Learning to Fuse Music Genres with Generative Adversarial Dual Learning

Zhiqian Chen*, Chih-Wei Wu†, Yen-Cheng Lu*, Alexander Lerch† and Chang-Tien Lu*
*Computer Science Department, Virginia Tech
Email:{czq,kevinlu,ctlu}@vt.edu
†Center for Music Technology, Georgia Institute of Technology
Email:{cwu307, alexander.lerch}@gatech.edu

***Abstract*—FusionGAN is a novel genre fusion framework for music generation that integrates the strengths of generative adversarial networks and dual learning. In particular, the proposed method offers a dual learning extension that can effectively integrate the styles of the given domains. To efficiently quantify the difference among diverse domains and avoid the vanishing gradient issue, FusionGAN provides a Wasserstein based metric to approximate the distance between the target domain and the existing domains. Adopting the Wasserstein distance, a new domain is created by combining the patterns of the existing domains using adversarial learning. Experimental results on public music datasets demonstrated that our approach could effectively merge two genres.**

I. Introduction

Computational creativity is a lively research area that focuses on understanding and facilitating human creativity through the implementation of software programs [2]. With the rapid advances in data-driven algorithms such as deep learning [7], the exploration into computational creativity via machine learning approaches has become increasingly popular. Examples showing the potential of deep learning for creative approaches are the artistic style transfer on images [4] and videos [12]. Music, with its complex hierarchical and sequential structure and its inherent emotional and aesthetic subjectivity, is an intriguing research subject at the core of human creativity. While there has been some work on generative models for music, research that investigates the capabilities of deep learning for creative applications such as style transfer is limited. This paper aims to fill this space by exploring the idea of style fusion in music with generative adversarial dual learning.

In the field of unsupervised generative learning, generative adversarial networks (GAN) [6] have recently gained considerable attention. It is important to note, however, that GANs are designed to learn from a single domain and cannot discover cross-domain knowledge. Inspired by dual learning in machine translation, several recent publications propose methods to pair unlabeled data points in different domains by optimizing bi-directional reconstruction errors [14], [8], [16]. Although these methods are designed to address the challenge of multimodal learning, they do not address the problem of domain fusion.

The goal of this paper is to provide a solution for fusing two or more groups of sequential patterns using unsupervised methods. Specifically, we focus on the problem of generating music with a fused music genre. To combine genres using generative adversarial learning, we propose a framework for integrating multiple GANs. Unlike previous work, our study targets the creation of an unknown domain by mixing two given domains; a multi-way GAN-based model is proposed to absorb existing patterns and yield a new mixture. With the utilization of the Wasserstein measure [1], we pose a distance constraint on the new domain that automatically balances its relation to the given domains. As a result, our model is capable of generating a mixed pattern after convergence. The main contributions of this paper are:

- **A novel framework for unsupervised music fusion**: The dual learning scheme is extended to involve multiple domains by leveraging mutual regularization. In this way, the proposed framework enables the new domain to keep equal similarities with the given domains and produce a mixture of existing sequential patterns.
- **A sequence fusion method using GANs**: To apply unsupervised fusion learning, we extend the generative adversarial model so that multiple generators and discriminators can be updated by one another.
- **Formulation of an objective function indicating fusion progress**: An objective function based on Wasserstein distance is designed to integrate the information from multiple domains and represent learning progress.

II. Related Work

Music Genre Fusion: Fusion is mostly known as the sub-genre of jazz that emerged in the late '60s, and that combines several musical styles such as funk, rock, and blues with the jazz harmony and improvisation [5]. The term can be generalized, however, to any combinations of music genres. The creative combination of two or more genres obviously requires not only intimate knowledge of the stylistic characteristics of the involved genres but also extensive compositional experience and skill to combine genres in a meaningful and satisfying way. Recently, Engel et al. [3] proposed a new approach of generating new musical sounds through interpolating the latent space learned by WaveNet [11] autoencoders. Additionally, Gatys et al. demonstrated the Deep Neural Networks' (DNNs) capabilities of learning the artistic styles by blending the style of the training materials to the testing images using ConvNet [4]. Both examples show the great potential of using DNNs for merging two abstract concepts without the need for domain knowledge and explicitly defined rules.

Cross-domain GAN: GAN [6], which has quickly risen to one of the most popular generative approaches, learns patterns without requiring adversarial examples or labels. In CGAN [10], the generator G has a conditional parameter c and will learn the conditional distribution of the data.

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.98

However, those conditions need to be provided manually, somewhat similar to supervised learning. CoGAN [9] is proposed to learn a joint distribution with only samples drawn from the marginal distributions. DiscoGAN [8] solves the cross-domain pairing by minimizing bi-directional reconstruction loss, while DualGAN [14] takes advantage of the dual learning paradigm in machine translation. CycleGAN [16] presents a method whose mapping functions are cycle-consistent and proposes a cycle consistency loss function to further reduce the space of possible mapping functions.

Different from the previous studies, our work raises a new problem, i.e., merging two sequential patterns into one. We propose a GAN-based framework that merges two domains without manually tuning the distance between the given domains and the target domain.

