4-1-2022

How can generative adversarial networks impact computer generated art? Insights from poetry to melody conversion

Sakib Shahriar
Zayed University

Noora Al Roken
American University of Sharjah

Recommended Citation

Shahriar, Sakib and Roken, Noora Al, "How can generative adversarial networks impact computer generated art? Insights from poetry to melody conversion" (2022). *All Works*. 4933.
https://zuscholars.zu.ac.ae/works/4933

Contents lists available at ScienceDirect

International Journal of Information Management Data Insights

journal homepage: www.elsevier.com/locate/jjimei

How can generative adversarial networks impact computer generated art? Insights from poetry to melody conversion

Sakib Shahriar[a,*], Noora Al Roken[b]

[a] College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates
[b] Department of Computer Science and Engineering, American University of Sharjah, Sharjah, United Arab Emirates

ARTICLE INFO

Keywords:
Generative adversarial networks
Deep learning
Melody generation
Music generation
Artificial intelligence
Computer generated art

ABSTRACT

Recent advances in deep learning and generative adversarial networks (GANs), in particular, has enabled interesting applications including photorealistic image generation, image translation, and automatic caption generation. This has opened up possibilities for many cross-domain applications in computer generated arts and literature. Although there are existing software-based approaches for generating musical accompaniment of a given poetry, there are no existing implementation using GANs. This work proposes a novel poetry to melody generation conditioned on poem emotion using GANs. A dataset containing pairs of poetry and melody based on three emotion categories is introduced. Furthermore, various GAN architectures including SpecGAN and WaveGAN were explored for automatic melody synthesis for a given class of poetry. Conditional SpecGAN produced the best melodies according to quantitative metrics. Melodies produced by SpecGAN were evaluated by volunteers who deemed the quality to be above average.

1. Introduction

Computer art was first discovered as a way to produce objective art pieces by relying on probability theories (Chamberlain et al., 2018). Since the 1970s, the artist Harold Cohen started testing with computer sketch generation, where he manually coded the painting structure into the program (Chamberlain et al., 2018). Unlike artificial intelligence (AI), regular computer programs could not intelligently produce a new art piece without the designer's intervention. With the rapid developments of AI, computer-generated drawings, music, and poetry have improved. Advanced learning algorithms and evolutionary models have generated complex and realistic artworks (Chamberlain et al., 2018). In 2018, the portrait of Edmond de Belamy was sold for $430,000 in an auction, at a much higher price compared to the anticipated price tag of $7,000-$10,000 for being a computer-generated art (Goold, 2021). The Edmond de Belamy case was the first case of an AI-generated artwork in an auction where the French creators fed 30,000 portraits to an available machine learning algorithm and selected the best-generated picture (Goold, 2021). Finally, the owners added their signatures to the bottom right corner. Furthermore, recent interests in adversarial training for artwork generation have grown significantly. The network tries to generate realistic outputs, and the critic or discriminator tries to distinguish between the real and generated samples. Earlier, the human played the critic's role, and the program was a form of an evolutionary algorithm or a neural network (Soderlund & Blair, 2018). A group of outputs (e.g., images or music) are presented to the critic, and a subgroup with the best results is selected and passed to the next stage or iteration. Such a process is time-consuming, and hence, the entire procedure has been automated by replacing the critic with a neural network that can learn discriminant features to distinguish between the inputs (Soderlund & Blair, 2018). Consequently, researchers and artists in recent times have started utilizing adversarial learning to generate various art forms including visual, textual, and audio arts.

Generating music that express a particular emotion conveyed by a poem helps the reader understand the piece of literature on a deeper level. Music accompaniments for poems can also be used to enhance the mood of the poetry readers. However, insufficient work has been carried out for generating musical pieces for poems while considering the emotion of the text. To the best of our knowledge, no research application has yet been proposed that relates to poem and melody generation using deep learning. The existing literature mainly focus on generating music for other art forms such as song lyrics. Hence, in this work, we present the first Arabic poetry to melody generation utilizing generative adversarial networks (GANs). GANs can generate new content based on a min-max game between two networks, the generator and the discriminator (Ruzafa, 2020). The generator tries to generate fake samples and improve its network to trick the discriminator. Meanwhile, the discriminator tries to distinguish between the real and fake samples. More sophisticated GAN architectures can make use of labels to

[*] Corresponding author.
E-mail address: sakib.shahriar@zu.ac.ae (S. Shahriar).

https://doi.org/10.1016/j.jjimei.2022.100066
Received 8 November 2021; Received in revised form 26 February 2022; Accepted 26 February 2022

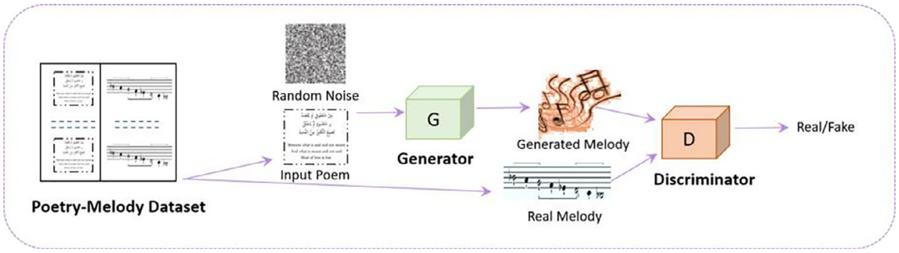

Fig. 1. Graphical illustration of the proposed application.

generate data points for a specific category. This is useful because in the standard GAN, data points are generated only based on the input noise and unconditional generation of data points is not suitable for many applications.

Although GANs have been widely utilized for various applications including 3D object generation, medicine, and traffic control (Aggarwal et al., 2021), their adoption in audio data and more specifically cross-domain audio generation including text to audio and image to audio generation is very limited. The specific problem of text to audio transformation can be highlighted further by a related research application, text-to-speech (TTS) transformation. Most TTS processes carried by the neural network models are auto-regressive (AR) which means that the speech segments are generated based on the previously synthesized segments. AR methods do not support parallelization (i.e., the use of GPU) due to their sequential nature (Prenger et al., 2019), (Li et al., 2019). Consequently, AR models are not efficient for real-time applications because of the consecutive calculations that require time (Okamoto et al., 2019; Ren et al., 2019) added how the commonly used AR TTS models such as Tacotron, Tacotron 2, Deep Voice 3, and Clarinet generate Mel-spectrograms from the text inputs for audio synthesis. The use of spectrograms further slows down the audio generation since each input is converted to a Mel-spectrogram, and the sequence length is hundreds or thousands. Mel-spectrograms are susceptible to word-skipping or repeating, which causes inconsistency between the text and generated speech (Ren et al., 2019). Besides the difficulties in TTS generation, melody or music generation comes with its own sets of challenges Dieleman et al. (2018) explain why music generation has received less attention than speech due to its complexity. MIDI and other symbolic music representations have been used in literature to represent the music concisely. However, many essential features that describe the instruments used are removed when using symbolic representations. The piano can be translated into MIDI sequences, but it is difficult to represent most instruments. It is more efficient to use raw music audio signals since it retains the needed information. However, it is even more difficult to model it i.e., capture musical structure over an extended period. Therefore, the proposed text to audio generation application investigates the competency of GANs with audio data. The objective of this study is to address some of the existing challenges in audio generation and further examine the audio generation from a given input text which has not received sufficient attention in existing research works.

A high-level graphical illustration of the proposed framework is presented in Fig. 1. The generator utilizes the poem from the dataset along with some noise to synthesize new data points that would ultimately resemble the melody corresponding to the input poem. The generator does so by optimizing the loss function that minimizes the difference between the generated sample and the real melody. The role of the discriminator, in this case, is to provide feedback to the generator by attempting to distinguish between the generated and the actual melody. After the training phase, the generator would ideally learn the representation between the input poetry and the corresponding melody. In Section 3, a detailed explanation of the type of GAN architecture as well as the experimental setup used in this work is presented.

