Application of Masked Token Transformer (MaskGIT) in high-quality and efficiency audio generation

Pengyu Zou
pz2272@columbia.edu

Yiru Gong
yg2832@cumc.columbia.edu

Baode Gao
bg2715@cumc.columbia.edu

Abstract

After the proposal of Transformer, different methods has been widely adopted in the areas of images synthesis and audio generation. Inspired by a novel image synthesis paradigm using a bidirectional transformer decoder termed MaskGIT, which accelerates autoregressive decoding by up to 64x, we adopted this method to the audio generation area. In this project, we completed the Pytorch implementation of pre-trained MaskGIT model on ImageNet dataset (subset of 3900 pictures) and replaced the VQGAN model in the first stage with a pre-trained VQGAN on VAS dataset (subset of 2061 audio) called SpecVQGAN. During the training, our model takes the spectrograms of audio as input, encodes the spectrograms to latent tokens, and then generating the output simultaneously in parallel. Experimental results of our model based on MaskGIT are also conducted to validate the application value of this method in audio generation. We evaluate our model with Fréchet Inception Distance (FID) and Inception Score (IS). The results indicate that MaskGIT has great potential to be applied in audio generation area.

1 Introduction

Thanks to the rapid progress of deep generative models such as Generative Adversarial Networks (GANs) [1], we have witnessed an amazing improvement in fidelity and modeling speeds in the field of deep image synthesis and audio generation. However, the limitation in lack of sample diversity, introduced by training instability and mode collapse of the GAN-based models, is always a problem in research. Plus, the best generative transformer models so far, however, still treat an image naively as a sequence of tokens, and decode an image sequentially following the raster scan ordering. Audio generation and image synthesis are similar tasks that shared the similar problems that GAN-based audio generation models generate sounds conditioned on visual input from multiple classes with a restricted time budget. Addressing these issues in these two areas still remains open research problems.

Inspired by the success of Transformer[16] and GPT[2] in NLP, generative transformer models have received growing interests in image synthesis and audio generation. To address the problems mentioned above, recent studies[4, 7, 13] tried to adopt autoregressive models (e.g., transformers) in NLP to image generation tasks by quantizing images into sequential tokens and training the model for autoregressive decoding to generate new tokens, namely generative transformers. These approaches consider modeling the input as a sequence and apply autoregressive model to generate output. Both audio generation and image synthesis can be divided into two stage: the first stage is to quantize the input(images or spectrograms) to a sequence of discrete tokens (or visual words). In the second stage, an autoregressive model (e.g., transformer) is learned to generate tokens sequentially based on the previously generated result. Unlike the subtle min-max optimization used in GANs, these models are

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

learned by maximum likelihood estimation. Because of the design differences, existing works have demonstrated their advantages over GANs in offering stabilized training and improved distribution coverage or diversity that pose great potential to be applied in audio generation area.

In order to employ generative transformer models in audio synthesis, most approaches model the spectrograms as a sequence and use existing autoregressive models to generate the image[10, 5]. As a result, even the most advanced generative transformers still naively treat spectrograms as sequences, where images are scanned from left to right line-by-line[12, 8]. Considering that artworks often start with a sketch and then gradually refine the sketch by filling in or adjusting details, which contrasts with the previous line-by-line printing, the article[3] provided a new bidirectional transformer framework in image generation (MaskGIT) which uses the non-autoregressive decoder to synthesize images in a constant number of steps. Their goal is to design a new paradigm for image synthesis using parallel decoding and bi-directional generation to handle the concern of images' spacial structure and improve training speeds. Inspired by the decoding idea in MaskGIT, we replaced the VQGAN of the first stage with a pre-trained VQGAN on VAS dataset called SpecVQGAN[10] to learn a prior in a form of the Vector Quantized Variational Autoencoder (VQVAE) codebook[14] and operate on spectrograms for efficiency.

