EXPLORING POLYPHONIC ACCOMPANIMENT GENERATION USING GENERATIVE ADVERSARIAL NETWORKS

Danae Charitou[3], **Christos Garoufis**[1,2,3], **Athanasia Zlatintsi**[1,2,3], **Petros Maragos**[2,3]
[1]Institute of Language and Speech Proc., Athena Research Center, Athens, Greece
[2]Institute of Robotics, Athena Research Center, Athens, Greece
[3]School of ECE, National Technical University of Athens, Athens, Greece
danaecharitou@gmail.com, cgaroufis@mail.ntua.gr, nancy.zlatintsi@athenarc.gr, maragos@cs.ntua.gr

ABSTRACT

Recently, various neural network architectures have shown capability of achieving compelling results in the field of automatic music generation. Motivated by this, in this work we design a generative framework that is structurally flexible and adaptable to different musical configurations and practices. At first, we examine the task of multi-track music generation without any human input, by modifying and proposing improvements to the MuseGAN architecture, an established GAN-based system, which we use as our baseline. Afterwards, we extend our developed framework to a cooperative human-AI setup for the generation of polyphonic accompaniments to user-defined tracks. We experiment with multiple structural variants of our model, and two different conditional instruments, namely piano and guitar. For both unconditional and conditional cases, we evaluate the produced samples objectively, using a set of widely used musical metrics, as well as subjectively, by conducting a listening test across 40 subjects. The experimental results, using the Lakh Pianoroll Dataset, reveal that our proposed modifications lead to improvements over the baseline from an auditory perspective in the unconditional case, and also provide useful insights about the properties of the produced music in the conditional setup, depending on the utilized configuration.

1. INTRODUCTION

Automatic music generation, i.e. the process of creating novel musical content with minimum human intervention, is undoubtedly one of the most exciting tasks within the scope of AI research. Music is generally perceived as a form of artistic expression of knowledge, experience, ideas, and emotions, but exact interpretations vary considerably around the world [1]. Without consensus over the foundation and the substance of the music itself, the act of conceiving a musical piece becomes even more challenging.

One of the major difficulties in creating realistic and aesthetically harmonic music lies behind the hierarchical arrangement of a musical composition. In general, a song consists of higher-level building blocks, which can be further subdivided into smaller building blocks [2]. Since the human brain focuses on such structural motifs, related to coherence, rhythm, tension, and emotion flow while listening to music [3, 4], a mechanism for incorporating the self-reference in multiple timescales is critical [5].

This hierarchy becomes even more complex when multiple tracks collectively unfold over time in an interdependent manner, preserving at the same time their own musical properties and dynamics. In this case, notes are typically presented in composite grouping formulations and polyphonic patterns that cannot be easily modeled by a computational machine.

In this paper, inspired by the latest advances in generative modeling, we attempt to tackle the aforementioned challenges by focusing on a hierarchical design that deals with polyphonic musical pieces of 5 distinct tracks: **Piano** (P), **Bass** (B), **Guitar** (G), **Strings** (S), and **Drums** (D).

We begin by examining the task of **Unconditional Generation**, i.e. generation that does not involve any supplementary information from the human user. For this purpose, we utilize MuseGAN [2], an existing model in the field of multi-track polyphonic music generation based on GANs, and propose some parameterized modifications to its architecture in order to become adaptable to different generative configurations and also capture more faithfully the hierarchical nature of music. In this way, we are able to experiment with multiple musical characteristics and further investigate its creativity.

Afterwards, we extend our model to the task of **Conditional Generation** by exploring practices for automatic accompaniment composition. In particular, given a human-composed track (as conditional information), the system learns how to generate the 4 remaining tracks by considering them as the rhythmic and harmonic support of the conditional one. For this setup, we develop 8 different variants of our generative framework that differ in terms of structural components, training procedure and type of conditional instrument.

For both tasks, we train our models using the Lakh Pianoroll Dataset [2] and apply both objective and subjective evaluation methods. For objective assessment, we utilize a set of 8 musical metrics that emphasize on tonal, rhythmical, texture, and harmonic attributes of the generated samples. For subjective assessment, we conduct a user study involving 40 listeners who evaluate the generated music from an auditory perspective. In all cases the produced results are promising, setting the basis for further investigation within the area of multi-instrumental music, particularly in the conditional setup.

Our main contributions can be summarized as follows:

- A parameterized MuseGAN-based architecture for unconditional and conditional generation of multi-track polyphonic music.
- Multiple variants of the above architecture differing in terms of generative configuration, training recipe, and GAN scheme.
- Objective and subjective validation of the proposed models using a set of musical metrics and a user study respectively.

The source code along with qualitative results and synthesized samples are available at: https://www.github.com/danae-charitou/MS_SMC23

2. RELATED WORK

In the last years, the particularly promising performance of Generative Adversarial Networks (GANs) on the creation of realistic pictures [6] as well as news articles [7] has inspired researchers to investigate their generation capabilities towards the domain of music.

The first comprehensive approach in the field of polyphonic music generation is the study of Mogren [8], who introduced C-RNN-GAN, a model for classical music generation. Later, Yu et al. [9] developed SeqGAN, a system that combines GANs with stochastic RL policies for the generation of monophonic music sequences. On similar ground, Dong et al. [2] introduced a GAN model with convolutional infrastructure for polyphonic music generation called MuseGAN. This model laid the foundation for other approaches [10, 11] and will be discussed in Sec. 3.1.