There are several challenges to solving this problem of melody generation for poetry. The most natural solution of employing human musicians and poets to collaborate in solving this task is not scalable due to high costs. Moreover, there is a challenge in terms of the language of the poem and the music associated with the culture of the written text. Therefore, for practical usage, the solution would be to utilize computers to generate melodies. The use of programming languages and extensive human supervision involved was a major drawback in older approaches to music generation by computers. In this context, the computer's role was to suggest syntheses and leave the human expert to complete the feedback loop and make the final decision of accepting or rejecting the suggestions (Noll, 1967). Functions for amplitude, frequency, and duration of a sequence of notes were drawn on a cathode-ray tube with a light pen and the computer provided synthesis by combining these functions using simple algorithms (Noll, 1967). Consequently, the melodies produced by such computer programs sounded uniform and artificial. The challenging aspect of the computer music generation comes from the fact that human supervision is required at some level. Besides the human supervision, it is not straightforward to program the computer to transform the poetry into melody to capture the mood of a poem. This would require further resources to first develop a program that can process the language within the poetry. The solution to both the need for human supervision as well as language to audio representation learning can only be achieved if computers are trained like humans, i.e., to learn from experience, using lots of data and without explicit programming. This is where deep learning and more specifically GANs are extremely useful. GANs are good at transforming inputs from one domain to another (Ruzafa, 2020), and consequently, in our work, it can help with transforming the poem, which is a text input into a representative melody, an audio output. Research in images generated using GANs have already shown to be realistic where humans cannot tell whether a particular art piece was produced by a human artist or a machine (Mazzone & Elgammal, 2019). Therefore, GANs will not only provide a more flexible implementation but will also produce more realistic and diverse melodies that can capture the imagination of the poetry readers.

2. Background

This section presents the related works in the literature pertaining to melody and lyrics generation as well as background information on GAN architectures.

2.1. AI and computer generated art

GANs have been effectively used in various art generation applications including visual arts, written arts, music generations, calligraphy style recognition, and melody classification (Shahriar, 2021). A review of the literature in the context of related melody and music generation applications is presented next. Scirea et al. (2015) proposed a system using two Markov chains to generate lyrics from academic research papers along with appropriate melodies to accompany the music. However, this work did not provide an evaluation case study on the quality of the lyrics and melodies generated. In Ackerman and Loker (2017), machine learning models including random forest were used to create an application that is capable of assisting musicians and amateurs in producing quality songs and music. Although three songs were created using the application, they were not evaluated by case studies. A system that can analyze the mood of the poem and generate a relevant musical component was presented in (Stere & Trăuşan-Matu, 2017). Firstly, a poem was analyzed by considering its rhythm, punctuation, and mood. Based on the analysis, a piano composition was generated to complement the mood of the poem. Machine learning was used to classify the mood of the poem. The author utilized a pre-trained Naïve Bayes classifier called Music-Mood (Raschka, 2016) to classify the poem text as “happy” or “sad”. The paper failed to address the accuracy of the mood classification as well as the quality of the generated music. Milon-Flores et al., 2019 presented an application to generate audiovisual summaries of literary texts by producing emotion-based music composition and graph-based animation. Natural language processing algorithms were used to extract emotions and characters from literary texts but no machine learning frameworks were used in this work. Fukayama et al. (2010) developed an automated Japanese song composer, "Orpheus." The application used the Galatea-Talk tool to convert the textual lyrics to speech and the hidden Markov model to synthesize singing voice. A survey was conducted on 1378 people concluded that 70.8% of the songs were attractive. Monteith et al., (2012) generated melody for different genres including nursery rhymes and rock songs by extracting the rhythm and pitch from the corresponding lyrics. The results from a case study indicated that the original melodies were more familiar, but the differences were not significant. Generation of music from text was proposed in a system called TransPose (Davis and Mohammad, 2014), which exploited the relationship between various characteristics of music (such as loudness, melody, and tempo) and the emotions they invoke. For example, Loud music corresponds to power or anger while fear or sadness is linked with soft music. One drawback of this approach is that mid-piece key changes cannot be captured with this technique. For further comparison of symbolic music generation techniques, readers are encouraged to refer to the following survey (Briot et al., 2020).

Mishra et al. (2019) proposed an RNN-based generative model for melody generation. The mode utilized LSTM architecture to learn the input structure from a text file and was trained to predict the next character in the sequence. It was concluded that the human-generated and machine-generated melodies could not be distinguished from a survey. In Welikala and Fernando (2020), musical instrument digital interface (MIDI) files, which contain data to specify the musical instruction such as note’s notation and pitch, were used to train a hybrid variational autoencoder and GAN to generate musical melody for a specific genre. It was concluded that the consistency of generated melodies was not up to the same level as human composition. Similarly, Xu et al. (2020) proposed a melody generation framework using GAN, consisting of a Bi-LSTM generator and an LSTM discriminator. The proposed model on average obtained a score of 3.27 out of 5 by 19 human evaluators, outperforming the baseline models. Yu et al. (2020). used conditional GANs for melody generation. The lyrics encoder tokenized the lyrics into a vector containing a series of word and syllable tokens as well as additional noises. The output of the layer was fed into the conditioned-lyrics GAN. MIDI sequence turner was then used to quantize the MIDI notes produced by the generator. A similar study was conducted in (Conditional LSTM-GAN for Melody Generation from Lyrics, 2021), which used conditional-LSTM GANs. Comparison with random and maximum likelihood estimation baseline showed that the proposed model outperformed the baselines in terms of BLEU score. Bao et al., 2019 proposed a melody composer, "songwriter" to generate lyrics-conditioned melody and its alignment. The authors divided the lyrics into sentences that are processed in sequence. The set of generated melodies was aggregated for each lyric to generate the complete melody. The evaluation using F1 and BLEU scores showed that the proposed model outperformed existing models on all metrics. Dias and Fernando (2019) introduced a system with an interface that generates melody using a sequence of musical notes given English lyrics without any human intervention. The LSTM was used to train and predict the melody target data. The human evaluation result showed that the generated melody was pleasant to listen to because of the well-developed notes. The authors in Deep Cross-Modal Correlation Learning for Audio and Lyrics in Music Retrieval | ACM Transactions on Multimedia Computing, Communications, and Applications (2021) introduced a multimodal music retrieval that retrieves audio from lyrics data and vice versa. MFCC features were extracted from music audio signals using CNN, and Doc2vec is used to embed lyrics documents. Based on PixelCNN (van den Oord et al., 2016), WaveNet (van den Oord et al., 2016) is a model for audio generation. It is possible to generate raw audio waveforms with this model. Although case study results were presented, evaluation metrics were not reported. In Performance RNN: Generating Music with Expressive Timing and Dynamics (2021), an LSTM models the music in symbolic form using MIDI files. Yamaha e-Piano Competition dataset was used for training and to supplement the training data, the augmentation of each record (e.g., speed and transposition) were also used. Similarly, Boulanger-Lewandowski et al. (2012) explored the application in polyphonic music generation using the restricted Boltzmann machine.

2.2. Deep learning and adversarial networks

Deep learning is a subset of machine learning, where computer systems learn from experience by getting exposure to training data and without requiring explicit programming. Deep learning architectures are different from traditional machine learning ones because they utilize artificial neural networks (ANNs), a biologically inspired computing technique that replicates the way neurons function in intelligent organisms (Kar, 2016). In many applications that are of high complexity, other forms of bio-inspired computation may be combined with neural networks to offer a hybrid decision support system (Chakraborty & Kar, 2017). In machine learning, an extensive selection of input features to train the algorithms for making intelligent decisions is required which makes the process time-consuming. Meanwhile, deep learning models generally learn features from the input training data which makes them far more effective for tasks such as image and audio classification. While simple ANNs contain a basic structure and models with less than three layers are known as shallow networks, deep learning models have an increasing depth of more than three layers (Bouwmans et al., 2019). The organization in layers allows more complex representations of a problem to be represented in terms of simpler representations. Other types of deep learning models are suitable for specific sets of problems depending on the nature of the dataset. For instance, convolutional neural networks (CNNs) utilize convolution operation to merge two sets of information and a subsequent pooling layer reduces the dimensionality and complexity (Dhillon & Verma, 2020). CNNs are therefore effective in image and video applications such as action recognition from video footage (Yao et al., 2019). One of the factors that enabled the recent success of deep learning models is the massive volume of available data allowing big data analytics which is the process of obtaining meaningful information and insights from a variety of data sources (Grover & Kar, 2017). The idea of learning from experience to classify or predict variables can be extended to generating new data points using adversarial networks.

