AudioComposer: Towards Fine-grained Audio Generation with Natural Language Descriptions

Yuanyuan Wang[1,*], Hangting Chen[2,†], Dongchao Yang[1], Zhiyong Wu[1,3], Xixin Wu[1,†]
[1] The Chinese University of Hong Kong, Hong Kong SAR, China
[2] Tencent AI Lab, Audio and Speech Signal Processing Oteam, China
[3] Shenzhen International Graduate School, Tsinghua University, Shenzhen, China

***Abstract*—Current Text-to-audio (TTA) models mainly use coarse text descriptions as inputs to generate audio, which hinders models from generating audio with fine-grained control of content and style. Some studies try to improve the granularity by incorporating additional frame-level conditions or control networks. However, this usually leads to complex system design and difficulties due to the requirement for reference frame-level conditions. To address these challenges, we propose AudioComposer, a novel TTA generation framework that relies solely on natural language descriptions (NLDs) to provide both content specification and style control information. To further enhance audio generative modeling, we employ flow-based diffusion transformers with the cross-attention mechanism to incorporate text descriptions effectively into audio generation processes, which can not only simultaneously consider the content and style information in the text inputs, but also accelerate generation compared to other architectures. Furthermore, we propose a novel and comprehensive automatic data simulation pipeline to construct data with fine-grained text descriptions, which significantly alleviates the problem of data scarcity in the area. Experiments demonstrate the effectiveness of our framework using solely NLDs as inputs for content specification and style control. The generation quality and controllability surpass state-of-the-art TTA models, even with a smaller model size.** [1]

***Index Terms*—audio generation, natural language description, style control, flow-based, diffusion**

I. Introduction

Text-to-audio (TTA) generation focuses on generating authentic and accurate audios corresponding to the information specified in text inputs [1]. TTA plays a crucial role in producing various sound effects and applies to diverse fields including movie sound effect creation, virtual reality, game design, audio editing, and interactive systems [2], [3]. In recent years, there have been remarkable advancements in deep generative models [4]–[6], which have substantially contributed to the development of audio generation. Some recent works have made considerable progress by employing diffusion models [1], [7]–[12] or autoregressive models [13]–[15]. These existing approaches primarily concentrate on audio generation based on coarse text content descriptions, which in turn restricts the style controllability of generating fine-grained audio. For instance, they are unable to specify and generate accurate temporal locations of sound events, which is a crucial limitation for applications such as video dubbing.

To achieve such detailed control, the availability of fine-grained text-audio pair data is a vital prerequisite. Nevertheless, such paired text-audio data is generally difficult to obtain, especially with fine-grained instructions. Recently, Make-An-Audio2 [2] employed large language models (LLMs) to augment structured captions into natural language captions, which alleviates the issue of insufficient temporal paired data. However, this approach is constrained to the simple instructions with temporal orders (e.g., "*sound A and then sound B*") and fails to specify more detailed information, like precise timestamps and durations (e.g., "*sound A starts from 2.5 seconds and lasts for 3 seconds*"). Another research line for improving fine-grained controllability proposes to incorporate extra control conditions into the TTA systems. PicoAudio [16] utilized frame-level timestamp information as complementary conditions to text inputs. Guo *et al.* [3] designed a specialized encoder to extract control information from frame-level conditions and a Fusion-Net for integrating the fine-grained control information in the generation process. However, these approaches increase model complexity and also bring difficulties during inference as extra frame-level conditions from reference audios are required. Therefore, current controllable TTA research still faces three main challenges: (1) **scarcity** of fine-grained audio-text data; (2) **complexity** in incorporating control information; (3) lack of **precision** in control capabilities.

In this paper, we propose AudioComposer, a fine-grained audio generation framework based on flow-based diffusion transformers with only text as inputs for both content specification and style control. First, to mitigate the issue of data scarcity, we introduce an innovative online data simulation pipeline that enables fine-grained style annotations, including information of timestamps, pitch, and energy, with natural language descriptions (NLDs). Second, leveraging the automatic data simulation pipeline, AudioComposer exclusively depends on fine-grained natural language guidance for controllable TTA generation. This approach eliminates the need for additional frame-level conditions or complex control networks, achieving high simplicity and efficiency. Finally, inspired by the success of flow matching and diffusion transformers [17], [18], considering the need to preserve fine-grained text rep-

[1]Demo and code are available in https://lavendery.github.io/AudioComposer/.
*Work performed during an internship at Tencent AI Lab. †Corresponding author. This work was supported by National Natural Science Foundation of China (62306260,62076144).

