

---

# Supplementary Materials & Appendix

---

Anonymous Author(s)

Affiliation

Address

email

1 These supplementary materials accompany submission #5331: “**Model-Guided Dual-Role Alignment for High-Fidelity Open-Domain Video-to-Audio Generation**.” We provide additional details  
2 on the experimental setup, extended results supporting the main paper, and a deeper analysis of the  
3 proposed MGAudio model.  
4

## 5 A Model Guidance: Derivation and Integration

6 In our framework, MGAudio, we incorporate the *Model Guidance* (MG) strategy proposed by [1] as  
7 a training-time objective that complements the conventional Classifier-Free Guidance (CFG). While  
8 CFG modifies the inference trajectory by combining outputs from conditional and unconditional  
9 branches, MG introduces an auxiliary loss that explicitly accounts for the posterior dependency  
10 between conditions and noisy inputs. This enables improved condition alignment without incurring  
11 additional inference-time overhead.

### 12 A.1 Model Guidance for Flow Matching

13 Our model is based on the flow-matching mechanism [2] for the denoising process, main backbone  
14 being SiT [3]. Flow-Matching aims to learn a conditional velocity field  $u_\theta(\mathbf{x}_t, t, \vec{v})$  given condition  $\vec{v}$   
15 (condense feature vector extracted from the silent video), such that it matches the ground-truth flow:

$$u_t(\mathbf{x}_t | \mathbf{x}_0) = \frac{d\mathbf{x}_t}{dt} = \mathbf{x}_0 - \epsilon, \quad (1)$$

16 where  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$  and  $\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\epsilon$ . The standard flow-matching loss is:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\mathbf{x}_0, \epsilon, t, \vec{v}} \|u_\theta(\mathbf{x}_t, t, \vec{v}) - u_t(\mathbf{x}_t | \mathbf{x}_0)\|_2^2. \quad (2)$$

17 While Eq. 2 learns to model the conditional distribution  $p_\theta(\mathbf{x}_t | \vec{v})$ , diffusion models often underutilize  
18 the conditioning information in practice. To address this, MG proposes to include the posterior term  
19  $p_\theta(\vec{v} | \mathbf{x}_t)$ , resulting in the joint distribution:

$$\tilde{p}_\theta(\mathbf{x}_t | \vec{v}) = p_\theta(\mathbf{x}_t | \vec{v}) \cdot p_\theta(\vec{v} | \mathbf{x}_t)^w, \quad (3)$$

20 where  $w$  is the guidance scale controlling the strength of the posterior term. The score of this joint  
21 distribution becomes:

$$\nabla_{\mathbf{x}_t} \log \tilde{p}_\theta(\mathbf{x}_t | \vec{v}) = \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t | \vec{v}) + w \cdot \nabla_{\mathbf{x}_t} \log p_\theta(\vec{v} | \mathbf{x}_t). \quad (4)$$

22 Using Bayes’ rule, we express the posterior as:

$$\log p_\theta(\vec{v} | \mathbf{x}_t) \propto \log p_\theta(\mathbf{x}_t | \vec{v}) - \log p_\theta(\mathbf{x}_t), \quad (5)$$

23 which leads to the posterior gradient:

$$\nabla_{\mathbf{x}_t} \log p_\theta(\vec{v} | \mathbf{x}_t) \propto u_\theta(\mathbf{x}_t, t, \emptyset) - u_\theta(\mathbf{x}_t, t, \vec{v}), \quad (6)$$

24 where  $u_\theta(\mathbf{x}_t, t, \emptyset)$  denotes the velocity predicted without conditioning.

25 This gives rise to the modified target velocity:

$$\mathbf{u}' = \mathbf{u} + w \cdot \text{sg} \left( u_{\theta}(\mathbf{x}_t, t, \vec{v}) - u_{\theta}(\mathbf{x}_t, t, \emptyset) \right), \quad (7)$$

26 where  $\mathbf{u} = \mathbf{x}_0 - \epsilon$  is the ground-truth velocity and  $\text{sg}(\cdot)$  denotes the stop-gradient operation, used to  
27 stabilize training. The final Model Guidance loss is:

$$\mathcal{L}_{\text{MG}} = \mathbb{E}_{\mathbf{x}_0, \epsilon, t, \vec{v}} \|u_{\theta}(\mathbf{x}_t, t, \vec{v}) - \mathbf{u}'\|_2^2. \quad (8)$$

Table 1: **MGAudio on VGGSound**. Audio generation quality across state-of-the-art methods on the VGGSound test set is presented. Bold indicates the best performance, and an underline denotes the second-best. CFG: *Classifier-Free Guidance*, MG: *Model-Guidance*. \*We train MMAudio from scratch solely on the VGGSound dataset, and set the text input to *None* during inference for fairness.

