
C3LLM: Conditional Multimodal Content Generation Using Large Language Models

Zixuan Wang*
HKUST

zwanggk@connect.ust.hk

Qinkai Duan*
HKUST

qduanaa@connect.ust.hk

Yu-Wing Tai
Dartmouth College
yuwing@gmail.com

Chi-Keung Tang
HKUST
cktang@cse.ust.hk

Abstract

We introduce C3LLM (Conditioned-on-Three-Modalities Large Language Models), a novel framework combining three tasks of video-to-audio, audio-to-text, and text-to-audio together. C3LLM adapts the Large Language Model (LLM) structure as a bridge for aligning different modalities, synthesizing the given conditional information, and making multimodal generation in a discrete manner. Our contributions are as follows. First, we adapt a hierarchical structure for audio generation tasks with pre-trained audio codebooks. Specifically, we train the LLM to generate audio semantic tokens from the given conditions, and further use a non-autoregressive transformer to generate different levels of acoustic tokens in layers to better enhance the fidelity of the generated audio. Second, based on the intuition that LLMs were originally designed for discrete tasks with the next-word prediction method, we use the discrete representation for audio generation and compress their semantic meanings into acoustic tokens, similar to adding “acoustic vocabulary” to LLM. Third, our method combines the previous tasks of audio understanding, video-to-audio generation, and text-to-audio generation together into one unified model, providing more versatility in an end-to-end fashion. Our C3LLM achieves improved results through various automated evaluation metrics, providing better semantic alignment compared to previous methods.

1 Introduction

Conditional multimodal generation is the task of generating output that incorporates different modalities, such as text, image, video, and audio [24, 48, 21, 22]. Essentially, this multimodal task can be seen as a translation task among different modalities, and thus, challenges arise for making inferences from cross-modal representations and dealing with potential modality gaps [29].

Multimodal Large Language Models (MM-LLMs) have recently gained significant interest in research due to their ability to understand and follow user instructions. Most work focuses on contextual understanding across various modalities like video-to-text [52, 40], audio-to-text [20, 25] and image-to-text [31]. However, the area of audio generation, particularly video-to-audio [26, 10] or image-to-audio generation [39], remains underexplored. This is partly because video contains excessive visual information not always needed for audio generation, while images lack the temporal information crucial for audio. Video-to-audio tasks also face synchronization challenges, with recent solutions like temporal masking [50] proving inadequate for complex scenarios. Additionally, current methods often encode video features by extracting a few random frames [50, 7], which hinders learning temporal information.

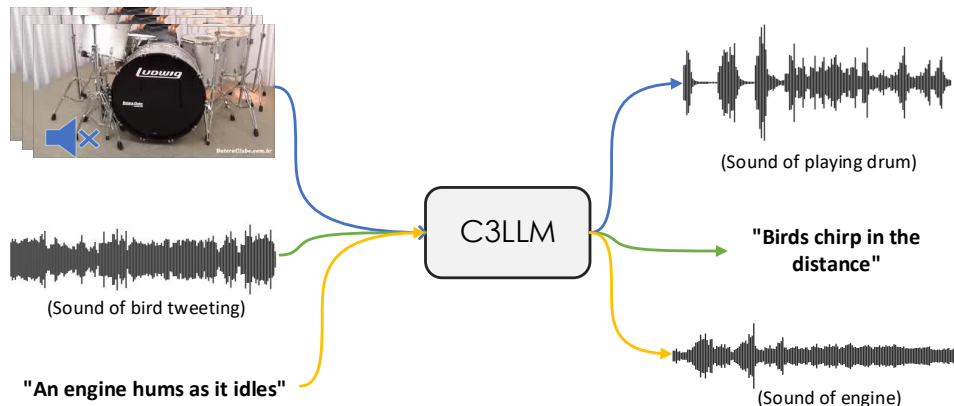


Figure 1: C3LLM is capable of video-to-audio, audio-to-text, and text-to-audio. Examples of different tasks are illustrated in arrows of different colors.

Audio generation is less intuitive compared to other tasks, as it is harder to precisely measure the generated sound quality using human ears. Additionally, previous works mainly focus on generating music-related audio, which is more structured compared to naturally occurring audio [6, 32]. Few works have focused on generating visually guided open-domain audio clips [4, 55]. Moreover, most existing models [26, 51] are only limited to generating audio-only content, consequently constrained to specific downstream tasks. While contemporary work CoDi [44] achieves some form of any-to-any generation, their result simply uses linear interpolations in the latent space. C3Net [53] adapted three ControlNet [54] architectures on top of CoDi’s design, still it also relies on interpolation when predicting the final output.