III. Learning to Blend Music with FusionGAN

A. Problem Definition

This paper aims to characterize music fusion across different genres in an unsupervised manner. Based on the data $\mathbf{X}_\mathrm{A}$ and $\mathbf{X}_\mathrm{B}$ from existing music domains $\mathcal{D}_A, \mathcal{D}_B$ respectively, our goal is to create a new domain $\mathcal{D}_F$ which resembles both $\mathcal{D}_A$ and $\mathcal{D}_B$. In the first phase, the new domain $\mathcal{D}_F$ is expected to learn from $\mathcal{D}_A, \mathcal{D}_B$ and keep an equal distance from both. Specifically, the generator $\mathbf{G}_\mathrm{F}$ of $\mathcal{D}_F$ obtains feedback from the discriminators $\mathbf{D}_\mathrm{A}, \mathbf{D}_\mathrm{B}$ of existing domains $\mathcal{D}_A, \mathcal{D}_B$, while the discriminator $\mathbf{D}_\mathrm{F}$ collects data from $\mathbf{G}_\mathrm{F}$ and $\mathcal{D}_A, \mathcal{D}_B$ for iterative updating. The $\mathbf{G}_\mathrm{F}$ from the new domain minimizes the distance from the domains $\mathcal{D}_A, \mathcal{D}_B$, respectively, and keeps the same distance from $\mathcal{D}_A$ and $\mathcal{D}_B$.

B. FusionGAN

First, the framework initializes individual GAN models for existing domains. Maximum likelihood estimation using Long Short Term Memory (LSTM) is applied to initialize the sequence generators $\mathbf{G}_\mathrm{A}(\mathbf{G}_\mathrm{B})$. Then $\mathbf{D}_\mathrm{A}(\mathbf{D}_\mathrm{B})$ are trained given the domain data and sequence sampled from $\mathbf{G}_\mathrm{A}(\mathbf{G}_\mathrm{B})$. Then GAN is employed to iteratively enhance the $\mathbf{G}_\mathrm{A}(\mathbf{G}_\mathrm{B})$ and $\mathbf{D}_\mathrm{A}(\mathbf{D}_\mathrm{B})$. After pre-training, we build a GAN model for the new domain using the feedback from the models constructed in the pre-training procedure. First, a new domain $\mathcal{D}_F$ is initialized randomly and trained by both $\mathcal{D}_A$ and $\mathcal{D}_B$. Following the dual learning strategy, $\mathcal{D}_A$ is enhanced by $\mathcal{D}_B$ and $\mathcal{D}_F$ in the second phase. Similarly, $\mathcal{D}_B$ is improved by $\mathcal{D}_A$ and $\mathcal{D}_F$. Such learning proceeds until convergence. The framework overview is shown in Figure 1.

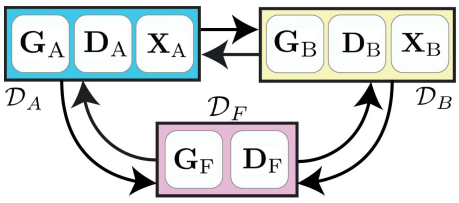

Figure 1: FusionGAN framework

In FusionGAN, we employ the Wasserstein-1 distance for all discriminators because it is sensible when the learning distributions are supported by low dimensional manifolds [1]. To avoid the intractable infimum in Wasserstein-1 distance, we resort to its Kantorovich-Rubinstein duality [13]: $W\left(\mathbb{P}_r, \mathbb{P}_\theta\right) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}\left[f\left(x\right)\right] - \mathbb{E}_{x\sim\mathbb{P}_\theta}\left[f\left(x\right)\right]$ For the vanilla GAN, the goal is to find the optimal configuration of the parameters ϕ of discriminator ($f = \mathbf{D}$). When the discriminator is optimized, the maximized Wasserstein distance can be used as reward in the policy gradient process of the generator. Similarly, we define a three-way Wasserstein distance as the objective function: $W\left(\mathbb{P}_{\mathbf{X}_\mathrm{A}}, \mathbb{P}_{\mathbf{X}_\mathrm{B}}, \mathbb{P}_\theta\right) = \mathcal{L}$ to measure the integrated distance between $\mathbb{P}_\theta$ and $\mathbb{P}_{\mathbf{X}_\mathrm{A}}, \mathbb{P}_{\mathbf{X}_\mathrm{B}}$. This measure should consider the relationship between every pair of domains. The framework consists of three pairs of generators and discriminators, i.e., $\mathbf{G}_\mathrm{A} - \mathbf{D}_\mathrm{A}, \mathbf{G}_\mathrm{B} - \mathbf{D}_\mathrm{B}, \mathbf{G}_\mathrm{F} - \mathbf{D}_\mathrm{F}$, and two input data, i.e., $\mathbf{X}_\mathrm{A}, \mathbf{X}_\mathrm{B}$. For each discriminator, there are five possible inputs, i.e., $\mathbf{X}_\mathrm{A}, \mathbf{X}_\mathrm{B}, \mathbf{G}_\mathrm{A}, \mathbf{G}_\mathrm{B}, \mathbf{G}_\mathrm{F}$. Collecting all possible inputs w.r.t. the three discriminators, the objective function to maximize is defined as:

$$
\begin{aligned}
\mathcal{L} = \quad & W\left(\mathbb{P}_{\mathbf{X}_\mathrm{A}}, \mathbb{P}_{\mathbf{X}_\mathrm{B}}, \mathbb{P}_\theta\right) \\
& \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{A}}}\left[\mathbf{D}_\mathrm{A}\left(x\right)\right] - \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{B}}}\left[\mathbf{D}_\mathrm{A}\left(x\right)\right] - \\
& \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right)\right] - \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{B}\left(z\right)\right)\right] - \\
& \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)\right] - \\
& \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{A}}}\left[\mathbf{D}_\mathrm{B}\left(x\right)\right] + \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{B}}}\left[\mathbf{D}_\mathrm{B}\left(x\right)\right] - \\
& \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{B}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right)\right] - \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{B}\left(\mathbf{G}_\mathrm{B}\left(z\right)\right)\right] - \\
& \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{B}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)\right] + \\
& \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{A}}}\left[\mathbf{D}_\mathrm{F}\left(x\right)\right] + \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{B}}}\left[\mathbf{D}_\mathrm{F}\left(x\right)\right] + \\
& \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right)\right] + \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{B}\left(z\right)\right)\right] - \\
& \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)\right],
\end{aligned}
\tag{1}
$$

where the first five terms are for $\mathbf{D}_\mathrm{A}$, the second five terms are for $\mathbf{D}_\mathrm{B}$, and the last five terms are for $\mathbf{D}_\mathrm{F}$. This loss is used to update all the generators and discriminators. The following two subsections, III-C and III-D, will present the update rules using Eq. 1.