Method	Train Type	Inference Type	FAD↓	FD↓	IS↑	KL↓	Align. Acc.↑
Diff-Foley [NeurIPS'23] [4]	CFG	CFG	6.25	23.07	10.85	3.18	93.94
See and Hear [CVPR'24] [5]	CFG	CFG	5.51	26.60	5.47	2.81	58.14
FRIEREN [NeurIPS'24] [6]	CFG	CFG	1.38	12.36	<u>12.12</u>	<u>2.73</u>	97.25
MDSGen [ICLR'25] [7]	CFG	CFG	1.40	17.42	9.66	2.84	96.88
MMAudio [CVPR'25]* [8]	CFG	CFG	<u>0.71</u>	<u>6.97</u>	11.09	<b>2.07</b>	92.28
<b>MGAudio</b>	<b>MG</b>	<b>No CFG</b>	0.80	7.89	9.90	2.77	90.80
<b>MGAudio</b>	<b>MG</b>	<b>CFG</b>	<b>0.40</b>	<b>6.16</b>	<b>12.82</b>	2.76	95.65

## 28 A.2 Inference-Time Integration of CFG into MG-Trained Models

29 At inference, we integrate classifier-free guidance (CFG) to explicitly control generation fidelity.  
30 Although the MG loss is utilized solely during training to enhance model sensitivity to conditioning  
31 signals, employing CFG at inference further boosts fidelity without additional retraining. Empirical  
32 results demonstrate that combining MG with CFG achieves superior perceptual quality and alignment,  
33 effectively balancing diversity and fidelity across key metrics such as FAD.

34 Tab. 1 shows that our best MGAudio model (131M parameters), trained for 1.1M steps without  
35 CFG, achieves a competitive FAD score of 0.80 and alignment accuracy of 90.80%, outperforming  
36 strong baselines like MDSGen (131M parameters) and FRIEREN (157M parameters), and closely  
37 approaching MMAudio (157M parameters), all utilizing CFG at inference. Notably, applying CFG  
38 significantly enhances our model’s performance, achieving state-of-the-art results with an FAD of  
39 0.40 and alignment accuracy of 95.65%, underscoring CFG’s critical role in improving alignment  
40 quality.

## 41 B More Training Details

42 **Training Configurations.** We adopt the same model architecture and hyperparameters as SiT [3].  
43 For all experiments, the guidance scale factor in Eq. 6 of the main paper is set to  $w = 1.45$ , following  
44 the default in [1]. For the initial 10,000 training steps, we set  $w = 0$  to stabilize early training. All  
45 models are optimized using AdamW [9] with a weight decay of 0 and betas (0.9, 0.999). We use  
46 a constant learning rate of  $1 \times 10^{-4}$ , a batch size of 64, and train for 1.1 million steps in the main  
47 experiments (Table 1 and Figure 5). Data augmentation follows the same protocol as Diff-Foley [4].  
48 We do not tune learning rates, apply warm-up or decay schedules, modify AdamW parameters, add  
49 extra augmentations, or apply gradient clipping. For ablation studies in Sections 4.3.1–4.3.5, models  
50 are trained for 300,000 steps with a batch size of 16, while keeping all other settings unchanged.

51 **Sampling Configurations.** We utilize an exponential moving average (EMA) of model weights with  
52 a decay factor of 0.9999 and employ EMA checkpoints for all sampling, which consistently achieving  
53 improved performance. By default, sampling is performed using the Euler-Maruyama solver with  
54 50 denoising steps and a classifier-free guidance (CFG) value of 1.45. For the comparative analysis  
55 presented in Figure 5 of the main paper, we adopt the Euler solver with a reduced step count of 25,  
56 aligning our methodology with that of FRIEREN [6] to ensure a fair and accurate comparison.

57 **Metric Calculation** We utilize audio evaluation tools provided by AudioLDM [10] for FAD, FD, IS,  
58 and KL. For alignment accuracy, we use the code provided by Diff-Foley [4].