In this connection, we aim at utilizing the versatility of LLM to align between different modalities. With sufficient data, transformers or LLMs have shown to be effective in serving as a unified backbone on different modality tasks even with simple design [42]. We thus propose Conditioned-On-Three-Modalities Large Language Model, or *C3LLM* in short. *C3LLM* mainly comprises a LLM backbone serving as the bridge among three different modalities, and a hierarchy audio tokenizer adapted from Encodec [11] for decoding. Our model first encodes the respective modality, either audio, text or video, to be processed by the LLM backbone. For video, we extract the dense information and project it to the LLM’s embedding space. For audio, we use the audio tokenizer to convert the information into discrete representation and translate the corresponding indices from the tokenizer codebooks into LLM special tokens, which are extended as part of LLM’s vocabulary beforehand. The semantic information is further processed by the LLM. For tasks involving audio generation, We treat the LLM prediction as coarse acoustic tokens. The preliminary result is further extended to fine-grained acoustic tokens and combined to generate the final audio output, and thus multimodal output not limited to a single modality.

To sum up, our contributions are as follows: 1) *C3LLM* utilizes the versatility of LLM for conditional multimodal generation tasks, where the LLM treats encoded audio information as additional acoustic vocabulary; 2) *C3LLM* uses a discrete tokenizer for modeling audio representation hierarchically, which better suits the nature of LLM while preserving the quality of the generated audio; 3) Consequently, this paper provides a uniform model for three different tasks involving video, audio and text, see Figure 1. Through extensive evaluation, our work demonstrates on-par results compared to the state-of-the-art models in their respective domains, providing further insight into conditional multimodal tasks.

2 Related Work

Multimodal Alignment Mokady et al. [33] has utilized the powerful Contrastive Language-Image Pre-Training (CLIP) [37] model to project the image into an image-text shared latent space. Given image embedding, they further apply a GPT-2 [38] to generate caption. Wu et al. [17] trained CLAP model on LAION-Audio-630K, a large collection of 633,526 audio-text pairs from different data sources, to obtain a robust and general result on all types of audio clips. Similar to them, we encode

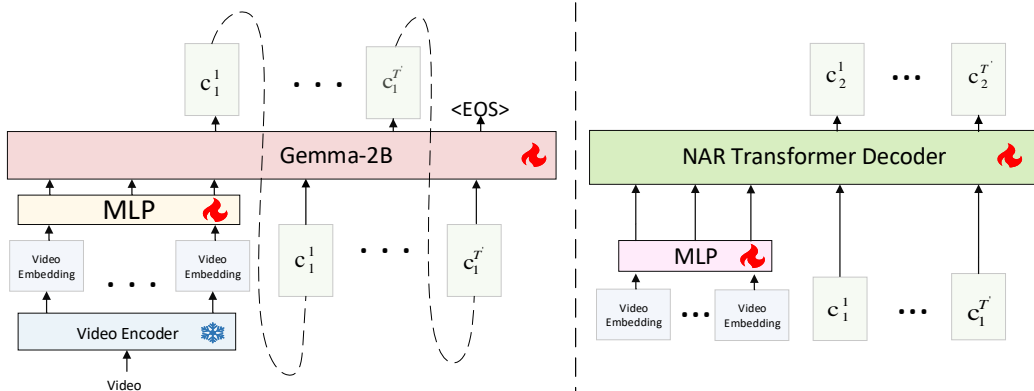


Figure 2: Overview of the **video** encoding part and the LLM. On the right is the non-autoregressive (NAR) transformer to further extend the coarse acoustic tokens into fine-grained acoustic tokens. We freeze the encoder, train a MLP, and finetune LLM using LoRA. For text condition, we directly input them into LLM after tokenization and also treat them as the condition for NAR transformer decoding.

video, audio and text data into a shared latent space using contrastive learning and projection. This approach enables an audio encoder to project audio into a space shared with image and text, aligning more closely with the hidden space of a large language model. For long audio understanding, local features and overall information are fused.

Multimodal LLM LLM is prevailing in the AI community. ChatGPT [34] and GPT-4 [35] have shown great power in understanding and reasoning tasks. Other open source LLMs, such as Llama [36], Vicuna [5], Alpaca [45], and Gemma [16] have greatly contributed to the research community and inspired many innovations. With these advances, progress has been made in using LLMs for understanding multimodal information, improving the performance of MM-LLMs [31].

