EgoSonics: Generating Synchronized Audio for Silent Egocentric Videos

Aashish Rai Srinath Sridhar

Brown University

https://ivl.cs.brown.edu/research/egosonics

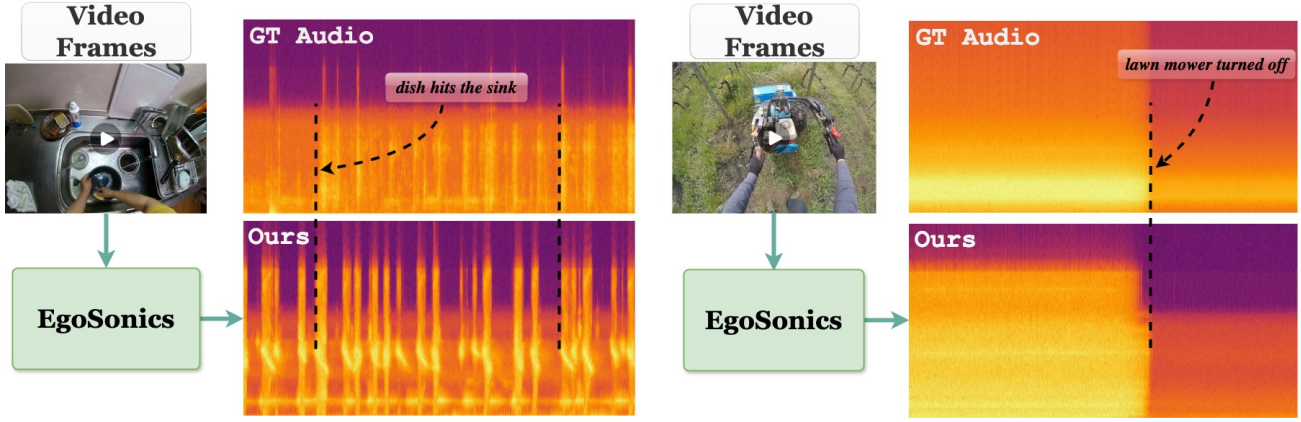

Figure 1. *We present **EgoSonics**, a method to synthesize audio tracks conditioned on silent in-the-wild egocentric videos. Our method operate on videos at 30 fps, and can synthesize audio that is semantically meaningful and synchronized with events in the video ("dish hits the sink" or "lawn mower turned off"). We also propose a new method to evaluate audio-video synchronization quality.*

EGOSONICS - SUPPLEMENTARY

A. User Study

We conducted an user study for a subjective evaluation of our synthesized audios. With randomly selected 15 participants from different backgrounds and asked them to do a 10 minutes survey using Google Forms. The survey presented 16 videos with audios to the participants and asked them to rate their realism on a scale of 1-5 (1 being "Doesn't Sound Real" and 2 being "Sounds Real"). Participants were also asked if the audio is contextually relevant to the video, if the events in the video and the corresponding audio are synchronized. To verify that our method doesn't overfit to one category of sounds, we also asked the participants to evaluate if the audio seems to be from a different category (e.g., a carpentry sound coming from a vacuum cleaner).

The following are the outcomes of this study:

1. 80% of the users believe that 90% of our audios are realistic with an Mean Opinion Score (MOS) [11] of more than 4.0. This means that EgoSonics is able to generate realistic audios.

2. In 100% of the cases, users preferred our audio over the current SOTA V2A model (Diff-Foley). This means all the 15 users believe that all of our audios are better that the baseline.

3. In more than 75% cases, users were unable to distinguish our audio from the GT audio, and rated both to be realistic. This again proves that EgoSonics is able to generate realistic and synchronized audio.

4. In more than 80% of the cases, users did not find our audios to be belonging to another category, meaning that EgoSonics doesn't overfit to any particular class and is able to generate all kinds of daily activity audios.

