Automatic Lyric Transcription and Automatic Music Transcription from Multimodal Singing

XIANGMING GU, LONGSHEN OU, WEI ZENG, JIANAN ZHANG, NICHOLAS WONG, and YE WANG, National University of Singapore, Singapore

Automatic lyric transcription (ALT) refers to transcribing singing voices into lyrics while automatic music transcription (AMT) refers to transcribing singing voices into note events, i.e., musical MIDI notes. Despite these two tasks having significant potential for practical application, they are still nascent. This is because the transcription of lyrics and note events solely from singing audio is notoriously difficult due to the presence of noise contamination, e.g., musical accompaniment, resulting in a degradation of both the intelligibility of sung lyrics and the recognizability of sung notes. To address this challenge, we propose a general framework for implementing multimodal ALT and AMT systems. Additionally, we curate the first multimodal singing dataset, comprising N20EMv1 and N20EMv2, which encompasses audio recordings and videos of lip movements, together with ground truth for lyrics and note events. For model construction, we propose adapting self-supervised learning models from the speech domain as acoustic encoders and visual encoders to alleviate the scarcity of labeled data. We also introduce a residual cross-attention mechanism to effectively integrate features from the audio and video modalities. Through extensive experiments, we demonstrate that our single-modal systems exhibit state-of-the-art performance on both ALT and AMT tasks. Subsequently, through single-modal experiments, we also explore the individual contributions of each modality to the multimodal system. Finally, we combine these and demonstrate the effectiveness of our proposed multimodal systems, particularly in terms of their noise robustness.

CCS Concepts: • **Applied computing** → **Sound and music computing**; • **Information systems** → **Music retrieval**; **Speech / audio search**; • **Computing methodologies** → *Neural networks*; • **Human-centered computing** → Ubiquitous and mobile computing systems and tools.

Additional Key Words and Phrases: singing, automatic lyric transcription, automatic music transcription, multimodality, dataset, self-supervised-learning

ACM Reference Format:
Xiangming Gu, Longshen Ou, Wei Zeng, Jianan Zhang, Nicholas Wong, and Ye Wang. 2024. Automatic Lyric Transcription and Automatic Music Transcription from Multimodal Singing. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1 (March 2024), 28 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Singing contains both textual and musical information. As an important component of singing voice analysis, automatic transcription of singing voice includes automatic lyric transcription (ALT) and automatic music transcription (AMT). The former is the task of recognizing textual information, while the latter is the task of identifying musical information, including onsets/offsets/pitch of note events. The above two tasks facilitate solving many downstream music information retrieval problems. For instance, ALT can be applied to lyric alignment [29], query by singing [33], audio

Authors' address: Xiangming Gu; Longshen Ou; Wei Zeng; Jianan Zhang; Nicholas Wong; Ye Wang, National University of Singapore, Singapore, {xiangming,oulongshen,w.zeng,e0950471,wong_nicholas}@u.nus.edu,wangye@comp.nus.edu.sg.

1551-6857/2024/3-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

indexing [20], music subtitling [18], and singing pronunciation evaluation [59]. AMT can be applied to sight-singing evaluation [80], music therapy [72], and human-computer interaction [58, 77]. Furthermore, they can also be employed in singing voice synthesis [37, 45], which is a topic that has recently been actively studied in the singing field.

Traditionally, ALT and AMT systems are built **only on audio modality** and treated as separate tasks with distinct objectives. However, they encounter certain common challenges, which motivate developing a generalized solution.

Insufficient robustness for noise. Audio recordings of singing may be accompanied with noise, e.g. background music. In challenging signal-to-noise ratio (SNR) environments, the intelligibility of singing in the audio modality will be drastically reduced, thus affecting the information retrieval of lyrics and musical note events. In our previous work [27], we showed that attempting the ALT task solely on audio recordings in noisy environments yields unsatisfactory performance. Additionally, [26, 36] showed that low SNR environments greatly harm the performance of pitch estimation from speech. Considering that singing and speech share similarities in terms of the sound production mechanism, it is reasonable to surmise that attempting audio-only AMT from singing voices would probably meet the similar challenge of noise robustness.

Limited data for complex tasks. Singing transcription is notably much more difficult compared to speech-related recognition tasks due to the scarcity of labeled data and the intricate intertwining of textual and musical information within singing. Speech recognition benefits from large-scale annotated datasets such as LibriSpeech [63], which comprises 960 hours of annotated speech recordings. In contrast, DSing [8, 13], a widely used ALT dataset, has about 150 hours of data, and the largest AMT dataset, MIR-ST500 [76], only contains around 30 hours. The scarcity of labeled data arises from the time-consuming process of manual annotation, where extensive musical knowledge is necessary. Additionally, singers inevitably have to adjust or compromise certain linguistic features, such as word stress and articulation, to accommodate properties or constraints of singing that are not present in regular speech, such as melody, tempo, or deliberate timbre adjustments. As a result, singing tends to be less intelligible as compared to speech [67], thereby further complicating the transcription process.

The perception of both speech and singing extends beyond the auditory realm, as exemplified by the McGurk effect [50]. This phenomenon highlights the significant impact of visual information on auditory perception. Inspired by this, we assume that incorporating more modalities in singing will enhance the performance of both ALT and AMT systems, particularly concerning noise robustness. In our previous work [27], we developed the first multimodal ALT system, MM-ALT, capable of processing audio, video, and IMU inputs. Comparative analyses between MM-ALT and its single-modal counterparts revealed that supplementary modalities, especially videos of lip movements, significantly contribute to noise robustness. However, the realm of AMT from multimodal singing has not been explored yet. In a position paper [77], the potential of multimedia fusion approaches in improving AMT from music or singing was mentioned. To address this research gap and validate our assumption, we extend our previous work [27] to accommodate for both multimodal ALT and AMT. In developing our multimodal system, we propose adapting self-supervised learning models, e.g. wav2vec 2.0 [3] and AV-HuBERT [68] from the speech domain to the singing domain. This approach addresses the issue of limited data availability for tasks of audio-only ALT and AMT. In this manner, we harness the abundance of speech data. Furthermore, to enhance the integration of representations from various modalities, we introduce a residual cross-attention mechanism, which combines self-attention and cross-attention to effectively utilize the strengths of each modality and exploit the complementary relationships among different modalities. To summarize, our contributions are four-fold:

- We present a general framework for ALT and AMT from multimodal singing. Our framework incorporates both audio and video modalities. To support the development of these systems, we curate the first multimodal singing dataset, consisting of N20EMv1 for ALT and N20EMv2 for AMT. By introducing the video modality, our systems demonstrate increased noise robustness. With severe perturbations of musical accompaniment (-10 dB SNR), our systems outperform their audio-only counterparts by large margins.
- We adapt self-supervised learning (SSL) models from the speech domain to the singing domain, employing our proposed adaptation method. Consequently, our audio-only systems achieve state-of-the-art performance for both ALT and AMT tasks on widely used benchmark singing datasets, including DSing [8, 13], DALI [54, 55], Jamendo [70], Hansen [31], Mauch [49], MIR-ST500 [76], TONAS [25], and ISMIR2014 [57].
- We initialize new tasks of lyric lipreading and note lipreading utilizing only video information. Our systems are capable of extracting language-related information (lyrics) and music-related information (note events) from only video modality.
- We introduce Residual Cross Attention (RCA), a new feature fusion method to better fuse the multimodal singing features, leveraging both self-attention and cross-attention mechanisms.

Our previous work [27] focused on the construction and evaluation of the multimodal ALT system. This article extends it in the following aspects: (1) We propose a generalized problem setting for both ALT and AMT from multimodal singing voice and we focus on the audio and video two modalities. (2) Based on the data collected in [27], we curate a new dataset named N20EMv2 with the annotations tailored for AMT. (3) We propose a novel adaptation strategy for AMT. (4) We conduct extensive experiments for single-modal and multimodal AMT systems. (5) We incorporate more comparison experiments and ablation studies to demonstrate the effectiveness of our methods.

2 RELATED WORK

2.1 Automatic Lyric Transcription

Automatic lyric transcription (ALT), the counterpart task of automatic speech recognition (ASR) in the field of music information retrieval, has evolved with various approaches. The initial work [33] developed a Japanese ALT system by adapting a Hidden-Markov-Model (HMM). [53] investigated the impact of in-domain lyric language models (LM) on transcription performance. Additionally, [51] leveraged the repetitive patterns of songs to enhance the consistency and accuracy of their transcription system. The advent of deep learning and benchmark datasets DSing [8, 13] and DALI [54, 55] enabled data-driven deep learning approaches for ALT. Notably, [8, 13, 15] proposed employing a DNN-HMM framework with factorized time-delay neural network (TDNN-F) or its variations as the feature encoder. Additionally, [16] adopted a connectionist temporal classification (CTC) architecture, utilizing a CRNN as the encoder, while [4, 29] implemented the hybrid CTC-Attention framework [78] in this task.

Despite the efforts, the progress in developing ALT systems is hindered by the limited availability of large-scale singing datasets [81]. Although DSing and DALI provide some support for ALT, they still fall short in scale. Moreover, the issue of copyright protection surrounding singing recordings restricts the sharing and accessibility of such data. Data augmentation emerges as a viable solution to alleviate the data scarcity problem. Previous research explored techniques like random time stretching, pitch adjustment [41], and vocoder-based synthesis [4] to transform speech data into a more "song-like" form. Moreover, [81] proposed a method that aligns lyrics with melodies before adjusting duration and pitch during data augmentation. However, these methods complicate the workflow of building transcription systems, making the training more computationally intensive. To resolve the problem of limited data more efficiently, we leverage the similarities between speech and singing. In

our previous work [27], we proposed adapting self-supervised-learning (SSL) models, e.g. wav2vec 2.0 [3], from the speech domain as acoustic models for ALT. Building upon this, the subsequent wav2vec 2.0-based ALT system achieved state-of-the-art performance on all benchmark singing datasets and exhibited few-shot capabilities [61]. Subsequently, [23] proposed a semi-supervised learning method to further improve the few-shot performance using the same ALT system.

Although ALT is the counterpart task of ASR, it still presents unique research problems that must be overcome to successfully adapt ASR systems for ALT. One major challenge is that singing is typically accompanied by musical instruments, resulting in polyphonic inputs. [29] introduced a genre-informed acoustic model for ALT systems under polyphonic scenarios. Follow-up research efforts enhanced this framework with genre adapters [22], and multi-task setting [21], etc. However, all these methods only consider the audio modality and do not incorporate additional information. When faced with challenging signal-to-noise ratio (SNR) environments or other types of sound contamination, these approaches may struggle to accurately transcribe lyrics. This, therefore, motivates the use of multimodal approaches for ALT.

2.2 Automatic Music Transcription

Automatic Music Transcription (AMT) involves three subtasks: musical note's onset detection, offset detection, and pitch estimation. Initial research focused on fundamental frequency (F0) estimation. One of the representative works is YIN [10], which utilized auto-correlation to estimate F0 from speech or music signals. pYIN, an extension of YIN, improved pitch estimation with multiple candidates and HMM-based refinement [48]. With the emergence of data-driven deep learning techniques, CREPE introduced a CNN architecture for frame-level pitch estimation and achieved state-of-the-art performance [39]. Simultaneously, PatchCNN used a patch-based CNN for pitch contour extraction [71]. Afterwards, SPICE introduced a self-supervised task for pitch estimation without relying on large labeled datasets [24]. Previous approaches primarily focused on predicting pitch values in frequency, while TONet considered tone (pitch name) and octave as the pitch targets [5]. These aforementioned works concentrate on pitch estimation, thus neglecting the other aspects of note events, i.e. onsets and offsets.

We narrow our focus on AMT from singing. Notably, AMT shares similarities with audio-to-score conversion [2, 60, 65], whose targets are symbolic representations that reflect what musicians read. However, our current work focuses on transcribing note events, rather than musical scores. In the pre-deep learning era, Tony, a software tool based on HMM, was developed to transcribe note onsets, offsets, and MIDI pitch values from singing recordings [47]. Afterwards, HCN [19] and VOCANO [34] adopted PatchCNN [71] for pitch estimation and integrated a note segmentation network for onset and offset detection. VOCANO also utilized virtual adversarial training [56] to leverage unlabeled singing data to improve performance. Recently, there has been a growing interest in end-to-end frameworks. For instance, [76] used an Efficient-Net [73] architecture to transcribe singing notes in an end-to-end manner and introduced the MIR-ST500 dataset, which is the largest singing dataset with human annotations for AMT. Besides, [42] employed a pretrained pitch estimation network and a quantization algorithm to generate frame-level pseudo labels, training an end-to-end AMT system using the noisy student framework [79]. It is noted that [42] directly transcribed singing notes from polyphonic singing while other approaches relied on source separation, like Demucs [12], as a preprocessing step. Regardless, both approaches still struggle to transcribe/separate singing audio when the instrumental musical accompaniment is much louder, i.e. a challenging SNR environment. Additionally, [28] found that AMT systems tend to perform better on females and proposed an approach to alleviate this fairness issue.