In the area of conditional generation, we can distinguish the work of Yang et al. [12], who proposed MidiNet, a convolutional GAN for melody generation either from scratch or by conditioning on prior information, e.g., a chord progression or previous melodic lines. Liu et al. [13] also utilized a convolutional GAN infrastructure combined with recurrent units to generate lead sheets and their multi-track arrangements. Similarly, Trieu and Keller [14] developed JazzGAN, an RNN-based GAN, which improvises monophonic jazz melodies conditioned on chord progressions.

This brief overview of existing works indicates that there aren't many GAN-based conditional models capable of capturing the hierarchy of polyphonic music without relying on simplifications of the task (e.g. generating single-track or monophonic lines). This fact provides us with a strong motive for further exploration of this approach.

In the context of non-GAN-based conditional generation, Jiang et al. [15] introduced RL-Duet, a model that employs Reinforcement Learning for interactive accompaniment generation in a human-machine duet setup. Another generation framework for pop music is PopMAG [16], a transformer-based model that aims at addressing the challenges of harmonic structure and long-term dependencies in multi-track accompaniments. More recent approaches mainly focus on exploring different types of conditional information, such as SingSong [17], a system that creates instrumental accompaniments for input vocals, and the novel lyrics-to-rhythm framework of Zhang et al. [18] that utilizes a Transformer architecture designed to capture dependencies between syllables and notes.

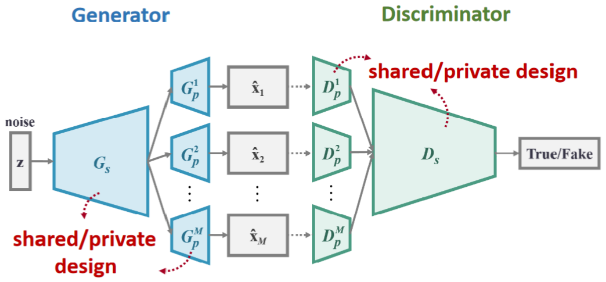


Figure 1: Architecture of our unconditional model.

3. METHOD

3.1 MuseGAN

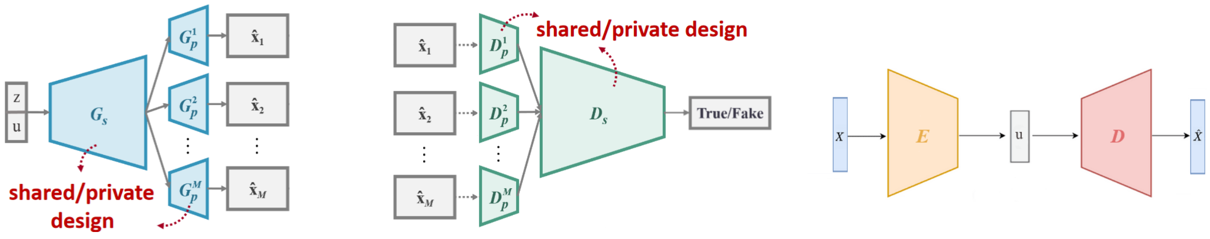
As the name suggests, MuseGAN [2] is a framework for symbolic multi-track polyphonic music generation, based on GANs. A GAN model typically consists of two modules: the Generator (G), which creates novel samples by mapping random noise to the target data space, and the Discriminator (D), which evaluates both real and generated instances in terms of authenticity by predicting the probability that the input is derived from the ground-truth distribution. These two networks are involved in an adversarial learning procedure, where the Discriminator is trained to distinguish the real samples from the fake ones, while the Generator aims at "fooling" its opponent by counterfeiting the ground-truth data as best as possible.

Dong et al. [2] exploited this mechanism in order to model the two main compositional approaches in accordance with the human experience. The first one is the jamming approach, which involves a group of musicians or instrumentalists improvising music without extensive preparation or predefined arrangements. The second one is the composer approach, where a composer arranges the various instruments according to harmonic and orchestration principles.

Based on these compositional modes, they proposed three distinct models capable of capturing the interdependency among the tracks: the jamming model (one Generator and one Discriminator for each track), the composer model (a single Generator and a single Discriminator for all tracks), and the hybrid model (one Generator per track, and a single Discriminator for all tracks). All the aforementioned models are implemented as deep CNNs and can generate musical segments with duration up to one bar. Therefore, in order to produce samples of longer duration, they combined them with a temporal unit, which produces the consecutive bars in a sequential manner.

3.2 Unconditional Modifications

In terms of the unconditional generation task, we focus on developing a GAN model that is capable of generating mu-



(a) Generator **(b)** Discriminator **(c)** Encoder and Decoder

Figure 2: Structural components of our conditional model.

sical phrases of variable length in a more compact setup. To this end, based on the hybrid multi-track module of MuseGAN and also inspired by a later work of Dong and Yang [10], we incorporate a shared-private design for both Generator and Discriminator, as demonstrated in Figure 1.

In more detail, our Generator module consists of a *shared* network G_s, followed by M *private* subnetworks G_p with each one corresponding to a different track (in this case $M = 5$). G_s can be regarded as the composer that arranges a common high-level musical idea, while the private parts represent the musicians improvising on their tracks in order to transform the abstract idea into the final musical result. Our Discriminator module mirrors the structure of the Generator, as it consists of M *private* subnetworks D_p followed by a *shared* network D_s. In this case, the private parts are responsible for extracting low-level features from each track, while the shared part aggregates their outputs in order to form a common, high-level abstraction of the final musical representation.