C. $\mathcal{D}_F$ Update of FusionGAN

Removing the terms that are unrelated to $\mathcal{D}_F$ in Eq. 1 (zero derivative w.r.t. $\mathbf{D}_\mathrm{F}$ and $\mathbf{G}_\mathrm{F}$), we have the objective function for updating $\mathbf{D}_\mathrm{F}$ and $\mathbf{G}_\mathrm{F}$:

$$
\begin{aligned}
\mathcal{L}_F = \quad & \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{A}}}\left[\mathbf{D}_\mathrm{F}\left(x\right)\right] + \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{B}}}\left[\mathbf{D}_\mathrm{F}\left(x\right)\right] + \\
& \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right)\right] + \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{B}\left(z\right)\right)\right] - \\
& \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)\right] - \\
& \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)\right] - \\
& \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{B}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)\right].
\end{aligned}
\tag{2}
$$

First, $\mathcal{L}_F$ is calculated w.r.t. $\mathbf{G}_\mathrm{F}$ so as to update $\mathbf{G}_\mathrm{F}$. Considering differentiating $\nabla_\theta \mathcal{L}_F$, the optimal $\mathbf{G}_\mathrm{F}$ is approximated as its convergence. The proof is provided below to show that this derivative is principled under the optimality assumption.

Theorem III.1 (FusionGAN Optimality Theorem)**.** *Let $\mathbb{P}_{\mathbf{X}_\mathrm{A}}$ and $\mathbb{P}_{\mathbf{X}_\mathrm{B}}$ be any distribution, Let $\mathbb{P}_\theta$ be the distribution of $\mathbf{G}_{\mathrm{F}\theta}(Z)$ with Z, a random variable with density p and $\mathbf{G}_{\mathrm{F}\theta}$ a function satisfying $\mathbb{E}_{z\sim p}[L(\theta, z)] < +\infty$, where L is Lipschitz constants. All functions are well-defined. Then there is a solution $\mathbf{D}_\mathrm{F} : \mathcal{X} \to \mathbb{R}$ to the problem:*

$$
W\left(\mathbb{P}_{\mathbf{X}_\mathrm{A}}, \mathbb{P}_{\mathbf{X}_\mathrm{B}}, \mathbb{P}_\theta\right) = \max_{\|\mathbf{D}_\mathrm{F}\|_L \leq 1} \mathcal{L}_F\,. \tag{3}
$$

and we have:

$$
\begin{aligned}
\nabla_\theta \mathcal{L}_{\mathbf{G}_\mathrm{F}} =& \nabla_\theta W\left(\mathbb{P}_{\mathbf{X}_\mathrm{A}}, \mathbb{P}_{\mathbf{X}_\mathrm{B}}, \mathbb{P}_\theta\right) \\
=& -\mathbb{E}_{z\sim p(z)} \nabla_\theta \sum_{i\in\{A,B,F\}} \mathbf{D}_i\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)
\end{aligned}
\tag{4}
$$

Proof of Theorem III.1: Let us define $\mathcal{L}_F = V(\tilde{\mathbf{D}}_\mathrm{F}, \theta)$ where $\tilde{\mathbf{D}}_\mathrm{F}$ lies in $\mathcal{F} = \{\tilde{\mathbf{D}}_\mathrm{F} : \mathcal{X} \to \mathbb{R}, \tilde{\mathbf{D}}_\mathrm{F} \in C_b(\mathcal{X}), \|\tilde{\mathbf{D}}_\mathrm{F}\|_L \leq 1\}$ and $\theta \in \mathbb{R}_d$. By the Kantorovich-Rubinstein duality, there is an $f \in \mathcal{F}$ that satisfies:

$$W\left(\mathbb{P}_{\mathbf{X}_\mathrm{A}}, \mathbb{P}_{\mathbf{X}_\mathrm{B}}, \mathbb{P}_\theta\right) = \sup_{\tilde{\mathbf{D}}_\mathrm{F} \in \mathcal{F}} V(\tilde{\mathbf{D}}_\mathrm{F}, \theta) = V(\mathbf{D}_\mathrm{F}, \theta).$$

Let us define $X^*(\theta) = \{\mathbf{D}_\mathrm{F} \in \mathcal{F} : V(\mathbf{D}_\mathrm{F}, \theta) = W\left(\mathbb{P}_{\mathbf{X}_\mathrm{A}}, \mathbb{P}_{\mathbf{X}_\mathrm{B}}, \mathbb{P}_\theta\right)\}$, which shows that $X^*(\theta)$ is non-empty. According to Theorem 1 in [1]: (1) if $\mathbf{G}_\mathrm{F}$ is continuous in θ, so is $W\left(\mathbb{P}_{\mathbf{X}_\mathrm{A}}, \mathbb{P}_{\mathbf{X}_\mathrm{B}}, \mathbb{P}_\theta\right)$, (2) If $\mathbf{G}_\mathrm{F}$ is locally Lipschitz and satisfies $\mathbb{E}_{z\sim p}[L(\theta, z)] < +\infty$, then $W\left(\mathbb{P}_{\mathbf{X}_\mathrm{A}}, \mathbb{P}_{\mathbf{X}_\mathrm{B}}, \mathbb{P}_\theta\right)$ is continuous and differentiable for any $\mathbf{D}_\mathrm{F} \in X^*(\theta)$ when both terms are well-defined. Let $\mathbf{D}_\mathrm{F} \in X^*(\theta)$, which we knows exists since $X^*(\theta)$is non-empty for all θ. Then:

