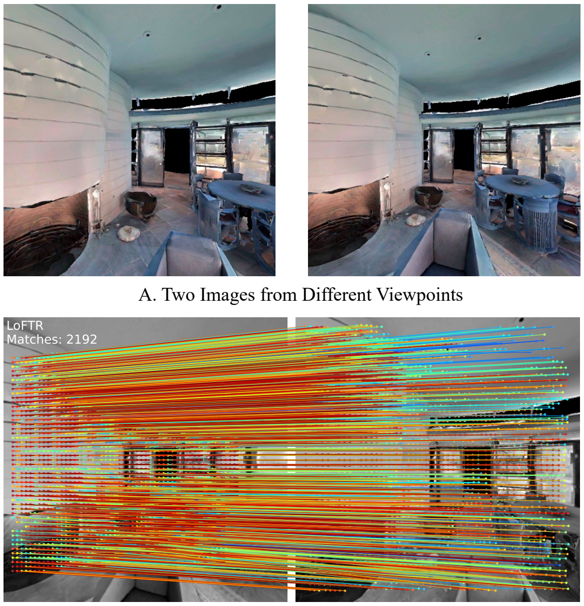
Sound3DVDet Supplementary Material

1. More Discussion on LoFTR


A. Two Images from Different Viewpoints

B. Matching Points found by LoFTR

Figure 1. **LoFTR extracted matching points visualization**. **A.** Two RGB images from different views. They contain large texture homogeneous area, such as wall and ceiling. **B.** LoFTR manages to give dense matching points even on these texture homogeneous areas. We utilize such characteristic to give robust "matchness" information to constrain the sound source to lie "on-the-surface".

In our paper, we adopt LoFTR [6] to extract RGB image feature embedding that is further used provide "matchness" information across multiview RGB images. LoFTR [6] is the model that is specifically trained for feature matching, so it is naturally suitable for our need. Benefiting from its coarse-to-fine learning strategy, LoFTR can find matching points on texture homogeneous area. We show such an example in Fig. 1, from which we can clearly see that dense matching points are generated on the wall and ceiling. It thus shows LoFTR [6] can provide dense"matchness" information across multiview RGB images, even in texture homogeneous area. This property guarantees LoFTR still provides "on-the-surface" matchness information on such texture homogeneous area. Moreover, The newly added Fully-connect layer is capable of further optimizing the LoFTR generated feature representation to better handle texture homogeneity situation.

2. Extra Experiment Advised by Reviewers

For all the newly added experiments, we train all models with the same experimental setting in the main paper. Each model is trained three times independently.

2.1. GCC-Phat with STFT and MFCC feature

In the original main paper, we concatenate 6-channel GCC-Phat with 4-channel LogMel scale spetrogram to represent one-view microphone array signal. As suggested by the reviewer, we further replace the LogMel spectrogram with STFT and MFCC respectively (but use the same GCCPhat feature). The result is given in Table 1, from which we can clearly see that using either STFT and MFCC inevitably leads to reduced performance.

Table 1. Overall quantitative result on different audio spectrogram representation versions across all object categories and sound classes.

Methods	**mAP** (↑)	**mAR** (↑)	**mALE** (↓)
Sound3DVDet (STFT)	0.277 ± 0.011	0.995 ± 0.001	0.597 ± 0.004
Sound3DVDet (MFCC)	0.261 ± 0.012	0.991 ± 0.013	0.627 ± 0.003
Sound3DVDet	**0.308** ± 0.011	**0.998** ± 0.000	**0.588** ± 0.001

2.2. Audio-only baselines

Table 2. Overall quantitative result on audio-only Sound3DVDet versions across all object categories and sound classes. We also show the three comparing methods result as all of them are audio-only based.

Methods	**mAP** (↑)	**mAR** (↑)	**mALE** (↓)
SELDNet [1]	0.101 ± 0.003	0.531 ± 0.000	0.912 ± 0.001
EIN-v2 [2]	0.111 ± 0.003	0.612 ± 0.001	0.877 ± 0.001
SoundDoA [3]	0.123 ± 0.001	0.701 ± 0.001	0.820 ± 0.003
SDVDet_mvSound	0.264 ± 0.032	0.994 ± 0.002	0.592 ± 0.008
SDVDet_oneSound	0.231 ± 0.021	0.891 ± 0.001	0.645 ± 0.014
Sound3DVDet	**0.308** ± 0.011	**0.998** ± 0.000	**0.588** ± 0.001