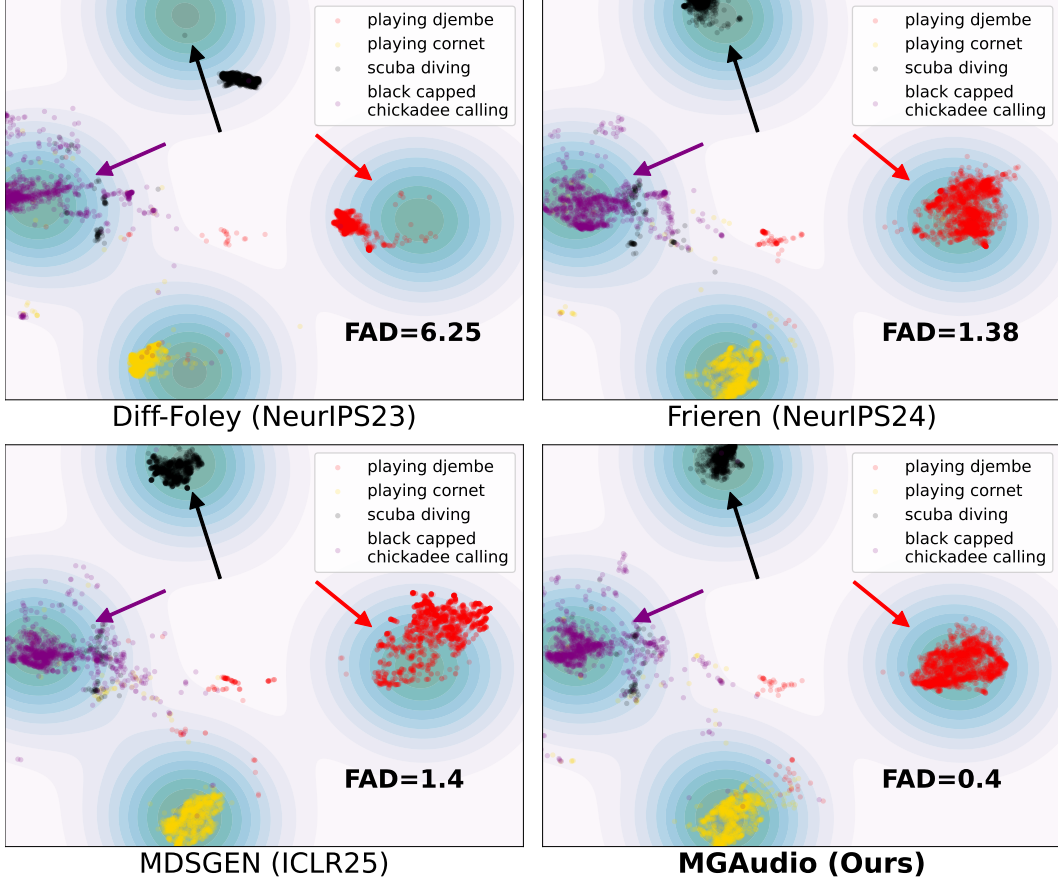


Figure 1: **Audio Distribution.** MGAudio generates audio samples that *more closely align* with the target distribution compared to other methods. For example, samples for classes “playing djembe” (red points) and “scuba diving” (black points) are tightly clustered around the center of the real sample distribution.

59 **Architectural Configurations.** We adopt the same transformer architecture as SiT [3], experimenting  
60 with four model scales: MGAudio-S, B, L, XL, varying in parameter count and computational cost. A detailed summary is provided in Tab. 2.

Table 2: **Model Configurations of MGAudio.** Four model sizes are used, following the setup in SiT [3].

Model	Layer N	Hidden dimension $d$	Head	Patch size	# Parameters (M)
MGAudio-S	12	384	6	2	34
MGAudio-B	12	768	12	2	131
MGAudio-L	24	1024	16	2	464
MGAudio-XL	28	1152	16	2	680

61

## 62 C Learned Data Distribution Comparison

63 We present a higher-resolution comparison of the generated and real audio sample distributions in  
64 Fig. 1. For each audio sample, both generated and real, we first extract sample-wise logits using  
65 the PANNs model [11]. These logits are then projected into a 2D space using UMAP [12] for  
66 visualization. The resulting embeddings are used to plot the distribution of generated samples, along

with contour plots of the real samples. The filled contours in Fig. 1 represent the density of real audio samples from VGGSound, focusing on four selected classes: *playing djembe*, *playing cornet*, *scuba diving*, and *black-capped chickadee calling*.

Table 3: Impact of Batch Size on MGAudio Performance Metrics

Batch Size	FAD↓	FD↓	IS↑	KL↓	Align. Acc.
16	1.14	13.09	8.35	<b>2.70</b>	93.67
32	0.71	12.07	8.61	2.72	94.41
<b>64</b>	<b>0.51</b>	<b>9.19</b>	<b>10.25</b>	2.75	<b>95.14</b>

## D Effect of Batch Size

We analyze the influence of varying batch sizes on the stability and quality of MGAudio training. As shown in Tab. 3, larger batch sizes generally improve key performance metrics, particularly FAD, FD, IS, and alignment accuracy, suggesting enhanced audio fidelity, diversity, and alignment quality.

For our main experiments, we select a batch size of 64, which yields the best overall performance metrics across all evaluated criteria. For computational efficiency in our extensive ablation studies, we opt for a smaller batch size of 16, which provides reasonable performance metrics despite some degradation, enabling faster experimentation cycles.

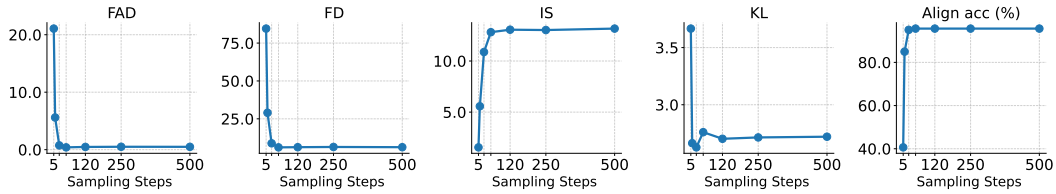


Figure 2: Performance Metrics of MGAudio Across Different Sampling Steps.