B. Syncronet

We propose *Syncronet*, a model to learn the correspondence between audio and video modalities by learning correlation between the input video embeddings and the audio frequencies for every time step t. ControlNet [13] is a current state-of-the-art neural network architecture that was introduced to enhance large pretrained text-to-image diffusion models with spatially localized, task specific image conditions providing pixel-level control. Over the time, ControlNet has been shown to work for a variety of controlled image generation tasks including but not limited to spatial conditions like *Canny edges, Hough lines, user scribbles, human key points, segmentation maps, shape normals, depths, cartoon line drawings*, etc. ControlNet basically works by processing these spatial conditioning and injecting additional control signals to the pretrained diffusion models. However, generating time-aware control signals to guide the

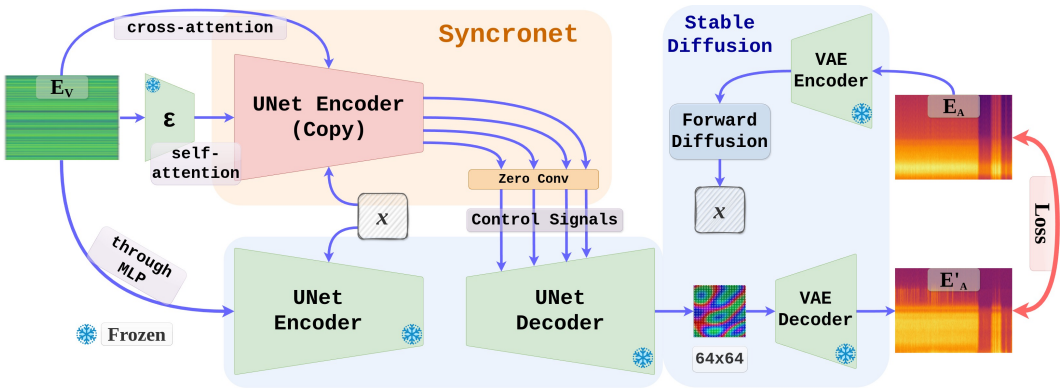

Figure 2. *Syncronet* training.

output in temporally consistent manner has not been extensively studied using ControlNet. In fact, to the best of our knowledge, only Music ControlNet [12] uses it to generate partially-specified time-varying control signals. In this paper, we modified the ControlNet architecture to operate over given video embedding in both encoded feature space, and time space to generate control signals that can provide local pixel-level control to the Stable Diffusion's UNet model to generate time-consistent audio spectrograms E_A. The generated spectrograms possess a strong correlation with the input video embedding and this results in highly synchronized audio of daily activity videos - where most previous methods fail due to design constraints.

We use *Syncronet* to provide control signals to Stable Diffusion (SD) 2.1. As a first step, we make a trainable copy of the entire UNet based encoder of SD along with the middle block and initialized them with the same pretrained weights of SD 2.1. The goal here is to use these trainable encoder to generate control signals that can be plugged into the pretrained SD's UNet decoder blocks providing pixel-level control to output (see Fig. 2). The trainable copy is connected to the frozen SD through zero convolution layers to avoid any influence of noisy control signals during the start of the training.

Similar to ControlNet, the input conditioning image (E_V) of size 512×512 is converted to a feature space of size 64×64, that matches the feature space of Stable Diffusion, through a pre-trained image encoder ϵ. We used the same convolution based image encoder as used in [13], and initialized it with the same weights. The image encoder is kept frozen throughout the training. The encoder ϵ extracts a feature space vector c_f from the input conditioning E_V.