In the ablation studies, we reported one audio-only *Sound3DVDet* version: S3DVDet_mvSound in which we use multiview audio data (microphone array) to supervise the whole neural network. We have observed a significant performance drop. To further test how *Sound3DVDet* performs when we just involve a single view audio data to supervise the whole neural network, we report another *Sound3DVDet* version with one single view audio-only data, we call this version S3DVDet_oneSound, the result is given in Table 2, from which we can clearly observe that removing multivew RGB images supervision inevitably reduces the performance. Involving multiview audio-only supervision (S3DVDet_mvSound) outperforms just involving one view audio supervision (S3DVDet_oneSound). This shows the necessity of involving crossmodal RGB images in detecting 3D sound sources.

2.3. Validation on the Usefulness of Deep Supervision

In the main paper, we show deep supervision improves *Sound3DVDet* performance. The reason why we can use deep supervision for *Sound3DVDet* is that the sound source queries appear multiple times throughout the whole neural network. In the three compared methods, however, the queries are predicted once. This is why we do not report the comparing methods' performance with deep supervision in the main paper during the review period. In order to test the impact of deep supervision on the three comparing methods, we explicitly add two extra Transformer encoder layers (the same setting as the Transformer encoder layer in *Sound3DVDet*) on top of the three comparing methods, then we add deep supervision to supervise all queries arising from both the three comparing methods' initial predictions and two Transformer encoder layers. During test, we just use the queries given by the Transformer layer to predict sound sources. The quantitative result is given in Table 3, from which we can clearly observe that involving deep supervision strategy to three comparing methods unanimously improves their performances, respectively (an average of 3% gain in mAP, 5% gain in mAR and 6% gain in mALE). It thus shows deep supervision strategy helps learn better sound source queries representation.

Table 3. Overall quantitative result on adding deep supervision to comparing methods across all object categories and sound classes. "_DeepSup" indicates training with deep supervision.

Methods	**mAP** (↑)	**mAR** (↑)	**mALE** (↓)
SELDNet [1]	0.101 ± 0.003	0.531 ± 0.000	0.912 ± 0.001
SELDNet_DeepSup	**0.137** ± 0.001	**0.552** ± 0.010	**0.867** ± 0.002
EIN-v2 [2]	0.111 ± 0.003	0.612 ± 0.001	0.877 ± 0.001
EIN-v2_DeepSup	**0.144** ± 0.013	**0.689** ± 0.001	**0.812** ± 0.005
SoundDoA [3]	0.123 ± 0.001	0.701 ± 0.001	0.820 ± 0.003
SoundDoA_DeepSup	**0.151** ± 0.002	**0.745** ± 0.013	**0.761** ± 0.002

2.4. Comparing with more baselines

We further compare with SALSA [5] and SALSA-Lite [7]. The two works focus on sound event detection and localization (SELD) task and propose to extract log-linear spectrograms and normalized principal eigenvector to represent the multi-channel microphone array data. We will add them to the main paper. In addition to the two methods, we further compare with SoundDet [4], which proposes to use learnable filters to automatically extract feature from multi-channel sound waveforms. The quantitative result is given in Table 4. From this table, we can observe that SALSA [5] achieves the best performance among all comparing methods, SALSA-Lite [7] gives slightly inferior performance. Given the the experimental finding in Sec. 2.3, we think introducing extra deep supervision may further improve their performance, respectively.

Table 4. Overall quantitative result on more existing methods across all object categories and sound classes.

Methods	**mAP** (↑)	**mAR** (↑)	**mALE** (↓)
SALSA [5]	0.147 ± 0.002	0.722 ± 0.002	0.793 ± 0.003
SALSA-Lite [7]	0.130 ± 0.012	0.712 ± 0.003	0.810 ± 0.002
SoundDet [4]	0.120 ± 0.012	0.674 ± 0.004	0.823 ± 0.003
Sound3DVDet	**0.308** ± 0.011	**0.998** ± 0.000	**0.588** ± 0.001

2.5. Noise Discussion

Thanks for pointing out the experiment with noisy data. The reason why we did not consider the noisy situation is that we assume the environment is absent of noise pollution (as the reviewer noted that the sound source is anechoic). In reality, there are many ways that environmental noise affects the data. Here we consider a simple case in which we add a Gaussian noise (with mean 0 and standard deviation σ, $\mathcal{N}(0, \sigma)$) to the microphone array multi-channel raw waveforms (normalized into $[-1., 1.]$). We test two noise levels with standard deviation $\sigma = 0.05$ and $\sigma = 0.10$ respectively. The quantitative result of *Sound3DVDet* on such noise data is given in Table 5, from which we can see that noise polluted sound data inevitably reduces the performance. We think extra process to explicitly suppress the noise is required to minimize noise impact, which remain as a future research direction.

