SPECMASKGIT: MASKED GENERATIVE MODELING OF AUDIO SPECTROGRAMS FOR EFFICIENT AUDIO SYNTHESIS AND BEYOND

Marco Comunità[*1,2] **Zhi Zhong**[*2] **Akira Takahashi**[2] **Shiqi Yang**[2] **Mengjie Zhao**[2]
Koichi Saito[3] **Yukara Ikemiya**[4] **Takashi Shibuya**[4] **Shusuke Takahashi**[2] **Yuki Mitsufuji**[2,3]
[1] Queen Mary University of London, UK [2] Sony Group Corporation, Japan
[3] Sony AI, US [4] Sony AI, Japan
m.comunita@qmul.ac.uk, Zhi.Zhong@sony.com

ABSTRACT

Recent advances in generative models that iteratively synthesize audio clips sparked great success in text-to-audio synthesis (TTA), but at the cost of slow synthesis speed and heavy computation. Although there have been attempts to accelerate the iterative procedure, high-quality TTA systems remain inefficient due to the hundreds of iterations required in the inference phase and large amount of model parameters. To address these challenges, we propose SpecMaskGIT, a light-weight, efficient yet effective TTA model based on the masked generative modeling of spectrograms. First, SpecMaskGIT synthesizes a realistic 10 s audio clip in less than 16 iterations, an order of magnitude less than previous iterative TTA methods. As a discrete model, SpecMaskGIT outperforms larger VQ-Diffusion and auto-regressive models in a TTA benchmark, while being real-time with only 4 CPU cores or even 30× faster with a GPU. Next, built upon a latent space of Mel-spectrograms, SpecMaskGIT has a wider range of applications (*e.g.*, zero-shot bandwidth extension) than similar methods built on latent wave domains. Moreover, we interpret SpecMaskGIT as a generative extension to previous discriminative audio masked Transformers, and shed light on its audio representation learning potential. We hope that our work will inspire the exploration of masked audio modeling toward further diverse scenarios.

1. INTRODUCTION

Text-to-audio synthesis (TTA) allows users to synthesize realistic audio and sound event signals by natural language prompts. TTA can assist the sound design and editing in the music, movie, and game industries, accelerating creators' workflow [1]. Therefore, TTA has earned increasing attention in the research community.

Recent advances in deep generative models, especially iterative methods such as diffusion [2–5] and auto-regressive models [6–8], have brought significant success to the sound quality and controllability in TTA tasks, but at the cost of slow synthesis speed. Since the synthesis speed of iterative methods is dominated by the number of iterations required at inference, techniques have been introduced to reduce iterations, *e.g.*, higher compression rate of raw audio signals [6] or more efficient diffusion samplers [4, 9]. Nevertheless, these iterative methods remain slow in synthesis speed and demanding for computing resources, as they typically require hundreds of iterations to synthesize a short audio clip. Moreover, the runtime of a single iteration increases due to the huge model size.

To further improve inference efficiency, Garcia *et al.* introduced the MaskGIT [10] synthesis strategy from computer vision to the audio domain and proposed VampNet [11]. Although VampNet can inpaint a 10-second clip with 24 iterations, 6 seconds are needed on GPU [11], which is still heavy for non-GPU environments. Moreover, VampNet is not compatible with text prompts or TTA tasks. Concurrent to our work, MAGNeT extended VampNet to text-conditional audio synthesis [12]. However, the method is less efficient as it requires 180 iterations, which is heav-

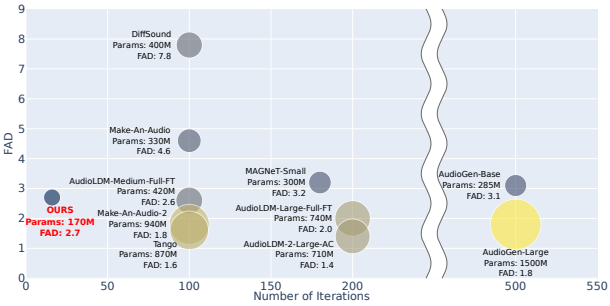

Figure 1. *Audio synthesis performance and number of synthesis iterations of different methods. The size of circle represents the model size. SpecMaskGIT achieves good quality with only 16 iterations and a small model size.*

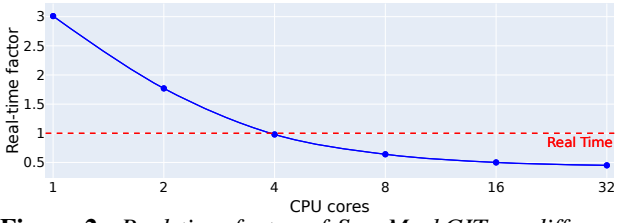

Figure 2. *Real-time factor of SpecMaskGIT on different Xeon CPU cores with standard Python implementation.*

*Equal contribution. Marco Comunità was an intern at Sony.

ier than some diffusion models that only require 100 iterations [4, 9, 13, 14]. Since both VampNet and MAGNeT work in a wave-domain latent space, it is difficult to conduct frequency-domain inpainting tasks such as bandwidth extension (BWE) in a zero-shot manner. Besides the aforementioned limitations, the audio representation learning potential of a masked generative Transformer has not been investigated yet.