Table 1. Data Collection and Processing for audio and video modalities. v1, v2 denote N20EMv1, N20EMv2.

Modality	Device	Resolution			Frequency (Hz)		
		Raw	v1	v2	Raw	v1	v2
Audio	Audio-Technica 4050	32-bit depth			44.1k	16k	
Video	Sony AX4	1920x1080 pixels	96x96 pixels		50	25	50

2.3 Multimodal Learning

Humans rely on multiple modalities, e.g., sight, hearing, taste, touch, and smell, to perceive and understand their surroundings. Each modality provides distinct and complementary information, enhancing the overall understanding. For instance, previous research showed that visual cues in speech provide valuable assistance in language learning [9, 52]. Inspired by this, many deep learning models are designed to enable multimodal input. Even though the original task may be designed for a single modality, the introduction of extra modalities brings empirical performance gains. In the speech domain, [46, 68, 69, 74] fused audio and video modalities to enhance the performance of speech recognition and speaker detection. In the vision domain, [35, 66] combined both RGB images and depth images to improve tasks like semantic segmentation. Besides the empirical findings of the general superiority of multimodal approaches over their single-modal counterparts, [38] derived the framework of multimodal learning problem from the theoretical perspective. Then they proved that learning with multiple modalities tends to have a better latent representation quality than that with a subset of modalities, thus providing a theoretical guarantee for better performance of multimodal systems.

3 MULTIMODAL SINGING DATASET: N20EMV1 AND N20EMV2

3.1 Singer Profile and Song Selection

Firstly, we recruited 30 participants from a local university to collect data, comprising 17 males and 13 females. To promote singer diversity, the participants were selected with varying accents, including a range of European, Indian, North American, East Asian, and Southeast Asian accents. Moreover, their abilities varied widely, from individuals with no formal vocal training to those who are amateur-level singers. To ensure song diversity, we chose 20 songs from [17] based on their rich phonemic coverage and variation in musical features, e.g. genre and tempo. Furthermore, these songs are easy for singers to learn. Although each participant was given the freedom to choose the songs according to their preferences, we made adjustments to ensure a limit of 10 singers for each song at maximum to balance the dataset.

3.2 Multimodal Singing Data Recording

To ensure a controlled and undisturbed environment, we conducted the singing data recording in a soundproof studio. The recording setup included specific equipment for each modality: an Audio-Technica 4050 condenser microphone with a pop filter for audio, a Sony AX4 video camera for video. Before the recording session, the singers were instructed to wear a monaural headset for playback of musical accompaniment. The video camera, accompanied by a ring light, was positioned in front of the singers, prioritising on the movements of the lower half of each singer's face, especially the jaw, lips, and tongue. The camera can also collect audio signals, which were only used for modality synchronization. Lyric sheets for the selected songs were provided on a music stand for reference.

During the recording, singers were advised to minimize bodily movements to reduce noise interference in the data. While the tempo and key of each song were predetermined, the singers had the flexibility to choose between male-vocal or female-vocal arrangements for songs that better suits their own vocals in pitch range and timbre. They also had some freedom in their rendition of pitch

Table 2. Statistics of our N20EMv1 dataset.

Split	Duration (min)	Num. of Utterances
Total	323	5116
Train	241	3803
Valid	35	616
Test	47	697

Table 3. Statistics of our N20EMv2 dataset.

Split	Duration (min)	Num. of songs
Total	502	157
Train	386	123
Valid	47	16
Test	69	18

and rhythm. Minor pronunciation errors were allowed as long as the clarity of the vocals remained intact. The recording process adhered to standard practices where singers were instructed to monitor the musical accompaniment through their monaural headsets, ensuring only vocal voices were captured. After each complete song recording, the singer moved on to the next song. Table 1 presents the resolutions and frequencies of the raw data for audio and video modalities. Following the recording, the raw data from the two modalities were synchronized using audio recorded from each device.

3.3 Multimodal Singing Data Processing

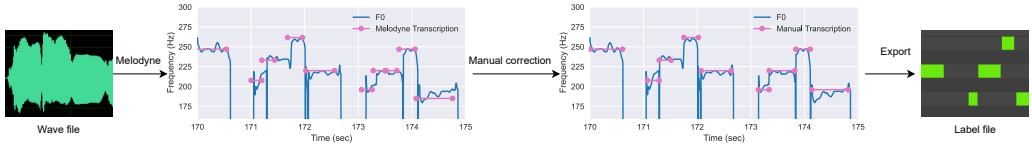
As presented in Table 1, we modified the resolutions and frequencies when curating N20EMv1 and N20EMv2 datasets. Specifically, the audio data was down-sampled to 16 kHz and transformed into a single channel for both two datasets to meet the input requirements of self-supervised learning (SSL) models from the speech domain, e.g., wav2vec 2.0 [3]. For raw video data, we followed the approach in [68] to crop regions-of-interest (ROIs) centered around the mouth region, resulting in a resolution of 96x96 pixels. This cropping technique not only reduces unnecessary information but also helps protect the privacy of the singers. The video data was down-sampled to 25 Hz for N20EMv1, adhering to the input specifications of AV-HuBERT [68]. However, for N20EMv2, we retained a frame rate of 50 Hz as a higher temporal resolution is crucial for accurate AMT.

Following the practices in benchmark ALT datasets, e.g. DSing [8, 13], and benchmark AMT datasets, e.g. MIR-ST500 [76], we curated N20EMv1 at the utterance-level and N20EMv2 at the song-level. As the raw data was already in the song-level form, it was directly used in N20EMv2 after the aforementioned pre-processing. For N20EMv1, we divided whole songs into utterances, and further details about this process are presented in the next section. Subsequently, the data was partitioned into train/valid/test splits, following the same division scheme used in our previous work [27] for N20EMv1 (different splits have no overlapping songs). The statistics of N20EMv1 and N20EMv2 can be found in Table 2 and Table 3. Notably, the total duration of N20EMv2 is longer than that of N20EMv1 due to the exclusion of silent utterances in N20EMv1.

3.4 Lyric Annotation for N20EMv1

The lyric annotation primarily focuses on the audio modality, as audio and video were already synchronized. To segment the whole song into utterances, an expert uses spectrogram information and the marker function in Adobe Audition software to annotate each utterance's starting and ending timestamps. The standards are established based on natural factors such as musical cadence, as well as practical considerations, including a preference for consonant boundaries over vowel boundaries between utterances. For each utterance, actually sung words between the starting and ending timestamps are served as the lyric annotation. In some cases where the sung words by the singers are different from the correct lyrics that should have been sung, the actual sung words are used in the annotations. We also provide the annotations for different types of errors, detailed in Appendix A. Following the completion of the annotation process, the recordings of the two modalities are segmented at the utterance level based on the provided annotations. Additionally,

Fig. 1. Illustration of our coarse-to-fine annotation procedure for the N20EMv2 dataset.



any instances of silence, breaths, or non-phonemic noise occurring between utterances are removed from the data. Similarly, the musical accompaniment is segmented accordingly.

3.5 Note Annotation for N20EMv2

The note annotation is also conducted on the audio modality. A coarse-to-fine method is used to enhance the annotation precision, as depicted in Fig. 1. In the first stage, we use Melodyne[1], a professional digital signal processing software, to obtain coarse annotations. Then in the second stage, a manual refinement is performed, involving the adjustment of onset/offset/pitch. This is achieved by concurrently playing and comparing the annotations and audio tracks simultaneously from an interface comprising spectrogram, waveform, and MIDI notes. Given that note annotation demands extensive musical knowledge and is time-consuming, two experts are assigned to complete the task. To ensure inter-rater reliability, several rules (detailed in Appendix A) are established as guidelines.

4 METHODOLOGY

4.1 Problem Formulation

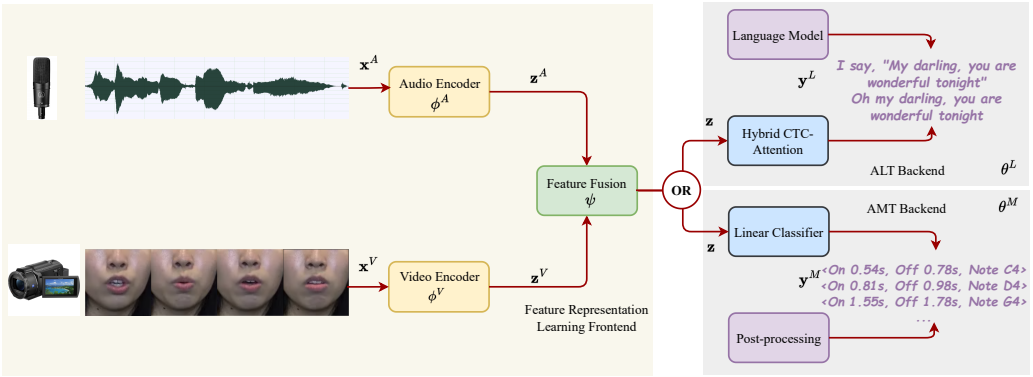
We consider a general setting for both automatic lyric transcription (ALT) and automatic music transcription (AMT) from singing. Specifically, given the synchronized singing recordings from multiple modalities (in this work, we consider audio and video modalities, $\boldsymbol{x}^A$ and $\boldsymbol{x}^V$, our framework can be seamlessly adopted to scenarios with more modalities), the ALT target is a sequence of tokens $\boldsymbol{y}^L = \{y_1^L, y_2^L, ..., y_{N_1}^L\}, y_n^L \in \mathbb{V}$, where N_1 is the length of output sequence and $\mathbb{V}$ represents the vocabulary comprising all possible tokens. Since lyrics belong to the textual modality, various tokenizers, such as characters, words, subwords, or phonemes can be used to represent tokens. In this work, we use a character tokenizer. Then the vocabulary has 26 English letters, 4 special characters (beginning of sentence < bos >, end of sentence < eos >, quotation <′>, and word boundary < >). AMT aims to produce a sequence of note events $\boldsymbol{y}^M = [(o_1, f_1, p_1), (o_2, f_2, p_2), ..., (o_{N_2}, f_{N_2}, p_{N_2})]$, where o_n and f_n are the onset/offset time of n-th note, $0 \leq o_1 < f_1 \leq o_2 < f_2 \leq ... \leq o_{N_2} < f_{N_2}$, p_n is the note pitch value and N_2 represents the number of note events. Consequently, the multimodal ALT system is a function that maps $\boldsymbol{x}^A$ and $\boldsymbol{x}^V$ into $\boldsymbol{y}^L$ while the multimodal AMT system is a function that maps into $\boldsymbol{y}^M$.

Each system consists of a feature representation learning frontend and a task-specific backend. Initially, modality-specific encoders ϕ^A and ϕ^V are employed to extract the feature representations for each modality input. The modality feature fusion module ψ first aligns the features from different modalities to ensure the features have the same number of frames and dimensions. Afterwards, ψ projects the features from different modalities into a shared latent space and integrates them to obtain more informative representations. Finally, θ^L and θ^M transform the fused representations into lyrics and note events, respectively.

Considering that the lengths of input modalities and output modalities do not possess fixed relationships, we formulate multimodal ALT and multimodal AMT as two sequence-to-sequence

[1]https://www.celemony.com/en/melodyne/what-is-melodyne

Fig. 2. Framework of our multimodal ALT system or multimodal AMT system.



problems. While these two systems share the same architectures (not their parameter weights) for encoders, they are trained separately. It is worth noting that (i) our systems can accommodate a single input modality or multiple input modalities, and (ii) our systems can be extended to output both lyrics and note events simultaneously. We direct readers to Sec. 6 for further discussions.

4.2 Modality-Specific Encoders

The audio encoder ϕ^A is designed to learn acoustic representations for audio modality. We propose the adaptation of self-supervised learning (SSL) models, specially wav2vec 2.0 LARGE [3], from the speech domain to the singing domain. The rationale behind this choice is that SSL models, pretrained on abundant speech data, exhibit strong generalization capabilities even provided with low-resource labeled data in new domains. wav2vec 2.0 consists of a CNN-based feature encoder and a Transformer-based context network. The feature encoder has seven temporal 1D convolutional blocks. It takes the raw waveform of the singing audio and produces latent singing representations. The latent singing representations are then fed into the context network. By capturing global temporal information, the context network transforms the latent singing representations into contextual singing representations. The resulting output $\mathbf{z}^A$ has a frame rate of approximately 49.8 Hz (equivalent to a frame length of about 20 ms), with each frame having 1,024 dimensions.