GANs primarily consist of two deep learning architectures namely a generator and a discriminator. The role of a discriminator can be likened to that of a deep learning classifier. In this context, the discriminator is predicting whether a given sample is from the real training set or it is generated by the generator. The performance of the discriminator is further used by the generator as a feedback as it tries to improve its learning parameters to fool the discriminator. Once the generator successfully learns the distribution of the dataset, it can produce images that are virtually indistinguishable from other samples in the real dataset. The ability of big data analytics to provide flexible management of information assets (Grover & Kar, 2017) can be further improved with GANs as they can effectively perform cross-domain transformation. Moreover, in recent times text-based information retrieval and multimedia-based information retrieval has become an integral research topic due to the growing amount of these data types (Kushwaha et al., 2021). Therefore, frameworks that can effectively perform cross-domain transfer offer great benefit in terms of information storage and retrieval as well as the extraction of novel insights and knowledge. The use of text-to-text transformation has resulted in unique application such as paraphrase generation (Palivela, 2021). Consequently, a more challenging aspect of transforming the text input to a different domain, i.e., audio is undertaken in this work using GANs.

2.3. GAN architectures and loss functions

WaveGAN and SpecGAN are two types of architectures that are suitable for the proposed application. Most of the other used GAN architectures are built for TTS generation and are not tested on music or melody generation. Meanwhile, the two considered models are tested for music generation according to the existing works. The output music samples are available online, and they sound clear and realistic. Furthermore, the realistic outputs of the WaveGAN and SpecGAN architectures found in Donahue et al. (2019), was due to the selected neural network model (i.e., DCGAN) and the loss function WGAN-GP. The DCGAN, proposed in Radford et al. (2016), combines the CNN and GAN architectures for unsupervised learning. CNN is a well-known model that acts as a feature extractor and a classifier. The convolutional layers extract basic and complex features, which helps to identify the input class. The generator and discriminator in Radford et al. (2016) both consist of four convolutional layers with no pooling layers. However, the WaveGAN and SpecGAN models have an extra layer. The original output of the DCGAN is 64 × 64 pixels which translate to 4096 samples. Therefore, Donahue et al. (2019) decided to add an extra layer for the generator and discriminator to increase samples to 16384.

In terms of the loss functions, the Wasserstein distance is different from the Kullback-Leibler divergence (KL), and Jenson Shannon (JS) distances since it measures the similarity between two probability distributions by comparing the distance between them (i.e., comparing the horizontal difference and not the vertical). The Wasserstein distance is helpful when comparing the probability distributions that are not overlapping. KL divergence is known to have an infinity value when the two probability distributions are not overlapping (i.e., the generated probability distribution is 0, and the real distribution is greater than 0) (Arjovsky et al., 2017). Meanwhile, JS divergence stabilizes the output to log 2 when the distributions are not meeting (Arjovsky et al., 2017). Unlike the previous methods, the Wasserstein distance provides a stable distance metric since the output value does not increase exponentially (Arjovsky et al., 2017).

The following conditions summarize the outputs of the distances in case the distributions are not overlapping (i.e., $\theta \neq 0$) and when they are entirely overlapping (i.e., $\theta = 0$):

$$W\left(\mathbb{P}_0, \mathbb{P}_\theta\right) = |\theta| \tag{1}$$

$$JS\left(\mathbb{P}_0, \mathbb{P}_\theta\right) = \{\begin{array}{ll} \log 2 & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0, \end{array} \tag{2}$$

$$KL\left(\mathbb{P}_\theta \parallel \mathbb{P}_0\right) = KL\left(\mathbb{P}_0 \parallel \mathbb{P}_\theta\right) = \begin{cases} +\infty & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0, \end{cases} \tag{3}$$

We can see that the Wasserstein distance is equal to the θ variable corresponding to the distance between the distributions. (Arjovsky et al., 2017) used the Wasserstein distance to calculate the loss of the GAN Eq. (4.) below presents WGAN:

$$V_{\text{WGAN}}\left(D_w, G\right) = \mathbb{E}_{\boldsymbol{x}\sim P_X}\left[D_w(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{z}\sim P_Z}\left[D_w(G(\boldsymbol{z}))\right] \tag{4}$$

The discriminator D_w no longer detects the real or fake samples, and instead, it calculates the Wasserstein distance. Hence, the discriminator is called a “critic” since it tells how close the two distributions are. The first function, $\mathbb{E}_{\boldsymbol{x}\sim P_X}[D_w(\boldsymbol{x})]$, calculates the Wasserstein distance with respect to the data sample x taken from the real data and the second equation with respect to the generated sample z. The network parameter weights for the critic and generator network are updated to minimize Wasserstein distance between the distributions.

The clipping method is used in the WGAN to satisfy the Lipschitz constraint by limiting the weight of the critic. However, it has some drawbacks, including taking a long time to reach the optimal point for the critic and having vanishing gradients if the number of layers is large or batch normalization was not used (Arjovsky et al., 2017). Hence, gradient penalty (GP) was introduced in Gulrajani et al. (2017) to better address the Lipschitz constraint by enforcing a softer limit on the critic's weight. The GP equation is as follows using the added equation:

$$L = \underbrace{\mathbb{E}_{\boldsymbol{x}\sim P_X}\left[D_w(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{z}\sim P_Z}\left[D_w(G(\boldsymbol{z}))\right]}_{\text{Original critic loss}} + \underbrace{\lambda \underset{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}{\mathbb{E}}\left[\left(\left\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\right\|_2 - 1\right)^2\right]}_{\text{Gradient penalty}} \tag{5}$$

where $\hat{x}$ is an interpolated value between the real and generated samples. The GP takes λ as a regularization parameter and the gradient norm of $\hat{x}$ is subtracted by 1 and squared. The result is the value of the norm of the gradient close to 1, which will satisfy the Lipschitz constraint.

3. Methodology

This section describes the overall methodology for the proposed application including GAN architectures used, data collection, and experimental setup.

3.1. GAN architecture selection

In this work, the audio dataset comprised of melodies in raw audio format instead of MIDI representations. WaveGAN and SpecGAN are two types of GAN architectures used for music generation that is compatible with raw audio file formats. We selected WaveGAN for the melody generation since it demonstrated better results on the speech commands zero through nine (SC09) dataset used in (Donahue et al., 2019). The SpecGAN has a higher chance of overfitting and the mean opinion scores (MOS) for the human quality, ease, and diversity evaluations are lower than WaveGAN (Donahue et al., 2019). WaveGAN is specifically suitable because of the melody data being in raw audio format. Other popular approaches related to melody and music generation utilized MIDI or symbolic datasets in which case the GAN architectures such as the one in Lim et al. (2017) would be suitable. Similarly, SpecGAN makes use of STFT from raw audio and uses an iterative Griffin-Lim algorithm for inversion to audio. So, in conclusion, the key difference between SpecGAN to WaveGAN is that the raw audio is converted to spectrograms first before training and performs inverse transformation to audio before saving the generations. The WaveGAN implementation is publicly available on GitHub (Donahue, 2021). For our experiments, we modified the following GitHub repository (Adrián, 2021) that is suited to conditional WaveGAN. For SpecGAN implementation, we modified the implementation found in (GitHub - naotokui/SpecGAN: SpecGAN- generate audio with adversarial training 2021).

3.2. Data collection

The application of Arabic poetry to melody is a novel one and consequently there do not exist any Arabic poem-emotion and melody-emotion datasets. Therefore, different sources were utilized to collect data to build a novel Arabic poem and melody datasets. To express the appropriate mood of the poetry and link it to its corresponding melody, poetry was collected based on emotions. In the future, this work can be extended to experiment with other attributes such as poem era and poem geographical location. To represent the three emotion classes of poems, the Arabic musical mode or *maqām* was used. These are patterns of melody and the *maqām* technique denotes the pitches, patterns, and construction of a piece of music, unique to Arabic art music (Touma & Touma, 2003). The use of *maqām* for selecting the melodies

Table 1
Dataset characteristics.

Dataset	Category	Total Samples	Attributes
Poetry	Sad	4064	Contains 9452 poems of varying length,
	Love	3443	with each row containing poem text and
	Joy	1945	corresponding emotion label.
Melody	Saba	8319	Contains 16,322 audio files of 4 seconds.
	Seka	4611	Each file is labeled with its corresponding
	Ajam	3392	maqam name.

is suitable due to the ability of various maqams to exclusively convey certain emotions (Shahriar & Tariq, 2021). According to Farraj and Shumays (2019) and Shokouhi and Yusof (2013), melodies belonging to *maqām Ajam* represents the emotion of joy, *Seka* represents love, and *Saba* represents melancholy and sadness. Therefore, the *maqām* denoting a particular emotion will be paired with its corresponding poem denoting the same emotion.