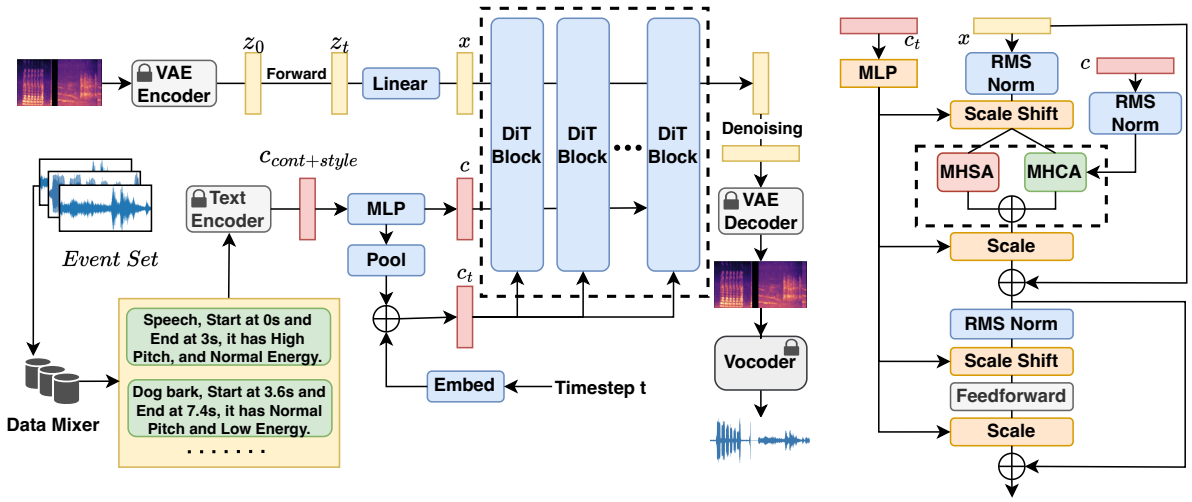

Fig. 1. The left picture shows the overview of AudioComposer, while the right picture illustrates the details of DiT Block. MHSA: Multi-Head Cross-Attention. MHCA: Multi-Head Self-Attention. $\oplus$ means add.

resentation [19], [20], we explore flow-based diffusion transformers by integrating text conditions through cross attention mechanisms, which can capture inherent connections between fine-grained text representation and latent audio tokens. This architecture not only accelerates the generation process but also enhances audio generative performance. In summary, the main contributions of this paper are as follows:

- We present a comprehensive automatic data simulation pipeline to generate fine-grained NLDs, which effectively tackles the issue of data scarcity in controllable TTA systems.
- Our method utilizes NLDs to enable precise control in TTA generation, eliminating the need for additional conditions or complex control networks.
- We employ flow-based diffusion transformers with the cross-attention mechanism, which improves generation speed, quality, and controllability.

II. Methodologies

This section presents our fine-grained TTA generation system AudioComposer. The overall architecture is demonstrated in Fig. 1, consiting of a variational auto-encoder (VAE) [21], a flow-based diffusion transformer (DiT) model [18], a text encoder [22], and a vocoder [23]. Following previous works [2], [7], we use pre-trained and frozen Mel-VAE and BigVGAN-based vocoder [23] in our framework.

A. Data Mixer for AudioComposer

Our data mixer can generate a dataset, referred to as AudioTPE, with fine-grained annotations which include event labels, timestamps, pitch, and energy information. The simulation process to obtain AudioTPE consists of the following steps. First, for datasets comprising a series of clean sound events, where typically each audio contains only one sound event, we initially calculate the average pitch and energy for each audio file, which are partitioned into high, normal, and low categories based on the 25%, 50%, and 75% quantiles. In this way, each audio possesses its distinct pitch and energy categories. We then randomly select these annotated audios to simulate mixed audios with durations less than 10 seconds. During this process, we record start time, end time, pitch category, and energy category of each event in simulated audios, which yields fine-grained annotated audio data. Finally, we generate NLDs based on these annotations using a template, e.g., "*Dog bark, Start at 3.6s and End at 7.4s, it has Normal Pitch and Low Energy.*"

B. AudioComposer

Flow matching [17], [20] has demonstrated powerful generation performance and efficiency in image processing fields. In this paper, we explore the ability of flow-based diffusion models in AudioComposer. Inspired by DiT [18], [19], [24], [25] and Sora, we also propose to combine flow-based diffusion formulation and transformer-based structure. In the following, we first review the flow matching and then show how to achieve fine-grained control using only NLDs.