$$\begin{aligned}
&\nabla_\theta W\left(\mathbb{P}_{\mathbf{X}_\mathrm{A}}, \mathbb{P}_{\mathbf{X}_\mathrm{B}}, \mathbb{P}_\theta\right) \\
=&\nabla_\theta V(\mathbf{D}_\mathrm{F}, \theta) = \nabla_\theta \mathcal{L}_F \\
=&-\nabla_\theta \mathbb{E}_{z\sim p(z)}[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right) + \mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right) + \mathbf{D}_\mathrm{B}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)],
\end{aligned} \tag{5}$$

since $\mathbf{D}_\mathrm{F}$ is 1-Lipschitz. Furthermore, $\mathbf{G}_{\mathrm{F}\theta}(z)$ is locally Lipschitz as a function of (θ, z). Therefore, $\mathbf{G}_{\mathrm{F}\theta}(z)$ is locally Lipschitz on (θ, z) with constants $L(\theta, z)$. By Radamachers Theorem, $\mathbf{D}_\mathrm{F}(\mathbf{G}_{\mathrm{F}\theta}(z))$ has to be differentiable almost everywhere for (θ, z) jointly. Rewriting this, the set $A = \{(\theta, z) : \mathbf{D}_\mathrm{F} \circ \mathbf{G}_\mathrm{F} \text{ is not differentiable}\}$ has measure 0. By Fubinis Theorem, this implies that for almost every θ the section $A_\theta = \{z : (\theta, z) \in A\}$ has measure 0. Let's fix a θ_0 such that the measure of A_{θ_0} is null. For this θ_0 we have $\nabla_\theta \mathbf{D}_\mathrm{F}(\mathbf{G}_{\mathrm{F}\theta}(z))|\theta_0$ is well-defined for almost any z, and since $p(z)$ has a density, it is defined $p(z)$ almost everywhere. Given the condition $\mathbb{E}_{z\sim p}[L(\theta, z)] < +\infty$, we have

$$\mathbb{E}_{z\sim p(z)}[\|\nabla_\theta \mathbf{D}_\mathrm{F}(\mathbf{G}_{\mathrm{F}\theta}(z))\|] \leq \mathbb{E}_{z\sim p(z)}[L(\theta_0, z)] < +\infty, \tag{6}$$

so $\mathbb{E}_{z\sim p(z)}[\nabla_\theta f(g_\theta(z))]$ is well-defined for almost every θ_0. To keep the notation simple, we leave it implicit in the following equation that the $\mathbb{E}$ subjects to $z \sim p(z)$. :

$$\begin{aligned}
&\frac{1}{\|\theta - \theta_0\|} \sum_i \Big[\mathbb{E}[\mathbf{D}_i(\mathbf{G}_{\mathrm{F}\theta}(z))] - \mathbb{E}[\mathbf{D}_i(\mathbf{G}_{\mathrm{F}\theta_0}(z))] - \\
&\langle(\theta - \theta_0), \mathbb{E}[\nabla_\theta[\mathbf{D}_i(\mathbf{G}_{\mathrm{F}\theta}(z))]]|_{\theta_0}\rangle\Big] \\
=&\frac{1}{\|\theta - \theta_0\|} \sum_i \mathbb{E}\Big[\mathbf{D}_i(\mathbf{G}_{\mathrm{F}\theta}(z)) - \mathbf{D}_i(\mathbf{G}_{\mathrm{F}\theta_0}(z)) - \\
&\langle(\theta - \theta_0), \nabla_\theta[\mathbf{D}_i(\mathbf{G}_{\mathrm{F}\theta}(z))]|_{\theta_0}\rangle\Big] \\
\leq&\frac{1}{\|\theta - \theta_0\|} \sum_i \left\| \|\theta - \theta_0\| L(\theta_0, z) + \|\theta - \theta_0\| \cdot \|\nabla_\theta[\mathbf{D}_i(\mathbf{G}_{\mathrm{F}\theta}(z))]\| \right\| \\
\leq&\sum_i 2L(\theta_0, z) = 6L(\theta_0, z),
\end{aligned} \tag{7}$$

where $i \in \{A, B, F\}$. By differentiability, the term inside the integral converges $p(z)$ to 0 as $\theta \to \theta_0$. Furthermore, and since $\mathbb{E}_{z\sim p(z)}[6L(\theta_0, z)] < +\infty$, we get by dominated convergence that Eq. 5 converges to 0 as $\theta \to \theta_0$. So the result of Eq. 5 equals to:

$$-\mathbb{E}_{z\sim p(z)}[\nabla_\theta \mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right) + \nabla_\theta \mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right) + \nabla_\theta \mathbf{D}_\mathrm{B}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)].$$

■

Therefore, the update rules for the parameters θ of $\mathbf{G}_\mathrm{F}$ is Eq. 4.