Table 5. Overall quantitative result on noise test across all object categories and sound classes. We omit the standard deviation report for succinct report.

Methods	Noise	**mAP** (↑)	**mAR** (↑)	**mALE** (↓)
Sound3DVDet	$\sigma = 0.10$	0.270	0.994	0.670
Sound3DVDet	$\sigma = 0.05$	0.281	0.995	0.645
Sound3DVDet	$\sigma = 0.0$	**0.308**	**0.998**	**0.588**

2.6. Model performance with increasing number of sources

In the dataset we have collected, we report result on the whole dataset where the number of sound sources ranges from 1 to 5. To further figure out our model's performance w.r.t. to varying number of sources, we re-compute the evaluation metrics based on sound source number. It is worth noting that the sound source class is independently chosen from the five sound classes corpus. The result is shown in Table 6, from which we can observe that our proposed framework does not exhibit obvious performance difference for different sound source number in the environment when the sound source number is not larger than 5. One future work remains to be done is to involve more sound source number in the environment. We cannot finish this test within the rebuttal period due to the time limit.

Table 6. Overall quantitative result on sound source number across all object categories and sound classes of our proposed *Sound3DVDet*.

Source Num.	**mAP** (↑)	**mAR** (↑)	**mALE** (↓)
1	0.307	0.997	0.587
2	0.310	0.998	0.588
3	0.312	0.997	0.587
4	0.309	0.997	0.589
5	0.310	0.998	0.588
Overall	**0.308**	**0.998**	**0.588**

3. More Discussion on Data Creation

We provide the statistics of our created dataset in Table 9 w.r.t. different physical object class. In this table, we can observe that the "wall" and "ceiling" consist of the largest portion of the dataset, which reflect the real scenario. We split whole 6.2K dataset into 5.0K/1.2K for train and test respectively.

It is worth noting that, although we lay the sound source on specific physical object surface in this work, the sound sources can lie on arbitrary physical surface. In another word, the sound source placement is independent of physical objects. In the main paper, we just provided four typical data samples. In order to give readers a more direct and intuitive understanding how the data and task look like, we provide more data samples visualization in Fig. 3.

4. More Details on Train and Test

During train, we randomly select a reference view among multiview inputs. The initial queries from multiview are optimized, and the corresponding sound source queries in the reference view is further optimized by passing through the detection backbone neural network ($\boldsymbol{\mathcal{B}}$). Since we randomly select the reference view for each iteration, every single view is guaranteed to be sampled as reference view during the multiple iterations training process.

During test, given a multiview input, we iterate over each single view and treat it as reference view to do the inference. Each inference is independent because we do the set-based prediction. In another word, we do `view-num` independent predictions for one multiview input.

5. Sound3DVDet Neural Network

5.1. Sound Source Query Generator Network

Sound source query generator $\mathcal{G}_{\text{mic}}$ takes as input 10 channel 2D feature map (of size $[10 \times 256 \times 256]$) that is originally constructed by one 4-channel microphone array input. It jointly outputs a 2D microphone array embedding feature $[512 \times 16 \times 16]$ and initial sound source queries $[16 \times 512]$. In our implementation, we adopt a sequence of `stride=2` 2D convolutions to sequentially reduce the feature map spatial resolution and accordingly increase the feature size (in channel dimension). Each 2D convolution is `2DConv+BatchNorm+ReLu` operation combination. In total, we build 8 of these 2D convolutions. While the output of the last layer is used as sound source queries, the output of the penultimate layer is used as sound microphone array embedding. The network architecture is given in Table 10 and we also provide the source code in the supplementary material.

5.2. Transformer-based Detection Backbone

For the Transformer-based detection backbone $\mathcal{B}$, we adopt the standard Transformer encoder network which consists of multi-head self-attention (MHSA) and feed forward network (FFN). In our implementation, we stack 6 such Transformer encoder layers. The detection backbone's hyperparameter selection is given in Table 11.