For the poem dataset, a web scraper was developed to collect poems with their corresponding emotions from *Al Diwan* (Al Diwan, 2021) website. This website was selected because it was the only accessible source where poems were categorized under several emotion categories. Other sources such as *Kaggle* contained Arabic poetry datasets classified by the era and locations of the poems. For simplicity, three emotion categories including sad, joy, and love were selected. However, in the future, the proposed framework is flexible and can easily be extended to other emotions. A total of 9452 poems were collected: 4064 sad, 3443 love, and 1945 joy. For the melody dataset, YouTube musical plays were used to collect three types of melodies from each of maqams *Saba, Seka*, and *Ajam* which correspond to sad, love, and joy, respectively. The melodies were split into 4-second segments, resulting in 16,322 audio files. This left us with a total of 8319 recordings for maqam Saba, 4611 recordings for Seka, and 3392 recordings for Ajam. Since there were more melodies available compared to poems, a poetry from each class was paired with a melody from its corresponding maqam class randomly. This meant that a big portion of melodies from the dataset were not used for training as they were left unpaired. In future work, more poems from the emotion classes can be collected and paired with the unpaired melodies to expand the training set size. The dataset is publicly available on GitHub (https://github.com/SakibShahriar95/Arabic-Poetry-Melody). The dataset attributes are summarized in Table 1.

3.3. Experimental setup

To implement the proposed application of Arabic poetry to melody generations, two different approaches were utilized. Both approaches are GAN-based and are used to generate melodies appropriate for a given input text. The key difference is that the first approach utilizes the text embedding from the poems to directly train the GAN whereas the second relies on a separate classifier to first determine the class that the poem belongs to before generating melody for that class. However, the result from both approaches will offer the proposed poetry to melody generation application.

3.3.1. End-to-end (E2E) poetry to melody generation using WaveGAN

In the first approach, no intermediary steps are required. This is because the input poem text in the form of word embeddings are being directly used to train the GAN architecture. Fig. 2. illustrates the experimental setup using the first approach.

During the training phase, the poems in the dataset is converted to corresponding word embeddings using the AraBERT (Antoun et al., 2020) model. To reduce the computational cost two steps were performed:

- The first 128 words of each poem were considered to generate the embedding. This was done to reduce computational cost and memory exhaustion.
- T-distributed stochastic neighbor embedding (TSNE) (van der Maaten & Hinton, 2008) is applied to the output embedding of AraBERT because the default embedding size was (768 X number of words). The number of output components for TSNE was set to 100 to balance it for concatenation with the latent vector size of 100.

Therefore, the input to the generator is a noise vector that is concatenated with the word embeddings (after applying TSNE). The discriminator receives the real melody data to train itself to distinguish between the real ones and the synthesized ones. After training, the architecture can generate an appropriate melody for a given text embedding. WaveGAN is built on top of DCGAN architecture. For up sampling, the generator uses transposed convolution (since audio signals are 1-D, a larger kernel of size 25 was used). The rest of the architecture is kept the same as standard DCGAN. For the generator, ReLU activation was used for all the first four layers whereas tanh activation was used for the final layer. Meanwhile, in the discriminator, no activation was used for the first four layers whereas the last layer used LeakyReLU activation. The training strategy used was WGAN-GP.

3.3.2. Poetry to melody generation using conditional GANs

In this approach, a conditional GAN-based method was used. There is an intermediary text classifier that is first responsible for generating the correct labels to be used by the conditional GAN. Fig. 3. displays the experimental setup for the second approach.

Two conditional GANs, namely conditional WaveGAN and conditional SpecGAN were experimented with. The generator for the WaveGAN is trained based on the input random noise and the labels of the poem texts which specify the emotion of the poem. The discriminator takes as input both the actual melody and the ones that were synthesized to learn to distinguish between them. The only difference for SpecGAN implementation is that the raw audio file is first converted to spectrogram representation for the discriminator to train on. The entire process is reversed during synthesis when the generated spectrograms are inverse transformed to raw audio representations. During the training phase, a text classifier is not required as the labeled data is available. However, once the conditional GANs have been trained successfully, we cannot simply input the text or its embedding to generate new melodies. Rather, the label of the poem category must be obtained first by running the text through a text classifier, AraBERT in this case. The label can then be passed on to the conditional GAN to generate a new melody. The most significant drawback to this approach is the performance of the text classifier plays a big role in the accuracy of the generated melodies. Moreover, an intermediary step is often undesirable in many applications.

Within the conditional approach, the first method utilizes the WaveGAN. The architecture of the conditional WaveGAN is the same as the one used for E2E generation. This includes the number of layers, activations, and the training strategy. The only difference is that in this approach the TSNE outputs for the word embeddings are not required. Rather, the label is concatenated to the noise vector such that the GAN architecture is trained to generate samples from the correct melody class. After the training phase, the generator can be called by passing the label for the classes as well as a random noise vector to obtain new melodies for all the three classes.

The second conditional approach utilizes SpecGAN. As mentioned previously and illustrated in Fig. 3, this approach makes use of a spectrogram representation. Therefore, a preprocessing step is employed before the training phase. During this stage, the audio files for each category are loaded. We then convert each audio into 128*128 Mel-spectrogram representations. We then save the spectrograms, the class labels, Mel feature mean, and standard deviation into .npz (NumPy) format for storage of array data. Thereafter, this file can directly be used for the training phase without running the preprocessing steps multiple times. During the training phase, the training ratio is set to 5 as per

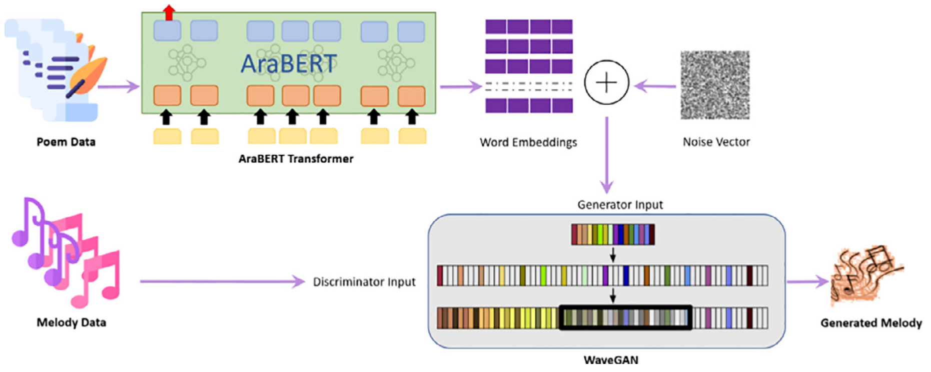

Fig. 2. End-to-end poetry to melody generation.

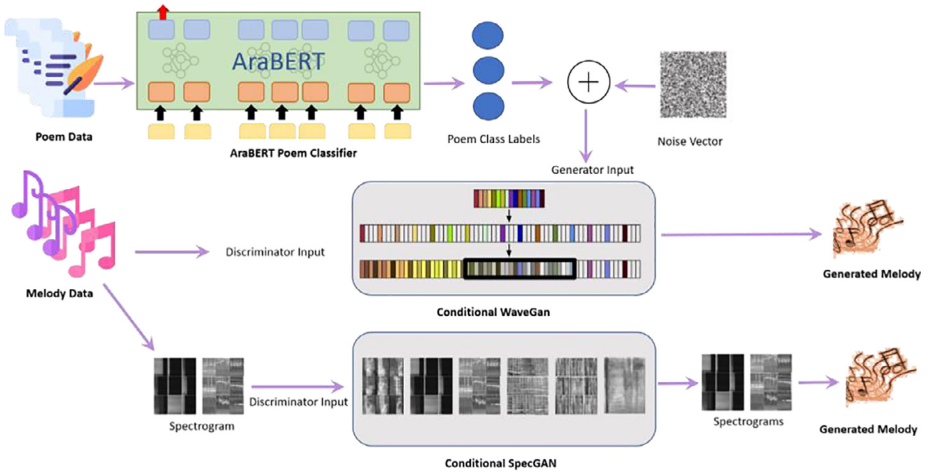

Fig. 3. Conditional poetry to melody generation.

the approach proposed in (Donahue et al., 2019). This ensures that the discriminator updates will happen 5 times per generator update. The generator consists of 11 layers as follows: A dense layer of 16384 units is followed by 5 blocks of UpSampling2D and Conv2D layers. All the activations are ReLU except for the last Conv2D layer which has tanh activation. The discriminator on the other hand consists of six layers. First, there are five Conv2D layers followed by LeakyReLU activations. Finally, after flattening, we have two dense layers. The first dense layer is responsible for determining whether the generated audio is real or fake, so it has a single output. However, no activation is used due to the use of the Wasserstein loss function. The other dense layer is responsible for generating the probabilities that the audio belongs to one of the categories. Therefore, it has three outputs with a softmax activation function. The generator learning rate is set to 0.0001 with Adam optimizer and the model will be optimized on Wasserstein loss. On the other hand, for the discriminator, the learning rate and optimizer are the same as that of the generator. In addition to the Wasserstein loss, the discriminator also has a gradient penalty loss and categorical cross-entropy loss for predicting the classes. First, we call the generator to predict the spectrogram by giving the input noise and the category label. Next, we call on the inverse Mel-spectrogram to obtain the raw audio files.