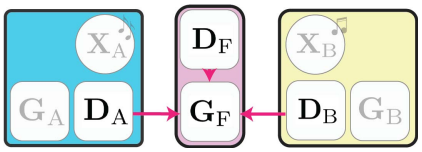

Figure 2: FusionGAN: update $\mathbf{G}_\mathrm{F}$

The physical meaning is that all three discriminators can offer feedback to generator $\mathbf{G}_\mathrm{F}$ as shown in Fig. 2. The left cyan rectangle indicates $\mathcal{D}_A$, the middle pink one is $\mathcal{D}_F$, and the right yellow one $\mathcal{D}_B$. Similarly, the gradient w.r.t. the parameters ϕ of $\mathbf{D}_\mathrm{F}$ is:

$$\begin{aligned}
\nabla_\phi \mathcal{L}_F = \quad &\nabla_\phi \big[\mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{A}}}\left[\mathbf{D}_\mathrm{F}\left(x\right)\right] + \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{B}}}\left[\mathbf{D}_\mathrm{F}\left(x\right)\right] + \\
&\mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right)\right] + \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{B}\left(z\right)\right)\right] - \\
&\mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)\right] \big].
\end{aligned} \tag{8}$$

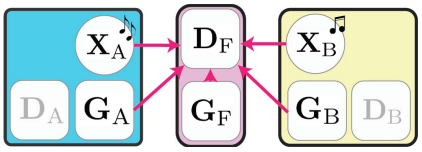

Figure 3: FusionGAN: update $\mathbf{D}_\mathrm{F}$

As shown in Fig. 3, the update for $\mathbf{D}_\mathrm{F}$ actually implicitly balances itself between $\mathcal{D}_A$ and $\mathcal{D}_B$. However, this update does not cover the degree of blending which may lead to an unbalanced mixture. Specifically, a good learning process for the discriminator $\mathbf{D}_\mathrm{F}$ should consider two factors: (1) minimizing the distance with $\mathcal{D}_A$, $\mathcal{D}_B$ and $\mathcal{D}_F$ simultaneously and (2) maintaining a equal distance from $\mathcal{D}_A$ and $\mathcal{D}_B$ so as to satisfy both domains. To satisfy (2) and push the discriminator $\mathbf{D}_\mathrm{F}$ to have a well-proportioned blending, an additional constraint is employed to further balance the ratio of existing domains, i.e., $\mathcal{D}_A, \mathcal{D}_B$:

$$\begin{aligned}
\mathcal{L}_{F-bal} = \quad &\left\| \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right)\right] - \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{B}\left(z\right)\right)\right] \right\| + \\
&\left\| \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{A}}}\left[\mathbf{D}_\mathrm{F}\left(x\right)\right] - \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{B}}}\left[\mathbf{D}_\mathrm{F}\left(x\right)\right] \right\|.
\end{aligned} \tag{9}$$

D. $\mathcal{D}_A$ and $\mathcal{D}_B$ Update of FusionGAN

After updating $\mathcal{D}_F$, FusionGAN will improve $\mathcal{D}_A$ and $\mathcal{D}_B$. Since $\mathcal{D}_A$ and $\mathcal{D}_B$ are symmetric in the framework, we only discuss $\mathcal{D}_A$ as shown in Fig. 9. Keeping related terms in Eq. 1, we get the loss function for $\mathcal{D}_A$:

$$\begin{aligned}
\mathcal{L}_A = \quad &\mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{A}}}\left[\mathbf{D}_\mathrm{A}\left(x\right)\right] - \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{B}}}\left[\mathbf{D}_\mathrm{A}\left(x\right)\right] - \\
&\mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right)\right] - \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{B}\left(z\right)\right)\right] - \\
&\mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)\right] - \\
&\mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{B}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right)\right] + \\
&\mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right)\right].
\end{aligned} \tag{10}$$

Taking the derivative of $\mathcal{L}_A$ w.r.t. $\mathbf{D}_\mathrm{A}$, we have

$$\begin{aligned}
\nabla_{\mathbf{D}_\mathrm{A}} \mathcal{L}_A = \quad &\nabla_{\mathbf{D}_\mathrm{A}} \big[\mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{A}}}\left[\mathbf{D}_\mathrm{A}\left(x\right)\right] - \mathbb{E}_{x\sim\mathbb{P}_{\mathbf{X}_\mathrm{B}}}\left[\mathbf{D}_\mathrm{A}\left(x\right)\right] - \\
&\mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right)\right] - \mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{B}\left(z\right)\right)\right] - \\
&\mathbb{E}_{z\sim p(z)}\left[\mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right)\right] \big].
\end{aligned} \tag{11}$$

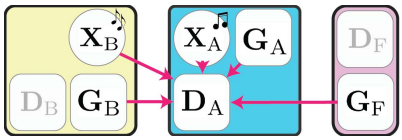

Figure 4: FusionGAN: update $\mathbf{D}_\mathrm{A}$

There are four negative terms and one positive term in Eq. 11 w.r.t. $\mathbf{D}_\mathrm{A}$. The training implicitly increases the positive terms and decreases the value of the negative terms. However, the inequalities in the two groups are not controlled. To further control the distance among the three domains, we propose an inequality constraint for $\mathbf{D}_\mathrm{A}$:

$$\mathbf{D}_\mathrm{A}\left(\mathbf{X}_\mathrm{A}\right) \geq \mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right) \geq \mathbf{D}_\mathrm{A}\left(\mathbf{X}_\mathrm{B}\right) \tag{12}$$

Then combining Eq. 12 into loss function Eq. 11, it can be rewritten as:

$$\mathcal{L}_{A-bal} = \| \mathbf{D}_\mathrm{A}\left(\mathbf{X}_\mathrm{A}\right) - \mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right) \| + \| \mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{F}\left(z\right)\right) - \mathbf{D}_\mathrm{A}\left(\mathbf{X}_\mathrm{B}\right) \|. \tag{13}$$