We notice that most LLMs understand multimodal information with the help of a well-trained encoder to bridge the gap between modality information and LLM hidden space. For example, PandaGPT [41] utilizes ImageBind [19] to align multiple modalities. Video-LLama [52] applies two Q-formers to transform video and audio information. Llava [31] directly applies a linear projection layer to give LLM image information. MiniGPT-4 [56] follows this approach, using linear layers to align Vicuna and Q-former. To fit the discrete and auto-regressive nature of LLMs, Large World Model [30] achieves long-context video understanding by combining VQGAN [12] to tokenize each frame of the video. However, these LLMs lack the capability to generate modalities other than language.

Multimodal Generation In multimodal generation, the goal is to generate various modalities like audio, text, images, and videos interchangeably. State-of-the-art approaches, such as Composable Diffusion (CoDi) [44] and C3Net [53], produce diverse modality combinations conditioned on other modalities. NExT-GPT [49] and CoDi2 [43] leverage LLMs to process and synthesize semantic information from different modalities, providing a wide range of meaningful conditions for the diffusion model. Diffusion models are crucial for generating and refining high-quality content, typically encoding each modality into a shared latent space and using MLPs to project information between the LLM hidden space and the diffusion latent condition space.

However, these methods often overlook the distinct features of each modality. For instance, NExT-GPT uses ImageBind [19] to encode videos into image space, losing temporal features. In audio generation, the diffusion model struggles to map audio timing accurately to corresponding video frames, making it challenging to incorporate the temporal dimension of video modality.

Conditioned Audio Generation To better focus on the time domain of audio, the transformer architecture [46] is well-suited due to its attention mechanism. In the context of audio generation, SpecVQGAN [26] has successfully employed a neural codebook to represent audio information, enabling the use of a transformer to predict discrete audio tokens based on video features.

Recently, VALL-E [47] and MusicLM [13] use multiple codebooks and Residual Vector Quantization (RVQ) [11] to create diverse audio representations. An auto-regressive transformer is employed to

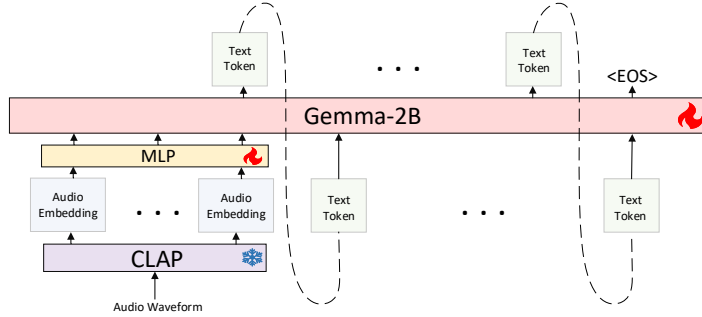


Figure 3: Overview of **audio** encoding part and the LLM. We use a pre-trained CLAP encoder and an MLP layer to align the audio information with LLM embedding.

predict the initial layers of audio indices based on the textual conditions, and a non-auto-regressive (NAR) transformer is followed to generate the subsequent layers, taking into account the previously generated tokens and conditions. It has been observed that the first audio index layer captures crucial information for overall audio quality, like rhythm and intonation, while subsequent layers add detailed information, enhancing richness.

3 Method

C3LLM comprises three major components: 1) Tokenizers for encoding corresponding modality information to be comprehended by LLM; 2) a multimodal LLM that serves as a powerful backbone that connects corresponding modalities. Especially for audio generation tasks, we have 3) a non-autoregressive (NAR) Transformer that further refines the generated coarse acoustic tokens from the LLM. Our model structure is illustrated in Figure 2 and Figure 3.

3.1 Audio Tokenizer and Video Tokenizer

For audio generation, we employ the EnCodec [11] pretrained codebooks and decoder, which utilizes the RVQ method to obtain eight codebooks that enable high-bandwidth audio reconstruction. To streamline the process and conserve computational resources, we focus on the first two codebooks. We leverage the LLM to predict the first layer, taking into account text or visual conditions, and we train the NAR transformer decoder specifically to predict the second layer of audio tokens. After the prediction of the first two layers of audio tokens, the pre-trained decoder decodes them back into audio waveform.

For the audio-to-text task, we leverage the powerful audio encoder of CLAP to convert audio waveforms into a shared space that is more similar to the hidden space of the LLM. We freeze the audio encoder and introduce an MLP to project the audio vectors into the hidden space of the LLM. This allows for a more effective integration of audio information into the text-generation process. By utilizing the capabilities of CLAP and LLM together, we achieve improved performance in generating detailed descriptions from audio inputs.