Within this setup, our proposed system can generate multiple bars of polyphonic multi-track music altogether and not in a sequential manner, requiring only a single noise vector as input. This is in contrast to MuseGAN, which employs 4 different kinds of inputs, each one representing distinct musical dependencies.

Moreover, MuseGAN is designed to process data of specific configuration in terms of time-related attributes, as well as tonal characteristics. In order to tackle this limitation and further investigate the capabilities of our proposed system, we implement both the Generator and the Discriminator as deep parameterized CNNs (generative modules apply transposed convolutional operations, discriminative modules apply typical convolutions) with respect to a group of musical characteristics (e.g. beat resolution, number of pitches) that define various generative configurations. In this way, our model becomes structurally adaptable to different musical properties.

We note that our parameterization exploits the hierarchical structure of music, especially regarding rhythm-related features, such as note durations: our Generator successively transforms notes with longer duration into smaller ones in accordance with the utilized resolution, by aligning the convolutions to the beats of each bar.

3.3 Conditional Modifications

In the context of the conditional generation task, we extend our unconditional framework, which we thoroughly described in the previous section, to a human-AI cooperation setup. This process requires some structural and functional modifications on the existing system components, as well as the incorporation of additional networks.

In particular, our Conditional Generator, which is graphically illustrated in Figure 2a, preserves the unconditional infrastructure but comprises only 4 *private* subnetworks instead of 5. The reason for this is that in the conditional setup the Generator is responsible for producing 4 tracks, which accompany the conditional one rhythmically and harmonically. We also modify properly its *shared* part in order to receive two distinct inputs: a random noise vector $\mathbf{z}$ sampled from a prior distribution, and an embedding $\mathbf{u}$ of the conditional track into the latent space of noise.

In order to acquire a general critic capable of measuring the fitness of the accompaniment parts for the corresponding conditional track, we incorporate the Unconditional Discriminator into our conditional model and refer to it as "Global". In this case, there are 5 *private* subnetworks, since our Global Discriminator assesses all 5 tracks collectively. We also include a second Discriminator, called "Local", which is responsible for evaluating only the accompaniment tracks as an independent musical composition. Structurally, it follows the design of the Global (Figure 2b) but comprises only 4 *private* subnetworks instead.

Apart from the typical GAN components, our conditional framework includes also an Encoder module, which produces embeddings of the conditional tracks into the latent space of the noise distribution. We also implement the corresponding Decoder, which decompresses the hidden representations of the conditional tracks into the original data space (Figure 2c). In this way, we are able to experiment with the training mode of the Encoder.

Similar to the unconditional case, all the structural components of our conditional framework are designed as configurable CNNs with respect to the same set of musical parameters.

4. EXPERIMENTAL SETUP

4.1 Data

4.1.1 Data Representation

Following [2, 10], we employ the *multi-track pianoroll* format for the representation of music samples. A multi-track pianoroll is defined as a set of binary-valued scoresheet-like matrices called pianorolls, each one corresponding to a different musical instrument. As demonstrated in Figure 3, the horizontal axis of each pianoroll indicates time in a symbolic format that discards tempo information, resulting in equally-sized timesteps, while the vertical dimension represents notes ordered from the low-pitched to the high-pitched ones. The binary values designate the presence (1) or absence (0) of notes over different timesteps.

4.1.2 Dataset

For the training of our models we follow our baseline and utilize the Lakh Pianoroll Dataset, a collection of 174,154 multi-track pianorolls derived from the Lakh MIDI Dataset [2, 19]. More specifically, we employ the *LPD-5-cleansed* version, which contains only the 5-track pianorolls with the higher matching confidence score to MSD entries [20], a “Rock” tag and 4/4 time signature. We also apply some preprocessing steps in order to segment the pianorolls into musical phrases of proper format in terms of tonal and rhythmical arrangement. This process involves temporal downsampling, removal of notes outside the desired pitch range and randomized selection of samples that contain an adequate amount of notes as specified by a fixed threshold. The resulting set of musical examples contains approximately 15,600 phrases from 7,323 songs.

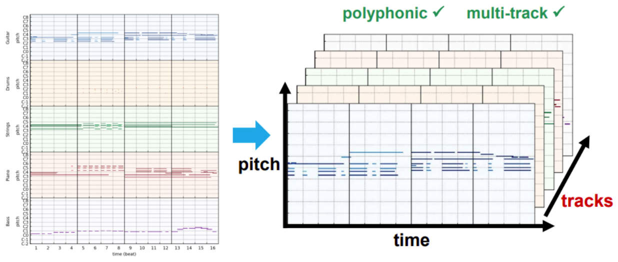


Figure 3: Multi-track pianoroll format [2].

4.2 Training Protocol

According to [21], the GAN mechanism can be modeled as a turn-based game between two opponents, the Generator (G) and the Discriminator (D). Mathematically, this adversarial setup can be described by the following minimax value function:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p_d}[\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[\log(1-D(G(\mathbf{z})))]. \tag{1}$$

Following [2], we employ a modified version of the above function that includes an additional gradient penalty term:

$$\begin{split}\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p_d}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[D(G(\mathbf{z}))] \\ + \mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{\mathbf{x}}}}[(\|\nabla_{\hat{\mathbf{x}}}D(\hat{\mathbf{x}})\|_2 - 1)^2],\end{split} \tag{2}$$

where $p_{\hat{\mathbf{x}}}$ is implicitly defined by uniform sampling along straight lines between pairs of points derived from the data distribution p_d and the generator distribution p_g. This term is found to ensure faster convergence to better optima and stabilize the overall training process [22].