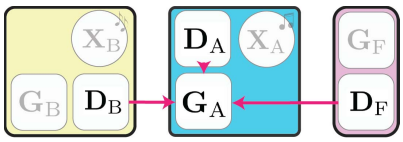

Figure 5: FusionGAN: update $\mathbf{G}_\mathrm{A}$

Some inequalities that have been implicitly included in GAN training are not covered in Eq. 12, such as $\mathbf{D}_\mathrm{A}\left(\mathbf{X}_\mathrm{A}\right) \geq \mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{A}\right)$. Similar to the $\mathbf{G}_\mathrm{F}$ update in Fig. 5, the derivative of $\mathcal{L}_A$ w.r.t. $\mathbf{G}_\mathrm{A}$ is:

$$\begin{aligned} \nabla_{\mathbf{G}_\mathrm{A}} \mathcal{L}_A = \quad & \mathbb{E}_{z\sim p(z)} \nabla_{\mathbf{G}_\mathrm{A}} \big[- \mathbf{D}_\mathrm{A}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right) - \\ & \mathbf{D}_\mathrm{B}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right) + \mathbf{D}_\mathrm{F}\left(\mathbf{G}_\mathrm{A}\left(z\right)\right) \big]. \end{aligned} \tag{14}$$

E. Algorithm Description

The Algorithm 1[1] initializes the parameters of the discriminators and generators (line 1). TextCNN [15] and LSTM are employed to build the discriminators and generators respectively. Given $\mathbf{X}_\mathrm{A}$, the generator is updated by maximum likelihood estimation (line 3). From lines 4 to 5, $\mathbf{D}_\mathrm{A}$ receives the sequences generated by $\mathbf{G}_\mathrm{A}$ as negative data, and real data as positive samples. Between lines 6 and 16, a GAN procedure is applied to optimize $\mathbf{D}_\mathrm{A}$ and $\mathbf{G}_\mathrm{A}$. $\mathbf{D}_\mathrm{A}$ returns regression scores that are treated as rewards for the policy gradient of $\mathbf{G}_\mathrm{A}$. Similarly, $\mathbf{D}_\mathrm{A}$ improves itself by the real world data $\mathbf{X}_\mathrm{A}$ and the data generated by $\mathbf{G}_\mathrm{A}$. Our method proceeds to update $\mathbf{D}_\mathrm{B}$ and $\mathbf{G}_\mathrm{B}$ by using the same process from lines 3 to 16. $\mathcal{D}_F$ repeats the same pre-training procedure from 3 to 16 with $\mathbf{D}_\mathrm{F}, \mathbf{G}_\mathrm{F}, \mathbf{X}_F = \mathbf{X}_\mathrm{A} + \mathbf{X}_\mathrm{B}$. From line 21 to 28, the framework iteratively updates the generators and discriminators from $\mathcal{D}_F, \mathcal{D}_A, \mathcal{D}_B$. Lines 21 to 24 are for $\mathcal{D}_F$, while lines 25 to 28 are for $\mathcal{D}_A$ and $\mathcal{D}_B$. The update rules for the parameters of $\mathbf{D}_\mathrm{F}$ and $\mathbf{G}_\mathrm{F}$ have been covered in subsection (III-C), while those for the parameters of $\mathbf{G}_\mathrm{A}$ and $\mathbf{D}_\mathrm{A}$ are described in subsection (III-D). Again, $\mathcal{D}_B$ follows the same process as that of $\mathcal{D}_A$. The algorithm stops when the discriminators converge.

[1]https://github.com/aquastar/fusion_gan

Algorithm 1: FusionGAN

Input: Input data $\mathbf{X}_\mathrm{A}, \mathbf{X}_\mathrm{B}$ from $\mathcal{D}_A, \mathcal{D}_B$
Output: a generator and a discriminator: $\mathbf{G}_\mathrm{F}$ *and* $\mathbf{D}_\mathrm{F}$