## E Effect of Sampling Step

We evaluate the performance of MGAudio under different sampling step budgets: 5, 25, 50, 120, 250, and 500 steps. As shown in Fig. 2, MGAudio achieves optimal performance for major metrics such as FAD, FD, and KL at 25-50 sampling steps. This indicates that further increasing sampling steps does not necessarily enhance results and may incur unnecessary computational overhead.

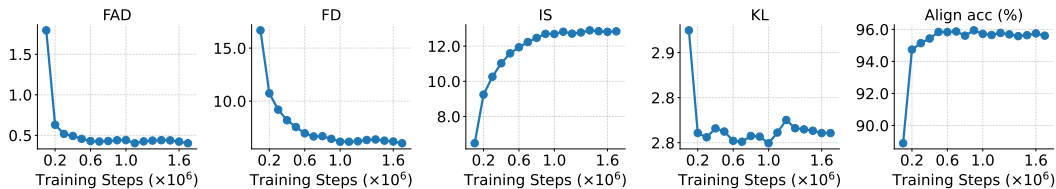


Figure 3: **Longer Training Effect.** We plot the evolution of five evaluation metrics, FAD, FD, IS, KL divergence, and alignment accuracy, over training steps. MGAudio exhibits consistent improvements across all metrics during the first 1M steps, after which most metrics begin to saturate. Notably, FAD and FD steadily decrease, and IS and alignment accuracy improve, indicating that both perceptual quality and semantic consistency benefit from prolonged training.

## F Effect of Longer Training

To study the effect of extended training, we continue training MGAudio up to 1.7M steps. As shown in Fig. 3, the model continues to improve steadily with longer training, with performance metrics



86 stabilizing beyond 1.1M steps. We select the 1.1 M-step checkpoint as our final model to balance  
87 training efficiency and generation quality.

88 Interestingly, prior work based on diffusion models such as MDSGen [7] reports that longer training  
89 can lead to degraded performance, likely due to overfitting or mode collapse. In contrast, our method  
90 maintains or improves quality throughout, highlighting its robustness and the effectiveness of the  
91 model-guided training objective with flow-matching learning.

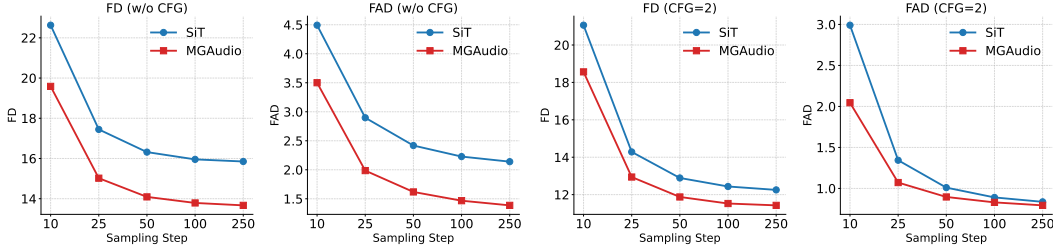


Figure 4: Comparison of FD and FAD metrics for SiT and MGAudio models, evaluated with and without CFG across different sampling steps.

## 92 G Model-Guidance vs. CFG baseline SiT

93 To further demonstrate the effectiveness of training with Model Guidance (MG), we conduct a  
94 controlled comparison between MGAudio and the CFG-only baseline SiT. Both models are trained  
95 for 300,000 iterations with a batch size of 32 under identical conditions, including model architecture,  
96 learning rate, and optimizer.

97 Figure 4 shows the FD and FAD results for each model under different sampling step settings, with  
98 and without CFG applied during inference. MGAudio consistently outperforms the SiT baseline in  
99 both scenarios. Notably, even without CFG, which reduces inference cost by requiring only a single  
100 model pass, MGAudio still achieves superior performance. These results underscore the robustness  
101 and efficiency gains of training with MG, validating its advantages over conventional CFG-only  
102 training approaches.

Table 4: Performance Comparison between ODE Euler and SDE Euler-Maruyama Samplers.

Sampler	Type	FAD↓	FD↓	IS↑	KL↓	Align. Acc. (%)
Euler	ODE	0.58	6.71	12.47	<b>2.69</b>	<b>95.98</b>
Euler-Maruyama	SDE	<b>0.40</b>	<b>6.16</b>	<b>12.82</b>	2.76	95.65

## 103 H Sampler Investigation

104 We investigate two widely-used sampler families: deterministic ODE solvers (Euler integrator) and  
105 stochastic SDE samplers (Euler-Maruyama integrator). Following prior works [3, 1], both samplers  
106 are evaluated using a consistent 50-step sampling scheme. As presented in Tab. 4, the SDE-based  
107 Euler-Maruyama sampler achieves better overall performance, indicating superior audio fidelity and  
108 diversity compared to the deterministic Euler sampler. Thus, for our main experiments, we utilize the  
109 Euler-Maruyama integrator due to its optimal balance between realism and variation.