When it comes to the video-to-audio task, we draw inspiration from the successful approach employed by SpecVQGAN [26]. To efficiently capture both visual and temporal information while compressing the video data, we employ a frame-wise feature extractor denoted as H . This feature extractor extracts RGB features (f^r) and optical flow features (f^o) from each frame of the video. By applying frame-wise concatenation, we obtain the video feature representation $F = \{f_i^r, f_i^o\}_{i=1}^N$, where N represents the total number of frames in the video. To further enhance the integration of visual information, we introduce an additional MLP that transforms the concatenated video features into embeddings suitable for the LLM.

Contrary to the discrete audio tokenization method, we choose to use the continuous representation for video. During our experiment, we observe that video involves too much visual information that audio generation may not necessarily need. A similar discrete method that uses VQGAN [12] to process frame-level information will result in excessive visual tokens, making learning visual information inefficient. The MLP projection layer that we employed will project the continuous video

feature into LLM embedding space, similar to previous methods [31]. For text input, we directly use the LLM’s tokenizer for tokenization.

3.2 Autoregressive Generation of Coarse Audio Tokens

Employing the EnCodec audio tokenizer allows us to represent continuous audio information in discrete form. We denote the continuous audio input as $a \in \mathbb{R}^{C \times L}$, where C is the number of channels and L is the time of the audio clip times sample rate. First, the audio input is encoded in a smaller representation in the form of $z = E(a) \in \mathbb{R}^{C \times N \times D \times Q}$, with Q denoting the number of quantizers used during the encoding process and D is the dimension of the codebooks. Our next step is to convert the representation into LLM-aware acoustic tokens. Specifically, we obtain the indices $s \in \mathbb{R}^{C \times N \times Q}$ from the encoded audio by comparing with the quantizer codebook.

To jointly model different modalities in a unified model, we further extended the LLM’s text vocabulary $V_t = \{v_i\}_{i=1}^{N_t}$ with acoustic vocabulary $V_a = \{v_j\}_{j=1}^{N_a}$. The extended audio-text vocabulary now becomes $V = \{V_t, V_a\}$. Contrary to previous audio generation models [26, 10] that involve the generation of a single modality, our method equipped the LLM backbone with the ability to understand and generate both audio and text information with a unified vocabulary.

To better differentiate the three kinds of modalities that condition the autoregressive generation, we further wrap the encoded feature with special tokens as modality indicators. To be more specific, we wrap the audio tokens with $\langle \text{Audio} \rangle, \langle / \text{Audio} \rangle$ indicators and video embedding in an embedded sequence of $\langle \text{Video} \rangle, \langle / \text{Video} \rangle$ indicators. In doing so, we avoid the possibility of confusing the LLM with different kinds of information.

To further elaborate on the conditional generation tasks performed by LLM: for audio-to-text and text-to-audio tasks, the source input $X_{a,t} = \{x_t^i\}_{i=1}^N$ is a sequence of either acoustic/text tokens. Here N is the number of tokens we have and $x_t \in V$; for video-to-audio task, the source input $X_v = \{x_e^i\}_{i=1}^N$ is a sequence of embeddings and $x_e \in \mathbb{R}^D$, where D is the embedding dimension of LLM. After LLM’s tokenization, the input tokens for audio and text will become input embeddings and fed into the LLM. Our LLM backbone is a decoder-only structure with the next token prediction method. The distribution of the predicted token in the first layer is given by $p_{\theta_{LLM}}(\mathbf{C}_1|X) = \prod_i p_{\theta_{LLM}}(c_1^i|X, \mathbf{C}_1^{<i})$ autoregressively. The objective has thus become:

$$\mathcal{L}_{LLM} = - \sum_{i=1}^{T'} \log p_{\theta_{LLM}}(c_1^i|X, \mathbf{C}_1^{<i}), \quad (1)$$

where T' is the number of acoustic tokens generated by LLM, θ_{LLM} is the parameter of LLM, c_1^i is the token generated at step i , $\mathbf{C}_1^{<i}$ are previous tokens, and X is the text or video condition.

During inference, the LLM will autoregressively predict the next token until $\langle eos \rangle$ is generated. Our LLM thus serves as the bridge for connecting between different pairs of modalities. The generated output will be decoded subsequently.

Due to the limited computation resource available, we use Gemma-2B [16], a lightweight open-source LLM developed by Google, which is claimed to have comparable performance with LLaMA-2-7B [14] on many QA and reasoning tasks. We use Low-Rank Adaptor (LoRA) [23] to finetune Gemma to make it understand vision/text conditions and generate audio tokens.