As shown in Fig. 2, the noisy sample x is generated through forward diffusion process, where the input audio spectrogram of size 512×512 is encoded to a feature vector of size 64×64 through pre-trained VAE encoder. We used the same pretrained VAE encoder-decoder network as SD and initialized them with the same weights. Noise is added to the feature vector to get a noisy sample x for $T = 1000$ steps. Then the encoded video embedding $\epsilon(E_V)$ is added to the input noisy data sample x. $h = x + \epsilon(E_V)$. The added sum is further enriched by passing it through a self-attention block.

$$Self - Attn(Q_h, K_h, V_h) = Softmax(\frac{Q_h K_h^T}{\sqrt{d_K}})V_h \quad (1)$$

$$h = h + Self - Attn(Q_h, K_h, V_h)$$

where, Q_h, K_h, V_h represents the Query, Key, and Value matrices derived from h. As we have seen in Table **??**, only using this self attention block alone is sufficient to generate good quality audio spectrograms with an FID of 34.33. However, as the alignment score suggests, this alone is not sufficient to guide the diffusion model generate temporally consistent synchronized audio.

Thus, to inject the temporal consistency to the control signals of *Syncronet*, we also apply the cross-attention between the original video embedding E_V and h to guide the audio spectrogram generation directly with the time steps of E_V. This helps the model effectively learn the synchronization between the time domain and the rich feature space of Stable Diffusion, as we can see through a significant improvement in the alignment score. There are two options to apply cross-attention, either by using Query from E_V and Key and Value matrices from h, or vice-versa. In our case, we use the latter approach as follows:

$$Cross - Attn(Q_h, K_V, V_V) = Softmax(\frac{Q_h K_T^T}{\sqrt{d_K}})V_V \quad (2)$$

$$h = h + Cross - Attn(Q_h, K_V, V_V)$$

where, Q_h, K_V, V_V represents the Query, Key, and Value matrices derived from h and E_V, respectively. After passing

Algorithm 1 Generate Control Signals

Require: x: Input Noisy Sample
Require: EV: Video Embedding
Require: $timesteps$: Timestep tensor
Ensure: $control\_signals$: List of control signals

$t\_emb \leftarrow$ timestep_embedding$(timesteps)$
$emb \leftarrow$ time_embed(t_emb)
$context \leftarrow EV$
$guided\_hint \leftarrow$ encoder(EV)
$control\_signals \leftarrow [\,]$
$h \leftarrow x$
for $(module, zero\_conv) \in$ (UNet.encoder_blocks, Syncronet.zero_convs) **do**
  if $module ==$ UNet.encoder_blocks.first_block **then**
    $h \leftarrow$ module(h)
    $h \leftarrow h + guided\_hint$
    $guided\_hint \leftarrow$ **None**
  else
    if $module$ is TimeStepBlock **then**
      $h \leftarrow$ module(h, emb)
    else if $module$ is SpatialTransformer **then**
      {Apply self-attention and cross-attention}
      $h \leftarrow$ module$(h, context)$
    end if
  end if
  Append zero_conv$(h, emb, context)$ to $control\_signals$
end for
$h \leftarrow$ UNet.middle_block$(h, emb, context)$
Append zero_conv(UNet.middle_block_out$(h, emb, context)$) to $control\_signals$
return $control\_signals$

through a *linear layer*, a *zero convolution* layer is applied to get the control signal c^n.

The Stable Diffusion's UNet architecture contains 12 encoder, 12 decoder and 1 middle blocks. Similar to [13], our trainable copy contains 12 encoder and 1 middle block consisting of several Vision Transformers (ViTs). Self-attention and cross-attention is applied in all the Spatial Transformers of encoder and middle blocks as described in algorithm 1.

These control signals are added to the 12 skip-connections and 1 middle block of the Stable Diffusion's UNet decoder block providing local pixel-level guidance at $64 \times 64, 32 \times 32, 16 \times 16, 8 \times 8$ resolutions.