## 110 I Training Time Analysis

111 Tab. 5 details the training durations for various MGAudio model sizes (small to extra-large). Training  
112 was conducted on a single A6000 GPU using mixed-precision optimization for 300,000 iterations with  
113 a batch size of 16. We select MGAudio-B as our primary model due to its balanced trade-off between  
114 performance and computational efficiency. The results highlight the increasing computational  
115 requirements associated with larger model architectures, illustrating practical considerations in  
116 choosing model complexity.

Table 5: **Training Time for Different MGAudio Model Sizes.** Wall-clock training durations on a single A6000 GPU for 300,000 iterations using mixed-precision training and a batch size of 16.

Model	Training Time (hours)
MGAudio-S	16.6
MGAudio-B	25.7
MGAudio-L	57.0
MGAudio-XL	77.1

Table 6: **Metrics from MMAudio [8]. MGAudio on VGGSound.** Audio generation quality across state-of-the-art methods on the VGGSound test set is presented. Bold indicates the best performance, and an underline denotes the second-best. \*We train MMAudio from scratch solely on the VGGSound dataset, and set the text input to *None* during inference for fairness.

Method	FD↓			KL↓		IS↑	IB-Score↑ (s)	# DeSync↓	# Params↓
	VGG	PANNs	PaSST	PANNs	PaSST	PANNs			
Diff-Foley [NeurIPS’23] [4]	6.25	23.07	358.92	3.17	3.04	10.77	19.88	0.91	860M
See and Hear [CVPR’24] [5]	5.51	26.60	227.67	2.82	2.78	5.71	<b>36.11</b>	1.20	1099M
FRIEREN [NeurIPS’24] [6]	1.38	12.36	107.57	2.72	2.84	12.12	22.83	0.85	157M
MDSGen [ICLR’25] [7]	1.40	17.42	114.27	2.83	2.80	9.68	22.53	1.23	<b>131M</b>
MMAudio [CVPR’25]* [8]	<u>0.71</u>	<u>6.97</u>	<b>51.00</b>	<b>2.08</b>	<b>1.97</b>	11.08	27.35	<b>0.50</b>	157M
<b>MGAudio</b> , CFG = 1.45	<b>0.40</b>	<b>6.16</b>	83.53	2.75	<u>2.56</u>	12.80	26.53	1.22	<b>131M</b>
<b>MGAudio</b> , CFG = 4.0	2.23	13.65	77.41	2.75	2.62	<u>14.41</u>	27.13	1.23	<b>131M</b>
<b>MGAudio</b> , CFG = 6.0	1.50	9.10	<u>69.75</u>	2.81	2.62	<b>17.36</b>	<u>28.77</u>	1.21	<b>131M</b>

## J Additional Metrics Comparison

To comprehensively evaluate audio quality beyond standard metrics such as FAD and FD, we follow [8] and include additional metrics shown in Table 6. These metrics encompass Fréchet Distance (FD) computed using multiple audio models (VGG [13], PANNs [11], PaSST [14]), and Kullback-Leibler divergence (KL) evaluated using PANNs and PaSST embeddings for distribution matching between generated audio and ground truth audio. And Inception Score (IS) [15] to assess audio quality. Semantic alignment is evaluated using IB-Score, where ImageBind [16] extracts visual and audio features, computing average cosine similarity as in [17]. Audio-visual synchrony is measured via DeSync scores using Synchformer to estimate temporal alignment errors.

As detailed in Table 6, MGAudio consistently outperforms competing methods in key distribution metrics (FD-VGG and FD-PANNs) and achieves the highest audio quality according to IS with a CFG of 6.0. Additionally, despite MMAudio achieving the lowest DeSync, MGAudio demonstrates a strong semantic alignment (IB-Score), particularly at higher CFG values, without the need for explicit test-time optimization as done by See and Hear. Furthermore, MGAudio accomplishes these superior results with fewer parameters (131M) compared to MMAudio (157M), underscoring its efficiency and effectiveness.

## K Qualitative Comparisons

We present qualitative comparisons between MGAudio and prior state-of-the-art methods on randomly selected videos from the VGGSound test set. Each example shows input video frames followed by the mel-spectrograms generated by different models. As illustrated in Fig. 5–8, MGAudio produces spectrograms with more realistic structure, temporal continuity, and acoustic richness, closely resembling the ground truth. Corresponding audio samples are included in the supplementary materials. These results highlight the benefits of the model-guided objective in generating coherent and semantically accurate audio. Beside the samples show at here, we have included more samples generated from different method in the supplementary material.

