training of the target estimator, the initial sample $\hat{\mathbf{x}}_T$ is drawn from the speech signal. Then, the vector field is calculated as

$$\begin{aligned}\boldsymbol{u}_\theta(\mathbf{x}_\theta, t, \mathbf{x}_1) = \frac{\sigma_t'}{\sigma_t} \left(\mathbf{x}_t - \boldsymbol{\mu}_t \left(\mathbf{x}_\theta(\mathbf{x}_t, \mathbf{X}_c, t), \mathbf{x}_1\right)\right) \\ + \boldsymbol{\mu}_t' \left(\mathbf{x}_1, \mathbf{x}_\theta(\mathbf{x}_t, \mathbf{X}_c, t)\right).\end{aligned} \tag{9}$$

The Euler ODE solver is employed to estimate terminal sample $\hat{\mathbf{x}}_0$ by iteratively computing

$$\hat{\mathbf{x}}_{T-n/N} \approx \hat{\mathbf{x}}_t + \boldsymbol{u}_\theta(\hat{\mathbf{x}}_t, t, \mathbf{x}_1)/N, \tag{10}$$

$$t = T - n/N, \tag{11}$$

where $\hat{\mathbf{x}}_t = \mathbf{x}_\theta(\mathbf{x}_t, \mathbf{X}_c, t)$, N denotes the number of timesteps, n denotes the current timestep, and t is set to T at the start.

We calculate the cosine similarity between $\hat{\mathbf{x}}_0$ and the embedding vectors associated with each class label b, and the predicted class is assigned by selecting the label corresponding to the highest similarity score

$$\hat{i}_b = \underset{b}{\operatorname{argmax}} \frac{\langle \hat{\mathbf{x}}_0, \mathbf{x}_0(b) \rangle}{\|\hat{\mathbf{x}}_0\| \, \|\mathbf{x}_0(b)\|}, \tag{12}$$

where $< \cdot, \cdot >$ denotes the inner product of two vectors.

In Algorithm 2, we summarize the inference process. Given a speech signal, we first extract the initial data sample $\mathbf{x}_1$ and the conditioning variable $\mathbf{X}_c$ using Eq. (1), and then the estimated terminal sample $\hat{\mathbf{x}}_0$ is obtained iteratively via the Euler ODE solver. During the classification stage, the predicted class is determined by the similarity between the estimated terminal sample and each class.

3. EXPERIMENTAL EVALUATIONS

3.1. Dataset

We mainly focus on emotion classification for English corpora and we use crema-d [17], emodb [18], TESS, savee [19], RAVDESS [20], MELD [21], and an in-house emotion classification dataset as our training set. There are over 52k data items and 48 hours in total. We choose MELD [21] test set to evaluate the model performance. For the baseline models, we choose two types of methods for comparison. The first type is deterministic model and follows the paradigm of encoder + classifier, such as emotion2vec [7] and WavLM [10]. The second type is a generative model and follows the paradigm of encoder + LLM, such as Qwen-audio [8], Qwen2-audio [9] and OSUM [22].

3.2. Implementation details

In Eq. (1), we use the chinese-hubert-large[1] model with 24 transformer layers. In the third stage of the neural network $\mathbf{x}_\theta$, we employ a transformer architecture comprising four layers with an embedding dimension of 1024 and a 16-head self-attention mechanism. Combined with the neural networks from the first and second stages, the model contains a total of 71.4 M parameters. Training is conducted with a batch size of 128 for 400k steps, using a learning rate of 5×10^{-4}. We report accuracy as the evaluation metric for each model.

[1] https://huggingface.co/TencentGameMate/chinese-hubert-large

Table 1: Accuracy for SER on MELD [21] test set. CLS denotes a classification layer.

Model	**Model Type**	**Accuracy(%)**
WavLM + CLS	Classification	50.6
Hubert + CLS	Classification	53.4
emotion2vec [7]	Classification	51.9
Qwen-audio [8]	LLM	55.7
Qwen2-audio [9]	LLM	55.3
OSUM [22]	LLM	53.4
SenseVoice-L [23]	LLM	**63.1**
Ours	Diffusion	56.5

Table 2: Accuracy for gender classification on Air-Bench [24]. Results of Qwen2-audio [9], Qwen-audio Turbo [8] and Soundwave [25] are from [25].