C. Dataset and Training

Ego4D [2] is a large scale multimodal dataset consisting of around 3600 hours of daily activity videos. However, not every video in the dataset comes with corresponding audio, due to privacy concerns or technical limitations. Only half of the dataset has the corresponding audio. Now, when we look at the dataset, it contains a large amount of person-to-person conversations in shops, homes, outdoor scenes, etc., which doesn't serve any purpose in our use case. Thus, we only selected those videos that belongs to certain categories like *cooking, carpentry, laundry, cleaning, working, farmer, mechanic, yardwork, blacksmith, etc.* We believe, only such categories are useful in learning corresponding between audio and video for day-to-day activities. Even, in these videos, not every section of the video is important as there's a lot of redundancy in the data. To overcome this limitation, we calculate the Root Mean Square (RMS) value of a 10 second clip randomly picked from the dataset as follows:

$$S_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$$

where, N is the total number of samples in audio waveform, and x is the value of each sample. After calculating the RMS value of each 10 second long audio sample, we compared it against a manually set threshold. If the sample's RMS value exceeded the threshold, we used it for training. This gave us a rich set of audio-video pair containing daily activities. We randomly picked a non-overlapping set of $150K$ such samples, which were used for training. Each 10 seconds long video is sampled at 30 frames per second and contains 300 frames in total. The corresponding audio is sampled at $22KHz$, and converted to audio spectrogram using Short-Time Fourier Transform [6]. The audio spectrogram is resized to 512×512 from 430×1024 to make it compatible with Stable Diffusion's encoder-decoder network.

To get a more useful and compact video representation, we use video embedding as the representation of videos. Video embedding is a feature rich image-like representation of the video where each video frame is represented as a vertical vector of shape 1×512. 300 such vectors are placed one after the other in the same sequence as frame number to get the video embedding of shape 300×512. Each video embedding is resized to 512×512 using bicubic interpolation to make sure it aligns with each time-step in the audio spectrogram.

Training of *Syncronet* has been done using the same loss functions as ControlNet [13]. We use DDIM [3] for faster and consistent sampling. Upto 1000 time steps were used in the forward process, and 20 during the denoising. Training was done using AdamW optimizer with a learning rate of $1e-4$.

E. Audio Super-Resolution

To upsample the generated audio spectrograms E_A from a resolution of 512×512 to 512×1024, we trained s small 5 layer Convolutional Neural Network (CNN) with

$16, 32, 64, 64, 32, 1$ output filters. Each layer is followed by ReLu activation and Batch Normalization. Sigmoid is used as the final activation to keep the values in $[0, 1]$. The model was trained using pairs of original audio spectrograms of shape 512×1024, and their down sampled version of shape 512×512 using Mean Square Error (MSE) loss. Adam optimizer was used for the optimization and a learning rate of 0.001 was used. The model was trained for 10 epochs with a dataset of $200K$ audio samples.

F. Video Summarization

Video Summarization is a long studied task which involves providing a short summary of a scene in a given video. It has been widely accepted that the audio contains rich information about the scenes and can even provide sufficient cues to reconstruct a scene geometry [7, 10]. We leverage this fact that audio provides additional cues about the scene through the easily identifiable sounds associated to improve the video summarization task. Our aim is to incorporate the audio generated from our method alongside the input video frames, and observe an improvement in the prediction accuracy of corresponding scene summary. We used a very simple setup to test this hypothesis as presented in Fig. 3. The input video frames are encoded to a rich feature space using the pretrained CLIP encoder. Initially, when the toggle switch is in "OFF" state, the Convolutional Neural Network (CNN) takes the video embedding E_V as the input, along with a zero vector as audio embedding e_A, and predicts the text embedding c'_t. The ground truth text embedding is estimated using the one sentence text summary after passing it through the CLIP text encoder. Ego4D dataset provides short narrations describing the activity of the scene. We used these narrations as the scene summary. The CNN is trained using MSE loss and the parameters are optimized using Adam optimizer with a learning rate of $5e^{-3}$.

When the toggle switch is "ON", that is, when the the zero vector e_A is replaced with the GT audio embedding, the CNN takes in this additional input and concatenates it with the video embedding vector through convolution. Similar as before, the model tried to predict the text embedding and the CNN is trained until convergence. We use a well known audio compression method EnCodec [1] to encode the audio waveform into a more rich neural codec representation.