In the unconditional case, our training strategy is established on consecutive interchanges between k optimization steps of the Discriminator and one update of the Generator. In this way, D is being maintained near its optimal solution, as long as G adjusts slowly enough [2, 21, 22].

In the conditional setup, we apply this practice for the GAN components, updating both Global and Local Discriminators during the same training steps and aggregating their feedback for the optimization of the Generator. Regarding the Encoder, we experiment with 2 training modes: **1-phase training**, where the Encoder is trained jointly with the GAN, following the Generator’s practice, and **2-phase training**, where the Encoder is pre-trained along with the Decoder, using a typical VAE loss: $\mathcal{L}_{\text{Enc}} = \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{KL}}$, where $\mathcal{L}_{\text{MSE}}$ represents the Mean Square Error between original and reconstructed conditional tracks and $\mathcal{L}_{\text{KL}}$ the Kullback-Liebler divergence between the embeddings and the Standard Normal Distribution $\mathcal{N}(0, \mathbb{I})$ [23]. Using a GeForce GTX 1080 Ti, the training time is nearly 8 hours for the 1-phase mode and 4 hours for the 2-phase mode.

4.3 Objective Evaluation

In the context of objective evaluation, we utilize 8 musical metrics that emphasize on tonal, rhythmical, and harmonic characteristics of the produced music samples (• for intra-track metrics, – for inter-track metrics):

• **Empty Bars (EB)**: ratio of empty bars (in %).

• **Used Pitch Classes (UPC)**: mean number of pitch classes used per bar (from 0 to 12).

• **Qualified Notes (QN)**: ratio of “qualified” notes (in %), i.e., notes with duration greater than a fixed number of timesteps (2 in this work).

• **Drum Pattern (DP)**: ratio of notes presented at the downbeats of 4/4 rhythm in accordance with the utilized resolution (in %).

– **Tonal Distance (TD)**: measures the harmonicity between a pair of tracks as the Euclidean distance between their chroma vectors projected in the interior space of a 6D polytope [24].

• **Used Pitches (UP)**: mean number of unique pitches used per bar, including all octaves in the predefined range.

• **Scale Ratio (SR)**: ratio of notes (in %) in the given music scale (C major).

• **Polyphonic Rate (PR)**: ratio of polyphonic timesteps (in %), i.e., timesteps where the number of pitches being played exceeds a specified threshold (2 in this work).

The first 5 are applied by Dong et al. [2] for the quantitative assessment of MuseGAN. We re-implement them in order to compare the performance of our proposed framework with the baseline on a common objective basis. Following the related literature [25], we also include the last 3, as they provide useful insights into tonal and texture elements of the produced music.

4.4 Subjective Evaluation

In the field of music generation, human evaluation is considered essential, since the objective metrics cannot precisely reflect the human perception over a piece of music. To this end, we conduct a user study in the form of a listening test across 40 subjects mainly recruited via our social circles. As demonstrated in Figure 4, the participants are characterized by diversity regarding various, not necessarily musical, aspects. Our survey is divided into two sections, corresponding to the unconditional and conditional tasks, respectively.

The section of Unconditional Generation aims at a comprehensive comparison between our developed framework and MuseGAN; thus, the respective questionnaire is structured on listening pairs. Each pair consists of two 4-bar musical phrases (ca. 12 sec) that are randomly selected from pools of samples generated by the two models and presented to the user in random order. Each subject evaluates 2 listening pairs, resulting in a total of 80 comparisons.

The section of Accompaniment Generation aims at a comprehensive comparison among multiple variants of our conditional framework, as well as comparison between real and fake accompaniment versions. In this case, the questionnaire is structured on listening triplets, with each one consisting of a conditional track and 2 matching accompaniments for it, derived either from the models or the ground-truth distribution. Both the order of the testing triplets and the sample order within each triplet are randomized for each user. Each subject evaluates 18 listening triplets, resulting in a total of 720 comparisons.

In both questionnaires, the evaluator is required to choose from each listening pair or triplet the sample or accompaniment version that they prefer in terms of:

- **Music Naturalness:** Could the musical segment be composed by human?
- **Harmonic Consistency:** Are the sounds produced by different instruments in musical consonance? Is the result acoustically pleasant?
- **Musical Coherence:** Are the various musical phrases associated somehow through time?

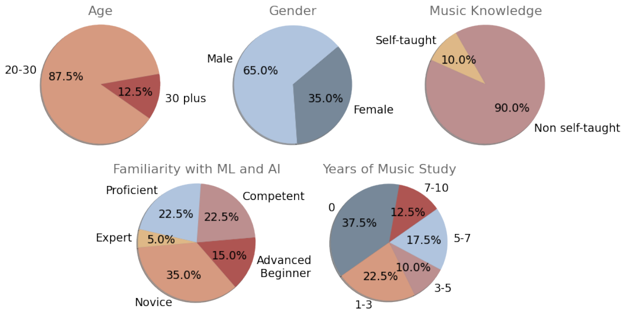


Figure 4: User study demographics.