Model	**Type**	**Accuracy(%)**
Fbank + CLS	Classification	86.6
WavLM + CLS	Classification	87.5
Qwen2-audio [9]	LLM	79.3
Qwen-audio Turbo [8]	LLM	82.5
Soundwave [25]	LLM	90.3
Ours	Diffusion	**90.5**

3.3. Experimental results

We compare the accuracy of the proposed method and baselines in Table 1. The first observation is that LLM-based methods are superior to the classification methods, potentially due to the large scale of training data and the mutual reinforcement between different training tasks of LLM. In addition, the proposed method outperforms both classification methods and most LLM-based methods, which shows that the proposed method can achieve efficient distribution transport. The performance of our method is not as good as SenseVoice-L [23] and potential reasons are that SenseVoice-L is trained on a large dataset comprising 30M items [22], and it can benefit from LLM's ability to understand semantic information.

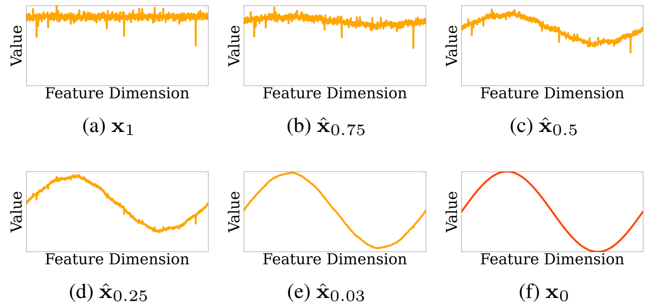
We applied the proposed method to the gender classification task. We use an in-house dataset over 750k items, amounting to 1023 hours. The experimental results are shown in Table 2. The proposed method outperforms other baselines. This shows that our method is versatile and can be extended to other speech understanding tasks.

3.4. Investigation on the distribution transport

Fig. 1a shows the initial embedding vector $\mathbf{x}_1$ extracted using the HuBERT model. Fig. 1f shows the target terminal embedding vector $\mathbf{x}_0$ generated using Eq. (2) with label index $i_b = 0$. Fig. 1b to Fig. 1e show the variation process of the estimated $\hat{\mathbf{x}}_t$ over the timestep. As the timestep decreases, we observe that the generated embedding vector $\hat{\mathbf{x}}_t$ progressively shifts from the initial embedding vector $\mathbf{x}_1$ to the target embedding vector $\mathbf{x}_0$. This result validates the effectiveness of the proposed generative distribution-shift-based classification model. One issue with generative models is their reasoning steps, and hence we investigate the impact of the number of reasoning steps on the proposed method. Table 3 illustrates the performance of the proposed method with respect to the number of reasoning steps. We observed that the number of reasoning steps steadily improves the accuracy, and the performance of single-step reasoning is already satisfactory.

Table 3: Accuracy with different reasoning steps (N)

N	1	2	4	10	20
Accuracy	0.5613	0.5617	0.5625	0.5628	0.5644



(a) $\mathbf{x}_1$ (b) $\hat{\mathbf{x}}_{0.75}$ (c) $\hat{\mathbf{x}}_{0.5}$

(d) $\hat{\mathbf{x}}_{0.25}$ (e) $\hat{\mathbf{x}}_{0.03}$ (f) $\mathbf{x}_0$

Fig. 1: Mean transformation process at different timesteps.

4. CONCLUSION

We have presented Gen-SER, a generative framework that recasts speech emotion recognition as distribution transport: discrete emotion labels are mapped into continuous embedding vectors via sinusoidal taxonomy encoding to serve as the diffusion targets for HuBERT speech features. The target-matching-based generative model with logistic mean and bridge variance schedules is adopted to realize efficient distribution transport. Experimental results demonstrate that the proposed method shows competitive performance in SER and is applicable in other tasks, demonstrating a new route for speech understanding.