As a summary, an audio synthesis method that is compatible with text prompts, highly efficient in synthesis speed, and flexible for various downstream tasks is yet to be explored. To this end, we propose SpecMaskGIT, an efficient and flexible TTA model based on the masked generative modeling of audio spectrograms. Our contributions lie in the following aspects:

- **Efficient and effective TTA.** SpecMaskGIT synthesizes a realistic 10-second audio clip in less than 16 iterations, which is one order of magnitude smaller than previous iterative methods (Fig. 1. As a discrete generative model, SpecMaskGIT outperforms larger VQ-Diffusion (DiffSound [2]) and auto-regressive (AudioGen-base [6]) models in a TTA benchmark, while being real-time with 4 CPU cores (Fig. 2) or even 30× faster on a GPU.
- **Flexibility in downstream tasks.** SpecMaskGIT is interpreted and implemented as a generative extension to previous discriminative audio masked Transformers [15–18]. The masked spectrogram modeling principle and architecture design similar to Audio Masked Autoencoder (MAE) [16–18] is believed to have contributed to the representation learning potential of SpecMaskGIT. Unlike prior art about finetuning MAE-like architectures for BWE [18,19], SpecMaskGIT enables BWE in a zero-shot manner.

We hope this efficient, effective and flexible framework paves the way to the exploration of masked audio modeling toward further diverse scenarios [20]. [1]

2. RELATED WORKS

Synthesizing audio signals in raw waveform is challenging and computationally demanding [21]. Therefore, the mainstream approach to audio synthesis is to first generate audio in a compressed latent space, and then restore waveforms from latent representations. Auto-regressive models such as Jukebox [22], AudioGen [6] and MusicGen [23] use vector-quantized (VQ) variational auto-encoders (VAE) [24] to tokenize raw waveforms into a discrete latent space. While AudioGen and MusicGen use a higher compression rate than Jukebox, 500 iterations are required to synthesize a 10-second clip, slowing generation down.

Advances in audio representation learning such as Audio MAE ([16–18]) indicate that Mel-spectrogram is an effective compression of raw audio signals, as it emphasizes acoustic features of sound events while maintaining sufficient details to reconstruct raw waveforms. Inspired by the above success of representation learning, several methods used discrete [2] or continuous [3,4,9,13,14] diffusion models upon the latent Mel-spectrogram space created by a VAE or SpecVQGAN [25]. These diffusion models require up to 200 iterations for high-fidelity synthesis, which is still challenging for low-resource platforms and interactive use cases. While distilling a diffusion model can effectively reduce the required iterations [26–28], we limit our discussion to non-distilled methods for a fair comparison. For Mel-based synthesis methods, waveforms are reconstructed from Mel-spectrogram with a neural vocoder, such as HiFiGAN [29] or BigVSAN [30].

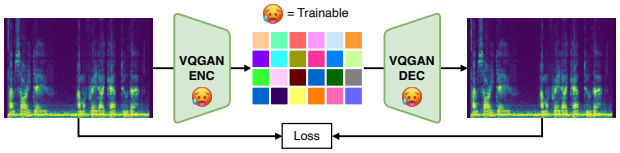

Figure 3. *SpecVQGAN encodes/decodes non-overlapping 16-by-16 time-mel patches into/from discrete tokens.*

In pursuit of higher synthesis efficiency, VampNet [11] and the concurrent MAGNeT [12] adopted the parallel iterative synthesis strategy from MaskGIT [10]. Originally proposed for class-conditional image synthesis tasks, MaskGIT uses a Transformer with bi-directional attention - instead of the uni-directional counterpart of auto-regressive methods - to reduce the required number of iterations. Although VampNet and MAGNeT reduced the number of iterations compared to their auto-regressive counterparts, VampNet does not support text prompts, while MAGNeT takes 180 iterations, which is even heavier than some diffusion models that only require 100 iterations [4, 9, 13, 14]. Moreover, it is difficult for methods built upon wave-domain latent spaces to address frequency domain tasks such as BWE, limiting their applications.

3. SPECMASKGIT

The efficiency, effectiveness and flexibility of SpecMaskGIT is due to a combination of efforts, including among other, the high compression rate in the tokenizer, the small model size, and the fast synthesis algorithm.

3.1 Spectrogram Tokenizer and Vocoder

A modified SpecVQGAN [25] is trained to tokenize non-overlapping 16-by-16 time-mel patches into discrete tokens, and recover the tokens back to Mel-spectrogram as in Fig. 3. Reconstructed Mel-spectrograms are then transformed to waveforms by a pre-trained vocoder. On top of the 3.2× compression offered by the wave-to-mel transform in our configuration, SpecVQGAN further offers 256× compression of the spectrogram, resulting in a total of over 800× compression of the raw waveform, effectively reducing the number of tokens to synthesize.

We utilize the standard Mel transform widely used in vocoders [29–32] for optimal Mel computation, as hyperparameters of Mel transform have an impact on tokenizers' performance [9]. To stabilize the training, we keep the spectrogram normalization in the original SpecVQGAN, which clips Mel bins lower than -80 dB or louder than 20 dB, and then maps the spectrogram into the [-1.0, 1.0] range. Our modified SpecVQGAN is shown competitive in reconstruction quality in Sec. 5.1.