The most relevant to our model is VQGAN[8, 14]. It consists of two components, learning the effective spectrograms constituents codebook by transformers and learning image composition with Transformers. As shown in Figure1, our model includes two stages. For the first stage, there is a Vector Quantised GAN model[8] from the SpecVQGAN[14] pre-trained model to learn discrete representations of spectrograms. In the second stage, trained Masked Visual Token Modeling predicts all tokens simultaneously in parallel, but keeps only the most deterministic ones. The remaining tokens will be masked and will be re-predicted in the next iteration. The goal of the model is to generate higher quality and diversity samples. In addition, the multi-directional nature of our model makes it easy to extend to image and audio manipulation tasks, like in-painting, out-painting and editing.

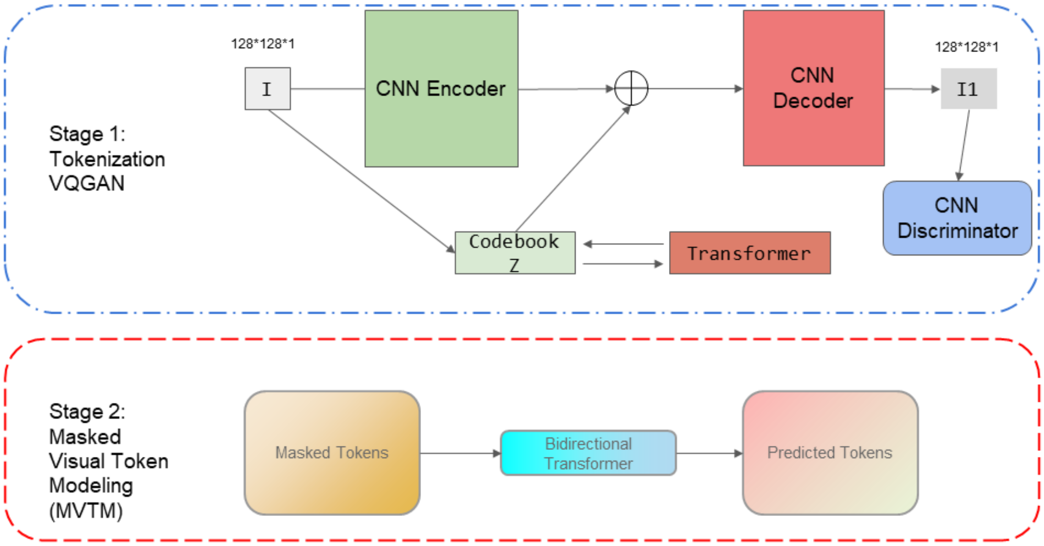

Figure 1: MaskGIT Pipeline Overview[3]

On the VAS benchmark, we empirically demonstrate our model is capable of generating higher quality samples and evaluate our model with Fréchet Inception Distance (FID) and Inception Score (IS). In addition, we transform the output spectrograms into audio. The demonstration will be presented in github.

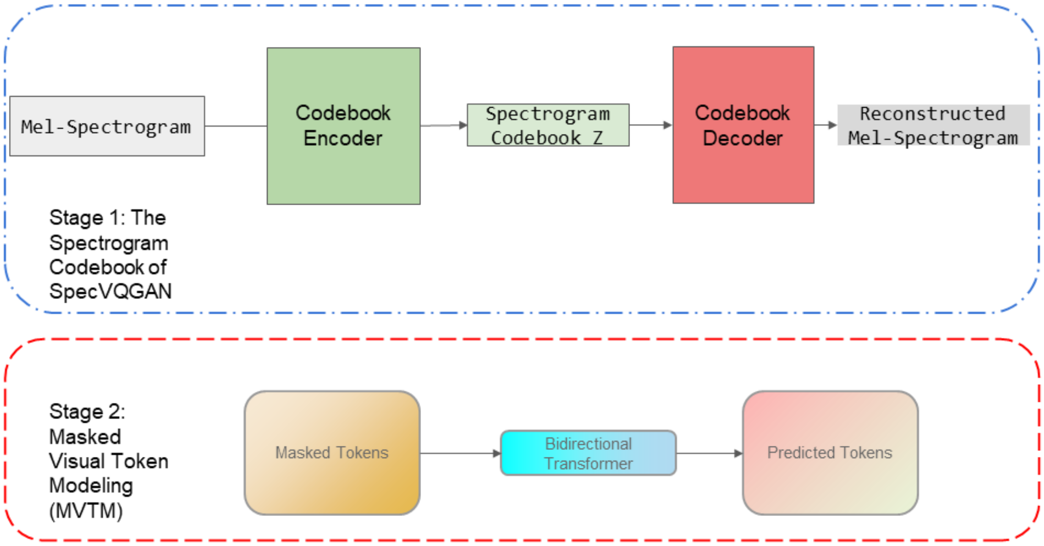
Mel-Spectrogram
Codebook Encoder
Spectrogram Codebook Z
Codebook Decoder
Reconstructed Mel-Spectrogram
Stage 1: The Spectrogram Codebook of SpecVQGAN
Stage 2: Masked Visual Token Modeling (MVTM)
Masked Tokens
Bidirectional Transformer
Predicted Tokens