3.3 Non-Autoregressive Transformer for Audio Refinement

In C3LLM, we propose an audio refinement method to further ensure the generated audio fidelity. Inspired by [47], we utilize a non-autoregressive Transformer (NAR) to transform coarse acoustic tokens from LLM’s output to fine-grained acoustic tokens. We treat the video embedding or text input as condition and concatenate it with the generated coarse acoustic tokens. In the original paper, the NAR is used to predict seven layers of acoustic tokens given by the first layer. However, we find this design very slow to converge during our experiment. We adopt a simpler design to only utilize two layers of codebooks, and train the NAR to predict the second layer given the first layer prediction generated by LLM. Thus probability distribution for the next layer is given by $p_{\theta_{NAR}}(\mathbf{C}_2|\mathbf{X}, \mathbf{C}_1)$,

and we want to minimize the objective function:

$$\mathcal{L}_{NAR} = -\log p_{\theta_{NAR}}(\mathbf{C}_2|\mathbf{X}, \mathbf{C}_1) = -\sum_{i=1}^{T'} \log p_{\theta_{NAR}}(c_2^i|\mathbf{X}, \mathbf{C}_1). \quad (2)$$

3.4 Detokenization for High Fidelity Output

The decoder combines LSTM and CNN architectures. The LSTM component emphasizes temporal consistency, while the CNN component reconstructs frequency information. The model employs a combination of L1 loss for the time domain and a set of L1 and L2 losses for the Mel-spectrogram in the frequency domain, across various time scales. To effectively preserve audio information, the model incorporates two strategies: 1) Utilizing a greater number of tokens to represent each second of audio, thereby increasing the sample rate. 2) Employing multiple codebooks to capture a wider bandwidth, enhancing the representation.

The audio waveform can experience interference from multiple sources. By employing more codebooks, the model can effectively decompose overlapping signals, with each codebook capturing audio of different frequencies.

4 Experiments

4.1 Training Datasets

We apply corresponding datasets for the three tasks. For video-to-audio task, we finetune our model on the VGGSound [3] dataset, which contains over 310 classes of 200,000+ videos, capturing challenging real-world acoustic scenarios. Concerning the large size, we use around half of the common version of VGGSound containing 164 classes. The resulting number of training video samples is 63,853.

For audio-to-text and text-to-audio tasks, we use the AudioCaps [8] dataset. AudioCaps [8] dataset is a large-scale dataset of about 46K audio clips paired with human-written text collected via crowdsourcing on the AudioSet [18] dataset. The dataset contains a diverse range of 10-second audio samples from various natural sources, including vehicles, animals, weather, etc. We filter all the failed links and produce 45,028 sound files in train split. For audio-to-test task, we notice that in some papers such as EnCLAP [27], multiple referencing ground truth are used for evaluation. As the original AudioCaps paper [8] only contains one caption per audio clip, we choose to only use one caption as the referencing ground truth.

4.2 Evaluation Metrics

The evaluation metrics are summarized as follows: For video-to-audio and text-to-audio tasks, we use the Inception score (ISc) and Frechet audio distance (FAD) to evaluate audio fidelity. For audio-video relevance, we utilize the MKL metric [26] and we use KL for text-to-audio task. For audio-to-text task, we use the CIDEr (Consensus-based Image Description Evaluation), SPIDEr (SPeech-to-Image Description Evaluation), and SPICE (Semantic Propositional Image Caption Evaluation). Furthermore, to evaluate the synchronization of the generated audio in the video-to-audio setting, we use the same evaluation metrics as CondFoleyGen [10], namely # Onset Accuracy [10], and Onset AP [10].

4.3 Evaluation and comparison

We mainly compare our model with CoDi [44], which is the current state-of-the-art model combining different multimodal content generation tasks. We download the pretrained fp16 version of CoDi [44] model and evaluate on the same test set. The training and evaluation are conducted on NVIDIA GeForce RTX 4090. The main result is presented in Table 1

For audio-to-test task, we use the open-sourced Audio Captioning metrics [28] for evaluation. we observe that CoDi’s performance is lower by a large margin, which might be the result that LLM is more capable of captioning tasks due to the next-word prediction method. We include some test results in Table 3.