[1] Demo: https://zzaudio.github.io/SpecMaskGIT

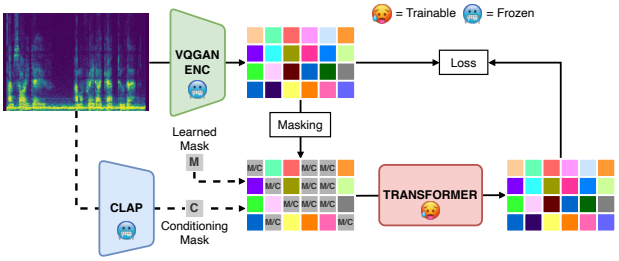

Figure 4. *Self-supervised training of SpecMaskGIT. The Transformer is trained to reconstruct SpecVQGAN token sequences - randomly masked with varying ratios - unconditionally via a learned mask token ("M"); or conditioned on a semantic token from the CLAP encoder ("C").*

3.2 Masked Generative Modeling of Spectrograms

We train a masked generative Transformer upon the discrete latent space of the pretrained SpecVQGAN as in Fig. 4. First, the pretrained CLAP encoder [33] maps the input audio to a semantic embedding aligned with its corresponding text descriptions. Meanwhile, the input audio is tokenized by SpecVQGAN. Finally, similar to representation learning such as Audio MAE [16–18], a bi-directional Transformer is trained to reconstruct Mel-spectrogram token sequences from a randomly masked input.

There are two major differences from Audio MAE. First, the masking ratio is *not* a fixed value but sampled on-the-fly from a truncated Gaussian distribution that is centered at 55% [34] and ranges from 0% to 100% [10]. As a result, although in each training step SpecMaskGIT behaves similarly to Audio MAE, it learns the training data distribution from various masking ratios, hence gaining the ability to iteratively refine audio tokens by gradually decreasing the masking ratio across multiple iterations, which is explained in Sec. 3.4. Second, while Audio MAE works on raw Mel-spectrogram, optimizing the mask reconstruction by mean square error; SpecMaskGIT works in a discrete latent space, which means the reconstruction of a masked position evolves to retrieval of the correct code from the SpecVQGAN codebook, *i.e.*, a multi-class single-label classification procedure. Therefore, the loss function becomes the cross entropy (CE) loss with label smoothing equal to 0.1. Following Audio MAE, visible positions in the input are not considered in the loss calculation:

$$\text{Loss} = \text{CE}(\text{prediction}[\text{mask}], \text{label}[\text{mask}]). \quad (1)$$

3.3 Text Conditioning via Sequential Modeling

Similarly to [4], we train SpecMaskGIT without audio-text pairs by using a pretrained CLAP model [33], for which audio and text embeddings are aligned in a shared latent space. Leveraging such alignment, after training with the audio branch of CLAP (see Fig. 4), we can directly condition our pretrained model with the text branch as shown in Fig. 5. We use a publicly available CLAP checkpoint ("630k-audioset-best.pt" [33]) for better reproducibility.

Although the above design is inspired by AudioLDM [4], SpecMaskGIT is different in the way CLAP embeddings are injected. Besides the FiLM mechanism ([35]) used in AudioLDM, prior works inject text conditions via cross-attention [2, 3, 9, 13, 14], even for methods based on sequential modeling such as AudioGen [6] and MAGNeT [12], which inevitably involves efforts to modify basic DNN modules. We believe that reusing modules, such as the Vision Transformer (ViT) [36], across different tasks is beneficial for efficient development, so we choose to achieve text-conditional audio synthesis by pure sequential modeling, *i.e.*, prepending the CLAP embedding to the input sequence to the Transformer. Note that the CLAP embedding is mapped to the same dimension as the Transformer by a linear layer in advance. As a result, SpecMaskGIT can be implemented with the same ViT used in Audio MAE [16–18], thus we view SpecMaskGIT as a generative extension to previous discriminative masked spectrogram modeling methods. We hypothesize the masked modeling and ViT implementation similar to Audio MAE has contributed to the representation learning potential of SpecMaskGIT, as is shown in Sec. 5.2.

While the common practice in [10, 16–18] is to use a learnable but input-independent token to indicate which parts in the sequence are masked ("M" in Fig. 4), the mask reconstruction task is challenging as the input-independent mask offers no hint for a better reconstruction. To further guide the mask reconstruction procedure, we propose to directly use the input-dependent CLAP embedding as a conditional mask ("C" in Fig. 4), which offers semantic hints like "a dog barking sound" to the model, and is found beneficial to TTA performance in Sec. 5.1.

3.4 Iterative Synthesis with Classifier-free Guidance

We follow the parallel iterative synthesis strategy proposed in MaskGIT [10] in general, but additionally employ classifier-free guidance (CFG) [37] to improve the synthesis quality. This iterative algorithm allows SpecMaskGIT to synthesize multiple high-quality tokens at each iteration, reducing the number of iterations to a value one order of magnitude smaller than previous TTA methods.

To enable CFG, we replace the CLAP embedding with the learned mask token on a random 10% of training steps. At inference phase, both the conditional (ℓ_c) and unconditional (ℓ_u) logits for each masked token are computed. The final logits ℓ_g are made by a linear combination of the two logits based on t, the guidance scale:

$$\ell_g = \ell_u + t(\ell_c - \ell_u). \quad (2)$$

Intuitively, CFG balances between diversity and audio-text alignment. Inspired by [38], we introduce a linear scheduler to the guidance scale t, which linearly increases t from 0.0 to an assigned value through the synthesis iterations. This allows the result of early iterations to be more diverse (unconditional) with low guidance, but increases the influence of the conditioning for late iterations, and is proved beneficial to synthesis quality in Sec. 5.1.