Figure 2: SpecVQGAN+MVTM

2 Method

The process of generating audio can be split into two stages[11, 15]. In the first stage, which is known as tokenization, it tries to compress into discrete latent space. In the second stage, it first predicts the latent priors of the visual token using deep autoregressive models, and then use the decoder from the first stage to map the token sequences into image pixels.

In the MaskGIT, the authors follow the two-stage recipe and propose a new image synthesis paradigm utilizing parallel decoding and bi-directional generation. Their goal is to improve the second stage, so they employ the same setup for the first stage for the first as in the VQGAN model[8]. Hence, we employ the same setup for the first stage as in the VQGAN model. For the second stage, they propose to learn a bidirectional transformer by Masked Visual Token Modeling (MVTM).

In our implementation of MaskGIT, we first use VQGAN pre-trained model on ImageNet dataset with f=16 (16x compression per spatial dimension) and with a larger codebook (16384 entries) for the tokenization. This VQGAN is further replaced with the SpecVQGAN model on VAS dataset for audio generation. In the second stage, we change the input dimension of tokens according to the codebook in the first stage and train the model on VAS.

2.1 SpecVQGAN

The transformer requires the input to be represented as a sequence. A direct operation on wave samples or raw spectrogram pixels, however, quickly becomes intractable due to the quadratic nature of the dot-product attention. Alternatively, one could apply an encoder such as VQVAE but the quantized bottleneck representation would be still infeasibly large. In our method, we employ an efficient autoencoder that allows decoding an spectrograms from a smaller-size representation than of VQVAE.

For this Spectrogram VQVAE, the model inputs a spectrogram $x \in R^{F \times T}$ and outputs a reconstructed version of it $\hat{x} \in R^{F \times T}$. First, the input x is encoded into a small-scale representation $\hat{z} = E(x) \in R^{F^{'} \times T^{'} \times n_z}$ where n_z is the dimension of the codebook entries and $F^{'} \times T^{'}$ is reduced frequency and time dimension. Next, the elements of the encoded representation $\hat{z}$ are mapped onto the cloest items in a codebook $Z = \{z_k\}_{k=1}^{K} \subset R^{n_z}$, forming a quantized representation $z_q \in R^{F^{'} \times T^{'} \times n_z}$:

$$z_q = q(\hat{z}) := (\underset{z_k \ inZ}{\arg\min} ||\hat{z}_{ft} - z_k|| \quad for\ all\ (f,t)\ in\ (F^{'} \times T^{'}))$$

In our project, we implemented the dataset initialization and input the VAS data into our VQGAN. Then, the codebook generated by VQGAN is fed by us to next stage model.

2.2 MVTM in Training

In the second stage, instead of conditioning only on previous tokens in the order of raster scan, bidirectional self-attention allows the model to generate new tokens from generated tokens in all directions.