Once the model is trained, we compared it's performance on the test dataset using various methods. The results are presented in Table **??**.

G. Video-to-Text Embedding MLP

We trained a small two layer MLP that takes a normalized video embedding E_V, and generates a vector of shape 512, which acts as a text embedding c_t to the stable diffusion model. The MLP was trained using $200K$ pair of video and text embedding using MSE loss and Adam optimizer.

H. Synchronization Metrics

An effective way of measuring the audio-video alignment is missing in the field of audio-visual learning. Thus, inspired by Diff-Foley [5], we introduced a Vision Transformer based metrics (Alignment Score) that can calculate the synchronization between audio and video. Unlike Diff-Foley, we used ViT-B32 as a feature extractor to get audio features from E_A, and video features from E_V, and then use 5 Linear layers, each followed by a ReLu activation. We uses a pre-trained ViT-B32 trained on $IMAGENET1K\_V1$. The linear layers were trained using MSE loss and Adam optimizer with a learning rate of 0.0001. Our training dataset consists of $200K$ samples, out of which $100K$ were labelled as 1 and the remaining $100K$ as 0. Audio samples belonging to the same video were all labeled as 1 and accounts for 50% of the training data. 25% data is the audio samples randomly assigned with any video other than the original video. These were labeled as 0. Remaining 25% samples came from randomly shifting the audio anywhere between $1-5$ seconds from the true audio-video pair. These were also labeled as 0. Our classifier reached an accuracy of 97% on testing dataset comprising of 20% of the training data kept separately. If the trained model classifies an audio-video pair as anything close to 1, it means the two modalities are accurately aligned. *EgoSonics* scores an average accuracy of 92% on test dataset meaning that we significantly outperforms the existing methods in better synchronizing audio.

For calculating Alignment Score (AS) at 15 FPS, we replaced the alternate image embeddings E_V^i with their previous ones E_V^{i-1}, to ensure it's consistent with the trained model. Similarily, we did for testing at 4 FPS.

H. Comparison with Baselines.

We compared our model against 3 different baselines: Diff-Foley, Im2Wav, and Make-an-Audio [4, 5, 9]). Im2Wav samples the input audio at 16 KHz and generates an audio of length 5 seconds. Make-an-Audio also samples at 16 KHz, but generates an audio of length 9-10 seconds with the intermediate spectrogram representation of 80×624. Diff-Foley also samples at 16 KHz to generate an audio of length 8 seconds via audio spectrogram of shape 256×128. On the contrary, we samples the input audio waveform at a higher 22 KHz sampling rate and also generates longer audio samples of 10 seconds. Using Stable Diffusion [8] allowed us to generate high resolution spectrograms that can fit more frequency bins and longer temporal length. For calculating the metrics of baselines, we make sure to use the same length ground truth audio wave as they generate for a fair comparison. We fine-tuned the Im2Wav, Make-an-Audio, and Diff-Foley's CAVP model for a few iterations on our