5. OBJECTIVE EVALUATION RESULTS

5.1 Unconditional Generation

In order to examine the effectiveness of our proposed modifications to the baseline on the task of unconditional generation, we compare our model with MuseGAN using the objective metrics presented in Sec. 4.3. In particular, we select 2 experimental configurations (see Table 1) and compare them with the 4 multi-track models of MuseGAN (the 4th is an ablated version of the composer model which corresponds to the absence of Batch Normalization). We note that $\mathbf{C_1}$ corresponds to MuseGAN's generative setup, applied in our implementation. Following [2], we generate 5,000 4-bar musical phrases with each model and calculate the mean of the objective metrics. Table 2 summarizes the produced results.

		C_1	C_2
Generation parameters	Number of Pitches	84	72
	Beat Resolution	24	4
	Number of Bars	4	4
	Lowest Pitch	24	24
	Samples per song	8	8
	Latent Dimension	128	128
Training parameters	Number of Steps	10000	10000
	Batch Size	16	16
	Number of Phrases	4	4
	Steps per G update	6	6
	Steps per Evaluation	50	50
	Learning Rate	0.001	0.001
	Betas	(0.5, 0.9)	(0.5, 0.9)

Table 1: Experimental configurations.

Regarding the intra-track metrics, we report a small difference between the statistics of the training data used for our proposed architectures and the original MuseGAN [2]. This results from the stricter criteria that were set in our case for selecting random candidate phrases. However, both of our models accomplish to approximate the statistics of the real distribution (bold values denote greater proximity). Moreover, in the case of QN and DP, where the training divergence is negligible, we remark that our framework outperforms almost all the baseline variations to a large degree (colored cells). This fact confirms the beneficial contribution of our parameterized architecture to the rhythm-related attributes of the generated music.

When it comes to the inter-track metric TD (smaller values are considered better), our $\mathbf{C_1}$ model surpasses all the baseline architectures by generating extremely harmonic samples (TD around 0.2 for all pairs). The performance of the $\mathbf{C_2}$ model, even though weaker than $\mathbf{C_1}$, is also conside-rable, especially regarding the harmonic distance between a monophonic track (B) and a chord-like track (P, G, S). We attribute this improvement to the *shared* modules of our architectures, which are responsible for handling the dependencies among the tracks.

5.2 Conditional Generation

In the context of accompaniment generation, we experiment with 8 distinct variants of our conditional framework. As demonstrated in Table 3, these models differ in terms of the employed Discriminator scheme ("✓" for the inclusion of Local Discriminator), the training mode of the Encoder ("-" for 1-phase training, "✓" for 2-phase training) and the type of conditional instrument (Piano, Guitar). We note that all conditional models are designed and trained with the same configuration and parameters as the $\mathbf{C_2}$ model.

5.2.1 *Quantitative Analysis*

In order to investigate the generation capabilities of the 8 conditional models, we utilize the objective metrics presented in Sec. 4.3 and apply the inference process described in Sec. 5.1. Table 4 summarizes the produced results.

In the Piano case, we observe that the utilization of the 2-phase training mode (models P_{10} and P_{11}) benefits some musical characteristics of the generated samples, such as

		EB					UPC				QN				DP	TD (↓)					
Instruments		B	D	G	P	S	B	G	P	S	B	G	P	S	D	B-G	B-S	B-P	G-S	G-P	S-P
training data	*baseline*	8.06	8.06	19.4	24.8	10.1	1.71	3.08	3.28	3.38	90.0	81.9	88.4	89.6	88.6	-	-	-	-	-	-
	ours	1.6	1.1	4.1	5.1	3.2	2.48	4.16	4.2	4.57	91.7	85.3	89.7	89.7	83.1	-	-	-	-	-	-
Baseline	jamming	6.59	2.33	18.3	22.6	6.10	1.53	3.69	4.13	4.09	71.5	56.6	62.2	63.1	93.2	1.56	1.60	1.54	1.05	0.99	1.05
	composer	0.01	28.9	1.34	0.02	0.01	2.51	4.20	4.89	5.19	49.5	47.4	49.9	52.5	75.3	1.37	1.36	1.30	0.95	0.98	0.91
	hybrid	2.14	29.7	11.7	17.8	6.04	2.35	4.76	5.45	5.24	44.6	43.2	45.5	52.0	71.3	1.34	1.35	1.32	0.85	0.85	0.83
	ablated	92.4	100	12.5	0.68	0.00	1.00	2.88	2.32	4.72	0.00	22.8	31.1	26.2	0.0	-	-	-	-	-	-
Ours	C_1	0.0	**0.7**	0.4	1.3	1.2	3.63	4.67	**4.64**	5.29	55.6	**75.8**	**74.1**	**75.9**	59.5	0.2	0.22	0.2	0.21	0.2	0.21
	C_2	**0.3**	0.0	**0.9**	**1.9**	**2.1**	**2.89**	**4.4**	4.88	**5.14**	**59.0**	58.2	57.2	60.8	**79.6**	0.86	0.91	0.9	0.98	0.99	0.97

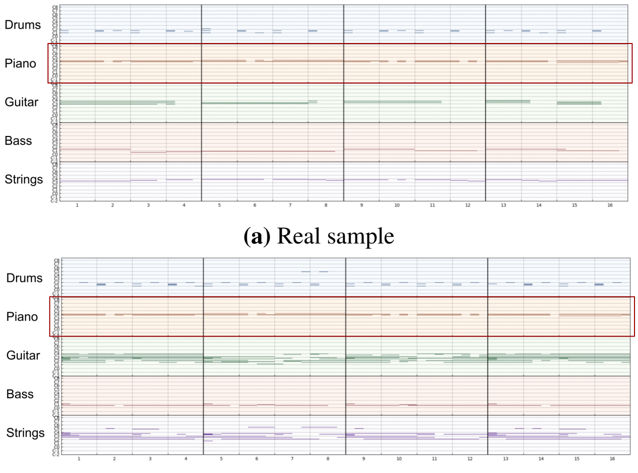
Table 2: Results of the objective evaluation of the baseline models and our proposed framework. For the intra-track metrics values closer to those of the training data are better. For the inter-track metric smaller values are better.