1) Conditional Flow Matching: Conditional flow matching (CFM) [17] aims to learn a mapping between samples $\varepsilon \sim \mathcal{N}(0, I)$ from noise distribution to samples $x \sim p(x)$ from data distribution through the following interpolation-based forward process between the time interval $[0, 1]$:

$$x_t = \alpha_t x + \beta_t \epsilon, \tag{1}$$

where $\alpha_0 = 0, \beta_0 = 1, \alpha_1 = 1$, and $\beta_1 = 0$. With different choices of α_t and β_t, different interpolation schedules are obtained, e.g., the linear interpolation $x_t = tx + (1-t)\epsilon$ with $\alpha_t = t, \beta_t = 1 - t$. The goal is to employ a trainable neural network $v_t(x_t; \theta)$ to approximate the time-dependent velocity field $u_t(x_t) = \alpha'_t x + \beta'_t \epsilon$, where α'_t and β'_t denote the derivatives of α_t and β_t with respect to time t. The training loss can be defined as [20]:

$$\mathcal{L}_{\texttt{flow}} = \mathbb{E}_{t\sim\mathcal{U}[0,1],\epsilon\sim\mathcal{N}(0,I),p(x)}||v_t(x_t;\theta) - u_t(x_t)||^2, \tag{2}$$

where $\mathcal{U}[0, 1]$ is a uniform distribution, sharing similarity with the noise prediction or score prediction losses in diffusion.

With the trained network, we can transform noise samples into data samples by solving the flow ordinary differential equation (ODE) from $t = 0$ to $t = 1$:

$$\mathrm{d}x_t = v_t(x_t;\theta)\mathrm{d}t. \quad (3)$$

2) Fine-grained Control with Natural Language Descriptions: To ensure capturing content specification and style control information, we utilize the pre-trained T5 [22] as the text encoder based its outstanding natural language understanding capability. The content and style information is extracted by T5 from NLDs into the representation $c_{cont+style}$. Based on the pre-trained VAE, z_0 is generated from mel-spectrograms. The diffusion process is conducted to gradually add noise to z_0 to obtain the noisy tokens z_t, which is further encoded to a latent representation x using linear layers. As depicted in the left picture of Fig. 1, we employ latent representations x and text representations of c and c_t as inputs to the DiT blocks. Following the standard DiT that utilizes a modulation mechanism [19] to condition the network on both the timesteps of the diffusion process and the class labels, we combine embeddings of the timestep t and pooled representations of $c_{cont+style}$ into c_t as inputs for the DiT blocks. Considering that the pooled text representations retain merely coarse-grained information about text inputs [26], while our AudioComposer needs fine-grained text representations to achieve accurate control [19], [20], we also incorporate the sequence representation c transfromed from $c_{cont+style}$ by multi-layer perceptron (MLP) without pooling into each DiT block.

The structure of the DiT blocks is shown in the right picture of Fig. 1. The queries x_q of latent audio tokens are used to aggregate information from keys and values of text representations c using a cross-attention mechanism, which helps capture inherent connections between text and latent audio tokens and contributes to fine-grained information extraction. Given audio queries x_q, keys x_k, and values x_v with text keys c_k and values c_v, the final output of self-attention and cross-attention is computed as:

$$A = softmax\left(\frac{\widetilde{x}_q\widetilde{x}_k^T}{\sqrt{d}}\right)x_v + tanh(\alpha)softmax\left(\frac{\widetilde{x}_q c_k^T}{\sqrt{d}}\right)c_v,$$

where $\widetilde{x}_q$ and $\widetilde{x}_k$ means applying Rotary Position Embedding (RoPE) [27] to audio queries and keys, d represents the dimension of queries and keys, α denotes the zero-initialized learnable parameter in the gated cross-attention [28]–[30]. Using the outputs of the last DiT block, we can recover the latent tokens by solving ODE during inference. Lastly, we can get waveforms with the help of VAE decoder and Vocoder.