1 Randomly initialize $\mathbf{G}_\mathrm{A}, \mathbf{D}_\mathrm{A}, \mathbf{G}_\mathrm{B}, \mathbf{D}_\mathrm{B}, \mathbf{G}_\mathrm{F}, \mathbf{D}_\mathrm{F}$, generators/discriminators employ LSTM/TextCNN;
2 `// Pre-training`
3 Train $\mathbf{G}_\mathrm{A}$ using maximum likelihood estimation on real data $\mathbf{X}_\mathrm{A}$;
4 Generate negative samples $\mathbf{N}_\mathrm{A}$ using $\mathbf{G}_\mathrm{A}$;
5 Train binary classifier $\mathbf{D}_\mathrm{A}$ with $\mathbf{N}_\mathrm{A}$ and $\mathbf{X}_\mathrm{A}$;
6 **repeat**
7 **for** $\mathbf{G}_\mathrm{A}$ *training* **do**
8 Sample a sequence $\mathcal{S}_{1:T} = (s_1, ..., s_T) \sim \mathbf{G}_\mathrm{A}$;
9 Derive Q value as rewards $\mathbf{R}$ from $\mathbf{D}_\mathrm{A}$;
10 Update parameters θ of $\mathbf{G}_\mathrm{A}$ by policy gradient: $\theta \leftarrow \theta + \alpha \nabla_\theta \mathbf{R}$;
11 **end**
12 **for** $\mathbf{D}_\mathrm{A}$ *training* **do**
13 Generate negative examples $\sim \mathbf{G}_\mathrm{A}$;
14 Train $\mathbf{D}_\mathrm{A}$ with negative samples and $\mathbf{X}_\mathrm{A}$;
15 **end**
16 **until** $\mathbf{D}_\mathrm{A}$ *converges*;
17 Repeat line 3 to 16 with $\mathcal{D}_B$, i.e., $\mathbf{D}_\mathrm{B}, \mathbf{G}_\mathrm{B}, \mathbf{X}_\mathrm{B}$;
18 Repeat line 3 to 5 with $\mathcal{D}_F$, i.e., $\mathbf{D}_\mathrm{F}, \mathbf{G}_\mathrm{F}, \mathbf{X}_F = \mathbf{X}_\mathrm{A} + \mathbf{X}_\mathrm{B}$;
19 `// FusionGAN training`
20 **repeat**
21 **for** $\mathcal{D}_F$ *training* **do**
22 Update parameters of $\mathbf{D}_\mathrm{F}$ using the sum of Eq. 8 and Eq. 9 : $\theta_{\mathbf{D}_\mathrm{F}} \leftarrow \theta_{\mathbf{D}_\mathrm{F}} + \alpha \nabla_{\mathbf{D}_\mathrm{F}} (\mathcal{L}_F + \mathcal{L}_{F-bal})$;
23 Update parameters of $\mathbf{G}_\mathrm{F}$ using Eq. 4 $\theta_{\mathbf{G}_\mathrm{F}} \leftarrow \theta_{\mathbf{G}_\mathrm{F}} + \alpha \nabla_{\mathbf{G}_\mathrm{F}} \mathcal{L}_F$;
24 **end**
25 **for** $\mathcal{D}_A$ *training* **do**
26 Update parameters of $\mathbf{D}_\mathrm{A}$ using the sum of Eq. 11 and Eq. 13 : $\theta_{\mathbf{D}_\mathrm{A}} \leftarrow \theta_{\mathbf{D}_\mathrm{A}} + \alpha \nabla_{\mathbf{D}_\mathrm{A}} (\mathcal{L}_A + \mathcal{L}_{A-bal})$;
27 Update parameters of $\mathbf{G}_\mathrm{A}$ using Eq. 14: $\theta_{\mathbf{G}_\mathrm{A}} \leftarrow \theta_{\mathbf{G}_\mathrm{A}} + \alpha \nabla_{\mathbf{G}_\mathrm{A}} \mathcal{L}_A$;
28 **end**
29 Repeat line 25 to 28 with $\mathcal{D}_B$, i.e., $\mathbf{D}_\mathrm{B}, \mathbf{G}_\mathrm{B}, \mathbf{X}_\mathrm{B}$;
30 **until** $\mathbf{D}_\mathrm{A}, \mathbf{D}_\mathrm{B}, \mathbf{D}_\mathrm{F}$ *converges*;

IV. Evaluation

Training data includes two datasets called the Weimar jazz dataset[2] and the Essen Associative Code (EsAC) folk dataset[3]. Both of these datasets contain symbolic data of single-voiced tracks, which support our objective of generating solo melodies. The baseline methods include:

1) **Random Mixing (RM)**: **RM** randomly selects notes and duration from the pitch and duration distribution

[2]http://jazzomat.hfm-weimar.de
[3]http://www.esac-data.org

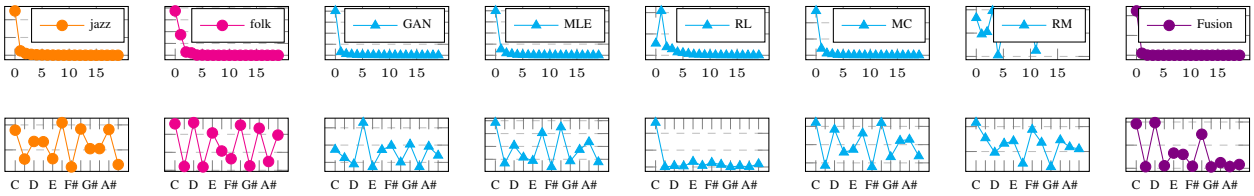

Figure 6: upper line: **DD** (**x-axis**: duration length;**y-axis**: percentage); lower line: **NPD** (**x-axis**: pitch class; **y-axis**: percentage)

on the combination of $\mathbf{X}_\mathrm{A}$ and $\mathbf{X}_\mathrm{B}$.

2) **Monte Carlo Sampling (MC)**: After calculating the note pitch and duration distributions on the combination of $\mathbf{X}_\mathrm{A}$ and $\mathbf{X}_\mathrm{B}$, **MC** samples notes and duration subject to these distributions.
3) **Maximum Likelihood Estimation (MLE)**: Employing LSTM, **MLE** directly learns a pattern from the mixture of $\mathbf{X}_\mathrm{A}$ and $\mathbf{X}_\mathrm{B}$ by predicting the next element.
4) **GAN**: The model directly learns a pattern from the mixture of $\mathbf{X}_\mathrm{A}$ and $\mathbf{X}_\mathrm{B}$ using GAN.
5) **Reinforcement Learning (RL)**: After applying GAN on $\mathbf{D}_\mathrm{A} - \mathbf{G}_\mathrm{A}, \mathbf{D}_\mathrm{B} - \mathbf{G}_\mathrm{B}$, we exchange the discriminators $\mathbf{D}_\mathrm{A}$ and $\mathbf{D}_\mathrm{B}$ which are treated as rewards function to $\mathbf{G}_\mathrm{B}$ and $\mathbf{G}_\mathrm{A}$ respectively.

A. Quantitative Study

Due to the aesthetic nature of the task, a quantitative evaluation can only give a first impression of the power of the generative system. Here, we investigate the distance between pitch and note duration distributions, knowingly discarding all sequential information of the melodies.

The ideal output should keep equally close to the given domains and minimize this distance. Two distributions of properties are selected for the study.