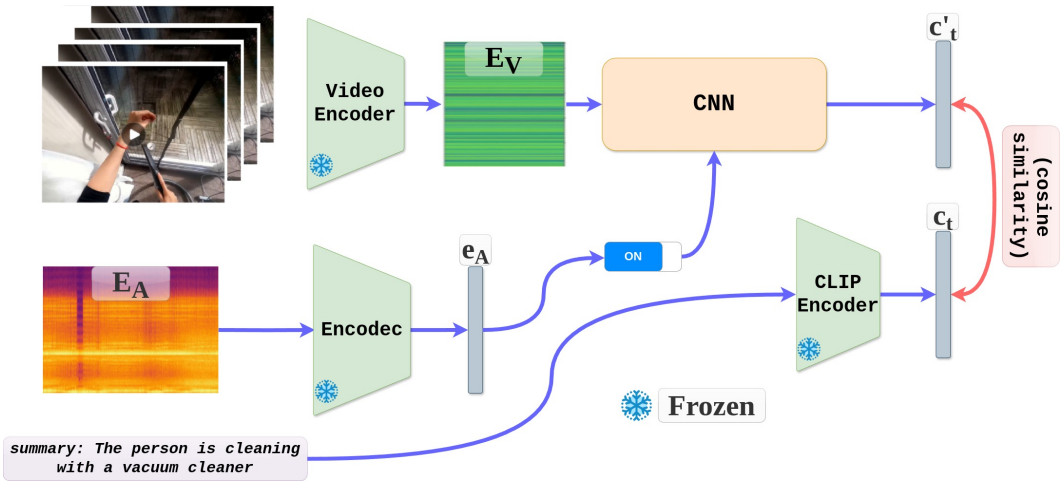

Figure 3. Video Summarization.

dataset before testing.

I. More Results

Fig. 5 shows more results generated from our model. We have used a different color scheme for spectrograms for a different perspective.

J. Failure Cases, Limitations, and Ethical Considerations

We also analyzed the failure cases. Most of the failure cases can be classifier into two categories: temporal misalignment and context-level misalignment. The temporal misalignment refers to cases where the model is able to predict contextually meaning audio, however, it's misaligned with the input video. Fig. 4(a) shows some of the misaligned results. The main factor contribution to the misalignment is the lack of rich visual information is most cases. For example, in the first case, a carpenter is polishing a steel bar with a rotating brush and the sound is made when they both are in contact. However, from the video, it's not very clear if the steel bar is actually in contact with the rotating brush or no.

The other type of failure happens when the model is not able to predict the contextually acceptable audio. This is mainly a reason of lack of data. For example, since there are a very few samples of musical instruments in the Ego4D dataset, our model doesn't perform very well on such videos. The similar thing happens if the model encounters people interacting. We believe that such challenges can be solved by training our model on a large amount of dataset comprising millions of audio-video pairs.

EgoSonics, being a generative model capable of accurately predicting audio from muted video, should be restricted to applications in research, the development of interactive AR/VR technologies, and assistive technologies for individuals with impairments. It is imperative to enforce strict adherence to ethical guidelines to prevent the misuse of this model for unethical purposes.