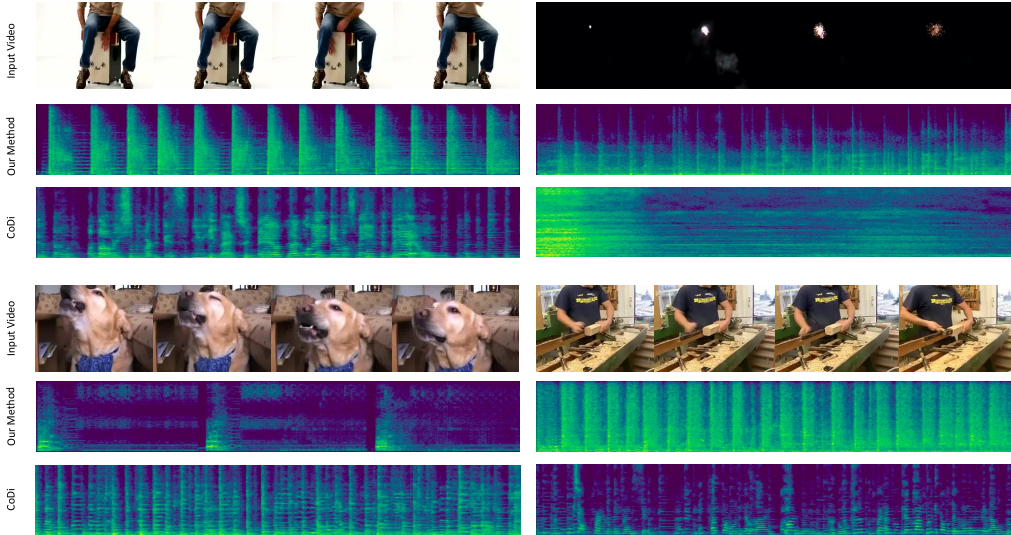


Figure 4: Comparison with baseline for video-to-audio generation task. CoDi failed to generate semantic-aligned audio and the generation is not clean, often mixed with human speaking or noise. Our method can produce aligned audio with clear synchronization.

Task	Method	Metric		
		KL↓	ISc↑	FAD↓
V2A	Codi [44]	3.6449	3.6941	11.5036
	Ours (w/o NAR)	3.5873	2.1301	17.1095
	Ours	3.5522	2.5783	10.2217
A2T	Codi [44]	0.0726	0.0812	0.0640
	Ours	0.3150	0.4721	0.1579
		SPIDEr↑	CIDEr↑	SPICE↑
T2A	CoDi [44]	3.1786	4.5047	10.1597
	Ours (w/o NAR)	3.4865	2.5732	22.7320
	Ours	3.6765	2.9606	18.1615

Table 1: Quantitative comparison with baseline on three tasks.

For video-to-audio task, our result is better than the baseline except for the ISc metric. The ISc metric measures the diversity of the generated audio. As our model is only fine-tuned on Gemma-2B [16] backbone, we believe our result will be improved with more power backbone and tunable parameters, given more training resources. Additionally, our method achieves synchronization with the input video, as shown in Figure 4. The quantitative evaluation is presented in Table 2

We notice that for text-to-audio, CoDi is trained on multiple datasets such as AudioCaps [8], AudioSet [18], BBC Sound Effect, Soundnet [2], and Freesound. On the other hand, our model is only trained on AudioCaps. The total number of training examples for CoDi is significantly larger than ours. Furthermore, there is a huge domain gap between text and audio. Audio waveform has time information while text does not, so it is hard to map a sentence to a specific audio token in each time frame. Besides, we utilize LoRA [23] to finetune LLM which is not capable of bridging the gap with so few trainable parameters. These are the reasons for the comparison result.

4.4 Ablation Studies

We hereby conduct experiments to test how the non-autoregressive transformer will refine the output coarse acoustic tokens. We include our results in Table 1. As shown in the table, the NAR plays a central role in further improving the result.

Method	Metric	
	# Onset Accuracy \uparrow	Onset AP \uparrow
Codi [44]	0.097	0.535
Ours	0.142	0.670

Table 2: Results for evaluating video-audio synchronization on VGGSound dataset

Ground truth	Our method	CoDi [44]
continuous snoring of a person	a person snoring	dog sleeping on a bed
church bells ringing	bells are ringing	angel is in a coat by in red hat
a car engine is revving while driving	a vehicle engine accelerates	driving for speed going for a crossing
a telephone ringing	a telephone rings several times	phone of the phone
A cat meowing a few times	a cat is meowing	catwoman’s cats are coming cat of the cats in a cat calendar
spinning tires on pavement	a car is skidding wildly	car for just avoiding a speeding car wreck with cold scary highway blare

Table 3: Samples output obtained from AudioCap test set. We observe that CoDi often fails to capture the semantic meaning, and the generated captions are more like describing visual input rather than acoustic sound. Additionally, the output is not sufficiently fluent. We thus believe LLM structure is more capable for captioning tasks

Table 4: Additional A2T task conducted on Clotho dataset

Method	Metric		
	SPIDER \uparrow	CIDEr \uparrow	SPICE \uparrow
Codi [44]	0.0640	0.0766	0.0514
Ours	0.2088	0.3097	0.1078

Table 5: Additional V2A task conducted on VAS dataset

Method	Metric		
	KL \downarrow	ISc \uparrow	FAD \downarrow
Codi [44]	4.54874	3.12170	11.80060
Ours	3.97517	2.69774	10.35411

To mitigate the possible effect of using only one referencing ground truth for evaluating audio-to-text task, we tested our model on the Clotho dataset [9]. The Clotho dataset contains 4981 audio clips and 3938 clips in train split. Audio waveforms are from 15 to 30s duration, and each audio has 5 captions which are 8 to 20 words long. We perform similar processing as the AudioCaps dataset for training and evaluation. Table 4 tabulates the results.