After obtaining the latent tokens of the inputting spectrograms from the SpecVAGAN $Y = [y_i]_{i=1}^{N}$, where N denote the length of the reshaped token matrix. Like the pre-training method of BERT[6], we choose a subset of latent tokens and replace them with a special $[MASK]$ token. Let $M = [m_i]_{i=1}^{N}$ denote the mask matrix corresponding to the token matrix. If $m_i = 1$, y_i will be masked with the $[MASK]$ token, otherwise, y_i remains.

In MaskGIT, the sampling process is scheduled by a mask scheduling function $\gamma(r) \in (0,1]$ which executes as follows: we sample a ratio from 0 to 1 and uniformly select $\lceil\gamma(r) \times N$ tokens in Y. These selected tokens are replaced with $[MASK]$ tokens, and we obtain $Y_{\overline{M}}$. The ratio here will be generated by the mask scheduling function in the following subsection.

Once we obtain the results after masking, $Y_{\overline{M}}$, the training objective is to minimize log-likelihood of the masked tokens:

$$L_{mask} = -E_{Y\ inD}[\sum_{\forall i \in [1,N], m_i=1} \log p(y_i | Y_{\overline{M}})]$$

We feed the $Y_{\overline{M}}$ into a multi-layer bidirectional transformer to predict the probability of each mask token in $Y_{\overline{M}}$. In this process, we employ negative log-likelihood above to compute the cross-entropy between the predicted token and ground truth.

2.3 Masking Design

As mentioned in the paper of MaskGIT, the mask scheduling (i.e. fraction of the latent tokens masked each iteration) significantly affects generation quality. The mask function is responsible for generating the ratio of masked tokens to be decoded. During inference time, we our mask scheduling function determines in how many iteration the tokens can be predicted.

The authors of MaskGIT proposed that the masking scheduling function $\gamma(r)$ should have following properties. First, this function needs to be a continuous function bounded between 0 and 1. Second, this function $\gamma(r)$ needs to be monotonically decreasing with respect to r, which ensure that decoding process will converge. The authors employed three masking scheduling function to compare, include Linear function, Concave function and Convex function. We adopt the cosine function which performs the best in the experiment result of MaskGIT and implement the code in Pytorch.

2.4 Decoding

In the decoding of traditional autoregressive model, the tokens are generated sequentially based the previous results. However, this process is unnecessarily slow when dealing with images or spectrograms which is much larger than than language sequence. Thanks to the bi-directional self-attention mechanism of Transformer in MTVM, we can decode use a novel decoding method where all tokens are generated simultaneously in parallel.

As we obtain the results of masking latent token $Y_{\overline{M}}$ from SpecVQGAN, we can runs the following algorithm for t iterations:

1. **Predict.** At t iteration, the model predicts the probability given the $Y_{\overline{M}}$ for all the masked locations in parallel.
2. **Sample.** After obtaining the probabilities of all masked tokens, we sample one token y_i^t based on the probability over all possible tokens in the codebook. The corresponding prediction score will be treated as "confidence" score to represent the confidence of our model in this prediction. For the unmasked tokens, their scores are set to 1.0.
3. **Mask Schedule.** According to the mask scheduling function $\gamma(r)$ (here we employ cosine function), we can compute the number of tokens to be masked.
4. **Mask.** For $t+1$ iteration tokens, the mask matrix can be computed based on the mask matrix of t iteration as follows: $m_i^{(t+1)} = 1$ if $c_i < sorted_j(c_j)[n]$. Otherwise, $m_i^{(t+1)} = 0$. c_i denote the confidence score corresponding token.

The decoding algorithm generates an spectrograms in T steps. At each iteration, the model predicts all tokens simultaneously but only keeps the most confident ones. The remaining tokens are masked out and re-predicted in the next iteration. The mask ratio is made decreasing until all tokens are generated within T iterations.