References

[1] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022. 4

[2] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra

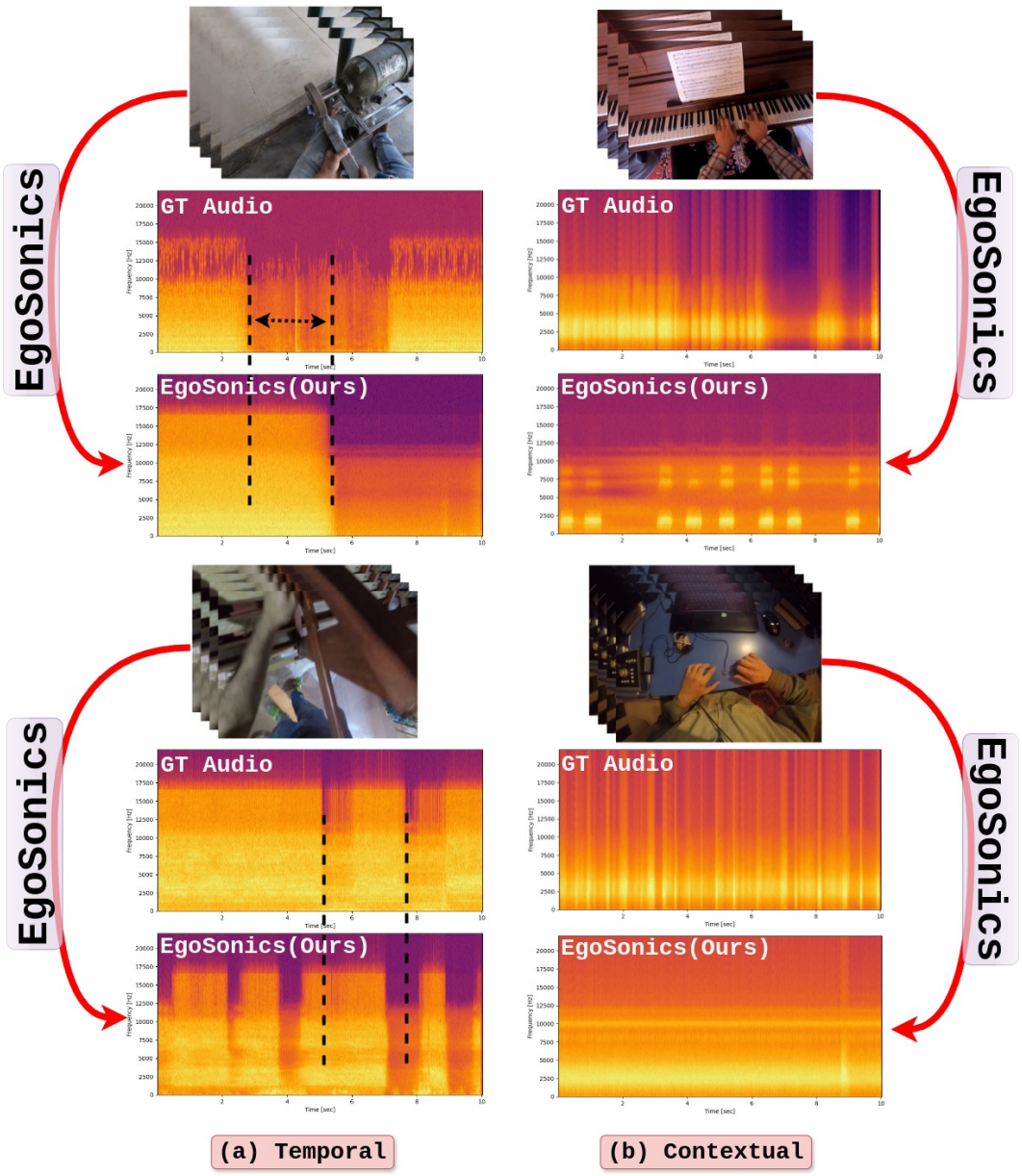

Figure 4. Failure Cases. There are two types of failure cases: (a) Temporal misalignment, (b) Contextual misalignment.

Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, 2022. 3

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[4] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *International Conference on Machine Learning*, pages 13916–13932. PMLR, 2023. 4

[5] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 4

[6] Nathanaël Perraudin, Peter Balazs, and Peter L Søndergaard. A fast griffin-lim algorithm. In *2013*