We also conducted an additional evaluation of our model on the VAS [15] dataset for video-to-audio task, as shown in Table 5. We obtain a similar result as VGGSound. Our model continues to outperform the baseline.

5 Conclusion and Discussion

5.1 Limitation and Future Work

Due to limited computation power, we can only take one modality input and generate text or audio. Next, we want to condition on two or more modalities and generate video, and we want to generate long audio/video. Moreover, to bridge the modality gap between text-to-audio generation, we will adopt semantic tokens of audio to give well-aligned information. We will utilize more powerful LLM backbones such as LLama3 [1]

5.2 Conclusion

In this paper, we present C3LLM, a unified structure that can perform three tasks namely video-to-audio, audio-to-text and text-to-audio. Our model capitalizes on the power of LLM for translating and aligning between different modalities. We also propose a non-autoregressive transformer for audio refinement. Through extensive experiments, we show that our model can synthesize high-fidelity audio, ensuring semantic alignment with input, especially synchronization with the visual condition.

Although our model demonstrates excellent results through evaluation, challenges exist that can restrain the performance. Specifically for the audio-to-text task, our model relies on the pre-trained CLAP encoder that poses an upper-bound for modeling more complex scenarios. A more efficient way for audio encoding is worthwhile in future research.

References

- [1] AI@Meta. Llama 3 model card, 2024.
- [2] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, 2016.
- [3] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP*, 2020.
- [4] Peihao Chen, Yang Zhang, Mingkui Tan, Hongdong Xiao, Deng Huang, and Chuang Gan. Generating visually aligned sound from videos. *IEEE Transactions on Image Processing*, 2020.
- [5] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.
- [6] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *arXiv:2306.05284*, 2024.
- [7] Hao-Wen Dong, Xiaoyu Liu, Jordi Pons, Gautam Bhattacharya, Santiago Pascual, Joan Serra, Taylor Berg-Kirkpatrick, and Julian McAuley. Clipsonic: Text-to-audio synthesis with unlabeled videos and pretrained languagevision models. *WASPAA*, 2023.
- [8] Kim Chris Dongjoo, Kim Byeongchang, Lee Hyunmin, and Kim Gunhee. Audiocaps: Generating captions for audios in the wild. In *NAACL-HLT*, 2019.
- [9] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: an audio captioning dataset. In *Proceedings of the ICASSP*, pages 736—740, 2020.
- [10] Yuexi Du, Ziyang Chen, Justin Salamon, Bryan Russell, and Andrew Owens. High fidelity neural audio compression. In *CVPR*, 2023.
- [11] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv:2210.13438*, 2022.
- [12] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.
- [13] Andrea Agostinelli et. al. Musiclm: Generating music from text. *arXiv:2301.11325*, 2023.
- [14] Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.
- [15] Peihao Chen et al. Generating visually aligned sound from videos. *TIP*, 2020.
- [16] Thomas Mesnard et al. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. *arXiv:2403.08295*, 2024.
- [17] Yusong Wu et al. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. *arXiv:2211.06687*, 2022.
- [18] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.
- [19] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023.