The parallel iterative synthesis of SpecMaskGIT shown in Fig. 5 is explained as follows:

1. Estimating. For each masked position, the Transformer estimates the probability of each code in the SpecVQGAN codebook to be the correct one, *i.e.*, the categorical distribution in the SpecVQGAN latent space.

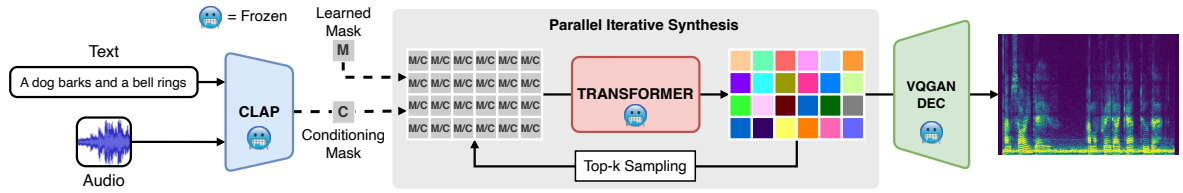

Figure 5. *The iterative text-to-audio synthesis in SpecMaskGIT.*

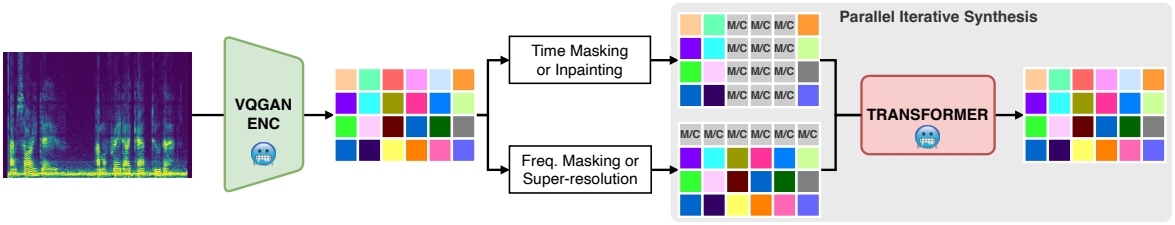

Figure 6. *Zero-shot time inpainting and bandwidth extension for general audio data via SpecMaskGIT.*

2. Unmasking. Given the categorical distribution over the codebook for each masked position, a code is randomly sampled. This step is different from the deterministic unmasking in Audio MAE.

3. Scheduling. Although SpecMaskGIT can unmask all positions at once, the quality of the synthesized audio is low. To iteratively refine the synthesis, we need to re-mask the result to a masking ratio that is lower than the current iteration. We follow the common practice in [10–12, 34] to use a cosine scheduler to decide the masking ratio at each iteration. The cosine scheduler re-masks a larger portion of the synthesized tokens for early iterations, which is intuitive as the quality in earlier iterations is lower.

4. Top-k sampling. Given the masking ratio for the next iteration, we know k tokens are going to be re-masked. The log-likelihood of unmasked tokens is used to decide the k worst tokens. Since it is observed that a deterministic top-k retrieval leads to the synthesis of monotonous images in [39], we follow [11, 34] and add Gumbel noise to the log-likelihood, making the top-k sampling stochastic:

$$\text{confidence} = \log(p) + t_{\text{gumbel}} \cdot n_{\text{gumbel}}, \qquad (3)$$

where p is the probability of each unmasked token calculated from the CFG logits in Eq. 2, n_{gumbel} is the Gumbel noise, and t_{gumbel} is the noise temperature. Following [34], we linearly anneal t_{gumbel} by a coefficient defined as iter/num_iter, with "iter" index of the current iteration and "num_iter" the total number of scheduled iterations.

5. Repeating. Repeat all steps until the cosine scheduler reduces the masking ratio to 0.

For TTA, SpecMaskGIT starts the above iterative procedure from a fully masked sequence as in Fig. 5. Nevertheless, the iterative algorithm is also valid when the masking ratio of an input sequence is lower than 100%, which automatically enables zero-shot inpainting in both time and frequency domain as is shown in Fig. 6. It is worth noticing that since VampNet [11] and MAGNeT [12] employ a wave-domain tokenizer, frequency inpainting or bandwidth extension (BWE) are difficult.

4. EXPERIMENTS

We pretrained the SpecVQGAN [25] and two vocoders (HiFiGAN [29] & BigVSAN [32]) on AudioSet (AS) unbalanced and balanced subset [40] for 1.5M steps. The AS we collected contains around 1.8 million 10-second audio segments of diverse sound sources and recording environments. AS has been widely used in general audio representation learning [16–18]. We followed the "VGGSound" configuration in the original SpecVQGAN repository [25] without using LPAPS loss as suggested in the repository itself. Our SpecVQGAN has around 75M parameters, and a codebook of 1024 codes, each of which is represented by a 256-dim embedding. As mentioned in Sec. 3.1, the standard Mel-spectrogram transform from vocoders [29, 30] is utilized, which transforms a 10-second audio clip at sampling rate 22.05kHz into 848 frames with 80 Mel bins. The Mel-spectrogram is further tokenized into 265 tokens.