3 Results

The MVTM is trained on ImageNet 256×256 for image generation tasks and on VAS (.mp4) for audio generation task. We evaluate MaskGIT with Fréchet Inception Distance (FID) and Inception Score (IS) on 500 ImageNet pictures and 500 VAS audio. Training the model on the entire ImageNet data requires 170 hours per epoch, and 50 epochs in total are set for model training. Due to the time and GPU resource limitation, we decided to train on a subset of ImageNet (3900 pictures) and baby VAS audio (2061 audio). In this section, we show our results of FID and IS as well as the synthesized image and spectrogram. Project page at `https://github.com/HawkDukez/W4995DL-Final`.

3.1 Evaluation Metrics

- Fréchet Inception Distance (FID)[9] is used to evaluate the quality of images created by generative models such as GAN. It evaluates the distribution of the generated images, and a lower score means that the two sets of images are more similar.

$$\text{FID} = \|\mu - \mu_w\|_2^2 + tr\left(\Sigma + \Sigma_w - 2\left(\Sigma^{1/2}\Sigma_w\Sigma^{1/2}\right)^{1/2}\right)$$

- Inception Score (IS) is an indicator to evaluate the quality of generated images based on clarity and diversity. The larger the better.

$$\mathbf{IS}(G) = \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}(p(y \mid \mathbf{x}) \| p(y))\right)$$

Table 1: Evaluation Score

Model	FID	IS
MaskGIT(image)	1.825	1.138 ± 0.047
SpecVQGAN+MaskGIT(audio)	1.068	1.003 ± 0.0011

The results are not ideal, because of the limited train size and test size.

3.2 Synthesized image

The four image indicate original image, fully reconstructed image, half-masked token, and fully-masked token from left to right. Here we showed that due to the fine-tuned VQGAN model, the reconstruction of image performs well. However, due to the training set limitation, the bidirectional

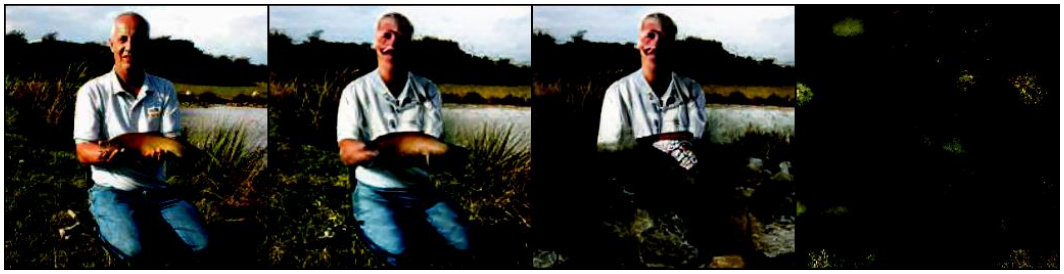

Figure 3: Synthesized Image

transformer is not adequately trained and thus perform worse. In the future, we could either trained the transformer model on a larger sample size or training our own first stage VQGAN model with a smaller code-book vector size.