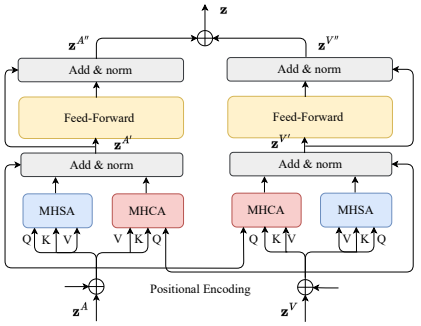
The video encoder ϕ^V is designed to learn visual representations of singing from videos of lip movements. We propose the adoption of AV-HuBERT LARGE [68] in our system, which is one of the state-of-the-art approaches for lip reading. Similar to wav2vec 2.0, AV-HuBERT consists of a CNN-based image encoder and a Transformer-based transformer encoder. The image encoder is constructed using a 3D convolutional front-end followed by a modified ResNet-18 block [32]. This component is responsible for extracting latent visual representations, which can be regarded as embeddings of the video frames. Then the transformer encoder operates on the video embeddings and captures contextual visual representations by considering the relationships among video frames in a large context. The frame rate of the final output $\boldsymbol{z}^V$ remains consistent with that of the input video clips, with each frame having 1,024 dimensions. In the original AV-HuBERT structure, the input video frame rate is set as 25 Hz. Hence, for ALT, we retain the same frame rate considering task similarity with ASR. However, the transcription of note events has higher resolution requirements, so we select an input frame rate of 50 Hz for our AMT systems.

4.3 Modality Feature Fusion

The modality feature fusion module ψ aims to exploit the complementary relationship and redundancy that are presented in the different modalities. Before fusing the acoustic representations $\boldsymbol{z}^A$

and the visual representations z^V, we unify the frame rates to about 50 Hz and the frame dimensions to 1,024 if necessary. Specifically, we up-sample z^V using nearest interpolation with a scale factor of 2. Afterwards, we introduce a new attention module called Residual Cross Attention (RCA) for fusing the unified features, as illustrated in Fig. 3. RCA is built upon Transformer block architecture, and its illustration can be found in Appendix D. There are M RCA blocks when considering M input modalities. Every RCA block takes input representations from all modalities. Within each block, one modality is designated as the source, providing keys and values, while the remaining modalities serve as references, providing queries. In addition to the multi-head self-attention (MHSA) [75] operation applied to the source modality, each RCA block adds extra shortcuts by performing the multi-head cross-attention (MHCA) operation between the source and each reference. The outputs of all RCA blocks are then aggregated to yield the final fused features z. RCA can be mathematically represented as follows:

Fig. 3. Illustration of modality feature fusion module.



$$z^{I_i'} = \mathrm{LN}(z^{I_i} + \mathrm{MHSA}(z^{I_i}) + \sum_{j \neq i} \mathrm{MHCA}(z^{I_i}, z^{I_j})), \quad I_i, I_j = A \text{ or } V, \tag{1}$$

$$z^{I_i''} = \mathrm{LN}(z^{I_i'} + \mathrm{FFN}(z^{I_i'})), \quad z = z^{A''} + z^{V''}, \quad I_i = A \text{ or } V, \tag{2}$$

where "LN" denotes layer normalization, and "FFN" refers to a positional-wise feed forward network.

4.4 Automatic Lyric Transcription Backend

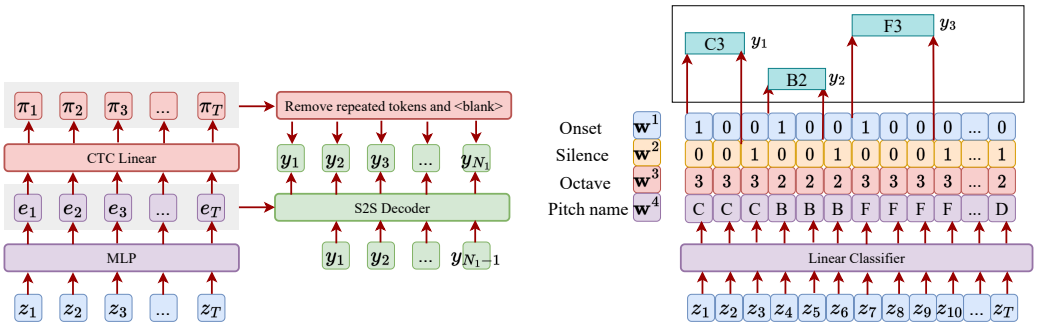
For ALT systems, we design a hybrid CTC-Attention backend to address the sequence-to-sequence (S2S) problem inspired by [78], as present in Fig. 4(a). Initially, the ground truth lyrics are converted into a sequence of tokens $\boldsymbol{y}^L = \{y_1^L, y_2^L, ..., y_{N_1}^L\}$, $y_n^L \in \mathbb{V}$ and $\mathbb{V}$ represents the character vocabulary comprising 30 tokens. The ALT backend θ^L aims to predict $p(\boldsymbol{y}^L|\boldsymbol{z})$ and consists of a two-layer MLP, a CTC linear layer, and an S2S decoder. Firstly, the MLP with 1,024 hidden neurons further encodes the fused features $\boldsymbol{z}$ into $\boldsymbol{e} \in \mathbb{R}^{T \times 1024}$, where T denotes the number of frames. Subsequently, there are two network branches to compute $p(\boldsymbol{y}^L|\boldsymbol{z})$, equivalently $p(\boldsymbol{y}^L|\boldsymbol{e})$.

The first branch is a CTC linear layer, which maps $\boldsymbol{e}$ to output probabilities for each frame $p_{\mathrm{CTC}}(\pi_t|e_t), \pi_t \in \mathbb{V} \cup \{<\text{blank}>\}, t = 1, 2, ..., T$, where <blank> is the blank token. In CTC, each frame's prediction is considered independent, leading to the probability of a sequence $\pi_{1:T}$ being $p(\pi_{1:T}|\boldsymbol{e}) = \prod_{t=1}^{T} p(\pi_t|e_t)$. The final predictions for output sequence $\boldsymbol{y}^L$ are derived from the alignment $\pi_{1:T}$ by eliminating repeated tokens and <blank> tokens. The operation is represented as $\mathcal{B}$. To supervise the CTC predictions, it is required to convert the ground truth labels into all possible CTC alignments. We use $\mathcal{B}^{-1}(\boldsymbol{y}^L)$ to represent all CTC paths mapped from $\boldsymbol{y}^L$, and then $p(\boldsymbol{y}^L|\boldsymbol{e}) = \sum_{\pi_{1:T} \in \mathcal{B}^{-1}(\boldsymbol{y}^L)} p(\pi_{1:T}|\boldsymbol{e})$. Therefore, the CTC loss is written as

$$\mathcal{L}_{\mathrm{CTC}} = -\log p_{\mathrm{CTC}}(\boldsymbol{y}^L|\boldsymbol{e}) = -\log \sum_{\pi_{1:T} \in \mathcal{B}^{-1}(\boldsymbol{y}^L)} \prod_{t=1}^{T} p(\pi_t|e_t). \tag{3}$$

The second branch is parameterized by a location-aware attention-based GRU decoder [6]. In contrast to the CTC formulation, the S2S formulation does not assume independence among predictions. Instead, it directly computes $p(\boldsymbol{y}^L|\boldsymbol{e}) = \prod_{n=1}^{N_1} p(y_n^L|y_{1:n-1}^L, \boldsymbol{e})$ following the chain rule. To

Fig. 4. (a) Hybrid CTC-Attention ALT backend; (b) AMT backend.



predict each target token y_n^L, the S2S decoder takes previously predicted tokens $y_{1:n-1}^L$ as input and utilizes a location-aware attention mechanism to derive a contextually weighted $\boldsymbol{e}$. This attention mechanism enables the model to focus on specific parts of $\boldsymbol{e}$ that are relevant for predicting the current token y_n^L. Then the S2S loss is written as

$$\mathcal{L}_{\text{S2S}} = -\log p_{\text{S2S}}(\boldsymbol{y}^L|\boldsymbol{e}) = -\log \prod_{n=1}^{N_1} p_{\text{S2S}}(y_n^L|y_{1:n-1}^L, \boldsymbol{e}). \tag{4}$$

As we employ a hybrid system, the overall loss function is a weighted sum of the two aforementioned loss terms: $\mathcal{L}^L = (1-\lambda)\mathcal{L}_{\text{S2S}} + \lambda\mathcal{L}_{\text{CTC}}$. To balance the losses, we set $\lambda = 0.2$ in this work.

During inference, in addition to the hybrid CTC-Attention structure mentioned above, we leverage a character-level LSTM language model. This allows us to predict the most likely lyrics by considering the output of three components:

$$\boldsymbol{y}^{L*} = \arg\max_{\boldsymbol{y}^L}\{\alpha \log p_{\text{CTC}}(\boldsymbol{y}^L|\boldsymbol{e}) + (1-\alpha)\log p_{\text{S2S}}(\boldsymbol{y}^L|\boldsymbol{e}) + \beta \log p_{\text{LM}}(\boldsymbol{y}^L)\}, \tag{5}$$

where α and β are hyper-parameters used to balance three log-probability terms during the beam search. We set the beam size as 512. To evaluate the performance of our ALT systems, we report word error rate (WER), which is a widely used metric for this task.

4.5 Automatic Music Transcription Backend

For AMT systems, we reformulate the sequence-to-sequence problem as a frame-level classification problem, inspired by [76]. The ground truth note events $\boldsymbol{y}^M = [(o_1, f_1, p_1), (o_2, f_2, p_2), ..., (o_{N_2}, f_{N_2}, p_{N_2})]$ are transformed into onset/silence/pitch name/octave frame-level targets, represented as $\boldsymbol{w}^1, \boldsymbol{w}^2, \boldsymbol{w}^3, \boldsymbol{w}^4$. This transformation enables us to classify each frame of the fused features $\boldsymbol{z} \in \mathbb{R}^{T\times 1024}$ into corresponding labels, as visualized in Fig. 4(b). Since directly predicting offsets is challenging, our AMT backend predicts silence instead, and the offsets $f_1, f_2, ..., f_{N_2}$ are determined as the beginnings of silence frames. We employ a pitch name and an octave to denote each note pitch.

To construct $\boldsymbol{w}^1$, frames covering the onsets $o_1, o_2, ..., o_{N_2}$ are labeled as 1, while other frames are labeled as 0. Similarly, silence frames are assigned a label of 1 in $\boldsymbol{w}^2$, while other frames are assigned a label of 0. As a result, we can use binary values to indicate the state of each frame in $\boldsymbol{w}^1, \boldsymbol{w}^2$. In conventional practice, pitch values $p_1, p_2, ..., p_{N_2}$ are represented as MIDI note numbers ranging from C2 (MIDI number 36, 65.41 Hz) to B5 (MIDI number 83, 987.77 Hz). Here "B" and "C" are the pitch names while "2" and "5" are the octaves. According to music theory, there are 12 notes

(C, $D\flat$, D, $E\flat$, E, F, $G\flat$, G, $A\flat$, A, $B\flat$, B) in each octave. We consider a pitch range from C2 to B5, resulting in a total of 4 octaves. Additionally, we introduce an octave class and a pitch name class to represent silence. Consequently, each frame of $\boldsymbol{w}^3$ has 13 possible values, and each frame of $\boldsymbol{w}^4$ has 5 possible values. During inference, the frame-level predictions are transformed back into the note events. It is noted that the transformation between note events and frame-level targets introduces temporal quantization errors. Therefore, the frame resolution significantly impacts the AMT accuracy.

The AMT backend θ^M consists of a linear layer with 20 output neurons, allocating 1, 1, 13, 5 neurons for $\boldsymbol{w}^1, \boldsymbol{w}^2, \boldsymbol{w}^3, \boldsymbol{w}^4$, separately. The output probabilities can be expressed as $p(\boldsymbol{w}^i|\boldsymbol{z}) = \prod_{t=1}^{T} p(w_t^i|z_t), i = 1, 2, 3, 4$. To train the AMT system, we combine the loss terms for the four targets:

$$\mathcal{L}^M = -\log p(\boldsymbol{y}^M|\boldsymbol{z}) = \sum_{i=1}^{4} -\log \prod_{t=1}^{T} p(w_t^i|z_t), \tag{6}$$

where we employ binary cross-entropy (BCE) loss for targets $\boldsymbol{w}^1, \boldsymbol{w}^2$ and cross-entropy (CE) loss for targets $\boldsymbol{w}^3, \boldsymbol{w}^4$. Notably, we set a positive weight of 15.0 in the BCE loss for onset prediction to amortize the effects of imbalanced distribution in $\boldsymbol{w}^1$.