		AutoEncoder	Local Discriminator
Piano	P_{00}	-	-
	P_{01}	-	✓
	P_{10}	✓	-
	P_{11}	✓	✓
Guitar	G_{00}	-	-
	G_{01}	-	✓
	G_{10}	✓	-
	G_{11}	✓	✓

Table 3: Configurations of conditional models.

the note density (EB), the rhythmical patterns (DP), the general tonality (UPC) and foremost the harmonicity between a melody-like and a chord-like track (TD). However, it seems to negatively affect the form of the Bass track by making it more sparse than the original (EB equal to 17.4%). On the other hand, the inclusion of Local Discriminator in the architecture (models P_{01} and P_{11}) has a positive impact on the overall tonality (SR, UP), as well as the fragmentation (QN) and the polyphonicity (PR) level of each track. This outcome confirms that the extra feedback over the quality of the accompaniment parts actually improves the Generator's performance.

In the Guitar case, we observe that similarly to the Piano models the utilization of the 2-phase training mode (models G_{10} and G_{11}) benefits musical characteristics related to overall note density (EB) and tonality (UP, UPC, SR), especially for the chord-like instruments. On the other hand, the inclusion of Local Discriminator results in stronger harmonic relations between the tracks (TD) and improves the rhythm (DP) as well as texture elements such as PR.



(a) Real sample

(b) Generated sample

Figure 5: Examples of pianorolls with common conditional track.

5.2.2 *Qualitative Analysis*

Figure 5 illustrates 2 pianorolls representing 4-bar musical phrases with a common conditional Piano track. The first one is a real sample, while the second has been created by the model P_{11} (see Table 3). We can observe the following: **(a)** The fake Drum track follows a rhythmic pattern that resembles the original. **(b)** The fake Bass track is mainly monophonic playing the lowest pitches. **(c)** The fake Strings and Guitar tracks tend to play the chord-like parts, but are quite noisy. We attribute this difference to these tracks being more rhythmically complex and polyphonic as compared to Drums and Bass. **(d)** All tracks usually play in the same music scale. Overall, our model is capable of learning some musical properties of the accompaniments, but tends to incorporate more noise than the original samples.

6. SUBJECTIVE EVALUATION RESULTS

6.1 Unconditional Generation

The results of our subjective testing for the unconditional case are graphically illustrated in Figure 6. The bar-plot represents the evaluators' preferences between the compared models under the examined musical criteria in the form of percentages (%). As can be seen, our developed framework outperforms MuseGAN with respect to all the examined musical aspects.

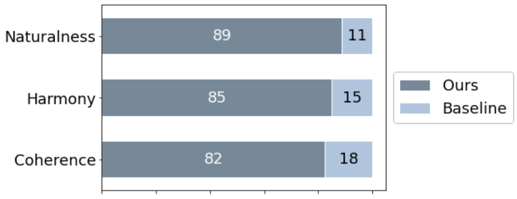


Figure 6: Subjective results for unconditional generation.

This outcome confirms the contribution of our proposed modifications and extensions to the quality of the generated music. More specifically, we attribute the improvement in Naturalness and Coherence to our parameterized architecture that emphasizes on rhythmical attributes, since an evident beat pattern is capable of creating a sense of cohesion and connectivity among the various parts of a music piece and hence is considered a key feature of human-composed songs. We also claim that the stronger harmonic relations among the tracks and the enhanced overall tonality are the results of the shared/private design we employ for both Generator and Discriminator modules.

	EB					UPC				QN				UP			
Instruments	B	D	G	P	S	B	G	P	S	B	G	P	S	B	G	P	S
Piano train	1.6	1.0	5.0	5.6	3.7	2.47	4.09	4.19	4.5	91.6	85.6	90.0	89.7	2.71	5.68	5.85	6.71
Guitar train	1.8	0.9	4.3	5.2	3.6	2.47	4.21	4.14	4.49	91.8	87.5	91.6	90.5	2.7	5.85	5.84	6.75
P_{00}	0.6	0.0	2.2	-	2.4	2.71	3.93	-	**4.33**	51.4	56.5	-	58.9	2.94	5.79	-	**6.28**
P_{01}	0.2	0.0	1.8	-	1.5	2.57	**4.09**	-	4.76	**58.2**	56.1	-	**61.7**	2.94	**5.77**	-	7.17
P_{10}	17.4	**0.2**	**3.0**	-	**4.4**	1.68	3.9	-	4.3	50.7	49.2	-	55.1	1.74	5.05	-	6.07
P_{11}	**1.6**	0.0	0.7	-	0.9	**2.56**	4.19	-	5.16	54.8	**56.6**	-	51.0	**2.84**	5.43	-	7.3
G_{00}	0.8	0.0	-	2.1	1.8	**2.51**	-	5.04	**4.59**	**62.5**	-	49.3	**60.3**	**2.77**	-	7.31	6.91
G_{01}	0.0	0.0	-	3.1	0.0	3.05	-	4.31	5.28	57.6	-	52.4	59.6	3.36	-	6.18	7.69
G_{10}	**1.6**	0.0	-	1.8	**3.5**	2.35	-	**4.28**	4.01	50.2	-	**59.5**	58.6	2.59	-	**6.13**	5.88
G_{11}	0.4	**0.2**	-	**3.3**	0.6	2.32	-	4.62	4.66	55.6	-	47.8	57.9	2.46	-	6.4	**6.68**