4. Evaluation and results

In this section, results from the different approaches will be compared using qualitative evaluation and common quantitative evaluation for GANs. The following sections present an explanation of these metrics and the corresponding results.

4.1. *Quantitative validation*

The leave-one-out (LOO) and GAN-train/test metrics will be used to validate the model performance. LOO uses all the n-1 sample points for training where n is the number of samples and the remaining sample for testing. The class (i.e., fake or real) of the nearest sample to the testing point is given to it in terms of distance. The objective is to train a k-nearest neighbor (K-NN) algorithm with a K value of 1 (1-NN) and apply the LOO approach for training K-NN. For the GAN, LOO of 0.5 is the best where the discriminator has a 50% chance of correctly labeling the point. If the LOO value was less than 0.5, it will likely lead to overfitting since the generated and real points are very close. On the other hand, a LOO value greater than 0.5 means that the discriminator can easily distinguish between real and fake samples. Furthermore, to calculate the distance between the sampled point and the nearest neighbor, the Euclidean distance will be used. Given two points, x and y we can get the Euclidean distance between them using Eq. (6):

$$Euclidian\,(x, y) = \sqrt{\sum\nolimits_{i=1}^{n} \left(y_i - x_i\right)^2} \tag{6}$$

Moreover, the dynamic time warping (DTW) (Salvador and Chan, 2007) algorithm will be used to calculate the distance between the distributions of the real examples and the generated ones. The LOO method using the DTW distance will also be utilized in the K-NN algorithm.

GAN-test is when the classifier is trained on the actual samples and tested on the generated sample, and GAN-train is the opposite (Konstantin_Shmelkov_How_good_is_ECCV_2018_paper.pdf, 2021). From the GAN-test, it can be determined whether the model is overfitting, or does not capture the actual distribution, or generating

Table 2
Architecture parameters and conditions for each experiment.

Approach	Batch Size	Input Layers (G, D)	Target Length (Sec.)	Sample Rate (Hz)
E2E-WaveGAN	64	(6,8)	4	8192
Conditional-WaveGAN	32	(6,8)	4	8192
Conditional-SpecGAN	64	(11,6)	1	16000
Approach	Noise Vector	Cond. Input Format	Batch Norm.	Epochs
E2E-WaveGAN	(100, 1)	Arabert+tSNE (100,1)	True	9200
Conditional-WaveGAN	(64, 1)	Label joined with Embedding (16,1)	True	15000
Conditional-SpecGAN	(97, 1)	One-hot Encoded (3,1)	False	3000

realistic samples (Konstantin_Shmelkov_How_good_is_ECCV_2018_paper.pdf, 2021). If the testing accuracy is greater than the validation, the generated samples capture the fine details of the training set and hence produce similar samples. Meanwhile, the diversity of the generated samples can be extracted using GAN-train accuracy. The closer the testing accuracy to the validation accuracy, the similar the generated results are to the real ones, and the more diverse they are. For building both networks, ResNet34 (ResNet-34, 2021) architecture was used. The same architecture was used because the goal in this evaluation was to estimate the similarity between the generated and real samples by assessing the GAN-train and GAN-test accuracies. Optimizing the architectures further would potentially lead to higher scores for both. However, in this case, the objective was to obtain a general idea of the accuracies and not necessarily optimize them.

4.2. Qualitative validation

To validate the quality of the generated melodies, a qualitative case study was conducted. We invited twelve volunteers who were university students to evaluate and score both the generated as well as the original melodies. The participants were presented with a total of four questions after being presented with ten audio samples from which five of them belonged to the real melodies and five of them were GAN synthesized. The generated samples were presented from the SpecGAN as the other models produced poor-quality melodies. For each audio sample, the evaluators were asked the following questions and asked to rate between 1 to 5 for each question, indicating very poor (1) to very good (5).

- The overall quality of the melody played (1 poor quality, 5 great quality).
- The melody played is enjoyable (1 not very enjoyable, 5 very enjoyable).
- The melody played is interesting (1 not very interesting, 5 very interesting).
- The melody played sounds artificial (1 sounds very artificial, 5 sounds very realistic).

Following the survey, the scores of each question for both the real and the generated melodies will be aggregated, and comparison will be made. Although due to the small sample size which limits the possibility of the results being conclusive, we can still obtain a general perception in terms of the quality of the generated melodies. We are particularly interested to note if the generated melodies were of good quality and if they did not sound artificial. Moreover, if the evaluators found them to be interesting and enjoyable, it would further add to the success of the generated melodies.

4.3. Results

Three different experiments for the purpose of poetry to melody generation application was conducted. In Table 2, we the parameters and conditions that each of the models was experimented on are outlined. The architecture in greater details was already presented in the previous section.

4.3.1. Quantitative results

The first experiment was end-to-end using conditional WaveGAN, and the model was trained for 9200 epochs in batch sizes of 64. With a sampling rate of 8192, output of 4 seconds was obtained which was the size of the files in the audio dataset. However, the generated melodies from the embeddings in this approach were of poor quality as they sounded like random noisy instruments. We believe this is due to the transformation of the word embedding vector, which for computational reasons could not be directly used as an input and was required to be down sampled. Using AraBERT, we obtained embeddings of 254 by 768 for each input text. In contrast, CIFAR-10 images are of 28 by 28 size. The extremely large size of the AraBERT model meant that sufficient memory could not be allocated for training. As a result, the input size was significantly reduced. Moreover, to match the dimensions a TSNE transformation was performed which would mean that further information was lost. Therefore, we believe in the future smaller text embedding models and a more sophisticated downscaling approach are required. The LOO results using Euclidean indicates that the model is leaning more towards the optimal result with a value of 0.58. However, the discriminator has a higher chance of distinguishing between the real and fake samples than the generator, which tries to trick the classifier. Meanwhile, the value of the LOO using the DTW distance is 0.91, which shows an entirely different outcome where the model is highly capable of distinguishing between the samples. Finally, the distance between the actual and generated distributions is 361382, which is high. Consequently, this means that the generator is not creating realistic samples that would fool the discriminator.

In the second experiment, the same WaveGAN architecture was utilized but with a different input representation. In this case, the class labels were used instead of the word embeddings. The model was trained for 15000 epochs with a batch size of 32. The other configurations are the same as the first experiment. The results obtained using this approach sounded more coherent, although a post-processing noise filter was required to denoise the generated melody. It is possible to further improve the performance of this architecture by using additional data and hyperparameter tuning. Due to only being able to run about 10 epochs every minute, the amount of tuning that could be done was restricted. The LOO with the Euclidean distance shows an optimal value of 0.51 which means that the discriminator has a 0.5 probability of correctly classifying the sample. The LOO with the DWT has similar behavior to the previous architecture where the value is higher (i.e., worst) than the Euclidean distance, and in this model, the value is 0.82. The DWT distance is 150063, which is lower than the E2E-WaveGAN. Hence the LOO-DWT value is slightly better. The GAN-test accuracy of 0.36 is higher than the GAN-train accuracy of 0.31. Subsequently, this reveals that the generated samples are less diverse. Overall, the GAN-test and GAN-train accuracies are very low, which indicates that the produced samples have poor quality and diversity.