SpecMaskGIT employs the ViT implementation widely used in previous audio masked Transformers [15, 16, 18, 41]. To be consistent with the image MaskGIT [10], 24 Transformer blocks are used, in which the attention dimension is 768 with 8 heads and the feedforward dimension is 3072, resulting in around 170M parameters. We trained SpecMaskGIT on AS for 500k steps with a batch size of 112. When training the model on AudioCaps (AC) [42], we train for 250k steps with a batch size of 48, as AC only contains 50k 10-second audio clips. To stably train SpecMaskGIT, we follow the common practice in [16–18] to employ a linear warmup and then a cosine annealing of the learning rate (LR). We warmup 16k steps for AS and 5k steps for AC. The base LR is set to 1e-3, and the LR equates to the base LR times the batch size divided by 256 [17, 34]. The iterative synthesis algorithm is based on the open-source implementation of [34].

To evaluate the **TTA synthesis** quality of SpecMaskGIT, we benchmark on the AudioCaps (AC) test set with the text prompts released by [4] for fair comparison. To investigate the flexibility of SpecMaskGIT in downstream tasks, we use the checkpoint trained on AS for 500k steps in the following tasks: **Zero-shot time inpainting**. We manually mask out the 25th to 35th Mel-spec frames (around 1.9s) of AC test set, and employ SpecMaskGIT to inpaint the lost regions in a zero-shot manner, *i.e.*, no task-specific finetuning. **Zero-shot audio bandwidth extension**. The top 16 Mel-spec bins (*i.e.*, components beyond 4.3kHz) of AC test set are masked, which creates

Table 1. *Comparing SpecMaskGIT with other discrete TTA methods on AudioCaps test set.*

Method	Params	Text	Num_iter	FAD
Diffsound [2]	400M	Yes	100	7.8
MAGNeT-small [12]	300M	Yes	180	3.2
AudioGen-base [6]	285M	Yes	500	3.1
AudioGen-large [6]	1.5B	Yes	500	**1.8**
SpecMaskGIT (ours)				2.7
- w HiFiGAN				2.8
- w/o conditional mask	**170M**	**No**	**16**	3.2
- w/o CFG				3.1
- w/o CFG linear scheduler				3.1

a 2.5× BWE task. For all tasks above, we compute the Fréchet Audio Distance (FAD) using [43] since FAD is widely adopted to evaluate TTA [4, 9, 13, 14], time inpainting [4] and BWE [44] tasks. To investigate the representation learning potential of SpecMaskGIT, we further linear probe the model for the multi-label (genre, instrument and mood) **music tagging** task in MagnaTagATune (MTAT) [45] - a dataset widely used to evaluate music tagging models [46–49] - with ROC-AUC and mAP as metrics [46]. We use a single linear layer with batch normalization and 0.1 dropout as the probe.

5. RESULTS

5.1 Text-to-audio Synthesis

We report FAD scores of SpecMaskGIT in Tab. 1 together with other discrete models. Our model is first trained on AS for 500k steps and then finetuned on AC train set for 250k steps. The CFG scale is set to 3.0 empirically. SpecMaskGIT outperforms Diffsound (VQ-Diffusion), MAGNeT-small (similar to SpecMaskGIT but in latent wave domain), as well as AudioGen-base (autoregressive) in terms of FAD with one order of magnitude fewer iterations. The FAD score is achieved training without any audio-text pairs, which proves the effectiveness of such self-supervised approach for discrete models. We also find the proposed conditional mask described in Sec. 3.3 to improve FAD score without additional parameters or computations. Both CFG and its linear scheduler contribute to improve the FAD.

Given the small number of iterations and model size, SpecMaskGIT can synthesize realistic 10-second audio clips in real-time with only 4 cores of a Xeon CPU (Fig. 2), or 30× faster than real-time on an RTX-A6000 GPU, making it attractive for interactive applications and low-resource environments.

When compared to state-of-the-art (SOTA) continuous diffusion models in Tab. 2, SpecMaskGIT could not achieve a comparable FAD score, but we emphasize that the proposed method offers good performance with high efficiency, *i.e.*, smaller model size and fewer iterations, which can be clearly seen in Fig. 1. Overall, continuous methods are advantageous in terms of FAD with respect to discrete methods. We leave the further improvement of our discrete generative model as future work.

Ablation study: Gumbel noise and iterations number. We use HiFiGAN in all ablation studies. As mentioned in Sec. 3.4, Gumbel noise is essential to the top-k sampling during iterative synthesis. Fig. 7 shows that a temperature of 1.5 is optimal. SpecMaskGIT achieves good quality (FAD = 3.4) with only 8 iterations, and reaches its best (FAD = 2.8) with 16. More iterations do not improve performance, which is consistent with MaskGIT [10].