	TD (↓)						SR				PR					DP
Instruments	B−G	B−S	B−P	G−S	G−P	S−P	B	G	P	S	B	D	G	P	S	D
Piano train	-	-	-	-	-	-	75.9	74.4	74.1	72.8	1.1	15.2	55.7	61.8	62.3	82.9
Guitar train	-	-	-	-	-	-	75.4	73.5	73.4	73.1	0.8	15.5	59.7	61.0	62.6	85.0
P_{00}	0.82	0.83	0.88	0.87	0.95	**0.94**	81.7	**75.8**	-	77.1	**1.2**	13.3	40.6	-	44.2	**86.1**
P_{01}	0.79	0.81	0.85	**0.85**	**0.94**	**0.94**	**77.1**	76.3	-	75.6	1.5	**15.2**	**48.7**	-	**59.9**	86.3
P_{10}	**0.74**	**0.73**	**0.81**	0.94	1.02	1.01	82.2	80.6	-	79.0	0.2	10.1	22.2	-	30.2	87.0
P_{11}	0.83	0.92	0.97	0.99	1.12	1.17	80.7	77.6	-	**72.3**	1.9	9.7	38.2	-	56.3	86.2
G_{00}	**0.83**	0.85	0.9	0.96	1.01	0.98	84.7	-	80.9	**77.0**	1.1	10.9	-	53.9	53.4	87.1
G_{01}	0.87	0.87	**0.83**	**0.93**	**0.92**	**0.86**	86.7	-	83.6	83.9	2.8	**14.9**	-	**55.3**	**60.8**	**86.0**
G_{10}	0.84	**0.84**	0.84	**0.93**	0.95	0.89	82.0	-	79.8	85.4	**0.7**	6.0	-	37.5	44.0	91.7
G_{11}	0.89	0.87	0.88	1.06	1.09	0.97	**78.0**	-	**76.9**	80.5	**0.9**	9.7	-	42.1	54.4	83.7

Table 4: Results of the objective evaluation of our conditional models. For the intra-track metrics values closer to those of the training data are better. For the inter-track metric smaller values are better.

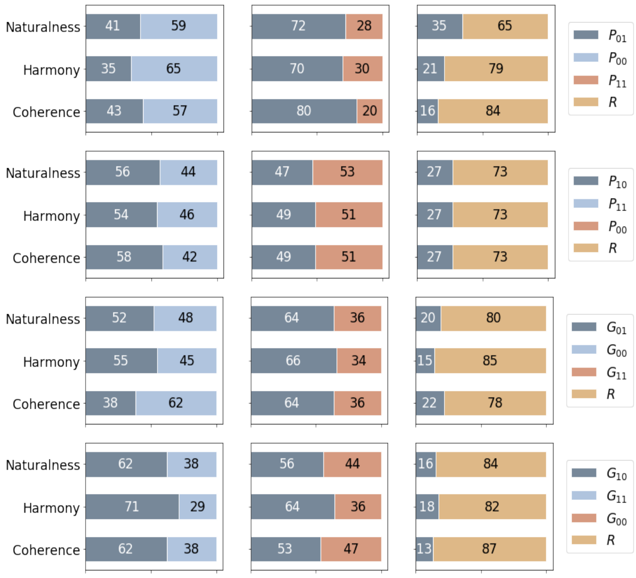


Figure 7: Subjective results for conditional generation.

6.2 Conditional Generation

The results of our subjective testing for the conditional models are illustrated in Figure 7. Each row demonstrates the comparisons of a model with two others, one differing in discriminator scheme (left column), and one differing in training procedure (middle column), as well as with real samples, denoted as R (right column). Similar to the unconditional case, each bar-plot represents the users' preferences (%) between the compared models for each criterion.

For the Piano models, we observe that the majority of fake accompaniments are easily distinguishable from the real ones. The highest favor proportion against human performance corresponds to model P_{01} for Naturalness and is equal to 35%, indicating that the additional feedback provided by the Local Discriminator can actually help the Generator to create samples that sound more natural to the human subjects. When it comes to the comparison between model variants, we notice that P_{01} outperforms P_{11} with respect to all the examined musical aspects, especially Coherence. This result suggests that the most suitable training practice for the architecture of both Discriminators is the 1-phase mode. We can also observe that P_{10} outperforms P_{11}, indicating that the proper structural design for the 2-phase mode includes only the Global Discriminator.