In the final experiment, a conditional SpecGAN architecture was used. Unlike the previous two experiments, a spectrogram representation is utilized here. Because the original implementation of SpecGAN was fixed to one-second representation (Donahue et al., 2019), audio synthesis of one-second length was produced using this approach. As per the recommendation in (Donahue et al., 2019), batch normalization

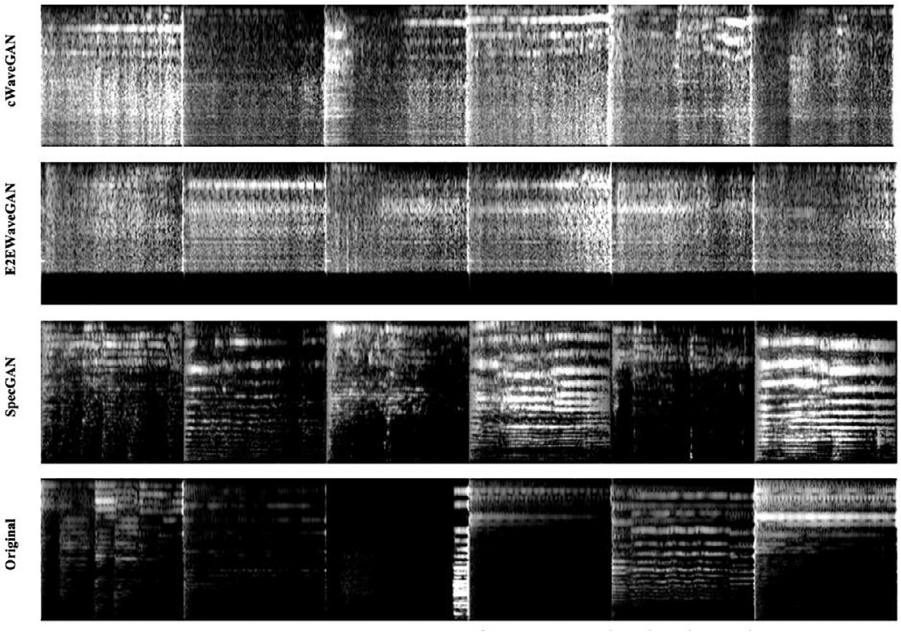

Fig. 4. Spectrogram comparison for six generated and real samples.

was not implemented for this architecture. The model was trained for a total of 3000 epochs in batch sizes of 64. The results obtained using this architecture sounded aesthetically more pleasant than the others. The only downside is that the length of the audio was one second as opposed to the input audio dataset files being four seconds long. A LOO score of 0.49 was obtained with the Euclidean distance, which is close to perfect, indicating that the discriminator has 0.5 probability of correctly classifying the sample. However, using the DTW distance with the LOO, the score was 0.78. This indicates that the generated samples are likely to be distinguished. Overall, the score is better compared to the previous two experiments. The GAN-train accuracy of 0.4 is higher than the GAN-test accuracy of 0.31. This indicates that the target distribution was not captured well and there is a lack of audio quality. Finally, the DTW distance of 92327 indicates that the generated distribution is closer to the actual one when we compare with the first two experiments. As a result, the best results were obtained using the conditional SpecGAN.

For further evaluation, a total of six spectrograms were selected that were generated by the three approaches as well as the spectrogram for the real samples. Fig. 4. displays six randomly selected Mel-spectrograms of original music and the music generated from conditional WaveGAN, conditional SpecGAN, and E2E WaveGAN. The original files contain a wide range of signals spanning over a large range of frequencies. This contrasts with the signals generated through E2E WaveGAN and to some extent conditional WaveGAN. The former has generated more monotonic signals with white noise while the latter shows the signals are modulated possibly through the provided label or the latent space variation. The closest waveforms are generated by SpecGAN which showed that the generated signals are more diverse and closer to the original one. This argument is also supported by the scores obtained through the qualitative assessment.

The quantitative results obtained from the three experiments using the aforementioned metrics is summarized in Table 3. Fig. 5. also illustrates the comparison of the DTW distance between the real and generated data distributions for each algorithm. A lower DTW distance indicates that the generated and real data points are closer and therefore lower value, in this case, indicates better performance. As can be seen from the table, in terms of LOO evaluation, both conditional WaveGAN and SpecGAN outperformed E2E WaveGAN. However, both these models obtained lower scores in terms of GAN-train and GAN-test accuracies. Since the E2E-WaveGAN generated data points based on text embedding, there was no label to perform the Gan-train and Gan-test evaluation on it. In terms of DTW distance, as displayed in Fig. 5, it can be observed that SpecGAN outperformed the other models. This was clear after assessing the samples generated from all three experiments that the conditional SpecGAN ones were the closest to real samples. In contrast, conditional WaveGAN produced noisy melodies and the E2E WaveGAN produced melodies that were not meaningful. Consequently, only the audios generated by conditional SpecGAN were selected for the qualitative case study.

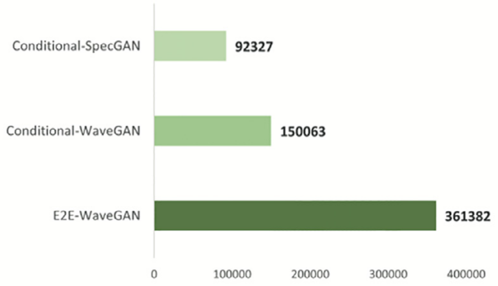

Fig. 5. DTW distance between real and generated distributions.

4.3.2. Qualitative results

To assess the quality of the melodies generated by the conditional SpecGAN, twelve volunteers were asked to answer four quality-related questions. The evaluation criteria were based on enjoyable, interesting, realistic, and the overall quality of the melodies to be scored between 1 and 5. The average score along with their standard deviation is summarized in Table 4. Sample melodies generated using conditional SpecGAN are available online (https://github.com/SakibShahriar95/Arabic-Poetry-Melody).

The table indicates that real melodies are better under all four criteria compared to the generated ones. However, for all the measures the generated melodies scored points that were above average (2.5). More interestingly, the highest of the four measures for the generated melodies was realistic with the participants on average rating it 2.75 out of 5 for being realistic. This means the participants think that the

Table 3
Quantitative results comparison.

Approach	LOO (Euclidean)	LOO (DTW)	GAN-Train	GAN-Test	DTW Distance
E2E-WaveGAN	0.58 ± 0.49	0.91 ± 0.29	N/A	N/A	361382
Conditional-WaveGAN	0.51 ± 0.50	0.82 ± 0.39	0.31	0.36	150063
Conditional-SpecGAN	0.49 ± 0.50	0.78 ± 0.41	0.40	0.31	92327

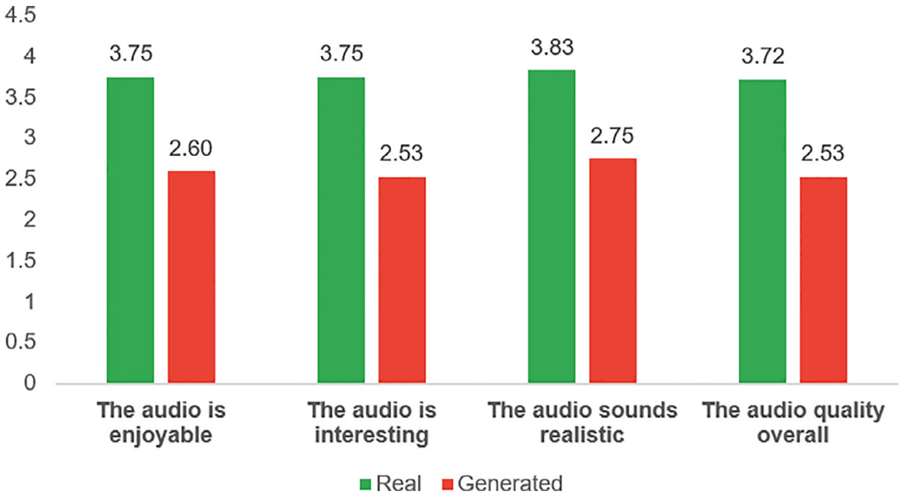

Fig. 6. Survey result comparison between real and generated melodies.

Table 4
Qualitative results for conditional SpecGAN.

Evaluation	Real Melodies	SpecGAN Generated
Enjoyable	3.73 ± 0.61	2.60 ± 0.28
Interesting	3.75 ± 0.45	2.53 ± 0.31
Realistic	3.83 ± 0.29	2.75 ± 0.38
Overall Quality	3.72 ± 0.32	2.53 ± 0.32

SpecGAN generated audios are more likely to be realistic than fake. This indicates to us that the overall results for the conditional SpecGAN are quite promising and can be improved upon in the future. Fig. 6. illustrates the comparison of the qualitative case study.

5. Discussion

Despite the comparatively limited research in GAN-based cross-domain audio generation, the results obtained for the novel poetry to melody generation were promising. In this section, some of the challenges, implications, and future work of this application are discussed.

5.1. Challenges

There were several notable challenges that were overcome to implement the proposed GAN-based poetry to melody generation framework. Firstly, there are no existing datasets for poetry and melodies paired based on suitable mood and emotion. As such, a significant contribution included the introduction of an Arabic poetry and melody dataset. Related works focused on generating melodies from given lyrics by aligning the lyrics with the musical notes, synthesizing the lyrics, and combining them with the generated melody to create suitable songs. Meanwhile, in this work, the objective is to generate a novel melody for each poem representing the emotion conveyed by the poem. The lack of cross-domain audio generation research in the literature made it further challenging.