In Fig. 4(b), we provide a visualization of the post-processing step to convert the predictions for $\boldsymbol{w}^1, \boldsymbol{w}^2, \boldsymbol{w}^3, \boldsymbol{w}^4$ into note events. We postpone the details to Appendix B. At a high level, we first identify pairs of onset and offset and then identify the pitch between the time. Unless otherwise stated, we maintain a fixed onset threshold of 0.4 and an offset threshold of 0.5. AMT systems are typically evaluated using F1-scores of COnPOff (Correct onset, pitch, and offset), COnP (Correct onset, pitch), and COn (Correct onset). Their definitions and implementations can be found in [57, 64]. To ensure fair comparisons with previous approaches, such as [19, 34, 42, 47, 76], we set the pitch tolerance to 50 cents, the onset tolerance to 50 ms, and the offset tolerance to the maximum of 50 ms and 0.2×note duration. Additionally, we use the F1-score of COff (Correct offset) metric to evaluate the performance of offset detection.

4.6 Training Strategy

We developed several training strategies for our multimodal ALT system and multimodal AMT system to address the following challenges. One key challenge is **adapting self-supervised learning (SSL) models from the speech domain to the singing domain**. In our approach, we utilize SSL models, namely wav2vec 2.0 [3] as audio encoder and AV-HuBERT [68] as video encoder. Originally, these models are pretrained on unlabeled speech data using SSL objectives. They are then finetuned on labeled speech data with ASR objectives. As we mentioned before, these SSL models have demonstrated the ability to generalize well to new domains, even in low-resource labeled scenarios, which can be attributed to their unsupervised learning on rich speech data. Given the similarities between speech and singing data, we hypothesize that these SSL models can also effectively generalize to our setting. For the ALT task, we initialize our audio encoder and video encoder with the SSL models pretrained and finetuned on speech data. This choice is motivated by the fact that ALT and ASR are analogous tasks with similar input-output pairs. We expect that both the pretraining and finetuning on speech data will yield benefits for the ALT task. However, the targets of the AMT task are the note events, rather than text in ALT and ASR. Hence, a question arises regarding the adaptation of the SSL models: will finetuning on speech data be advantageous for the AMT task?

Inspired by [43], we speculate that finetuning on speech data may distort the pretrained features of SSL models and bias them towards ASR, thus hindering their generalization to AMT. To address this concern, we propose a new adaptation strategy specifically tailored to the AMT task. We skip the finetuning step on speech data with ASR objectives. Instead, we conduct linear probing on the AMT backend θ^M, followed by full finetuning of the entire system. To further compare the

Algorithm 1 Adaptation of the SSL models from the speech domain to the **ALT** task

Require: SSL model ϕ^{I_i} (here I_i is either A or V) which has been pretrained under SSL objective, randomly initialized task-specific backend θ^L, learning rates γ_1, γ_2 for θ^L and ϕ^{I_i}, iterations K for full finetuning.

Finetuning ϕ^{I_i} on the ASR task.
for $k = 1$ **to** K **do**
  $\theta^L \leftarrow \theta^L - \gamma_1 \frac{\partial \mathcal{L}^L}{\partial \theta^L}$
  $\phi^{I_i} \leftarrow \phi^{I_i} - \gamma_2 \frac{\partial \mathcal{L}^L}{\partial \phi^{I_i}}$
end for

Algorithm 2 Adaptation of the SSL models from the speech domain to the **AMT** task

Require: SSL model ϕ^{I_i} (here I_i is either A or V) which has been pretrained under SSL objective, randomly initialized task-specific backend θ^M, learning rates γ_1, γ_2 for θ^M and ϕ^{I_i}, iterations K_1, K_2 for linear probing and full finetuning.

Skip finetuning ϕ^{I_i} on the ASR task.
for $k = 1$ **to** $K_1 + K_2$ **do**
  $\theta^M \leftarrow \theta^M - \gamma_1 \frac{\partial \mathcal{L}^M}{\partial \theta^M}$
  if $k \leq K_1$ **then**
    $\phi^{I_i} \leftarrow \phi^{I_i}$ ▷ Linear Probing
  else
    $\phi^{I_i} \leftarrow \phi^{I_i} - \gamma_2 \frac{\partial \mathcal{L}^M}{\partial \phi^{I_i}}$ ▷ Full Finetuning
  end if
end for

above two adaptation strategies, we outline the training pipeline for the single-modal singing ALT system and single-modal singing AMT system in Algorithm 1 and Algorithm 2, respectively (for single-modal system, the feature fusion module ψ can be omitted). Typically, we use a relatively smaller learning rate γ_2 than γ_1 to preserve pretrained features of modality-specific encoders.

Both wav2vec 2.0 and AV-HuBERT in our multimodal systems are large-scale. Consequently, to **mitigate high GPU memory demands**, we propose a two-stage training approach similar to [62]. In the first stage, we train single-modal systems independently, each of which consists of a modality-specific encoder and a task-specific backend. Then in the second stage, we freeze the modality-specific encoders and only train the feature fusion module and the task-specific backend. In this way, we eliminate the requirements to load and update all model weights simultaneously and take advantage of powerful singing representations learnt by single-modal systems. For more details, we refer readers to Appendix B.

5 EXPERIMENTS

In this section, we comprehensively evaluate our proposed systems for automatic lyric transcription (ALT) and automatic music transcription (AMT) tasks using both benchmark singing datasets and our curated multimodal singing dataset. To begin with, we conduct single-modal experiments for each task in a clean scenario (only vocal) to evaluate the efficacy of our modality-specific representation learning and assess the individual contributions of each modality to the task. Then we proceed with multimodal experiments to demonstrate the robustness of multimodal systems in the presence of sound noise contamination[2]. Finally, we conduct ablation studies to evaluate the effectiveness of our proposed methods. Additional information on the benchmark singing datasets and implementation details are presented in Appendix C and Appendix D, respectively.

5.1 Automatic Lyric Transcription Experiments

5.1.1 Audio-only ALT. To evaluate our audio encoder and ALT backend, we train and test our systems using benchmark datasets, including DSing [8, 13], DALI [54, 55], Jamendo [70], Hansen

[2]We have released our code and trained models for ALT and AMT through https://github.com/guxm2021/MM_ALT and https://github.com/guxm2021/SVT_SpeechBrain, respectively.

Table 4. WER(%) of different audio-only ALT systems on various datasets. The reported numbers with * are from [15] since the results in original papers are either absent or inferior. The best results and the second-best results are marked as **bold face** and underline, respectively.

Method	DSing valid	DSing test	DALI test	Jamendo	Hansen	Mauch
TDNN-F [8]	23.33	19.60	67.12*	76.37*	77.59*	76.98*
CTDNN-SA [13]	17.70	14.96	76.72*	66.96*	78.53*	78.50*
MSTRE-Net [15]	-	15.38	42.11	**34.94**	36.78	37.33
Genre-Informed [30]	-	56.90*	-	50.64*	39.00*	40.43*
Voice2Singing [4]	-	-	41.5	-	-	-
Pitch-Informed [16]	-	-	64.41	76.2	-	-
DE2 - segmented [14]	-	-	-	44.52	-	49.92
Ours	**13.26**	**14.56**	**32.71**	35.63	**18.55**	**29.47**

Table 5. WER(%) of our audio-only ALT systems on N20EMv1.

Train Data		WER (%) ↓	
N20EMv1	DSing	valid	test
√	×	12.74	19.68
√	√	**9.65**	**13.00**

Table 6. WER(%) of our video-only ALT system on N20EMv1 with ablated decoding configurations.

Method			WER (%) ↓	
CTC	S2S	LM	valid	test
√	×	×	63.52 (+15.61)	78.20 (+9.75)
×	√	×	55.72 (+7.81)	74.10 (+5.65)
√	√	×	55.80 (+7.89)	72.70 (+4.25)
√	√	√	**47.91**	**68.45**

[31], Mauch [49]. DSing and DALI are two large-scale datasets for model training while the other three datasets are only used for evaluation. We follow [11] to extract vocal parts.

Initially, we train and evaluate our ALT system on DSing. The audio encoder, wav2vec 2.0, was pretrained on LibriVox (LV-60k) and finetuned on LibriSpeech [63] before its fine-tuning on singing data. For inference, we train an LSTM language model using lyrics exclusively from the DSing train split, aiming for simplicity. We note that the previous methods incorporated a broader range of lyrics to train their language models (LMs). As indicated in Table 4, our system achieves state-of-the-art (SOTA) performance on DSing. Subsequently, we finetune the above system using the DALI train split and evaluate its performance on DALI test split/Jamendo/Hansen/Mauch. For inference, we train an LSTM LM on both the DSing and DALI train splits. We observe that our wav2vec 2.0-based ALT system surpasses previous approaches on DALI test split/Hansen/Mauch by large margins. While on Jamendo dataset, our system achieves comparable performance as MSTRE-Net [15].

Considering that our proposed wav2vec 2.0-based ALT system has achieved SOTA performance on the benchmark singing datasets, we adopt it to build a strong baseline for our curated N20EMv1 dataset. As for the training of LM, we use the lyrics from both the DSing train split and N20EMv1 train split. We also augment the LM using some texts from LibriSpeech. This LM will be used in all following experiments related to N20EMv1. Initially, we train the system using only the N20EMv1 train split. To further enhance the system's performance, we augment the training data by incorporating the DSing dataset. As present in Table 5, the system exhibits improved performance, which demonstrates that scaling more singing data during training enhances the system's generalization.

5.1.2 Video-only ALT. In this section, we initialize the new task of video-only ALT (or *lyric lipreading*). As this is the very first attempt, we train our video encoder and ALT backend to establish a benchmark system and assess the contribution of the video modality to the ALT task. Prior to

Fig. 5. (Left) Quantitative results (Right) qualitative results of audio-only and audio-visual ALT systems in different SNR scenarios on N20EMv1. "GT" refers to the ground truth. Correct words are marked in yellow color, insertions are highlighted using red color, and substitutions are highlighted using blue color.

finetuning our system on N20EMv1, the video encoder undergoes pretraining on LRS3 [1] and VoxCeleb2 [7], followed by finetuning on LRS3. We present the experimental results in Table 6. It is noted that lip videos inherently possess ambiguity in distinguishing between different characters, as singers may exhibit similar mouth shapes when pronouncing different characters. Therefore, the context relationships among consecutive characters are important. We validate this by conducting ablation studies on the decoding choice. Firstly, when we ablate the use of LM, we found that the performance drops a lot. Furthermore, when only using CTC backend (w/o. LM & w/o. S2S) or only S2S backend (w/o. LM & w/o. CTC) for decoding, we observe that S2S backend makes great contributions to the performance. We assume that the use of S2S backend and an external language model alleviates the ambiguity in the video modality, thus enhancing the performance of video-only ALT.

5.1.3 Multimodal ALT. To build our multimodal ALT system, we adhere to the training strategy detailed in Appendix B. For fair comparisons, we train our audio-only and audio-visual systems using the same training strategy, with the disabled modality set as zeros. Our experiments are conducted on the N20EMv1 dataset. In contrast to the previous sections, we simulate noisy environments by mixing the vocal singing with its corresponding musical accompaniment at different signal-to-noise ratio (SNR) levels, including $-10, -5, 0, 5, 10$ dB, as well as clean scenarios without accompaniment.

The quantitative results are reported in Fig. 5(Left). It is observed that the multimodal systems outperform the audio-only system by large margins, especially in challenging SNR environments. For instance, at -10 dB, the performance gap is about 30% WER. While at -5 dB, the performance gap is about 10% WER. However, with the increase of SNR, the benefits brought by additional modality gradually become limited, which is also observed in the comparison between audio-visual speech recognition and audio-only speech recognition in [46, 68]. The reason behind this is that with the absence of noise perturbations, the audio modality has sufficient information to retrieve textual information with limited aid from other modalities. Afterwards, we compute the average WER across the six scenarios as an evaluation metric for noise robustness. Consequently, on average, our audio-visual system shows a significant improvement over its audio-only counterpart, reducing WERs by 7.62% and 6.51% on the N20EMv1 valid and test splits, respectively. Therefore, we conclude that multimodal systems are more robust to noise disturbances than the single modal system.