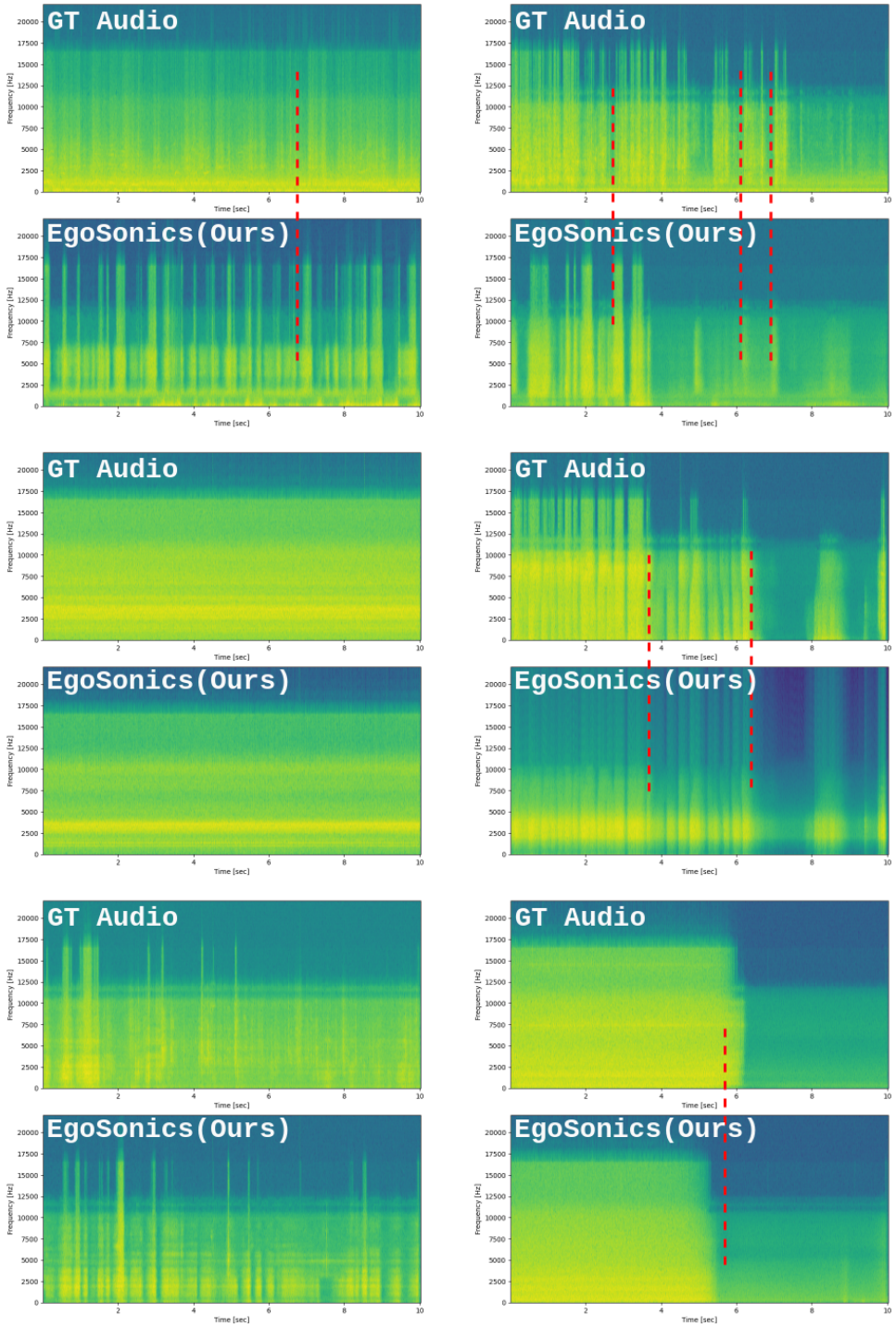

Figure 5. More Results (different color map).

IEEE workshop on applications of signal processing to audio and acoustics, pages 1–4. IEEE, 2013. 3

[7] Senthil Purushwalkam, Sebastia Vicenc Amengual Gari, Vamsi Krishna Ithapu, Carl Schissler, Philip Robinson, Abhinav Gupta, and Kristen Grauman. Audio-visual floorplan reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1183–1192, 2021. 4

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 4

[9] Roy Sheffer and Yossi Adi. I hear your true colors: Image guided audio generation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 4

[10] Arjun Somayazulu, Sagnik Majumder, Changan Chen, and Kristen Grauman. Activerir: Active audio-visual exploration for acoustic environment modeling. *arXiv preprint arXiv:2404.16216*, 2024. 4

[11] Robert C Streijl, Stefan Winkler, and David S Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227, 2016. 1

[12] Shih-Lun Wu, Chris Donahue, Shinji Watanabe, and Nicholas J Bryan. Music controlnet: Multiple time-varying controls for music generation. *arXiv preprint arXiv:2311.07069*, 2023. 2

[13] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1, 2, 3