5. REFERENCES

[1] S. K. Pandey, H. S. Shekhawat, and S. M. Prasanna, "Deep learning techniques for speech emotion recognition: A review," in *The 29th International Conference RADIOELEKTRONIKA*. IEEE, 2019, pp. 1–6.

[2] H. Xu, J. Gao, and J. Yuan, "Application of speech emotion recognition in intelligent household robot," in *2010 International Conference on Artificial Intelligence and Computational Intelligence*. IEEE, 2010, vol. 1, pp. 537–541.

[3] E. Lieskovská, M. Jakubec, R. Jarina, and M. Chmulík, "A review on speech emotion recognition using deep learning and attention mechanism," *Electronics*, vol. 10, no. 10, pp. 1163, 2021.

[4] Z. Du, C. Gao, Y. Wang, et al., "Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training," *arXiv preprint arXiv:2505.17589*, 2025.

[5] C. Gao, Z. Du, and S. Zhang, "Differentiable reward optimization for llm based tts system," *arXiv preprint arXiv:2507.05911*, 2025.

[6] N. Mustaqeem and S. Kwon, "A cnn-assisted enhanced audio signal processing for speech emotion recognition," *Sensors*, vol. 20, no. 1, pp. 183, 2019.

[7] Z. Ma, Z. Zheng, J. Ye, et al., "emotion2vec: Self-supervised pre-training for speech emotion representation," *arXiv preprint arXiv:2312.15185*, 2023.

[8] Y. Chu, J. Xu, X. Zhou, et al., "Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models," *arXiv preprint arXiv:2311.07919*, 2023.

[9] Y. Chu, J. Xu, Q. Yang, et al., "Qwen2-audio technical report," *arXiv preprint arXiv:2407.10759*, 2024.

[10] S. Chen, C. Wang, Z. Chen, et al., "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[11] A. C. Li, M. Prabhudesai, S. Duggal, et al., "Your diffusion model is secretly a zero-shot classifier," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2206–2217.

[12] Z. Chen, B. Han, S. Wang, et al., "Flow-tsvad: Target-speaker voice activity detection via latent flow matching for speaker diarization," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[13] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, et al., "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[14] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *Proc. ICLR*, Feb. 2023.

[15] T. Wang, R. Chen, T. Lei, et al., "Target matching based generative model for speech enhancement," *arXiv preprint arXiv:2509.07521*, 2025.

[16] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 4195–4205.

[17] H. Cao, D. G. Cooper, M. K. Keutmann, et al., "Crema-d: Crowd-sourced emotional multimodal actors dataset," *IEEE Transactions on Affective Computing*, vol. 5, no. 4, pp. 377–390, 2014.

[18] F. Burkhardt, A. Paeschke, M. Rolfes, et al., "A database of german emotional speech.," in *Interspeech*, 2005, vol. 5, pp. 1517–1520.

[19] P. Jackson and S. Haq, "Surrey audio-visual expressed emotion (savee) database," *University of Surrey: Guildford, UK*, 2014.

[20] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," *PloS one*, vol. 13, no. 5, pp. e0196391, 2018.

[21] S. Poria, D. Hazarika, N. Majumder, et al., "Meld: A multimodal multi-party dataset for emotion recognition in conversations," *arXiv preprint arXiv:1810.02508*, 2018.

[22] X. Geng, K. Wei, Q. Shao, et al., "Osum: Advancing open speech understanding models with limited resources in academia," *arXiv preprint arXiv:2501.13306*, 2025.

[23] K. An, Q. Chen, C. Deng, et al., "Funaudiollm: Voice understanding and generation foundation models for natural interaction between humans and llms," *arXiv preprint arXiv:2407.04051*, 2024.

[24] Q. Yang, J. Xu, W. Liu, et al., "Air-bench: Benchmarking large audio-language models via generative comprehension," *arXiv preprint arXiv:2402.07729*, 2024.

[25] Y. Zhang, Z. Liu, F. Bu, et al., "Soundwave: Less is more for speech-text alignment in llms," *arXiv preprint arXiv:2502.12900*, 2025.