5.3. Detection Head

Sound3DVDet detection head $\boldsymbol{\mathcal{H}}$ takes as input the query embedding to jointly predicts the the query 3D spatial position $[x, y, z]$ and class label c. In our implementation, we adopt two parallel multi-layer perceptron (MLP). The network architecture detail is given in Table 12.

6. More Experiment Result

6.1. More Quantitative Result

The detailed quantitative experimental result w.r.t. sound source class is given in Table 7. From this table, we can see that our proposed *Sound3DVDet* stays as the best-performing method among all comparing methods and other *Sound3DVDet* variants for five out of six sound sources class in terms of average precision (AP), six out of six for average recall (AR). It thus shows our proposed *Sound3DVDet* is suitable for 3D sound source detection task, it is capable of handling various sound source classes.

The detailed quantitative experimental result w.r.t. physical object class is given in Table 8. We can observe from this table that our proposed *Sound3DVDet* outperforms all other *Sound3DVDet* variants across all physical object classes, in terms of both mAP and mAR evaluation metrics. It thus

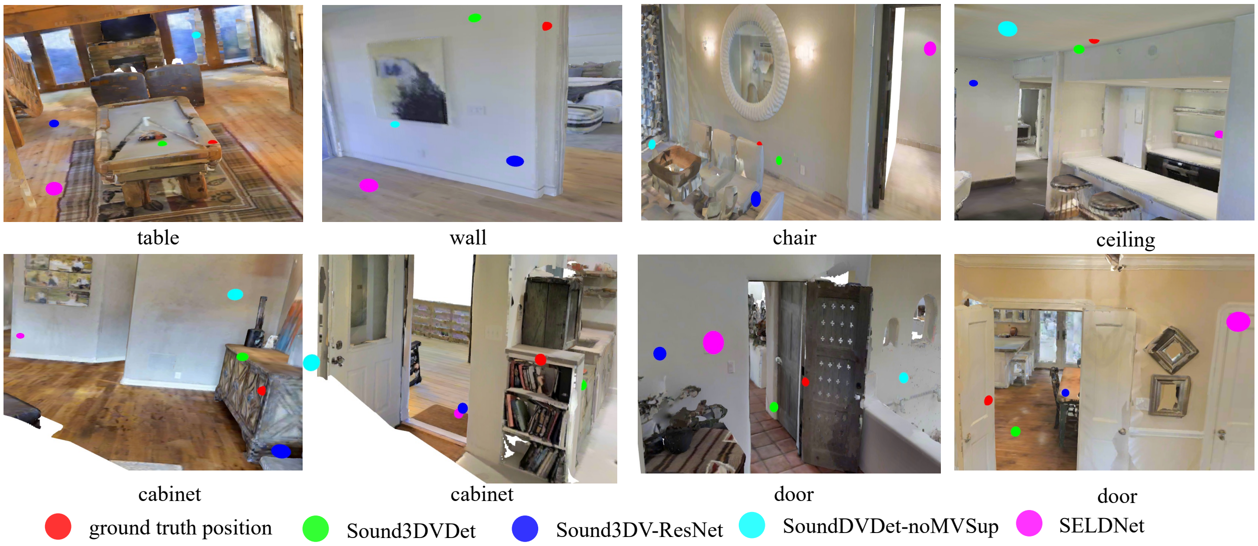


Figure 2. Qualitative Detection Result Visualization: We visualize the position of one detected sound source position by different methods as well as its ground truth position. We recommend to zoom in for better visualization.