- [20] Yuan Gong, Hongyin Luo, Alexander H. Liu, Leonid Karlinsky, and James Glass. Listen, think, and understand. *arXiv:2305.10790, 2023b.*, 2024.
- [21] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *arXiv:2210.02303, 2022.*
- [22] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv:2205.15868, 2022.*
- [23] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv:2106.09685, 2021.*
- [24] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Visual instruction tuning. *arXiv:2304.08485, 2024.*
- [25] Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, Yi Ren, Zhou Zhao, and Shinji Watanabe. Audiogpt: Understanding and generating speech, music, sound, and talking head. *ArXiv, abs/2304.12995, 2023.*
- [26] Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. *arXiv:2110.08791, 2021.*
- [27] Jaeyeon Kim, Jaeyoon Jung, Jinjoo Lee, and Sang Hoon Woo. Enclap: Combining neural audio codec and audio-text joint embedding for automated audio captioning. *arXiv preprint arXiv:2401.17690, 2024.*
- [28] Etienne Labbé. aac-metrics: Metrics for evaluating automated audio captioning systems for pytorch. <https://github.com/Labbeti/aac-metrics/>, 2013.
- [29] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the gap: Understanding the modality gap in multimodal contrastive representation learning. *arXiv:2203.02053, 2022.*
- [30] Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with ringattention. *arXiv preprint, 2024.*
- [31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv:2304.08485, 2024.*
- [32] Jan Melechovsky, Zixun Guo, Deepanway Ghosal, Navonil Majumder, Dorien Herremans, and Soujanya Poria. Mustango: Toward controllable text-to-music generation. *arXiv:2311.08355, 2023.*
- [33] R. Mokady, A. Hertz, and A. H. Bermano. Clipcap: Clip prefix for image captioning. *arXiv:2111.09734, 2021.*
- [34] OpenAI. Introducing chatgpt, 2022.
- [35] OpenAI. Gpt-4 technical report, 2023.
- [36] OpenAI. Llama: Open and efficient foundation language models, 2023.
- [37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
- [38] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.
- [39] Roy Sheffer and Yossi Adil. I hear your true colors: Image guided audio generation. In *ICASSP*, 2023.

- [40] Fangxun Shu, Lei Zhang, Hao Jiang, and Cihang Xie. Audio-visual llm for video understanding. *arXiv:2312.06720*, 2024.
- [41] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355*, 2023.
- [42] Quan Sun, Qiyang Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. *arXiv:2307.05222*, 2024.
- [43] Zineng Tang, Ziyi Yang, Mahmoud Khademi, Yang Liu, Chenguang Zhu, and Mohit Bansal. Codi-2: In-context, interleaved, and interactive any-to-any generation. *arXiv:2311.18775*, 2023.
- [44] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *arXiv:2305.11846*, 2023.
- [45] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.
- [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [47] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural codec language models are zero-shot text to speech synthesizers. *arXiv:2301.02111*, 2023.
- [48] Zhao Wang, Aoxue Li, Enze Xie, Lingting Zhu, Yong Guo, Qi Dou, and Zhenguo Li. Customvideo: Customizing text-to-video generation with multiple subjects. *arXiv:2401.09962*, 2024.
- [49] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *arXiv:2309.05519*, 2023.
- [50] Zhifeng Xie, Shengye Yu, Qile He, and Mengtian Li. Sonicvisionlm: Playing sound with vision language models. *arXiv:2401.04394*, 2024.
- [51] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *arXiv:2207.09983*, 2022.
- [52] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *CoRR, abs/2306.02858*, 2023.
- [53] Juntao Zhang, Yuehuai Liu, Yu-Wing Tai, and Chi-Keung Tang. C3net: Compound conditioned controlnet for multimodal content generation. *arXiv:2311.17951*, 2023.
- [54] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.
- [55] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L. Berg. Visual to sound: Generating natural sound for videos in the wild. In *CVPR*, 2018.
- [56] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

A Appendix / supplemental material

A.1 Audio-to-Text examples

In this section, we provide more examples of the predicted output of the audio-to-text task in Table 6, in addition to Table 3. We also include some examples from the Clotho dataset [9], shown in Table 7.

Ground truth	Our method	CoDi [44]
a person snoring	a person snores lightly	sleeping or a baby in a skiing
a river stream of water flowing	water is rushing by	brown birds outside in the forest
clicking followed by humming noise	an engine humming and clicking	highway worker hiking cold turkey
sounds of a river with man briefly mumbling	water is flowing and a man speaks	waterfall falls in water
several goats bleat	sheep bleat nearby	the brown goat
a police siren going off	a siren is emitted	clear weather warning direction as water officer does traffic safety direction while a traffic stop in her direction.

Table 6: Additional samples output obtained from AudioCap test set. We again notice that words describing colors or visual scenes exist in CoDi’s output which is unusual.

Ground truth 1	Our method	CoDi [44]
a radio dispatcher and an officer are communicating over the radio	a radio is tuned, as a person speaks over the radio	foreign radio is not time to watch someone on the phone to communications wire
lost of people are conversing in a very busy diner	a group of people are talking and laughing	people walking in the area
a machine is running in a humming manner while metal is buzzing	a buzzing electric engine that is trying to start up	pilot cable bowler take a caution steer light not flying < speed> engine.

Table 7: Additional samples output obtained from Clotho test set. Due to the space limit, we only include ground truth caption 1. Other referencing groundtruth can be found in the CSV file provided by the original paper [9].