III. EXPERIMENTAL SETUP

A. Dataset

In the experiments, we use three types of datasets: (1) **AudioTPE data**. We combine several datasets, including FSD50K [31], ESC50 [32], UrbanSound8K [33], ODEON_Sound_Effects[2], to simulate mixed data online. Each audio in the raw datasets typically encompasses a single sound event. With a total of 49k audios amounting to approximately 140 hours, we perform online simulation and fine-grained annotation of the data, as introduced in Section II-A to obtain the final AudioTPE dataset that contains fine-grained annotations on information of events, time, pitch, and energy. (2) **AudioCondition** [3]. This dataset is derived from AudioSet Strong [34], [35], which contains about 81k audios, in total about 230 hours, with temporally strong labels. (3) **AudioCaps** [36]. This is a large-scale dataset of about 46K audio clips to human-written text pairs, totaling around 120 hours. It contains only content information without style control.

TABLE I
RESULTS ON AUDIOCONDITION TEST SET. -S: SMALL, -L: LARGE.

Model	#Params	Objective(%) ↑		Subjective↑	
		$F1_{event}$	$F1_{seg}$	MOS_t	MOS_q
Ground Truth	-	43.36	63.46	4.01	4.24
AudioLDM-L-Full	739M	3.21	23.94	-	-
AudioLDM2	346 M	4.71	39.72	-	-
AudioLDM2-Large	712M	8.4	46.19	-	-
Tango	866M	1.6	26.51	-	-
Tango2	866M	4.04	39.41	1.69	2.8
MC-Diffusion [3]	1076M	29.07	-	-	-
Tango+LControl	866M	21.46	55.15	2.47	3.01
AudioComposer-S	272M	<u>43.51</u>	<u>60.83</u>	**4.6**	<u>3.81</u>
AudioComposer-L	742.79M	**44.4**	**63.3**	<u>4.51</u>	**4.02**

B. System Configuration

We train the VAE to compress mel-spectrograms into 20-dimension latent representations. The AudioComposer is trained on 8 NVIDIA V100 GPUs, using a batch size of 16 per GPU. We employ the AdamW optimizer [37] with a learning rate of 1e-4. All model is trained with 70k steps, and the transformer head is 32. The AudioComposer has two versions with different parameter sizes, i.e., AudioComposer-S with 6 transformer blocks and a hidden size of 768, and AudioComposer-L with 9 transformer blocks and a hidden size of 1024.

C. Evaluation Metrics

Objective metrics. For timestamp control, we employ a sound event detection (SED) model to pinpoint the locations of generated sound events. The open-source SED system PB-SED[3] [38] is used. We utilize event-based macro F1-score ($F1_{event}$), and segment-based macro F1-score ($F1_{seg}$) to evaluate event accuracy [39]. For pitch and energy control, we evaluate the accuracy (ACC) of pitch and energy categories across all audio clips. We also evaluate the mean absolute error (MAE) between frame-wise pitch and energy extracted from generated audios and those from ground-truth audios [40].

Subjective metrics. We conduct mean opinion score (MOS) assessment from multiple perspectives: (1) temporal controllability (MOS_t) for evaluating accuracy of timestamp control; (2) pitch controllability (MOS_p) for assessing accuracy of pitch control; (3) energy controllability (MOS_e) for measuring accuracy of energy control, and (4) audio quality (MOS_q) for evaluating naturalness of generated audios (without taking into account the accuracy of time, pitch, or energy). For each task, 10 test groups from each model are rated by 10 evaluators, and the mean score is calculated.

[2]https://www.paramountmotion.com/odeon-sound-effects

[3]https://github.com/fgnt/pb_sed

TABLE II
RESULTS ON AUDIOTPE TEST SET. -S DENOTES THE SMALL MODEL SIZE, AND -L DENOTES THE LARGE MODEL SIZE.

Model	#Params	Timestamp	Pitch		Enenrgy		Subjective↑			
		$F1_{seg}(\%)\uparrow$	$ACC(\%)\uparrow$	$MAE\downarrow$	$ACC(\%)\uparrow$	$MAE\downarrow$	MOS_t	MOS_p	MOS_e	MOS_q
Ground Truth	-	60.63	78.75	-	90.16	-	4.57	4.41	4.4	4.48
AudioLDM-L-Full	739M	23.11	29.98	108.97	37.81	34.4	-	-	-	-
AudioLDM2	346 M	41.54	33.55	129.05	43.62	36.61	-	-	-	-
AudioLDM2-Large	712M	39.72	33.56	118.87	42.05	34.56	-	-	-	-
Tango	866M	33.79	35.57	116.95	44.07	30.11	-	-	-	-
Tango2	866M	44.35	34.45	118.6	48.99	30.7	2.11	2.59	2.61	2.77
Tango+LControl	866M	47.91	39.37	113.9	52.13	**27.4**	4.14	3.41	3.93	3.68
AudioComposer-S	272M	50.97	**60.63**	91.25	63.53	37.33	4.5	**3.71**	3.63	3.82
AudioComposer-L	742.79M	**51.36**	56.6	**87.72**	**65.77**	37.03	**4.58**	3.64	**4.20**	**4.11**

TABLE III
ABLATION STUDIES ON AUDIOCONDITION TEST SET.