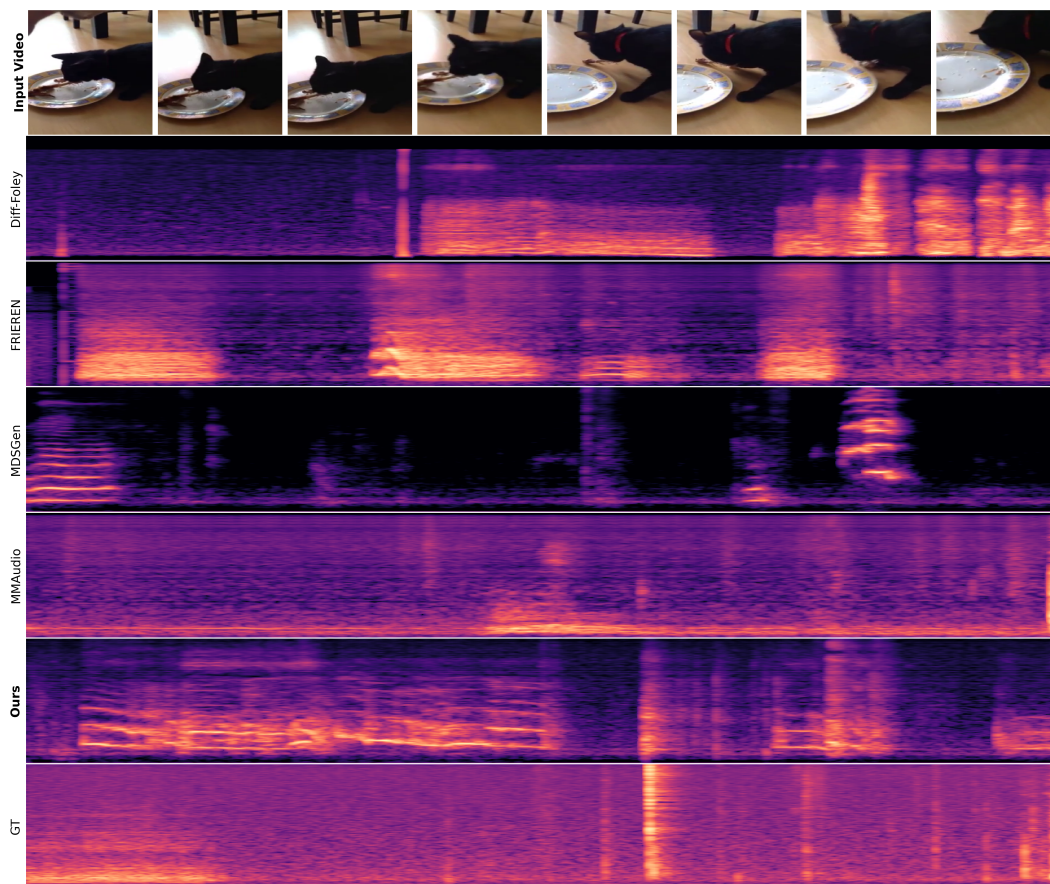


Figure 5: **Qualitative Comparison of Mel-Spectrograms.** The example is from the VGGSound test set (video: “A8rkIgn3N4A\_000028.mp4”), depicting a cat growling.