The quantitative results are also presented in Fig. 5(Right), where we showcase the comparisons among predicted lyrics of audio-only and audio-visual transcription systems. More case studies are included in Appendix E. Firstly, we would like to highlight that although a character-level tokenizer is used in our system, the word-level errors (e.g. insertions, substitutions, deletions) are analyzed as WER is used as the evaluation metric. As shown in Fig. 5(Right), in the clean scenario, the

Table 7. COnPOff/COnP/COn F1-score (%) of different audio-only AMT systems on MIR-ST500 test set/TONAS/ISMIR2014. The best results and the second-best results are marked as **bold face** and underline.

Dataset	Metric (%) ↑	Tony [47]	HCN [19]	VOCANO [34]	EffNet [76]	JDC [42]	Ours 1	Ours 2
MIR-ST500	COnPOff	-	-	-	45.78	42.23	52.39	**52.84**
	COnP	-	-	-	66.63	69.74	**70.73**	70.00
	COn	-	-	-	75.44	76.18	**78.32**	78.05
TONAS	COnPOff	-	-	-	9.57	-	12.71	**24.08**
	COnP	-	-	-	19.65	-	25.24	**36.87**
	COn	-	-	-	42.41	-	52.77	**64.38**
ISMIR 2014	COnPOff	50	59.4	**68.38**	49.55	-	52.36	62.42
	COnP	68	-	**80.58**	63.63	-	70.38	75.91
	COn	73	79.0	84.04	79.16	-	92.77	**93.02**

predictions of the audio-only system have one insertion error and two substitution errors while the audio-visual system corrects the insertion error. Similarly, in noisy environments, the audio-visual system corrects the insertion error of "is". While it fails to transcribe the words "Wonder" and "along", it exhibits fewer character-level errors compared to the audio-only system.

5.2 Automatic Music Transcription Experiments

5.2.1 Audio-only AMT. We validate our choice of audio encoder and AMT backend on N20EMv2 and benchmark singing datasets, which include MIR-ST500 [76], TONAS [25], and ISMIR2014 [57]. MIR-ST500 is the largest singing AMT dataset with human annotations for training and in-domain (ID) evaluation. TONAS and ISMIR2014 are two small datasets only for evaluation in out-of-domain (OOD) scenarios. We follow [76] to extract the vocal parts from singing if necessary.

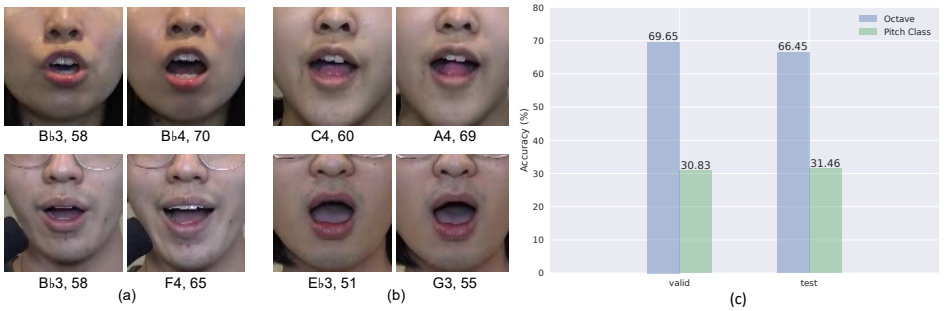
We first train an AMT system, referred to as "Ours 1" in Table 7, on the MIR-ST500 train split and evaluate its performance on MIR-ST500 test split, TONAS, and ISMIR2014. The audio encoder (wav2vec 2.0) has been pretrained on LibriVox (LV-60) without additional finetuning on a speech recognition task. For in-domain (ID) data, our system achieves a significant performance improvement over the previous SOTA. For OOD data, our system still outperforms the EffNet [76], thereby indicating the effectiveness of our model design and adaptation strategy. We note that the performance on TONAS is noticeably lower compared to the MIR-ST500 test set and ISMIR2014. This disparity can be attributed to the fact that TONAS predominantly consists of Flamenco songs, resulting in a substantial distribution shift when compared to the other datasets that primarily consist of pop songs.

Next, we proceed to train another AMT system, denoted as "Ours 2", using both the MIR-ST500 train split and the N20EMv2 train split. In Table 7, "Ours 2" not only maintains a high level of performance for in-domain (ID) data but also exhibits significantly improved generalization abilities when confronted with singing data from previously unseen domains. Specifically, "Ours 2" achieves state-of-the-art performance in terms of COnPOff/COnP/COn on TONAS and COn on ISMIR2014. Despite pitch quantization errors, the performance of "Ours 2" is comparable to the state-of-the-art [34] in terms of COnPOff/COnP on ISMIR2014. It is important to note that while the MIR-ST500/TONAS/N20EMv2 datasets are annotated in semitones, the pitch values in ISMIR2014 are annotated in cents (1 semitone = 100 cents), which puts our AMT system at a disadvantage. However, considering modern musical notation and following the approach in [76], our current design adopts a 12-tonal equal temperament system with semitonal resolution, which proves to be more practical in real-world applications. To summarize, the performance of our AMT systems

Table 8. COnPOff/COnP/COn/COff F1-score (%) of our audio-only/video-only AMT systems on N20EMv2.

Dataset	Metric (%) ↑	Audio	Video	
		Tolerance 1	Tolerance 1	Tolerance 2
N20EMv2 valid	COnPOff	61.83	4.45	9.27
	COnP	68.42	6.16	11.79
	COn	92.18	77.14	88.69
	COff	89.80	74.68	83.01
N20EMv2 test	COnPOff	73.06	6.84	15.25
	COnP	79.56	8.79	18.53
	COn	93.66	78.62	88.64
	COff	91.78	78.83	84.48

Fig. 6. Examples of (a) different mouth shapes (b) the same mouth shapes for the same pronunciation with different pitches. (c) Classification accuracy of our video-only AMT system for octaves and pitch names on N20EMv2.



("Ours 1" and "Ours 2") demonstrates that wav2vec 2.0 can learn excellent acoustic representations for the AMT task. Finally, we evaluate the performance of the system "Ours 2" on the N20EMv2 valid/test splits to establish a baseline for this new dataset. The results are presented in Table 8, where "Tolerance 1" denotes the default onset/offset/pitch tolerance.

5.2.2 Video-only AMT. In this section, we initialize the new task of video-only AMT (or *note lipreading*). To establish a baseline for N20EMv2, we train our video encoder and AMT backend. The video encoder has been pretrained on LRS3 [1] and VoxCeleb2 [7] without finetuning on a lip reading task. Experimental results in Table 8 demonstrated the effectiveness of utilizing video data for lip movements, achieving an F1-score of approximately 80% for onset and offset detection using the default tolerance. This performance is noteworthy as it competes with the performance of previous audio-only AMT systems in terms of these two metrics. Furthermore, we explore the potential of our video-only system by relaxing the tolerance settings as "Tolerance 2". Specifically, we set the onset tolerance to 100 ms, the offset tolerance to the maximum of 100 ms and 0.2×note duration, and the pitch tolerance to 100 cents. Consequently, the COn F1-score reaches about 89%, indicating that within a range of ±50 ms, our system can accurately detect almost all onsets. Regarding pitch estimation, our video-only system also provides hints for distinguishing different pitches, showcasing the power of AV-HuBERT in learning visual representations for AMT.

To interpret the high performance of our video-only system in onset/offset detection, we assume that it can detect transitions between consecutive note events by recognizing subtle changes in the mouth shape of singers. However, capturing acoustic information such as pitch solely from video is

Fig. 7. COnPOff/COnP/COn/COff F1-score (%) of our audio-only/audio-visual AMT systems on N20EMv2.

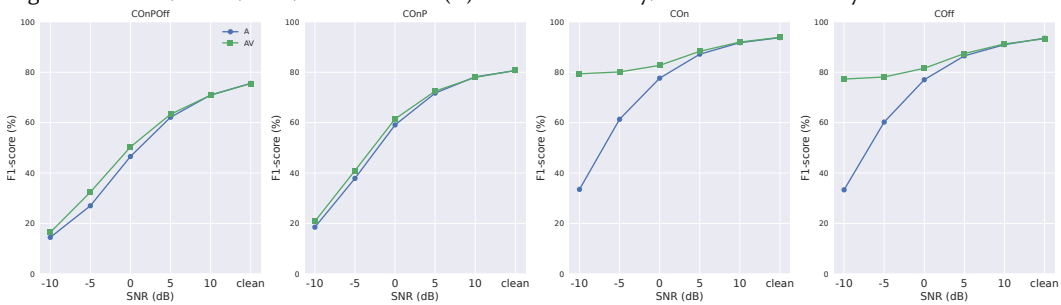
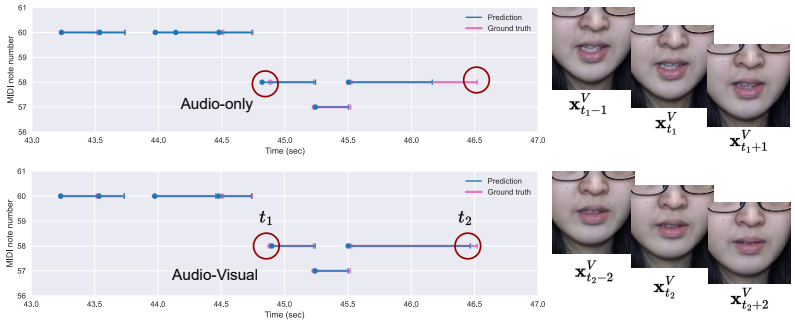


Fig. 8. Qualitative comparison of our audio-only AMT system and audio-visual counterpart.



challenging. The system demonstrates rough differentiation between mouth shapes. as indicated by the COnPOff/COnP performance. However, mouth shapes alone are insufficient for accurate pitch predictions. As depicted in Fig. 6(a) and Fig. 6(b), some cases show different mouth shapes corresponding to different pitch labels, while others exhibit identical mouth shapes but different ground truth MIDI numbers, resulting in potential failures. This issue resembles the character ambiguity problem in video-only ALT systems. We further evaluate the pitch name and octave classification accuracy of our video-only AMT system on N20EMv2, as shown in Fig. 6 (c). The accuracy of octave predictions ranges from 65% to 70%, while the accuracy of pitch name estimation is around 30%.

5.2.3 Multimodal AMT. Similar to multimodal ALT, we develop multimodal AMT systems using different combinations of modality inputs and conduct experiments in noisy environments using synchronized musical accompaniment on N20EMv2. Firstly, we compare the performance of the audio-only system with that of the audio-visual system. As illustrated in Fig. 7, the audio-visual system consistently outperforms the audio-only system across COnPOff/COnP/COn/COff metrics. The addition of the video modality yields significant improvements, particularly in low SNR scenarios. Especially for the COn and COff metrics, the audio-visual system surpasses the audio-only system by more than 40% F1-score at -10 dB. This result aligns with our assumption that the video modality excels in onset/offset detection but faces challenges in pitch estimation due to the inherent ambiguity. As the SNR increases, the performance gaps between the two AMT systems are narrowed, suggesting that the contributions of video modality are diluted in less noisy environments.

To further illustrate the advantages of incorporating the video modality, we visualize the predictions made by our audio-only system and audio-visual system in a 0 dB environment in Fig. 8. The selected song segment (42 s to 47 s) contains seven notes, and both systems accurately predict the pitch of the notes. However, the audio-only system fails to detect the onset (t_1) of the fifth note and the offset (t_2) of the seventh note, whereas these events are successfully detected by the

Table 9. Ablation study of model choices on the N20EMv2 dataset.

Models	N20EMv2 valid				N20EMv2 test			
	COnPOff	COnP	COn	COff	COnPOff	COnP	COn	COff
AV-HuBERT	44.99	51.45	85.47	83.49	57.77	64.89	88.01	86.11
wav2vec 2.0 (rand)	43.47	53.43	82.60	81.07	50.74	63.12	81.29	80.39
wav2vec 2.0	**59.54**	**64.89**	**91.45**	**90.65**	**69.77**	**76.04**	**93.02**	**91.94**

audio-visual system. To understand the decision-making process of the audio-visual system, we visualize consecutive video frames capturing the lip movements $\boldsymbol{x}^V_{t_1-1}, \boldsymbol{x}^V_{t_1}, \boldsymbol{x}^V_{t_1+1}$, corresponding to the onset of the fifth note. From $t_1 - 1$ to $t_1 + 1$, we observe a slight narrowing of the subject's mouth, indicating a transition from a higher-pitched note to a lower-pitched note. Similarly, from $t_2 - 2$ to $t_2 + 2$, we observe a gradual closure of the subject's mouth, signifying the transition from a note to silence after t_2. To conclude, the audio-visual system effectively captures the note transitions, allowing for precise onset and offset predictions.