Moreover, the representation of the audio signals is quite complex and spatially costly to represent in digital form. The music signals could form the melody at various timescales which cannot be captured algorithmically. The spatial complexity, i.e., the high-fidelity rate of audio signals, is a problem because it requires large computational power. These two problems of varying timescale consistency and high-fidelity make it very difficult to model the melodies. Various algorithms prioritize the global structure at the cost of high computational power whereas others focus on using proxy representations like spectrograms to conserve computational power. Therefore, the solutions to this tradeoff cannot be easily generalized and require significant domain knowledge and experimentation.

5.2. Contributions to literature

As discussed in the previous sections, this work presented the first poetry to melody generation using GANs. Consequently, a novel dataset was introduced containing paired poetry and melody based on emotions. The dataset can be used by researchers to further improve the melody generation. They can also experiment with new tasks such as melody to poetry generation as well as text and audio classification in the Arabic language. Moreover, this work presented a reasonable performance for melody generation using SpecGAN architecture and the implementation can be utilized for related applications. However, the results obtained in this work also highlight the immense challenge in dealing with raw audio especially with the text to audio transformation in an end-to-end manner. Therefore, necessary computation power and training time is required to further enhance the generation quality.

5.3. Implications for practice

The practical implications of this research are manifold. Firstly, the user experience for reading poetry will be greatly enhanced with the addition of melody. This can be achieved by integrating the proposed poetry to melody generation framework into a mobile or web application. Based on the poetry being read by the user, a melody matching the mood of the poem will be played in real-time to enhance the reading experience. Moreover, the proposed framework can be of great benefit to the media production industry. For instance, a video scene where a poem is being recited may have a background melody to accompany it. Instead of relying on generic music or investing valuable time and resources in finding a suitable melody for a specific poem, the proposed framework can be utilized to automatically generate a novel piece of melody that corresponds to the mood of the poem being recited. Similarly, for other related multimedia production applications, the GAN architectures can be retrained and adjusted. Furthermore, the proposed framework can be expanded to other similar applications such as melody generation for stories and other text forms. Given the fact that this research project deals with text to audio transformation, it further contributes to the TTS research community that provides numerous benefits in terms of accessibility. It also demonstrates the feasibility of cross-domain application in the Arabic language, which is not prominent in the literature. Finally, we hope that the novel melodies generated by the proposed framework can be a source of inspiration for poets and artists.

5.4. Future work

The SpecGAN, which obtained the best results, has less computational complexity since it receives spectrograms instead of raw audio files. However, further improvements could be made to the SpecGAN model. One-second melody file does not provide adequate information for analysis. Hence the model should be updated to generate longer melodies. Also, different hyperparameter values should be explored which could potentially lead to higher quality results. A longer training time in terms of the number of iterations is needed to train the model better. It was quite evident that the E2E model using AraBERT embedding performed the worst. However, in future, more experiment with the text embedding is necessary to provide direct generation of melodies from poems. It is also worth experimenting with other GAN architectures using the embedding from the text directly. Given the fact that conditional WaveGAN produced noisy melodies, it could be beneficial to train an autoencoder for post-processing the melodies generated to make them sound more realistic.

For related future applications, experiments on poems and melodies from different centuries or eras could be explored. In each century or era, the style and type of the poems and music change. Also, the work could be implemented on Arabic datasets with specific dialects since the preferences change from region to region. For example, the model could be built to generate Egyptian melody from Egyptian poetry. Researchers are also encouraged to introduce datasets in other languages and consequently implement the poetry to melody generation for different languages. Moreover, only three emotions were considered, and this does not represent a complete range of human emotions. In the future, additional discrete and complex emotions such as anger, fear, optimism, and longing could be explored. Finally, a reverse approach of generating poems from melodies considering the emotions could be implemented.

6. Conclusions

Despite the overwhelming popularity of GANs in recent times, their implementations in cross-domain audio generation have been limited. There are many research challenges including the representation of audio data. This paper presented a novel GAN-based melody generation framework from a given poetry. A dataset for Arabic poem to melody generation application was introduced. Three different GAN architectures were explored, namely E2E WaveGAN, conditional WaveGAN, and conditional SpecGAN. The results indicate that the samples produced by the SpecGAN were more diverse and realistic compared to the other two architectures. Moreover, a qualitative case study proved that human evaluators deem the generated conditional SpecGAN audios to be realistic.

Declarations of Competing Interest

The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript, and there is no financial interest to report. We certify that the submission is original work and is not under review at any other publication.

References

Ackerman, M., & Loker, D. (2017). Algorithmic songwriting with alysia. In *Proceedings of the international conference on evolutionary and biologically inspired music and art* (pp. 1–16).

Adrián, *Adrianbarahona/conditional_wavegan_knocking_sounds*. 2021. Accessed: Sep. 08, 2021). [Online]. Available: https://github.com/adrianbarahona/conditional_wavegan_knocking_sounds

Aggarwal, A., Mittal, M., & Battineni, G. (2021). Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights, 1*(1), Article 100004 Apr. doi:10.1016/j.jjimei.2020.100004.

Al Diwan, 2021, https://www.aldiwan.net (accessed May 20, 2021).

Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of theLREC workshop language resources and evaluation conference* (p. 9). 11–16 May.

Arjovsky, M., Chintala, S., & Bottou, L., "Wasserstein GAN," *ArXiv170107875 Cs Stat*, Dec. 2017, Accessed: May 27, 2021. [Online]. Available: http://arxiv.org/abs/1701.07875

Bao, H., et al. (2019). Neural Melody Composition from Lyrics. In *Proceedings of the natural language processing and Chinese computing* (pp. 499–511). doi:10.1007/978-3-030-32233-5_39.

Boulanger-Lewandowski, N., Bengio, Y., & Vincent, P. "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," *ArXiv12066392 Cs Stat*, Jun. 2012, Accessed: Feb. 28, 2021. [Online]. Available: http://arxiv.org/abs/1206.6392

Bouwmans, T., Javed, S., Sultana, M., & Jung, S. K. (2019). Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. *Neural Network, 117*, 8–66 Sep. doi:10.1016/j.neunet.2019.04.024.

Briot, J.P., Hadjeres, G., & Pachet, F.D., *Deep learning techniques for music generation*. 2020.

Chakraborty, A., & Kar, A. K. (2017). Swarm intelligence: A review of algorithms. In S. Patnaik, X. S. Yang, & K. Nakamatsu (Eds.), *Nature-inspired computing and optimization: Theory and applications* (pp. 475–494). Cham: Springer International Publishing. doi:10.1007/978-3-319-50920-4_19.

Chamberlain, R., Mullin, C., Scheerlinck, B., & Wagemans, J. (2018). Putting the art in artificial: Aesthetic responses to computer-generated art. *Psychology of Aesthetics, Creativity, and the Arts, 12*(2), 177.

"[1908.05551] Conditional LSTM-GAN for Melody Generation from Lyrics." https://arxiv.org/abs/1908.05551 (accessed Feb. 28, 2021).

Davis, H., & Mohammad, S. M., "Generating music from literature," *ArXiv14032124 Cs*, Mar. 2014, Accessed: Feb. 28, 2021. [Online]. Available: http://arxiv.org/abs/1403.2124

"Deep Cross-Modal Correlation Learning for Audio and Lyrics in Music Retrieval | ACM Transactions on Multimedia Computing, Communications, and Applications." https://dl.acm.org/doi/abs/10.1145/3281746 (accessed Feb. 28, 2021).

Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: A review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence, 9*(2), 85–112 Jun. doi:10.1007/s13748-019-00203-0.

Dias, D. S., & Fernando, T. G. I. (2019). Komposer – Automated Musical Note Generation based on Lyrics with Recurrent Neural Networks. In *Proceedings of the 1st international conference on artificial intelligence and data sciences (AiDAS)* (pp. 76–82). Sep. doi:10.1109/AiDAS47888.2019.8970710.

Dieleman, S., van den Oord, A., & Simonyan, K. (2018). The challenge of realistic music generation: modelling raw audio at scale. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 8000–8010). Dec.