Table 2. *Benchmarking on AudioCaps test set. Dis.: discrete methods. Con.: continuous methods.*

Method	Params	Dis.	Con.	Num_iter	FAD
Diffsound [2]	400M	✓		100	7.8
Make-an-Audio [3]	330M		✓	100	4.6
MAGNeT-small [12]	300M	✓		180	3.2
AudioGen-base [6]	285M	✓		500	3.1
AudioLDM-Medium-full-FT [4]	420M		✓	100	2.6
AudioLDM-Large-full-FT [4]	740M		✓	200	2.0
Make-an-Audio 2 [9]	940M		✓	100	1.8
AudioGen-large [6]	1.5B	✓		500	1.8
AudioLDM2-Small-AC [14]	350M		✓	200	1.7
TANGO-AC [13]	870M		✓	100	1.6
AudioLDM2-Large-AC [14]	710M		✓	200	**1.4**
SpecMaskGIT (ours)	**170M**	✓		**16**	2.7

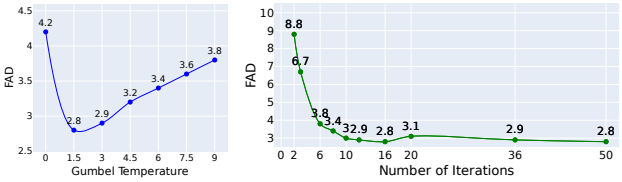

Figure 7. *Left: FAD vs. Gumbel temperature. Right: FAD vs. Number of iterations.*

Ablation study: Audio reconstruction quality. We evaluate the reconstruction FAD (rFAD) scores of two vocoders and SpecVQGAN in Tab. 3 with previous methods reported in [9]. Even with a similar architecture, rFAD of DiffSound and SpecMaskGIT can vary a lot due to different Mel computation and vocoder. Our pipeline achieves SOTA level rFAD scores for Mel-spectrogram methods while maintaining the highest compression rate (*i.e.*, the lowest latent rate) which helped SpecMaskGIT to outperform methods such as Diffsound and Make-an-audio by a large margin, yet with higher efficiency. We further analyze the rFAD of vocoders using ground truth input Mel-spectrograms, and find a significant performance gap between HiFiGAN and BigVSAN, which is not observed when vocoders are combined with SpecVQGAN. This indicates that SpecVQGAN is the bottleneck for reconstruction quality and asks for future improvements.

Ablation study: Bias in AudioCaps benchmark. The dataset gap between AC and other larger, more diverse datasets is investigated. It is observed in [4] that finetuning (FT) a TTA model on AC improves the TTA perfor-

Table 3. *rFAD of Mel-spectrogram VAEs and Vocoders on AudioCaps test set.* ***Bold****: best overall rFAD.*

Method	Mel-spec VAE	Vocoder	Latent rate	rFAD
Diffsound [2]	SpecVQGAN	MelGAN	27Hz	6.2
Make-an-audio [3]	VAE-GAN	HiFiGAN	78Hz	6.0
AudioLDM [4]	VAE-GAN	HiFiGAN	410Hz	1.2
Make-an-audio 2 [9]	VAE-GAN	BigVGAN	31Hz	**1.0**
SpecMaskGIT (ours)	-	HiFiGAN	27Hz	0.4
	SpecVQGAN			1.1
	-	BigVSAN	27Hz	0.1
	SpecVQGAN			**1.0**

Table 4. *Music tagging performance on MTAT.*

Method	CLMR [47]	MusiCNN [48]	MERT-330M [46]	MULE-contrastive [49]	Jukebox [22, 50]	SpecMaskGIT
mAP (%)	36.1	38.3	40.2	40.4	**41.4**	40.5
ROC-AUC (%)	89.4	90.6	91.3	91.4	**91.5**	**91.5**

Table 5. *AC test set performance w/ or w/out AC finetune.*

Method	Params	Num_iter	FAD	
			before FT	after FT
AudioLDM-Small-full [4]	180M	200	4.9	2.3
AudioLDM-Large-full [4]	740M	200	4.2	2.0
SpecMaskGIT (ours)	170M	16	4.2	2.8

Table 6. *Small-scale AudioCaps training results in better scores than large-scale dataset.*

Method	Params	Num_iter	FAD	
			Other datasets	AudioCaps
AudioLDM-Small [4]	180M	200	4.9	2.4
AudioLDM-Large [4]	740M	200	4.2	2.1
AudioLDM2-Small [14]	350M	200	2.1	1.7
AudioLDM2-Large [14]	710M	200	1.9	1.4
SpecMaskGIT (ours)	170M	16	4.2	2.9

mance in terms of FAD, though the model is pretrained on a larger dataset. We reproduced this phenomenon with SpecMaskGIT as shown in Tab. 5. We also observed that training on the small-scale AC alone brought better FAD score than the model trained with larger datasets in Tab. 6, which is consistent with [13, 14].

We hypothesize that there is a data distribution gap between AC and other datasets, such that when a model fully fits other datasets, the distribution of its synthesis deviates from AC, resulting in worse FAD. Therefore, we continued to train SpecMaskGIT on AS until 800k steps, and depict the "FAD vs. training step" curves on both the valid and test set of AC to verify our hypothesis. It is clear in Fig. 8 that SpecMaskGIT learns to synthesize audio in the early stage and keeps improving the FAD on AC. As the training goes on, SpecMaskGIT just fits toward AS, which worsens the FAD on AC.

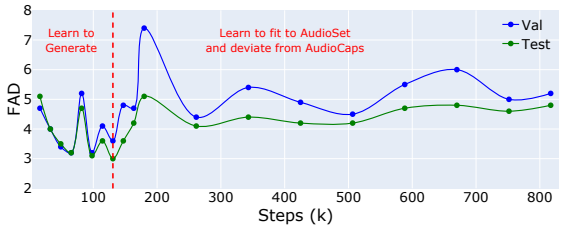

Figure 8. *FAD vs. AudioSet training steps.*

Inspired by audio classification tasks in which early stop is applied to prevent the model from overfitting to the train set, we propose to apply early stop to the SpecMaskGIT model trained solely on AS, and report the competitive FAD score with other methods that are without AC finetuning or AC-alone training in Tab. 7. We believe that a more comprehensive and less biased benchmark would contribute to future advances in TTA research.