Model	$F1_{event}(\%)\uparrow$	$F1_{seg}(\%)\uparrow$
Ground Truth	43.36	63.46
AudioComposer-S	**43.51**	**60.83**
w 200 inference steps	46.43	66.19
w/o flow matching	35.19	46.92
w/o AudioCaps	34.55	47.84

IV. RESULTS AND ANALYSES

A. Results on AudioCondition

In this experiment, we train AudioComposer on all datasets introduced in Section III-A and evaluate it on AudioCondition. We compare AudioComposer with mainstream generative models, including AudioLDM [8], AudioLDM2 [11], Tango [10], Tango2 [12], and MC-Diffusion [3], to assess the performance of temporal controllability, with the results provided in Table I. As Tango and AudioLDM generate audio without the need for temporal conditions, we train Tango [4] from scratch with language control as a more direct comparison baseline, namely Tango+LControl. Considering the lack of style control in open-source models and the cost of human resources, we only select Tango2 for the MOS evaluation.

Table I shows that our AudioComposer outperforms these baseline models across all metrics. Interestingly, AudioComposer even surpasses the ground truth in $F1_{event}$ and MOS_t. This can be attributed to the fact that the ground truth is derived from AudioSet, which contains some extraneous noise apart from specific events. These results demonstrate the outstanding performance of AudioComposer, even with fewer parameters in AudioComposer-S.

B. Results on AudioTPE

Similar to Section IV-A, we compare our AudioComposer with several mainstream baseline models in terms of time, pitch and energy controllability on the AudioTPE test set. As shown in Table II, AudioComposer exhibits significant performance advantages in nearly all metrics. It is noteworthy that on energy control, AudioComposer exhibits a notable performance advantage in the ACC metric, yet a inferior performance in the MAE metric. This can be attributed to the input control information of our approach being limited to only three categories: high, normal, and low, rather than frame-level pitch and energy contours. As a result, the ACC metric, which measures the accuracy of the three categories, demonstrates higher performance, while the frame-wise MAE metric performance is less satisfactory. However, the superior category accuracy also indicates the AudioComposor can generate pitch and energy within more appropriate ranges than baselines.

[4]https://github.com/declare-lab/tango/tree/master

C. Ablation Study

Table III provides further ablation studies on AudioCondition. Firstly, we remove flow matching and use the DDPM-based diffusion method, which results in a slight performance decline compared to AudioComposer-S. This suggests that the flow-based diffusion can enhance the model's control capability. In terms of inference efficiency, both DDPM-based diffusion and Tango need 200 steps, while our method can generate high-quality audio in just 25 steps. Our method also achieves better results with the same number of steps. Furthermore, when comparing our AudioComposer using DDPM-based diffusion transformer in Table III with Tango+LControl using U-Net diffusion from Table I, our diffusion transformer still achieves comparable performance, particularly in $F1_{event}$.

As shown in Table III, we remove the AudioCaps data and retrain the model AudioComposer-S, resulting in a performance decline. This validates that our method can utilize other coarse-grained data to enhance audio generation performance, as the models can be trained solely on NLDs in a holistic manner. Furthermore, AudioComposer can generate audio without style control, and we have evaluated it on the AudioCaps test set, as presented on our demo page due to page limitation.

V. CONCLUSIONS

In this study, we present a fine-grained audio generation approach with natural language descriptions using flow-based diffusion transformers. The proposed method does not require additional conditions or complex network structures, as it relies solely on natural language descriptions to provide content specification and style control information with simplicity and efficiency. We also propose a novel automatic data simulation pipeline that can construct fine-grained data and significantly alleviate the problem of data scarcity. Extensive experimental results prove that our approach enhances the speed, quality, and controllability of TTA generation and achieves state-of-the-art performances.

REFERENCES

[1] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, "Diffsound: Discrete diffusion model for text-to-sound generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[2] J. Huang, Y. Ren, R. Huang, D. Yang, Z. Ye, C. Zhang, J. Liu, X. Yin, Z. Ma, and Z. Zhao, "Make-an-audio 2: Temporal-enhanced text-to-audio generation," 2023.