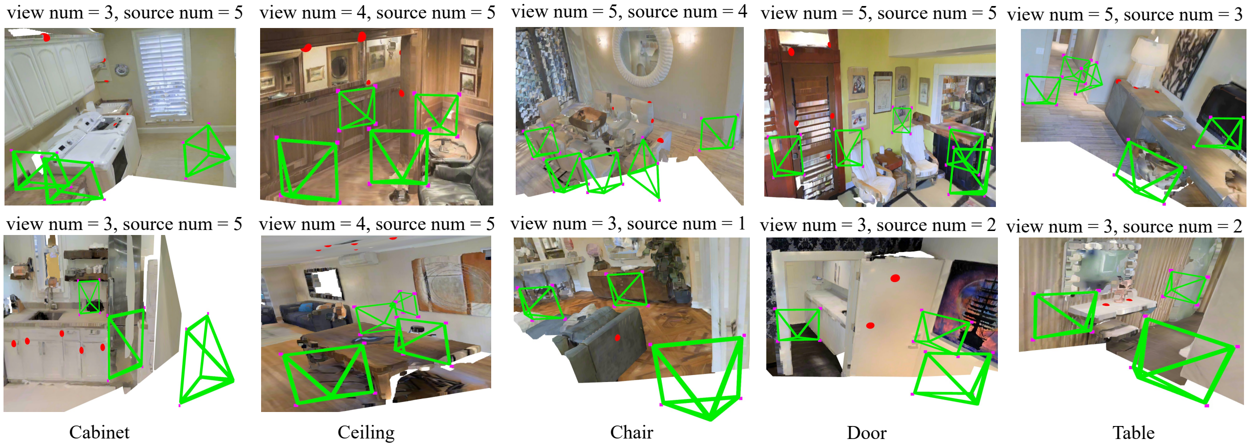

view num = 3, source num = 5
view num = 4, source num = 5
view num = 5, source num = 4
view num = 5, source num = 5
view num = 5, source num = 3
view num = 3, source num = 5
view num = 4, source num = 5
view num = 3, source num = 1
view num = 3, source num = 2
view num = 3, source num = 2
Cabinet
Ceiling
Chair
Door
Table

Figure 3. **More Data Sample Visualization**: Multiple 3D sound sources (red ball) are emitted by visually uninformative objects, we use an acoustic-camera device to record the multi-view, multi-modal visual-acoustic scene. Each recording consists of an RGB image at a known pose (green) and a four-channel microphone array (magenta). The number of sound sources and their classes are arbitrary. The sound sources arbitrarily lie on texture homogeneous (top row) or discriminative regions (bottom row).

shows *Sound3DVDet* is robust to the physical surface where the sound sources may lie on.

6.2. More Qualitative Result

We provide more qualitative result visualization in Fig. 2. From this figure, we can clearly see that *Sound3DVDet* is capable of accuractely detect 3D sound sources under various room scenarios. It is better at handling both texture-homogeneous and texture-discriminative situation.

Table 7. Quantitative result w.r.t. each sound source classes.

Methods	Telephone			Siren			Alarm			Fireplace			Horn-beeps			Overall		
	AP	AR	ALE	AP	AR	ALE	AP	AR	ALE	AP	AR	ALE	AP	AR	ALE	**mAP**	**mAR**	**mALE**
SELDNet [1]	0.091	0.526	0.915	0.092	0.530	0.914	0.103	0.527	0.914	0.109	0.537	0.908	0.108	0.536	0.909	0.101	0.531	0.912
EIN-v2 [2]	0.096	0.542	0.882	0.095	0.543	0.881	0.105	0.544	0.879	0.117	0.554	0.873	0.119	0.551	0.870	0.111	0.612	0.877
SoundDoA [3]	0.095	0.646	0.827	0.094	0.660	0.825	0.108	0.700	0.817	0.112	0.740	0.816	0.112	0.752	0.816	0.123	0.701	0.820
S3DVDet_ResNet	0.221	0.995	**0.588**	0.304	0.969	**0.553**	0.074	0.927	0.592	0.290	0.998	**0.584**	0.292	0.997	0.587	0.236	0.977	**0.581**
S3DVDet_mvSound	0.266	0.995	0.595	0.227	0.993	0.602	0.319	0.998	**0.570**	0.180	0.991	0.620	**0.330**	0.996	**0.570**	0.264	0.995	0.592
S3DVDet_noDeepS	0.095	0.984	0.648	0.109	0.993	0.633	0.227	0.996	0.594	0.268	0.998	0.597	0.137	0.996	0.608	0.167	0.994	0.616
S3DVDet_wMVIS	0.254	0.995	0.598	0.270	0.996	0.598	0.301	0.997	0.591	0.304	0.998	0.592	0.316	**0.997**	0.595	0.289	0.997	0.595
Sound3DVDet	**0.308**	**0.999**	0.603	**0.320**	**0.998**	0.579	**0.332**	**0.999**	0.581	**0.322**	**0.999**	0.587	0.222	**0.997**	0.588	**0.301**	**0.998**	0.588

Table 8. Quantitative result w.r.t. each physical object class.