3.3 Synthesized audio

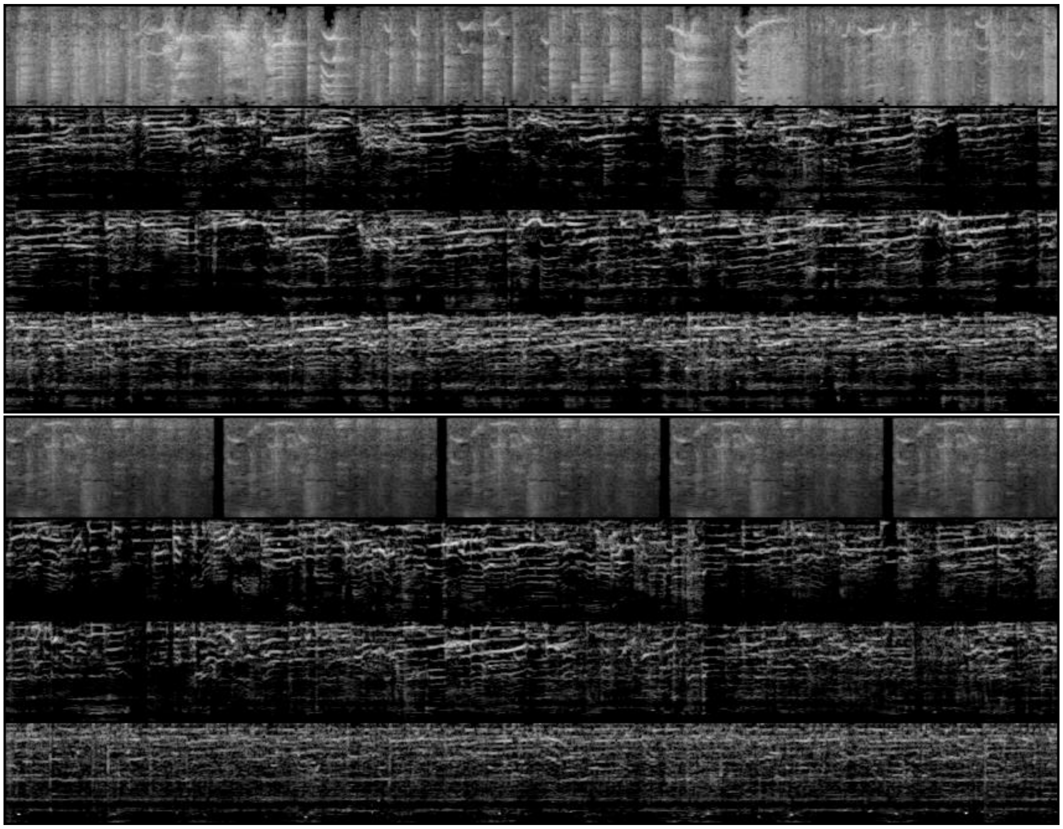

Figure 4: Synthesized Audio with log transformation (input, reconstruction, half-masked, new)

We applied log image transit in the transformer indices and compare the results of models with direct reconstruction, half-masked, and fully new indices examples (Figure 4). Unlike the image model, we found that the reconstruction audio did not recover the input audio well, and perform similarly to the half-masked indices. However, the entire new sample turned out with better results and present a more complete audio spectrum, which suggested a positive effect of the MaskGIT transformer.

The training process is also faster in audio, even though we use a 80*848 input size of spectrum image. The major reason is that a smaller code-book size of 265 is applied in audio, which significantly reduce the burden of transformer training. This is also the main reason of a more efficient transformer in audio model than image model. In the image model, we applied the fine-tuned pre-trained VQGAN model with a codebook vector size of 16384, which is more complicated than the code-book size indicated in the original article (1000 indices) and the usual LVTM model code-book size. With limited computational power and resources, it is thus more difficult to trained the transformer efficiently. However, in the audio model, we applied a much smaller codebook of 265 indices based on the pre-trained SpecVQGAN model. Therefore, the complexity of transformer is largely reduced and thus performs better under the situation of limited resources and training sample size.

4 Conclusions and limitations

In this project, we first implement pre-trained VQGAN and MVTM on image generation task. Then, SpecVQGAN and MVTM also are trained and evaluated by VAS data. The results showed a faster training speed and audio generating ability of MaskGIT model when comparing to other models under the same computer resources. However, due to the limited computational resources, the scale of training and testing is quite limited, which leads to worse performances. The future implementation thus lies in completing the training process on the entire ImageNet and VAS data. Meanwhile, the MaskGIT model is also flexible for adopting video, image, and word keys to facilitate audio generation. In general, we showed a new application of MVTM masking transformers in the audio generation field.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. *arXiv preprint arXiv:2202.04200*, 2022.

[4] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020.

[5] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *CoRR*, abs/2012.09841, 2020.

[8] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[10] Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. *arXiv preprint arXiv:2110.08791*, 2021.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[12] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[13] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. *CoRR*, abs/1906.00446, 2019.

[14] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016.

[15] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.