The results for the Guitar models are similar to those of the Piano case. More specifically, in terms of comparison with the real accompaniments, the fake versions are easily distinguishable under all musical criteria but with lower favor proportions ranging from 13 to 20%. This fact probably indicates that Guitar, which typically plays the chords in Rock songs, provides less conditional information than Piano, which includes some melodic features as well. We can also observe that G_{10} outperforms both G_{00} and G_{11} with respect to all musical aspects. This outcome suggests that the most effective combination of training practice and architectural design is the 2-phase mode applied in a GAN with only the Global Discriminator. We also notice that G_{01} surpasses G_{11}, indicating that the most suitable training practice for the architecture of both Discriminators is the 1-phase mode. However, we remark that the utilization of only the Global Discriminator with the 1-phase training setup seems to have a beneficial impact on the coherence of the generated accompaniments (G_{00} scores 62% in the pairwise comparison against G_{01}).

7. CONCLUSIONS

In this work, we have presented a configurable generative framework that is capable of creating multi-track polyphonic musical phrases from scratch and also generating multi-instrumental accompaniments for human-composed tracks. Our proposed architecture is established on MuseGAN and employs a hierarchical shared/private design for both Generator and Discriminator modules, which is adaptable to different generative configurations. We evaluate our models objectively, using a set of widely used musical metrics, and subjectively by conducting a user study across 40 listeners. The results demonstrated that our model outperforms MuseGAN in the unconditional setup under 3 musical criteria and also provided useful insights on training and structural schemes for conditional architectures that pave the way for further exploration of the accompaniment generation field. As future work, we aim at validating these findings on transformer-based architectures and using other feature representations.

Acknowledgments

This research was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "3rd Call for H.F.R.I. Research Projects to support Post-Doctoral Researchers" (Project Number: 7773).

8. REFERENCES

[1] I. Morley, *The Prehistory of Music: Human Evolution, Archaeology, and the Origins of Musicality*. Oxford University Press, 2013.

[2] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," in *Proc. AAAI 2018*, New Orleans, LA, USA, 2018.

[3] D. Herremans, C.-H. Chuan, and E. Chew, "A Functional Taxonomy of Music Generation Systems," *ACM Comp. Surveys (CSUR)*, vol. 50, no. 5, pp. 1–30, 2017.

[4] D. Herremans and E. Chew, "MorpheuS: Generating Structured Music with Constrained Patterns and Tension," *IEEE Trans. on Affective Computing*, 2017.

[5] C.-Z. A. Huang *et al.*, "Music Transformer: Generating Music with Long-Term Structure," in *Proc. ICLR 2018*, Vancouver, BC, Canada, 2018.

[6] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv preprint arXiv:1511.06434*, 2015.

[7] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending against Neural Fake News," in *Proc. NeurIPS 2019*, vol. 32, Vancouver, BC, Canada, 2019.

[8] O. Mogren, "C-RNN-GAN: Continuous Recurrent Neural Networks with Adversarial Training," *arXiv preprint arXiv:1611.09904*, 2016.

[9] L. Yu *et al.*, "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient," in *Proc. AAAI 2017*, San Francisco, CA, USA, 2017.

[10] H.-W. Dong and Y.-H. Yang, "Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation," *arXiv preprint arXiv:1804.09399*, 2018.

[11] F. Guan, C. Yu, and S. Yang, "A GAN Model with Self-Attention Mechanism to Generate Multi-Instruments Symbolic Music," in *Proc. IJCNN 2019*, Budapest, Hungary, 2019.

[12] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation," *arXiv preprint arXiv:1703.10847*, 2017.

[13] H.-M. Liu and Y.-H. Yang, "Lead Sheet Generation and Arrangement by Conditional Generative Adversarial Network," in *Proc. ICMLA 2018*, Orlando, Florida, USA, 2018.

[14] N. Trieu and R. Keller, "JazzGAN: Improvising with Generative Adversarial Networks," in *Proc. MUME 2018*, Salamanca, Spain, 2018.

[15] N. Jiang *et al.*, "RL-Duet: Online Music Accompaniment Generation using Deep Reinforcement Learning," in *Proc. AAAI 2020*, New York, NY, USA, 2020.

[16] Y. Ren, J. He1, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, "PopMAG: Pop Music Accompaniment Generation," in *Proc. ICME 2020*, New York, NY, USA, 2020.

[17] C. Donahue, A. Caillon, A. Roberts *et al.*, "SingSong: Generating Musical Accompaniments from Singing," *arXiv preprint arXiv:2301.12662*, 2023.

[18] D. Zhang, J.-C. Wang, K. Kosta, J. B. Smith, and S. Zhou, "Modeling the Rhythm from Lyrics for Melody Generation of Pop Song," *arXiv preprint arXiv:2301.01361*, 2023.

[19] C. Raffel, "Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching," Ph.D. dissertation, Ph.D. Thesis, Columbia University, 2016.

[20] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The Million Song Dataset," in *Proc. ICWWW 2012*, Lyon, France, 2012.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza *et al.*, "Generative Adversarial Nets," in *Proc. NeurIPS 2014*, Montreal, Canada, 2014.

[22] I. Gulrajani, F. Ahmed, M. Arjovsky *et al.*, "Improved Training of Wasserstein GANs," in *Proc. NeurIPS 2017*, Long Beach, CA, USA, 2017.

[23] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[24] C. Harte, M. Sandler, and M. Gasser, "Detecting Harmonic Change in Musical Audio," in *Proc. ICAMCM, Santa Barbara, CA, USA*, 2006.

[25] S. Ji, J. Luo, and X. Yang, "A Comprehensive Survey on Deep Music Generation: Multi-Level Representations, Algorithms, Evaluations, and Future Directions," *arXiv preprint arXiv:2011.06801*, 2020.