5.2 Downstream Inpainting, BWE and Tagging Tasks

Results of the time inpainting and audio BWE tasks are shown in Tab. 8. We utilize the pipeline in Fig. 6 unconditionally, with Gumbel temperature 1.5 and 16 iterations. SpecMaskGIT significantly improves the input signals in terms of FAD, validating its zero-shot ability in

Table 7. *Benchmarking on AudioCaps test set without AC finetuning or AC-alone training.*

Method	Params	Dis.	Con.	Num_iter	FAD
Diffsound [2]	400M	✓		100	7.8
AudioLDM-Small-full [4]	180M		✓	200	4.9
Make-an-Audio [3]	330M		✓	100	4.6
AudioLDM-Large-full [4]	740M		✓	200	4.2
MAGNeT-small [12]	300M	✓		180	3.2
AudioGen-base [6]	285M	✓		500	3.1
AudioLDM2-Small-full [14]	350M		✓	200	2.1
AudioLDM2-Large-full [14]	710M		✓	200	1.9
Make-an-Audio 2 [9]	940M		✓	100	**1.8**
AudioGen-large [6]	1.5B	✓		500	**1.8**
SpecMaskGIT-AS-EarlyStop (ours)	**170M**	✓		**16**	2.9

Table 8. *Zero-shot time inpainting and BWE FAD scores.*

	BWE	Time inpaint
Unprocessed	2.7	1.6
SpecMaskGIT (ours)	1.5	**1.2**
- w/ LFR	**0.4**	-
Ground truth	0.0	0.0

such tasks. BWE performance can be further improved by applying low-frequency replacement (LFR) [51, 52]. Unlike prior arts that finetune MAE-like architectures for BWE [18, 19], SpecMaskGIT achieves it zero-shot. In Tab. 4, the potential of SpecMaskGIT in representation learning is confirmed by the music tagging performance on the MTAT dataset. As a TTA model, SpecMaskGIT outperforms classification-specialized models such as CLMR, MusiCNN, MULE, and MERT (the MAE-like model in wave domain). SpecMaskGIT achieves an ROC-AUC comparable to Jukebox, which contains 5B parameters. We hypothesize the tagging capability comes from the masked spectrogram modeling and ViT implementation similar to Audio MAE, as explained in Sec. 3. We leave the in-depth investigation of SpecMaskGIT in downstream tasks as future work.

6. CONCLUSION

Generative models that iteratively synthesize audio clips sparked great success to text-to-audio synthesis (TTA). However, due to the hundreds of iterations required for inference and the large amount of model parameters, high-quality TTA systems remain inefficient. To address the challenges, we propose SpecMaskGIT, a light-weight, efficient yet effective TTA model based on masked generative modeling of spectrograms. SpecMaskGIT synthesizes realistic audio clips in less than 16 iterations, an order of magnitude less than previous iterative TTA methods. It also outperforms larger discrete models in a TTA benchmark, while being real-time with 4 CPU cores and 30× faster with a GPU. Compared to similar methods, SpecMaskGIT is more flexible for downstream tasks such as zero-shot bandwidth extension. Moreover, we interpret SpecMaskGIT as a generative extension to Audio MAE and shed light on its audio representation learning potential. We hope our work inspires the exploration of masked audio modeling toward further diverse scenarios.

7. ETHICAL STATEMENT

SpecMaskGIT is supposed to assist creators in the sound design and editing workflow. Our method presents a huge advancement in the efficiency of TTA technology, which makes TTA accessible to a broader range of users, including creators who do not have GPUs. Despite of the technical advances, there is concern for the potential reflection of training data biases. The model may not be able to maintain a consistent sound quality or audio-text alignment when prompted by text descriptions or audio clips that are rarely presented in the training data. We also pointed out that the benchmark widely used to evalute TTA models in the research community is biased, and hope our findings here can contribute to a less biased benchmark in the future. The challenge in dataset bias emphasizes the importance for in-depth consideration and collaboration with stakeholders across various communities.

8. REFERENCES

[1] Y. Zhang, Y. Ikemiya, G. Xia, N. Murata, M. Martínez, W.-H. Liao, Y. Mitsufuji, and S. Dixon, "Musicmagus: Zero-shot text-to-music editing via diffusion models," *arXiv preprint arXiv:2402.06178*, 2024.

[2] 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.

[3] 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," in *International Conference on Machine Learning*. PMLR, 2023, pp. 13 916–13 932.

[4] 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," *arXiv preprint arXiv:2301.12503*, 2023.

[5] M. Comunità, R. F. Gramaccioni, E. Postolache, E. Rodolà, D. Comminiello, and J. D. Reiss, "Syncfusion: Multimodal onset-synchronized video-to-audio foley synthesis," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 936–940.

[6] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," *arXiv preprint arXiv:2209.15352*, 2022.

[7] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, "Audiolm: a language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[8] G. Li, X. Xu, L. Dai, M. Wu, and K. Yu, "Diverse and vivid sound generation from text descriptions," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[9] 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," *arXiv preprint arXiv:2305.18474*, 2023.