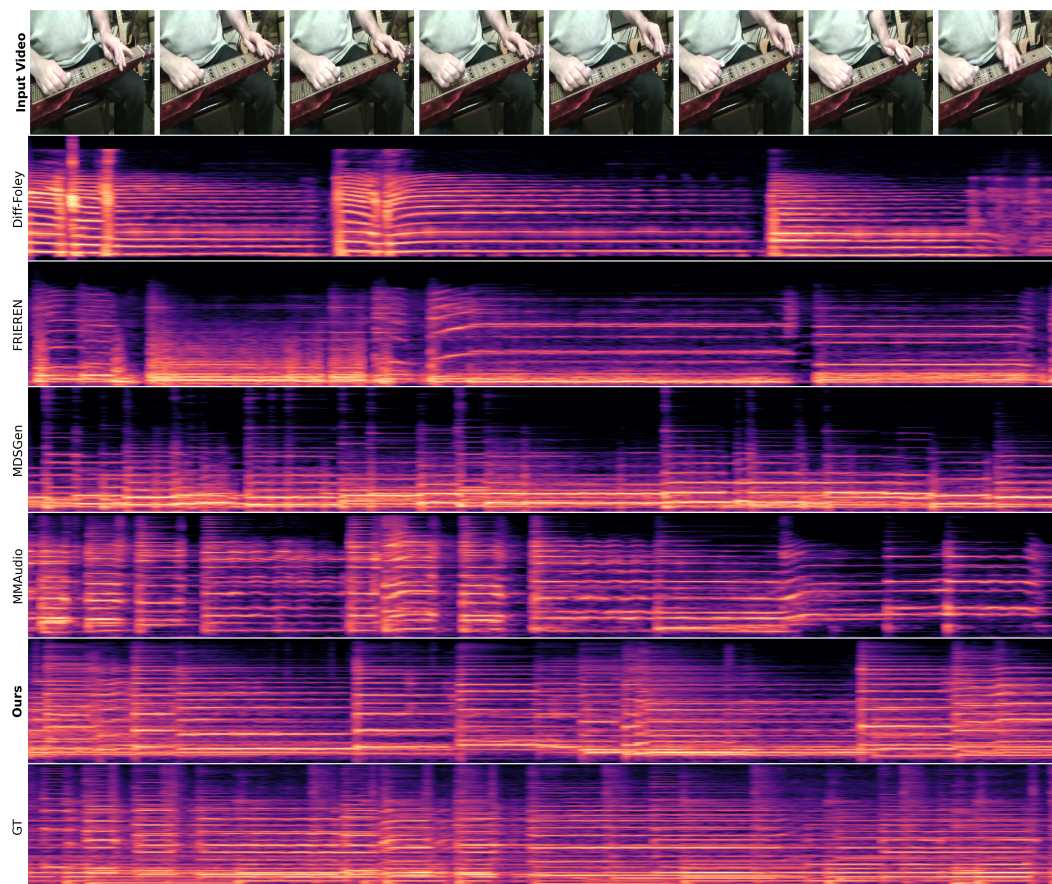


Figure 6: **Qualitative Comparison of Mel-Spectrograms.** The example is from the VGGSound test set (video: “GO2Tf8KLJ14\_000061.mp4”), depicting a person playing steel guitar, slide guitar.



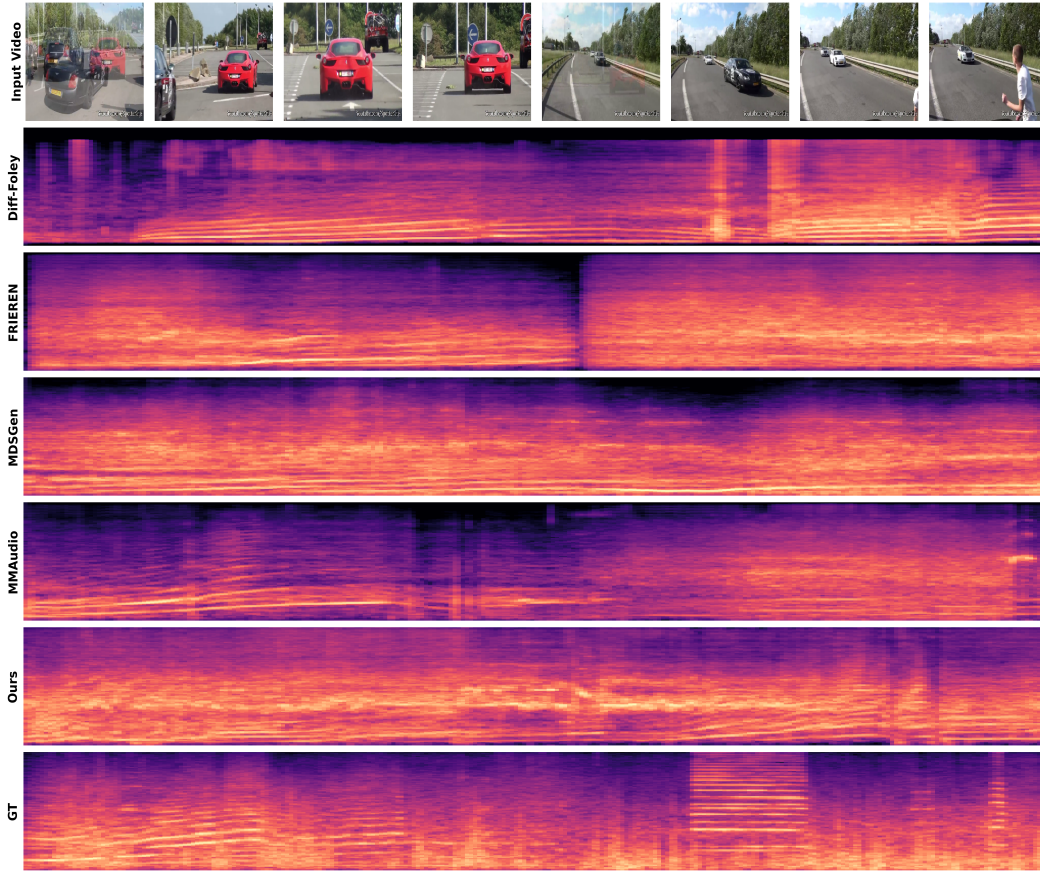


Figure 7: **Qualitative Comparison of Mel-Spectrograms.** The example is from the VGGSound test set (video: “hHbJRPjqgXQ\_000090.mp4”), depicting a race car, auto racing.