Donahue, C., McAuley, J., & Puckette, M., "Adversarial audio synthesis," *ArXiv180204208 Cs*, Feb. 2019, Accessed: May 20, 2021. [Online]. Available: http://arxiv.org/abs/1802.04208

Donahue, C., *Chrisdonahue/WaveGan*. 2021. Accessed: May 03, 2021. [Online]. Available: https://github.com/chrisdonahue/wavegan

Farraj, J., & Shumays, S. A. (2019). *Inside Arabic music: Arabic Maqam performance and theory in the 20th century*. Oxford University Press.

Fukayama, S., Nakatsuma, K., Sako, S., Nishimoto, T., & Sagayama, S. (2010). Automatic song composition from the lyrics exploiting prosody of Japanese language. In Proceedings of the *7th sound and music computing conference (SMC2010)* Jul. 21. doi:10.5281/zenodo.849727.

"GitHub - naotokui/SpecGAN: SpecGAN- generate audio with adversarial training." https://github.com/naotokui/SpecGAN (accessed Sep. 08, 2021).

Goold, P. (2021). *The curious case of computer-generated works under the copyright, designs and patents act 1988*. London, UK: The City Law School Working Paper City Law School Research Paper 2021/03[Online]. Available:https://openaccess.city.ac.uk/id/eprint/26210/.

Grover, P., & Kar, A. K. (2017). Big data analytics: A review on theoretical contributions and tools used in literature. *Global Journal of Flexible Systems Management, 18*(3), 203–229 Sep. doi:10.1007/s40171-017-0159-3.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A., "Improved training of Wasserstein GANs," *ArXiv170400028 Cs Stat*, Dec. 2017, Accessed: May 27, 2021. [Online]. Available: http://arxiv.org/abs/1704.00028

Kar, A. K. (2016). Bio inspired computing – a review of algorithms and scope of applications. *Expert Systems With Applications, 59*, 20–32 Oct. doi:10.1016/j.eswa.2016.04.018.

"Konstantin_Shmelkov_How_good_is_ECCV_2018_paper.pdf." Accessed: May 23, 2021. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/papers/Konstantin_Shmelkov_How_good_is_ECCV_2018_paper.pdf

Kushwaha, A. K., Kar, A. K., & Dwivedi, Y. K. (2021). Applications of big data in emerging management disciplines: A literature review using text mining. *International Journal of Information Management Data Insights, 1*(2), Article 100017 Nov. doi:10.1016/j.jjimei.2021.100017.

Li, N., Liu, S., Liu, Y., Zhao, S., & Liu, M. (2019). Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence: 33* (pp. 6706–6713). Jul. doi:10.1609/aaai.v33i01.33016706.

Lim, H., Rhyu, S., & Lee, K., "Chord generation from symbolic melody using BLSTM networks," *ArXiv171201011 Cs Eess*, Dec. 2017, Accessed: Sep. 08, 2021. [Online]. Available: http://arxiv.org/abs/1712.01011

Mazzone, M., & Elgammal, A. (2019). Art, creativity, and the potential of artificial intelligence. *Arts, 8*(1), 26.

Milon-Flores, D. F., Ochoa-Luna, J., & Gomez-Nieto, E. (2019). Generating audiovisual summaries from literary works using emotion analysis. In *Proceedings of the 32nd SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)* (pp. 31–38). 10.1109/SIBGRAPI.2019.00013.

Mishra, A., Tripathi, K., Gupta, L., & Singh, K. P. (2019). Long short-term memory recurrent neural network architectures for melody generation. In *Soft computing for problem solving* (pp. 41–55). Springer.

Monteith, K., Martinez, T., & Ventura, D. (2012). Automatic generation of melodic accompaniments for lyrics. In *Proceedings of the international conference on computational creativity* (pp. 87–94).

Noll, A. M. (1967). The digital computer as a creative medium. *IEEE Spectrum, 4*(10), 89–95.

Okamoto, T., Toda, T., Shiga, Y., & Kawai, H. (2019). Real-time neural text-to-speech with sequence-to-sequence acoustic model and WaveGlow or single Gaussian WaveRNN vocoders. *INTERSPEECH*, 1308–1312.

Palivela, H. (2021). Optimization of paraphrase generation and identification using language models in natural language processing. *International Journal of Information Management Data Insights, 1*(2), Article 100025 Nov. doi:10.1016/j.jjimei.2021.100025.

"Performance RNN: Generating Music with Expressive Timing and Dynamics," *Magenta*. https://magenta.tensorflow.org/performance-rnn (accessed Feb. 28, 2021).

Prenger, R., Valle, R., & Catanzaro, B. (2019). Waveglow: A flow-based generative network for speech synthesis. In *Proceedings of the IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 3617–3621). doi:10.1109/ICASSP.2019.8683143.

Radford, A., Metz, L., & Chintala, S., "Unsupervised representation learning with deep convolutional generative adversarial networks," *ArXiv151106434 Cs*, Jan. 2016, Accessed: May 27, 2021. [Online]. Available: http://arxiv.org/abs/1511.06434

Raschka, S., "MusicMood: Predicting the mood of music from song lyrics using machine learning," *ArXiv Prepr. ArXiv161100138*, 2016.

Ren, Y., et al. (2019). FastSpeech: fast, robust and controllable text to speech. In *Proceedings of the 33rd international conference on neural information processing systems* (pp. 3171–3180). Curran Associates Inc.

"ResNet-34." https://kaggle.com/pytorch/resnet34 (accessed May 23, 2021).

Ruzafa, E. R. (2020). *Pix2Pitch: Generating music from paintings by using Conditionals GANs*. Polytechnic University of Madrid Master's Thesis.

Salvador, S., & Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis, 11*(5), 561–580 Oct.

Scirea, M., Barros, G. A., Shaker, N., & Togelius, J. (2015). SMUG: Scientific music generator. *ICCC*, 204–211.

Shahriar, S., & Tariq, U. (2021). Classifying Maqams of Qur'anic recitations using deep learning. *IEEE Access, 9*, 117271–117281 doi:10.1109/ACCESS.2021.3098415.

S. Shahriar, "GAN computers generate arts? A survey on visual arts, music, and literary text generation using generative adversarial network," Aug. 2021, Accessed: Nov. 08, 2021. [Online]. Available: https://arxiv.org/abs/2108.03857v2

Shokouhi, M. A., & Yusof, A. (2013). The Influence of Islamic culture and holy Quran on performing arts: Relating to sacred vocal music (lahn). In Proceedings of the *3rd annual international quranic conference 2013*. Centre of Quranic Research [Online]. Available: http://eprints.um.edu.my/id/eprint/9577.

Soderlund, J., & Blair, A. (2018). Adversarial image generation using evolution and deep learning. In *Proceedings of the IEEE congress on evolutionary computation (CEC)* (pp. 1–8). Jul. doi:10.1109/CEC.2018.8477754.

Stere, C. C., & Trăuşan-Matu, Ş. (2017). Generation of musical accompaniment for a poem, using artificial intelligence techniques. *International Journal of User-System Interaction, 10*(3), 250–270.

Touma, H. H., & Touma, H. (2003). *The music of the Arabs*. Hal Leonard Corporation.

van den Oord, A., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., & Kavukcuoglu, K., "Conditional image generation with PixelCNN decoders," *ArXiv160605328 Cs*, Jun. 2016, Accessed: Apr. 07, 2021. [Online]. Available: http://arxiv.org/abs/1606.05328

van den Oord, A., et al., "WaveNet: A generative model for raw audio," *ArXiv160903499 Cs*, Sep. 2016, Accessed: Feb. 28, 2021. [Online]. Available: http://arxiv.org/abs/1609.03499

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*(86), 2579–2605.

Welikala, M. D., & Fernando, T. (2020). Komposer V2: A hybrid approach to intelligent musical composition based on generative adversarial networks with a variational autoencoder. In *Proceedings of the future technologies conference* (pp. 413–425).

Xu, Y., Yang, X., Gan, Y., Zhou, W., Cheng, H., & He, X. (2020). A music generation model based on generative adversarial networks with Bayesian optimization. In *Proceedings of the Chinese intelligent systems conference* (pp. 155–164).

Yao, G., Lei, T., & Zhong, J. (2019). A review of Convolutional-Neural-Network-based action recognition. *Pattern Recognition Letters, 118*, 14–22 Feb. doi:10.1016/j.patrec.2018.05.018.

Yu, Y. , Harscoët, F., Canales, S., Reddy M, G., Tang, S., & Jiang, J., "Lyrics-conditioned neural melody generation," in MultiMedia Modeling, Cham, 2020, pp. 709–714. doi:10.1007/978-3-030-37734-2_58.