[10] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 315–11 325.

[11] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, "Vampnet: Music generation via masked acoustic token modeling," *arXiv preprint arXiv:2307.04686*, 2023.

[12] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "Masked audio generation using a single non-autoregressive transformer," *arXiv preprint arXiv:2401.04577*, 2024.

[13] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, "Text-to-audio generation using instruction-tuned llm and latent diffusion model," *arXiv preprint arXiv:2304.13731*, 2023.

[14] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "Audioldm 2: Learning holistic audio generation with self-supervised pretraining," *arXiv preprint arXiv:2308.05734*, 2023.

[15] K. Koutini, J. Schlüter, H. Eghbal-Zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *arXiv preprint arXiv:2110.05069*, 2021.

[16] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked spectrogram modeling using masked autoencoders for learning general-purpose audio representation," *arXiv:2204.12260*, 2022.

[17] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, "Masked autoencoders that listen," *NeurIPS*, vol. 35, pp. 28 708–28 720, 2022.

[18] Z. Zhong, H. Shi, M. Hirano, K. Shimada, K. Tateishi, T. Shibuya, S. Takahashi, and Y. Mitsufuji, "Extending audio masked autoencoders toward audio restoration," in *IEEE WASPAA 2023*, 2023, pp. 1–5.

[19] S.-B. Kim, S.-H. Lee, H.-Y. Choi, and S.-W. Lee, "Audio super-resolution with robust speech representation learning of masked autoencoder," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 1012–1022, 2024.

[20] S. Yang, Z. Zhong, M. Zhao, S. Takahashi, M. Ishii, T. Shibuya, and Y. Mitsufuji, "Visual echoes: A simple unified transformer for audio-visual generation," *arXiv preprint arXiv:2405.14598*, 2024.

[21] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[22] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[23] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[24] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[25] V. Iashin and E. Rahtu, "Taming visually guided sound generation," *arXiv preprint arXiv:2110.08791*, 2021.

[26] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," *arXiv preprint arXiv:2303.01469*, 2023.

[27] D. Kim, C.-H. Lai, W.-H. Liao, N. Murata, Y. Takida, T. Uesaka, Y. He, Y. Mitsufuji, and S. Ermon, "Consistency trajectory models: Learning probability flow ode trajectory of diffusion," *arXiv preprint arXiv:2310.02279*, 2023.

[28] K. Saito, D. Kim, T. Shibuya, C.-H. Lai, Z. Zhong, Y. Takida, and Y. Mitsufuji, "Soundctm: Uniting score-based and consistency models for text-to-sound generation," *arXiv preprint arXiv:2405.18503*, 2024.

[29] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *NeurIPS*, vol. 33, pp. 17 022–17 033, 2020.

[30] T. Shibuya, Y. Takida, and Y. Mitsufuji, "Bigvsan: Enhancing gan-based neural vocoders with slicing adversarial network," *arXiv preprint arXiv:2309.02836*, 2023.

[31] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. De Brebisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[32] 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.

[33] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[34] T. Li, H. Chang, S. Mishra, H. Zhang, D. Katabi, and D. Krishnan, "Mage: Masked generative encoder to unify representation learning and image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2142–2152.

[35] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR 2021*, 2021.

[37] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[38] H. Chang, H. Zhang, J. Barber, A. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein *et al.*, "Muse: Text-to-image generation via masked generative transformers," *arXiv preprint arXiv:2301.00704*, 2023.

[39] V. Besnier and M. Chen, "A pytorch reproduction of masked generative image transformer," *arXiv preprint arXiv:2310.14400*, 2023.

[40] J. F. Gemmeke, D. P. 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 *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[41] R. Wightman, "Pytorch image models," https://github.com/rwightman/pytorch-image-models, 2019.

[42] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 119–132.

[43] "A lightweight library of frechet audio distance (fad) calculation," https://github.com/gudgud96/frechet-audio-distance.

[44] E. Moliner, M. Turunen, F. Elvander, and V. Välimäki, "A diffusion-based generative equalizer for music restoration," *arXiv preprint arXiv:2403.18636*, 2024.

[45] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International Conference Center, Kobe, Japan, October 26-30, 2009*, K. Hirata, G. Tzanetakis, and K. Yoshii, Eds. International Society for Music Information Retrieval, 2009, pp. 387–392. [Online]. Available: http://ismir2009.ismir.net/proceedings/OS5-5.pdf

[46] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Y. Guo, and J. Fu, "Mert: Acoustic music understanding model with large-scale self-supervised training," 2023.

[47] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," *arXiv preprint arXiv:2103.09410*, 2021.

[48] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," *arXiv preprint arXiv:1909.06654*, 2019.

[49] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. Ehmann, "Supervised and unsupervised learning of audio representations for music understanding," in *ISMIR 2022*, 2022.

[50] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," *arXiv preprint arXiv:2107.05677*, 2021.

[51] H. Liu, W. Choi, X. Liu, Q. Kong, Q. Tian, and D. Wang, "Neural vocoder is all you need for speech super-resolution," *arXiv preprint arXiv:2203.14941*, 2022.

[52] H. Liu, K. Chen, Q. Tian, W. Wang, and M. D. Plumbley, "Audiosr: Versatile audio super-resolution at scale," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1076–1080.