Methods	Table			Ceiling			Door			Chair			Wall			Cabinet		
	mAP	mAR	mALE	mAP	mAR	mALE	mAP	mAR	mALE	mAP	mAR	mALE	mAP	mAR	mALE	mAP	mAR	mALE
S3DVDet_ResNet	0.198	0.818	0.585	0.168	0.733	**0.475**	0.298	0.917	**0.556**	0.227	0.742	**0.487**	0.259	0.925	**0.571**	0.227	0.780	**0.538**
S3DVDet_mvSound	0.253	0.944	0.609	0.233	0.975	0.607	0.295	0.993	0.583	0.222	0.882	0.553	0.270	0.984	0.596	0.276	0.958	0.584
S3DVDet_noDeepS	0.200	0.939	**0.573**	0.166	0.957	0.628	0.170	0.961	0.608	0.193	0.866	0.557	0.144	0.969	0.625	0.184	0.924	0.614
S3DVDet_wMVIS	0.253	0.965	0.631	0.278	0.982	0.603	0.339	0.985	0.584	**0.237**	0.918	0.580	0.289	0.989	0.590	0.295	0.952	0.595
Sound3DVDet	**0.267**	**0.977**	0.616	**0.304**	**0.988**	0.601	**0.348**	**0.991**	0.578	**0.222**	**0.924**	0.611	**0.293**	**0.992**	0.582	**0.300**	**0.974**	0.583

Table 9. Created Multiview Microphone Array and RGB Images Dataset Summary w.r.t. each Physical Object Category.

Object	Texture-homo	Texture-disc.	source num.	view num
wall	975	717	1-5	4
ceiling	727	614	1-5	4
table	464	461	1-5	4
door	712	702	1-5	4
cabinet	286	292	1-5	4
chair	100	222	1-5	4
sum	3264	3008	/	/

Table 10. Sound Source Query Generator $\mathcal{G}_{\mathrm{mic}}$ network illustration. 2D convolution kernel size is 3×3 and the stride is 2.

in-channel num.	out-channel num.	feature size
10	32	$[32, 256, 256]$
32	64	$[64, 128, 128]$
64	128	$[128, 64, 64]$
128	256	$[256, 32, 32]$
256	256	$[256, 16, 16]$
MicArray Embed: Linear (256, 512): $[512, 16, 16]$		
256	512	$[512, 8, 8]$
Avg Pooling: $[512, 4, 4]$		
Query Representation: Reshape: $[16, 512]$		

Table 11. Sound3DVDet detection backbone $\mathcal{B}$ network illustration.

Transformer Encoder Layer Num.	6
Head Num.	8
Token (query) Num.	16
FFN dim.	1024

Table 12. Sound3DVDet detection head $\mathcal{H}$ network illustration.

Layer Name	input dim	output dim	output
Input: Queries $[16, 512]$			
Position Regression Head			
Linear + BN + ReLU	512	256	$[16, 256]$
Linear	256	3	$[16, 3]$
Classification Head			
Linear	512	256	$[16, \text{classnum}]$

References

[1] Sharath Adavanne, Pasi Pertilä, and Tuomas Virtanen. Sound event detection using spatial features and convolutional recurrent neural network. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. 1, 2, 5

[2] Yin Cao, Turab Iqbal, Qiuqiang Kong, Fengyan An, Wenwu Wang, and Mark D Plumbley. An Improved Event-Independent Network for Polyphonic Sound Event Localization and Detection. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021. 1, 2, 5

[3] Yuhang He and Andrew Markham. SoundDoA: Learn Sound Source Direction of Arrival and Semantics from Sound Raw Waveforms. In *Interspeech*, 2022. 1, 2, 5

[4] Yuhang He, Niki Trigoni, and Andrew Markham. SoundDet: Polyphonic Moving Sound Event Detection and Localization from Raw Waveform. In *International Conference on Machine Learning (ICML)*, 2021. 2

[5] Thi Ngoc Tho Nguyen, Karn N. Watcharasupat, Ngoc Khanh Nguyen, Douglas L. Jones, and Woon-Seng Gan. SALSA: Spatial Cue-Augmented Log-Spectrogram Features for Polyphonic Sound Event Localization and Detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022. 2

[6] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching with Transformers. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[7] Thi Ngoc Tho Nguyen, Douglas L. Jones, Karn N. Watcharasupat, Huy Phan, and Woon-Seng Gan. SALSA-Lite: A Fast and Effective Feature for Polyphonic Sound Event Localization and Detection with Microphone Arrays. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022. 2