5.3 Ablation Study

5.3.1 Ablation on Model Choices. AV-HuBERT can accept both audio and video modalities within its original structure [68]. However, our preliminary experiments indicate that AV-HuBERT struggles to learn powerful acoustic representations for our singing transcription tasks. To investigate this phenomenon further, we conducted an ablation study on N20EMv1 and N20EMv2. Specifically, we trained an audio-only ALT system based on AV-HuBERT using the N20EMv1 training split and an audio-only AMT system based on AV-HuBERT using the N20EMv2 training split. These models were subsequently evaluated on the corresponding valid/test splits, and their performance was compared to audio-only transcription systems based on wav2vec 2.0. As presented in Table 9 and Table 10, we observe that the transcription systems utilizing wav2vec 2.0 achieve significantly better performance in both ALT and AMT tasks compared to the systems relying on AV-HuBERT. Moreover, we validate the effectiveness of using the pre-trained model on both ALT and AMT tasks. Specifically, we re-train the wav2vec 2.0-based transcription systems following the same procedure except that we randomly initialize the model weights of wav2vec 2.0. As shown in Table 10, the resulting ALT hardly recognizes lyrics from audio as it achieves almost 100% WER. While for AMT, the performance also deteriorates significantly.

Table 10. Ablation study of model choices on N20EMv1.

Models	WER (%) ↓	
	valid	test
AV-HuBERT	31.21	38.64
wav2vec 2.0 (rand)	99.91	99.37
wav2vec 2.0	**12.74**	**19.68**

5.3.2 Ablation on Adaptation Strategy. To adapt self-supervised-learning (SSL) models from the speech domain to the AMT task, we propose Algorithm 2 considering both the domain shift and task difference. Specifically, we first skip the finetuning of SSL models on the ASR task and then directly finetune them on the AMT task in a linear probing and full finetuning manner. We conduct an ablation study on audio-only AMT systems to validate the effectiveness of this adaptation strategy. In addition to our proposed design, we create two variants for comparison. For "variant 1", we retain finetuning of SSL models on the ASR task, followed by full-finetuning on singing data. Conversely, for "variant 2", we skip the finetuning on the ASR task and directly proceed with full-finetuning on singing data. All transcription systems are trained using both the MIR-ST500 train split and N20EMv2 train split. The evaluation results are presented in Table 11 and Table 12. We note that our

Table 11. Comparison among different adaptation strategies for AMT on the N20EMv2 dataset.

Methods	N20EMv2 valid				N20EMv2 test			
	COnPOff	COnP	COn	COff	COnPOff	COnP	COn	COff
variant 1	56.89	63.39	91.50	89.09	70.16	77.25	93.08	91.22
variant 2	59.24	65.99	91.17	89.62	69.90	76.84	92.71	91.21
Ours	**61.83**	**68.42**	**92.18**	**89.80**	**73.06**	**79.56**	**93.66**	**91.78**

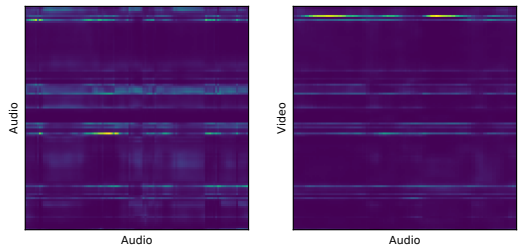
Table 12. Comparison among different adaptation strategies for AMT on MIR-ST500 test/TONAS/ISMIR2014.

Methods	MIR-ST500 test			TONAS			ISMIR2014		
	COnPOff	COnP	COn	COnPOff	COnP	COn	COnPOff	COnP	COn
variant 1	50.78	68.75	77.23	21.63	34.60	63.01	61.03	74.25	91.84
variant 2	51.43	68.89	77.98	22.55	36.72	63.48	57.97	72.21	92.16
Ours	**52.84**	**70.00**	**78.05**	**24.08**	**36.87**	**64.38**	**62.42**	**75.91**	**93.02**

Table 13. WER (%) of our multimodal ALT system with ablated fusion module in -10 dB SNR scenario. CA: Cross-Attention, SA: Self-Attention.

Fusion		WER (%) ↓	
CA	SA	valid	test
×	√	41.62 (+1.90)	65.15 (+2.17)
√	×	42.35 (+2.63)	63.99 (+1.01)
√	√	**39.72**	**62.98**

Fig. 9. Visualization of self-attention and cross-attention weights in RCA module of audio modality. We use brighter colors to highlight stronger attention.



adaptation strategy consistently outperforms the two variants across all evaluation sets, including ID data and OOD data, in terms of all metrics.

5.3.3 Ablation on Feature Fusion. We evaluate the effectiveness of our proposed RCA module for multi-modal ALT task. To highlight the differences, we evaluate the ALT performance with different feature fusion modules at −10 dB SNR scenario. As present in Table 13, we find that the absence of cross-attention shortcuts leads to an increase of 1.90% and 2.17% WER on the valid and test splits. While the absence of self-attention mechanism causes an increase of 2.63% and 1.01% WER, respectively. These results indicate that RCA contributes to improved feature fusion.

To further investigate the effectiveness of the RCA mechanism, we visualize the attention maps within the RCA module when the audio serves as the source modality. We include audio-audio self-attention and audio-video cross-attention, as shown in Fig. 9. We observe that both attention maps exhibit common attention patterns. Moreover, the cross-attention can extract additional relationships between frames that are not captured by the self-attention alone. This demonstrates that the RCA enhances feature fusion by incorporating complementary information from the other modality.

6 DISCUSSIONS AND FUTURE WORK

In this work, we consider multimodal singing automatic lyric transcription (ALT) and multimodal singing automatic music transcription as two distinct tasks, following previous literature. Our current system can be trained to seamlessly transcribe both lyrics and musical note events, resulting

in a multi-task system. However, the data of different modalities are highly imbalanced. Treating the training on a single dataset with predefined modality combinations and predefined learning objectives as an individual learning task, the challenge lies in striking a balance between different learning tasks to train a multitask and multimodal system that achieves high performance and high robustness simultaneously. This poses an open problem that requires further investigation. Furthermore, while our ALT system is designed for a single language, the AMT system is language-agnostic. Thus, combining these two systems would necessitate considering a multilingual setting. We leave this direction as a topic for future research.

7 CONCLUSION

In this work, we proposed a unified multimodal framework for transcribing lyrics and note events from singing voices. To develop our systems, we carefully curated the multimodal singing automatic lyric transcription (ALT) dataset N20EMv1 and the multimodal singing automatic music transcription (AMT) dataset N20EMv2. Then, we adapted self-supervised learning (SSL) models from the speech domain into the singing domain as acoustic encoders, yielding state-of-the-art performance. Additionally, we adapted SSL models initially used for lipreading tasks to serve as visual encoders, allowing us to initialize two novel tasks: lyric lipreading and note lipreading. Our results demonstrated that video modality can significantly contribute to both ALT and AMT tasks, despite the inherent challenges posed by ambiguity. Finally, we introduced residual cross attention (RCA), a new feature fusion method, to fuse features from different modalities to obtain the ultimate transcription. Through our comprehensive experiments, we unveiled the advantages of incorporating additional modalities, which led to improved transcription performance and enhanced robustness against sound contamination and perturbations.

ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for their valuable suggestions. This project is funded by a research grant MOE-T2EP20120-0012 from the Ministry of Education in Singapore.

REFERENCES

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018. LRS3-TED: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496* (2018).

[2] Víctor Arroyo, Jose J Valero-Mas, Jorge Calvo-Zaragoza, and Antonio Pertusa. 2022. Neural audio-to-score music transcription for unconstrained polyphony using compact output representations. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4603–4607.

[3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems* 33 (2020), 12449–12460.

[4] Sakya Basak, Shrutina Agarwal, Sriram Ganapathy, and Naoya Takahashi. 2021. End-to-end lyrics Recognition with Voice to Singing Style Transfer. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 266–270.

[5] Ke Chen, Shuai Yu, Cheng-i Wang, Wei Li, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2022. Tonet: Tone-octave network for singing melody extraction from polyphonic music. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 621–625.

[6] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. *Advances in Neural Information Processing Systems* 28 (2015).

[7] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. VoxCeleb2: Deep Speaker Recognition. *Proceedings of Interspeech* (2018), 1086–1090.

[8] Gerardo Roa Dabike and Jon Barker. 2019. Automatic Lyric Transcription from Karaoke Vocal Tracks: Resources and a Baseline System. In *Proceedings of Interspeech*. 579–583.

[9] Rebecca Davies, Evan Kidd, and Karen Lander. 2009. Investigating the psycholinguistic correlates of speechreading in preschool age children. *International Journal of Language & Communication Disorders* 44, 2 (2009), 164–174.

[10] Alain De Cheveigné and Hideki Kawahara. 2002. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America* 111, 4 (2002), 1917–1930.
[11] Alexandre Défossez. 2021. Hybrid Spectrogram and Waveform Source Separation. In *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*.
[12] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. 2019. Music source separation in the waveform domain. *arXiv preprint arXiv:1911.13254* (2019).
[13] Emir Demirel, Sven Ahlbäck, and Simon Dixon. 2020. Automatic lyrics transcription using dilated convolutional neural networks with self-attention. In *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
[14] Emir Demirel, Sven Ahlbäck, and Simon Dixon. 2021. Low Resource Audio-to-Lyrics Alignment From Polyphonic Music Recordings. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 586–590.
[15] Emir Demirel, Sven Ahlbäck, and Simon Dixon. 2021. MSTRE-Net: Multistreaming Acoustic Modeling for Automatic Lyrics Transcription. In *Proceedings of the 22rd International Society for Music Information Retrieval Conference (ISMIR)*. 151–158.
[16] Tengyu Deng, Eita Nakamura, and Kazuyoshi Yoshii. 2022. End-to-end lyrics transcription informed by pitch and onset estimation. In *The 23rd International Society for Music Information Retrieval Conference (ISMIR)*.
[17] Zhiyan Duan, Haotian Fang, Bo Li, Khe Chai Sim, and Ye Wang. 2013. The NUS sung and spoken lyrics corpus: A quantitative comparison of singing and speech. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 1–9.
[18] Georgi Dzhambazov et al. 2017. *Knowledge-based probabilistic modeling for tracking lyrics in music audio signals*. Ph. D. Dissertation. Universitat Pompeu Fabra.
[19] Zih-Sing Fu and Li Su. 2019. Hierarchical classification networks for singing voice segmentation and transcription. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*. 900–907.
[20] Hiromasa Fujihara, Masataka Goto, and Jun Ogata. 2008. Hyperlinking Lyrics: A Method for Creating Hyperlinks Between Phrases in Song Lyrics.. In *Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR)*. 281–286.
[21] Xiaoxue Gao, Chitralekha Gupta, and Haizhou Li. 2022. Automatic lyrics transcription of polyphonic music with lyrics-chord multi-task learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2022), 2280–2294.
[22] Xiaoxue Gao, Chitralekha Gupta, and Haizhou Li. 2022. Genre-conditioned acoustic models for automatic lyrics transcription of polyphonic music. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 791–795.
[23] Xiaoxue Gao, Xianghu Yue, and Haizhou Li. 2023. Self-Transriber: Few-Shot Lyrics Transcription With Self-Training. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.
[24] Beat Gfeller, Christian Frank, Dominik Roblek, Matt Sharifi, Marco Tagliasacchi, and Mihajlo Velimirović. 2020. SPICE: Self-supervised pitch estimation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 1118–1128.
[25] Emilia Gómez and Jordi Bonada. 2013. Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal* 37, 2 (2013), 73–90.
[26] Sira Gonzalez and Mike Brookes. 2014. PEFAC-a pitch estimation algorithm robust to high levels of noise. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22, 2 (2014), 518–530.
[27] Xiangming Gu, Longshen Ou, Danielle Ong, and Ye Wang. 2022. Mm-alt: A multimodal automatic lyric transcription system. In *Proceedings of the 30th ACM International Conference on Multimedia*. 3328–3337.
[28] Xiangming Gu, Wei Zeng, and Ye Wang. 2023. Elucidate Gender Fairness in Singing Voice Transcription. In *Proceedings of the 31st ACM International Conference on Multimedia*. 8760–8769.
[29] Chitralekha Gupta, Emre Yılmaz, and Haizhou Li. 2020. Automatic lyrics alignment and transcription in polyphonic music: Does background music help?. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 496–500.
[30] Chitralekha Gupta, Emre Yılmaz, and Haizhou Li. 2020. Automatic lyrics alignment and transcription in polyphonic music: Does background music help?. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 496–500.
[31] Jens Kofod Hansen and IDMT Fraunhofer. 2012. Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients. In *9th Sound and Music Computing Conference (SMC)*. 494–499.
[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
[33] Toru Hosoya, Motoyuki Suzuki, Akinori Ito, Shozo Makino, Lloyd A Smith, David Bainbridge, and Ian H Witten. 2005. Lyrics Recognition from a Singing Voice Based on Finite State Automaton for Music Information Retrieval.. In *ISMIR*. 532–535.