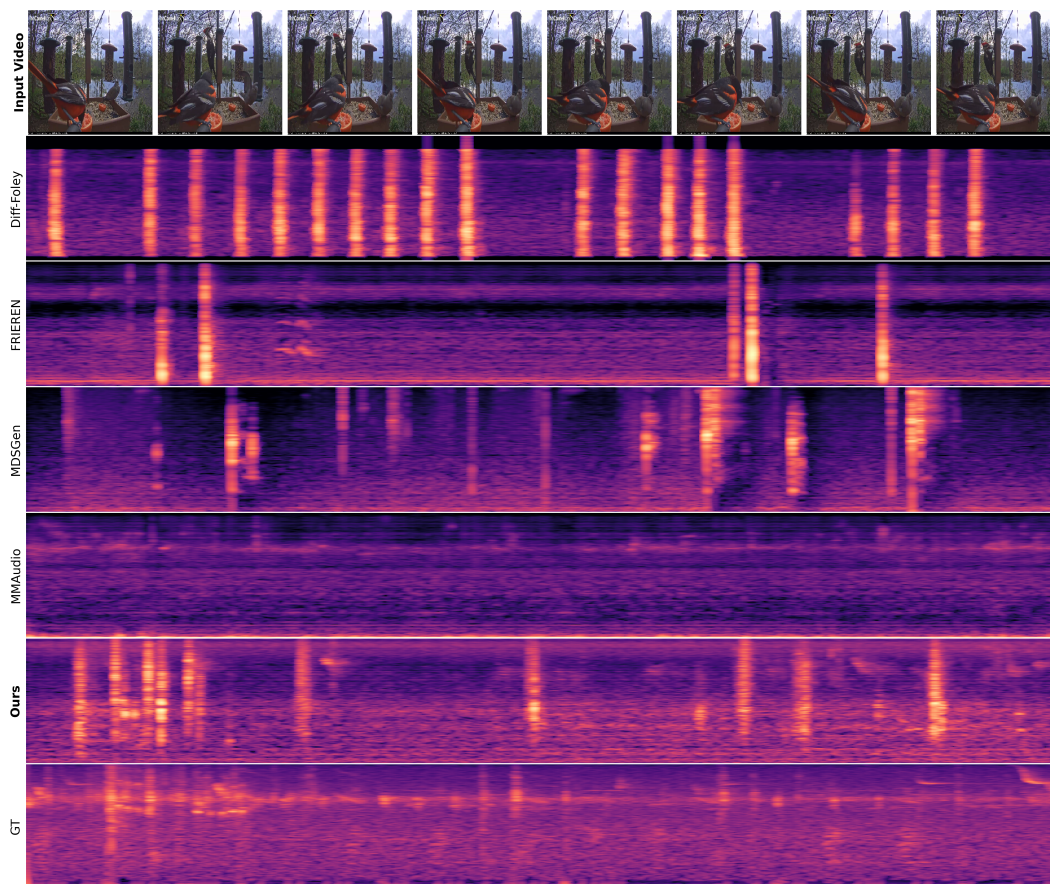


Figure 8: **Qualitative Comparison of Mel-Spectrograms.** The example is from the VGGSound test set (video: “mKEJRZtNx9o\_000044.mp4”), depicting a baltimore oriole calling his mate.



## References

- [1] Zhicong Tang, Jianmin Bao, Dong Chen, and Baining Guo. Diffusion models without classifier-free guidance. *arXiv preprint arXiv:2502.12154*, 2025. 1, 2, 5
- [2] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023. 1
- [3] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, 2024. 1, 2, 3, 5
- [4] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *NeurIPS*, 2023. 2, 6
- [5] Yazhou Xing, Yingqing He, Zeyue Tian, Xintao Wang, and Qifeng Chen. Seeing and hearing: Open-domain visual-audio generation with diffusion latent aligners. In *CVPR*, 2024. 2, 6
- [6] Yongqi Wang, Wenxiang Guo, Rongjie Huang, Jiawei Huang, Zehan Wang, Fuming You, Ruiqi Li, and Zhou Zhao. Frieren: Efficient video-to-audio generation network with rectified flow matching. *NeurIPS*, 2024. 2, 6
- [7] Trung X Pham, Tri Ton, and Chang D Yoo. MDSGen: Fast and efficient masked diffusion temporal-aware transformers for open-domain sound generation. In *ICLR*, 2025. 2, 5, 6
- [8] Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. Taming multimodal joint training for high-quality video-to-audio synthesis. In *CVPR*, 2025. 2, 6
- [9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2
- [10] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. In *ICML*, 2023. 2
- [11] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 28:2880–2894, November 2020. 3, 6
- [12] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 3
- [13] 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017. 6
- [14] Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 2753–2757. ISCA, 2022. 6
- [15] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 6
- [16] 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. 6
- [17] Ilpo Viertola, Vladimir Iashin, and Esa Rahtu. Temporally aligned audio for video with autoregression. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 6