[3] Z. Guo, J. Mao, R. Tao, L. Yan, K. Ouchi, H. Liu, and X. Wang, "Audio generation with multiple conditional diffusion model," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 18 153–18 161.

[4] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 2018.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[7] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," 2023.

[8] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," *Proceedings of the International Conference on Machine Learning*, pp. 21 450–21 474, 2023.

[9] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, "Text-to-audio generation using instruction-tuned llm and latent diffusion model," 2023.

[10] G. Deepanway, M. Navonil, M. Ambuj, and P. Soujanya, "Text-to-audio generation using instruction-tuned llm and latent diffusion model," *arXiv preprint arXiv:2304.13731*, 2023.

[11] H. Liu, Y. Yuan, X. Liu, X. Mei, Q. Kong, Q. Tian, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "Audioldm 2: Learning holistic audio generation with self-supervised pretraining," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2871–2883, 2024.

[12] N. Majumder, C.-Y. Hung, D. Ghosal, W.-N. Hsu, R. Mihalcea, and S. Poria, "Tango 2: Aligning diffusion-based text-to-audio generations through direct preference optimization," 2024.

[13] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," 2023.

[14] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu, X. Chang, J. Shi, S. Zhao, J. Bian, X. Wu, Z. Zhao, S. Watanabe, and H. Meng, "Uniaudio: An audio foundation model toward universal audio generation," 2023.

[15] D. Yang, H. Guo, Y. Wang, R. Huang, X. Li, X. Tan, X. Wu, and H. Meng, "Uniaudio 1.5: Large language model-driven audio codec is a few-shot audio task learner," 2024.

[16] Z. Xie, X. Xu, Z. Wu, and M. Wu, "Picoaudio: Enabling precise timestamp and frequency controllability of audio events in text-to-audio generation," 2024. [Online]. Available: https://arxiv.org/abs/2407.02869

[17] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *arXiv preprint arXiv:2210.02747*, 2022.

[18] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4195–4205.

[19] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel *et al.*, "Scaling rectified flow transformers for high-resolution image synthesis," in *Forty-first International Conference on Machine Learning*, 2024.

[20] P. Gao, L. Zhuo, D. Liu, R. Du, X. Luo, L. Qiu, Y. Zhang, C. Lin, R. Huang, S. Geng, R. Zhang, J. Xi, W. Shao, Z. Jiang, T. Yang, W. Ye, H. Tong, J. He, Y. Qiao, and H. Li, "Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers," 2024. [Online]. Available: https://arxiv.org/abs/2405.05945

[21] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[22] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[23] S.-g. Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "Bigvgan: A universal neural vocoder with large-scale training," *arXiv preprint arXiv:2206.04658*, 2022.

[24] D. Yang, D. Wang, H. Guo, X. Chen, X. Wu, and H. Meng, "Simple-speech: Towards simple and efficient text-to-speech with scalar latent transformer diffusion models," *Proc. INTERSPEECH*, 2024.

[25] D. Yang, R. Huang, Y. Wang, H. Guo, D. Chong, S. Liu, X. Wu, and H. Meng, "Simplespeech 2: Towards simple and efficient text-to-speech with flow-based scalar latent transformer diffusion models," 2024.

[26] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," *arXiv preprint arXiv:2307.01952*, 2023.

[27] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.

[28] R. Zhang, J. Han, C. Liu, P. Gao, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, and Y. Qiao, "Llama-adapter: Efficient fine-tuning of language models with zero-init attention," *arXiv preprint arXiv:2303.16199*, 2023.

[29] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue *et al.*, "Llama-adapter v2: Parameter-efficient visual instruction model," *arXiv preprint arXiv:2304.15010*, 2023.

[30] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.

[31] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2021.

[32] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, 2015, pp. 1015–1018.

[33] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 1041–1044.

[34] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[35] S. Hershey, D. P. W. Ellis, E. Fonseca, A. Jansen, C. Liu, R. Channing Moore, and M. Plakal, "The benefit of temporally-strong labels in audio event classification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 366–370.

[36] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *NAACL-HLT*, 2019.

[37] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[38] J. Ebbers and R. Haeb-Umbach, "Pre-training and self-training for sound event detection in domestic environments," 2022.

[39] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[40] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv preprint arXiv:2006.04558*, 2020.