[34] Jui-Yang Hsu and Li Su. 2021. VOCANO: A note transcription framework for singing voice in polyphonic music.. In *Proceedings of the 22rd International Society for Music Information Retrieval Conference (ISMIR)*. 293–300.
[35] Xinxin Hu, Kailun Yang, Lei Fei, and Kaiwei Wang. 2019. Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation. In *IEEE International Conference on Image Processing (ICIP)*. IEEE, 1440–1444.
[36] Feng Huang and Tan Lee. 2012. Pitch estimation in noisy speech using accumulated peak spectrum and sparse estimation technique. *IEEE transactions on audio, speech, and language processing* 21, 1 (2012), 99–109.
[37] Rongjie Huang, Chenye Cui, Feiyang Chen, Yi Ren, Jinglin Liu, Zhou Zhao, Baoxing Huai, and Zhefeng Wang. 2022. Singgan: Generative adversarial network for high-fidelity singing voice generation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 2525–2535.
[38] Yu Huang, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. 2021. What Makes Multi-modal Learning Better than Single (Provably). *Advances in Neural Information Processing Systems* 34 (2021).
[39] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. 2018. Crepe: A convolutional representation for pitch estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 161–165.
[40] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[41] Anna M. Kruspe. 2015. Training Phoneme Models for Singing with "Songified" Speech Data. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*. 336–342.
[42] Sangeun Kum, Jongpil Lee, Keunhyoung Luke Kim, Taehyoung Kim, and Juhan Nam. 2022. Pseudo-Label Transfer from Frame-Level to Note-Level in a Teacher-Student Framework for Singing Transcription from Polyphonic Music. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 796–800.
[43] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *International Conference on Learning Representations*.
[44] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, et al. 2023. MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training. *arXiv preprint arXiv:2306.00107* (2023).
[45] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. 2022. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11020–11028.
[46] Pingchuan Ma, Stavros Petridis, and Maja Pantic. 2021. End-to-end audio-visual speech recognition with conformers. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7613–7617.
[47] Matthias Mauch, Chris Cannam, Rachel Bittner, George Fazekas, Justin Salamon, Jiajie Dai, Juan Bello, and Simon Dixon. 2015. Computer-aided melody note transcription using the Tony software: Accuracy and efficiency. (2015).
[48] Matthias Mauch and Simon Dixon. 2014. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 659–663.
[49] Matthias Mauch, Hiromasa Fujihara, and Masataka Goto. 2011. Integrating additional chord information into HMM-based lyrics-to-audio alignment. *IEEE Transactions on Audio, Speech, and Language Processing* 20, 1 (2011), 200–210.
[50] Harry McGurk and John MacDonald. 1976. Hearing lips and seeing voices. *Nature* 264, 5588 (1976), 746–748.
[51] Matt McVicar, Daniel PW Ellis, and Masataka Goto. 2014. Leveraging repetition for improved automatic lyric transcription in popular music. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3117–3121.
[52] Andrew N Meltzoff and M Keith Moore. 1977. Imitation of facial and manual gestures by human neonates. *Science* 198, 4312 (1977), 75–78.
[53] Annamaria Mesaros and Tuomas Virtanen. 2010. Recognition of phonemes and words in singing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2146–2149.
[54] Gabriel Meseguer-Brocal, Alice Cohen-Hadria, and Geoffroy Peeters. 2019. Dali: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm. *arXiv preprint arXiv:1906.10606* (2019).
[55] Gabriel Meseguer-Brocal, Alice Cohen-Hadria, and Geoffroy Peeters. 2020. Creating DALI, a large dataset of synchronized audio, lyrics, and notes. *Transactions of the International Society for Music Information Retrieval* 3, 1 (2020).
[56] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 8 (2018), 1979–1993.
[57] Emilio Molina, Ana Maria Barbancho-Perez, Lorenzo Jose Tardon-Garcia, and Isabel Barbancho-Perez. 2014. Evaluation framework for automatic singing transcription. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*.
[58] Meinard Müller, Emilia Gómez, and Yi-Hsun Yang. 2019. Computational methods for melody and voice processing in music recordings (Dagstuhl seminar 19052). In *Dagstuhl Reports*, Vol. 9. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[59] Dania Murad, Riwu Wang, Douglas Turnbull, and Ye Wang. 2018. SLIONS: A karaoke application to enhance foreign language learning. In *Proceedings of the 26th ACM International Conference on Multimedia*. 1679–1687.
[60] Ryo Nishikimi, Eita Nakamura, Masataka Goto, and Kazuyoshi Yoshii. 2021. Audio-to-score singing transcription based on a CRNN-HSMM hybrid model. *APSIPA Transactions on Signal and Information Processing* 10 (2021), e7.
[61] Longshen Ou, Xiangming Gu, and Ye Wang. 2022. Transfer learning of wav2vec 2.0 for automatic lyric transcription. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*.
[62] Xichen Pan, Peiyu Chen, Yichen Gong, Helong Zhou, Xinbing Wang, and Zhouhan Lin. 2022. Leveraging Unimodal Self-Supervised Learning for Multimodal Audio-Visual Speech Recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 4491–4503.
[63] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5206–5210.
[64] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel. 2014. mir_eval: A transparent implementation of common MIR metrics. In *In Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*.
[65] Miguel A Román, Antonio Pertusa, and Jorge Calvo-Zaragoza. 2018. An End-to-end Framework for Audio-to-Score Music Transcription on Monophonic Excerpts.. In *ISMIR*. 34–41.
[66] Daniel Seichter, Mona Köhler, Benjamin Lewandowski, Tim Wengefeld, and Horst-Michael Gross. 2021. Efficient rgb-d semantic segmentation for indoor scene analysis. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 13525–13531.
[67] Bidisha Sharma and Ye Wang. 2019. Automatic evaluation of song intelligibility using singing adapted STOI and vocal-specific features. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2019), 319–331.
[68] Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. 2022. Learning audio-visual speech representation by masked multimodal cluster prediction. In *International Conference on Learning Representations*.
[69] Bowen Shi, Wei-Ning Hsu, and Abdelrahman Mohamed. 2022. Robust Self-Supervised Audio-Visual Speech Recognition. *arXiv preprint arXiv:2201.01763* (2022).
[70] Daniel Stoller, Simon Durand, and Sebastian Ewert. 2019. End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 181–185.
[71] Li Su. 2018. Vocal melody extraction using patch-based CNN. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 371–375.
[72] Cynthia Tam, Heidi Schwellnus, Ceilidh Eaton, Yani Hamdani, Andrea Lamont, and Tom Chau. 2007. Movement-to-music computer technology: a developmental play experience for children with severe physical disabilities. *Occupational therapy international* 14, 2 (2007), 99–112.
[73] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 6105–6114.
[74] Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li. 2021. Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3927–3935.
[75] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing systems* 30 (2017).
[76] Jun-You Wang and Jyh-Shing Roger Jang. 2021. On the preparation and validation of a large-scale dataset of singing transcription. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 276–280.
[77] Ye Wang and Bingjun Zhang. 2008. Application-specific music transcription for tutoring. *IEEE MultiMedia* 15, 3 (2008), 70–74.
[78] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing* 11, 8 (2017), 1240–1253.
[79] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10687–10698.
[80] Weiming Yang, Xianke Wang, Bowen Tian, Wei Xu, and Wenqing Cheng. 2022. A Multi-stage Automatic Evaluation System for Sight-singing. *IEEE Transactions on Multimedia* (2022).
[81] Chen Zhang, Jiaxing Yu, LuChin Chang, Xu Tan, Jiawei Chen, Tao Qin, and Kejun Zhang. 2022. PDAugment: Data Augmentation by Pitch and Duration Adjustments for Automatic Lyrics Transcription. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*.

A ADDITIONAL DETAILS ON THE N20EMV1 AND N20EMV2 DATASETS

During the annotation of N20EMv1, we mentioned in the main paper that the pronunciation errors are annotated. Although the annotations are not used in this work, we think they will prove valuable for future research endeavors, particularly in the area of singing pronunciation evaluation. As depicted in Table 14, there are four types of pronunciation errors, including mispronunciation, substitution, insertion, and deletion. We use distinct brackets to mark them.

Table 14. Annotation for different types of pronunciation errors.

Error	Example ("correct" - "wrong")	Annotation
Mispronunciation	“little” - “laytle”	/little/
Substitution	“the” - “a”	a [the]
Insertion	“” - “a”	{a}
Deletion	“and” - “”	(and)

During the annotation process of N20EMv2, we established several rules to ensure consistency between the two annotators. Firstly, notes are segmented based on both pitch and syllables. Each syllable was considered as a separate note, while specific guidelines for labeling onset/offset/pitch are outlined below:

- **Pitch**: Notes with a duration longer than a semiquaver are treated as individual notes, as perceived by the annotators. However, ornaments such as pitch bending at the beginning of a note or vibratos were not considered independent notes. The pitch of each note was annotated with semitonal precision.
- **Onset**: The onset time of each note was identified as the start of the vowel in each syllable. In cases where a syllable began with a non-vowel sonorant, the annotators deliberately determined when the vowel was pronounced and marked it as the onset. For instance, if the lyrics of a note were “last” [la:st], the onset would be placed at the beginning of “a” [a:] rather than “l” [l].
- **Offset**: The offset time of each note was determined based on the absence of significant patterns in the audio spectrogram or when the next note commenced.

Following the initial annotation, two experts carefully scrutinized each other’s labeling results to reach a final agreement.

B ADDITIONAL DETAILS ON METHODOLOGY

Post-processing for AMT systems. During the inference of AMT systems, we first obtain frame-level predictions $\boldsymbol{w}^{1*}, \boldsymbol{w}^{2*}, \boldsymbol{w}^{3*}, \boldsymbol{w}^{4*}$. These predictions are then transformed back into note-level predictions $\boldsymbol{y}^{M*}$ through a post-processing step. Firstly, given the pitch name predictions $\boldsymbol{w}^{3*}$ and octave predictions $\boldsymbol{w}^{4*}$, we determine the predicted MIDI number (or silence) for each frame. Next, we iterate all frames to identify all note events. For each note event, we first determine its onset. If the onset prediction w_t^{1*} surpasses 0.4 (onset threshold) and w_t^{1*} is a local maximum, we consider $(t-1)\frac{L}{T}$ as the onset time. Here L represents the duration of input and $\frac{L}{T}$ corresponds to the frame length or frame resolution. Then the offset time $(t'-1)\frac{L}{T}$ is determined under the condition that $t' = \arg\min(w_{t'}^{2*} > 0.5)$ and $t' > t$. The MIDI number assigned to this note is determined as the mode of the predicted MIDI numbers between the t-th and t'-the frames.

Training multimodal singing transcription systems. The complete algorithm for training multi-modal ALT system is presented in Algorithm 3. The training pipeline for the multimodal AMT

system can be similarly derived. Specifically, in stage I, we train single-modal systems using Algorithm 1 for ALT and Algorithm 2 for AMT. This training procedure results in two task-specific backends θ_A^L and θ_V^L (or θ_A^M and θ_V^M), corresponding to audio and video modalities. We observe that initializing the task-specific backend for the multimodal system using the task-specific backend from the best-performing modality (normally the audio modality) can expedite convergence and lead to empirical performance improvements. Then in stage II, we freeze the parameter updates for the modality-specific encoders and only train the feature fusion module and task-specific backend.

Algorithm 3 Training pipeline for multimodal singing ALT system

Require: Modality-specific encoders ϕ^A, ϕ^V, modality feature fusion module ψ, task-specific backend θ^L, learning rates γ_1, γ_2 for θ^L and ϕ^A / ϕ^V in stage I, learning rate γ_3 for ψ and θ^L in stage II, iterations K_1 and K_2 for stage I and stage II.

Train $\{\phi^A, \theta_A^L\}$ and $\{\phi^V, \theta_V^L\}$ via Algorithm 1 using hyperparameters γ_1, γ_2, K_1. ▷ Stage I

Evaluate $\{\phi^A, \theta_A^L\}$ and $\{\phi^V, \theta_V^L\}$ to decide the best-performing modality for ALT task.
Initialize $\theta^L \leftarrow \theta_A^L$
Freeze the weights of ϕ^A, ϕ^V.
for $k = 1$ **to** K_2 **do** ▷ Stage II
  $\theta^L \leftarrow \theta^L - \gamma_3 \frac{\partial \mathcal{L}^L}{\partial \theta^L}$
  $\psi \leftarrow \psi - \gamma_3 \frac{\partial \mathcal{L}^L}{\partial \psi}$
end for

Training on samples with uneven duration. To address the issue of uneven duration in singing samples and enable batch training, we employ a padding approach. Both the training samples and their corresponding frame-level targets are padded with zeros to match the duration of the longest sample in a batch. Suppose the numbers of frames in a batch are $T^1, ..., T^b, ..., T^B$, where B is the batch size, then the frame number of the padded batch is denoted as $T_{\max} = \max\{T^1, ..., T^b, ..., T^B\}$. We then construct a mask $\boldsymbol{M} \in \mathbb{R}^{B \times T_{\max}}$ for each batch. Each element $M_t^b = 1$ if $t \leq T^b$. Otherwise, $M_t^b = 0$. Suppose the loss for each frame in a single sample is l_t^b, then the masked loss is computed as:

$$\mathcal{L} = \frac{1}{B} \sum_{b=1}^{B} \sum_{t=1}^{T_{\max}} M_t^b l_t^b \quad \text{or} \quad \mathcal{L} = \frac{1}{\sum_{b=1}^{B} \sum_{t=1}^{T_{\max}} M_t^b} \sum_{b=1}^{B} \sum_{t=1}^{T_{\max}} M_t^b l_t^b, \tag{7}$$

where the choice between the two forms depends on whether the loss is averaged over the frame axis. To reduce the padding ratio, we sort all the samples in ascending order based on their duration during training.

Training on song-level data. The experiments of AMT are conducted at the song level. However, loading an entire song is highly demanding for GPU memories, given the typical duration of songs ranging from 3 to 5 minutes. To address this, we divide each song into segments, each comprising 5 seconds, except for the last segment, which may range from 2.5 to 7.5 seconds. During the evaluation, the predictions for all segments are combined to compute song-level metrics.

C BENCHMARK SINGING DATASETS

ALT datasets. The DSing dataset [8, 13] provides official train/valid/test splits. Specifically, the train split has three subsets, namely DSing1, DSing3, and DSing30, each varying in size. Throughout our work, we utilize the DSing30 subset as the train split. We divide DALI v2 [55] into train and

valid splits, and regard a subset of DALI v1 as the test split following [15]. These two datasets are the most large-scale ones for ALT. Jamendo, Hansen, Mauch are three small datasets only for evaluation. Among them, Jamendo comprises English songs with different genres while the other two datasets consist of Western pop songs. The statistics of these datasets are shown in Table 15.

AMT datasets. The MIR-ST500 [76] dataset is the largest singing AMT dataset with human annotations. It comprises 500 Chinese pop songs, amounting to a total duration of about 30 hours. The dataset is divided into a train split with 400 songs and a test split of 100 songs. TONAS [25] and ISMIR2014 [57] are two small datasets that we used only for evaluating out-of-domain (OOD) scenarios, due to their distinct styles, languages, and annotation processes. TONAS has 72 Flamenco songs while ISMIR2014 encompasses 14 songs sung by children, 13 by male adults, and 11 by female adults. The statistics of these datasets are shown in Table 16.

Table 15. Statistics of benchmark ALT datasets.

Data	Split	Num. of Utter.	Duration
DSing	train	81,092	149.1 h
	valid	482	41 min
	test	480	48 min
DALI	train	268,392	183.8 h
	valid	1,313	55 min
	test	12,471	9 h
Jamendo	-	921	49 min
Hansen	-	634	34 min
Mauch	-	878	54 min

Table 16. Statistics of benchmark AMT datasets.

Data	Split	Num. of Songs	Duration
MIR-ST500	train	400	27.6 h
	test	100	6.8 h
TONAS	-	72	36 min
ISMIR2014	-	38	19 min

D IMPLEMENTATION DETAILS

D.1 Automatic Lyric Transcription Experiments

In Section 5.1.1 Audio-only ALT, we first train our ALT system on the DSing dataset. We follow Algorithm 1 and employ a learning rate of $\gamma_1 = 3 \times 10^{-4}$ for the ALT backend and a learning rate of $\gamma_2 = 1 \times 10^{-5}$ for the acoustic encoder. The model is trained using Adam optimizer [40] for 10 epochs with a batch size of 4. During the inference, we train an RNN language model (RNNLM), which has the embedding size of 128, 2 RNN layers with 2,048 RNN neurons, as well as 1 DNN block with 512 DNN neurons. This RNNLM is trained on the lyrics from DSing train split. We select $\alpha = 0.4$ for CTC decoding weight and $\beta = 0.4$ for LM decoding weight when evaluating the system on DSing valid/test splits. We mark this system as "System 1".

When we train our ALT system only on the N20EMv1 dataset, we keep the same training configuration except that the RNNLM is trained on the lyrics from N20EMv1/DSing/LibriSpeech train splits. We mark this ALT system as "System 2". In the main paper, we further augment the training data by incorporating the DSing dataset. To achieve this, we directly fine-tune "System 1" on the N20EMv1 dataset, which is called "System 3". This ALT system is further used in our multimodal experiments.

To evaluate our proposed framework on the DALI/Jamendo/Mauch/Hansen datasets, we fine-tune "System 1" on the DALI train split for 2 epochs. We enhance the ability of RNNLM by increasing the number of RNN layers to 3, the number of DNN blocks to 2, and DNN neurons to 1,024. We

employ $\alpha = 0.3$ for CTC decoding weight and $\beta = 0.2$ for LM decoding weight when evaluating the system on the above datasets. This system is labeled as "System 4".

In Section 5.1.2 Video-only ALT, we train our transcription system based on only video modality. We follow the same configurations as "System 2" for implementation. In Section 5.1.3 Multimodal ALT, we train our transcription system following Algorithm 3 for 10 training epochs with a learning rate of 1×10^{-4}. Since we freeze the parameters of our acoustic encoder and visual encoder, we increase the batch size to 24.

D.2 Automatic Music Transcription Experiments

In Section 5.2.1 Audio-only AMT, we first train our AMT system on the MIR-ST500 dataset. We follow Algorithm 2 and employ a learning rate of $\gamma_1 = 3 \times 10^{-4}$ for the AMT classifier and a learning rate of $\gamma_2 = 5 \times 10^{-5}$ for the acoustic encoder. Then we train the system for 2 epochs under the linear probing stage and 8 epochs under the full fine-tuning stage. The batch size is set as 8. The resulting system is marked as "Ours 1" in the main paper. We follow the same training configurations to train "Ours 2" and our video-only AMT system with only modification on training data.

In Section 5.2.2 Video-only AMT, we train our video-only AMT system on N20EMv2 using the same training pipeline as our audio-only AMT system except that the input is the video modality. In Section 5.2.3 Multimodal AMT, we train our transcription system following Algorithm 3 for 10 training epochs with a learning rate of 3×10^{-4}.

E MORE RESULTS

E.1 Adaptation of Models from the Music Domain

In the main paper, we have highlighted that singing and speech share similarities, which is our main motivation to adapt wav2vec 2.0 and AV-HuBERT from the speech domain into the singing domain. It is worth of mentioning that singing and music are also similar, especially in terms of musical perspective in the audio signal. Therefore, we replace wav2vec 2.0 with MERT, a recent large-scale self-supervised-learning model from the music domain [44], in our transcription systems to evaluate its ability to extract linguistic/musical information for ALT and AMT tasks. MERT shares a similar model architecture as wav2vec 2.0 [3] but with different training paradigms and data resources. We consider four model variants of MERT[3] and compare the best performance we can achieve to wav2vec 2.0. To facilitate fair comparison, we only make minimal modifications on training configurations to meet the specifications of input sampling rate and output frame rate.

Table 17. Comparison between wav2vec 2.0 and MERT for the ALT task on N20EMv1.

Models	WER↓	
	valid	test
MERT	57.48	74.55
wav2vec 2.0	**12.74**	**19.68**

For ALT, we train and evaluate audio-only systems on the N20EMv1 dataset. As shown in Table 17, we observe that when replacing wav2vec 2.0 with MERT in our ALT framework, its performance deteriorates drastically. This is expected since MERT was trained on music data and prone to focus on the musical part of audio input instead of textual information.

For AMT, we train audio-only systems on the combination of MIR-ST500 and N20EMv2 train splits. Afterwards, we test their performance on various singing datasets. Afterwards, we test their performance on various singing datasets. We found that in most cases, the performance of wav2vec 2.0 exceeds that of MERT, especially on the MIR-ST500, TONAS and ISMIR2014 datasets, as present in Table 18. While on the N20EMv2 dataset, it seems that MERT performs better on pitch estimation while wav2vec 2.0 performs better on onset and offset detection, as shown in Table 19. Therefore,

[3]MERT-v1-330M/MERT-v1-95M/MERT-v0-public/MERT-v0 in https://huggingface.co/m-a-p.

Table 18. Comparison between wav2vec 2.0 and MERT for AMT on the MIR-ST500/TONAS/ISMIR2014 datasets.

Models	MIR-ST500 test			TONAS			ISMIR2014		
	COnPOff	COnP	COn	COnPOff	COnP	COn	COnPOff	COnP	COn
MERT	50.76	69.62	76.61	19.62	30.30	61.08	**62.59**	74.91	90.91
w2v 2.0	**52.84**	**70.00**	**78.05**	**24.08**	**36.87**	**64.38**	62.42	**75.91**	**93.02**

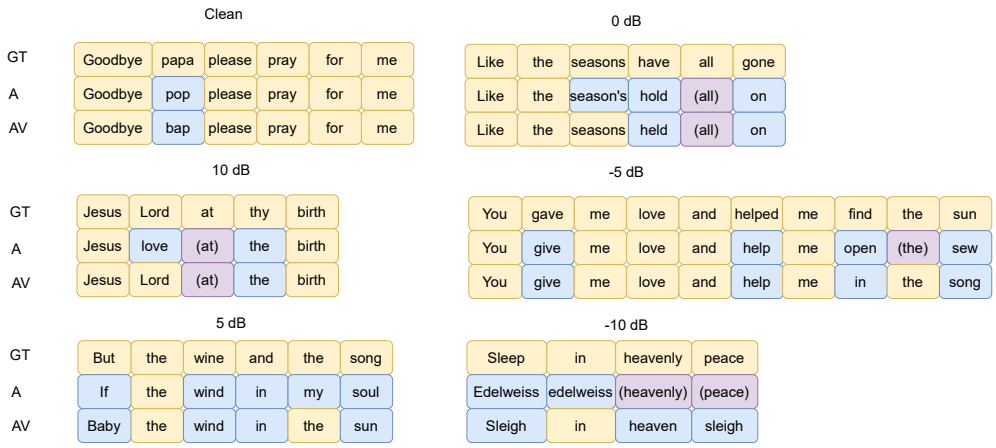
Table 19. Comparison between wav2vec 2.0 and MERT for AMT on the N20EMv2 dataset.

Models	N20EMv2 valid				N20EMv2 test			
	COnPOff	COnP	COn	COff	COnPOff	COnP	COn	COff
MERT	**63.08**	**71.18**	89.29	87.89	72.06	**79.89**	91.78	90.50
w2v 2.0	61.83	68.42	**92.18**	**89.80**	**73.06**	79.56	**93.66**	**91.78**

we conclude that wav2vec 2.0 has superiority over MERT. However, we think further explorations on the adaptation of MERT will be beneficial to this task. For instance, it is possible to ensemble wav2vec 2.0 and MERT to extract better representations from singing audio.

E.2 More Qualitative Results for our ALT systems.

Fig. 10. More qualitative results of our audio-only/audio-visual ALT systems under different SNR environments. Deletions are marked using brackets and purple color, and substitutions are marked using blue color.



Received 29 July 2023; revised 25 December 2023; accepted 26 February 2024