1) **Duration Distribution (DD)**: Notes are categorized by their durations.
2) **Normalized Pitch Distribution (NPD)**: All tokens are categorized into C/C#/D/D#/E/F/F#/G/G#/A/A#/B.

To quantify the similarity between distributions, two symmetric metrics, the Euclidean distance (**EUD**) and the Wasserstein-1 metric (Earth Mover distance (**EM**)), are employed to calculate the distance between the distributions of generated sequences and those of domains $\mathbf{X}_\mathrm{A}$ and $\mathbf{X}_\mathrm{B}$. Intuitively, the distance between $\mathbf{X}_\mathrm{A}$ and $\mathbf{X}_\mathrm{B}$, between $\mathbf{X}_\mathrm{A}, \mathcal{D}_F$, and between $\mathbf{X}_\mathrm{B}, \mathcal{D}_F$ should satisfy the triangle inequality, and the optimal configuration should minimize the inequality, i.e.,

$$\min \quad Diff = \| \mathbf{X}_\mathrm{A} - \mathbf{G}_\mathrm{F}(z) \| + \| \mathbf{X}_\mathrm{B} - \mathbf{G}_\mathrm{F}(z) \| - \| \mathbf{X}_\mathrm{A} - \mathbf{X}_\mathrm{B} \|,$$

$$Ratio = \frac{\| \mathbf{X}_\mathrm{A} - \mathbf{G}_\mathrm{F}(z) \| + \| \mathbf{X}_\mathrm{B} - \mathbf{G}_\mathrm{F}(z) \|}{\| \mathbf{X}_\mathrm{A} - \mathbf{X}_\mathrm{B} \|}.$$

Figure 6 (top) shows the note duration distributions (DD) and Table I (top) gives the corresponding distance measures. The note duration distribution of jazz and folk have shapes that resemble the power law distribution. The shape of **RM** is very different from jazz or folk. Visually, **RL**'s duration distribution does not match that of jazz and folk because the portion of notes at the length of 0 and 1 (the peak) were relatively higher than that of jazz or folk. The other baselines (**GAN**, **MLE**, **MC**) appear similar to the given domains because they conform to the power law distributions. **Fusion** is roughly at the same level of **MC**. Note that, given our evaluation metrics, **MC** has the best performance in this experiment since it directly samples notes from the true distribution. This disregard of sequential information, however, results in unnatural rhythmic qualities of the music clips generated by **MC**.

Table I: Distance between Distributions (the lower the better)

	EUD		**EM**	
	Diff	Ratio	Diff	Ratio
		DD		
RM	39742.2	1.375	2757.6	1.461
MLE	24546.4	1.231	2005.2	1.335
GAN	24765.2	1.233	2064.3	1.345
RL	37971.2	1.358	2629.0	1.439
MC	13988.7	1.132	1289.2	1.215
Fusion	19452.6	1.183	1831.9	1.306
		NPD		
RM	19586.6	1.647	5231.0	1.643
MLE	15921.4	1.526	4147.5	1.510
GAN	16807.1	1.555	4098.0	1.504
RL	16927.1	1.559	4399.2	1.541
MC	11175.0	1.369	2660.2	1.327
Fusion	11564.6	1.382	3182.5	1.391

Figure 6 (bottom) shows the pitch distributions (NPD) and Table I (bottom) gives the corresponding distance measures. **RL**'s pitch distribution is —due to overfitting— different from that of jazz and folk, which shows its low capacity in modeling pitch distribution. The single peak at D# as generated by **GAN** does not match either the jazz or folk distributions. The distributions of the other baselines are difficult to categorize as a good or bad match. Since **MC** is directly sampled from real distributions, its configurations should be exactly the same as that of the optimal fusion. **MC** shows two valleys at C# and F#, and two peaks at C and G, which should be expected. **RM** satisfies one of them (F# is a valley), while **RM** meets only one standard (C is the peak). **GAN** fails in all the characteristic pitch classes, and both **MLE** and **Fusion** meet all of them. Quantitatively, the ratio of **RM** in **NPD** is the highest which means that **RM** is the worst in modeling pitch. **GAN** and **RL** improve **RM** by around 10%. As one of the two distributions that look most similar to the original distributions, **MLE** outperforms **RM**, **RL**, and **GAN**. Our method **Fusion** has the second lowest ratio after **MC**. However, as mentioned above, the **MC** does not consider the sequential consistence, resulting in bad evaluations in the listening test described in Sect. IV-B.

B. User Study via Listening Test

To evaluate the subjective quality of the generated sequences, a user study was conducted via Amazon Mechanical Turk. The listening test was divided into three steps:

1) **Qualification test**: *"Listen to the music, and choose one genre below that most matches the music"*. This is to test the evaluators' qualification level. The provided choices are *(A) jazz, (B) folk, (C) neither*.
2) **Fusion recognition**: *"Listen to the attached music, and then choose the choice that most matches the music"* Possible answers are *(A) pure jazz, (B) pure folk, (C) mixture of jazz and folk, (D) neither*. Each of the pieces of music generated by our proposed method and the baselines are assigned to one such question.
3) **Musicality**: *"Listen to the attached music in 2nd step, who do you think its composer"*, the allowed answers are *(A) expert, (B) newbie, (C) robot*. Similarly, each generated sample is associated with one such question.

We randomly drew from the training data and prepared 800 sets for the first question. In the first step, 66.5% of candidates chose the correct answer, which is an acceptable rate given the inherent difficulties of genre identification for average listeners. The statistics resulting from the valid answers are shown in Table II. Each of the methods generated 500 samples, which were randomly selected for the second and third questions.

As shown in Table II, the best system based on the percentage of *mixture* is **MLE**. A closer look at **MLE**, however, reveals the unbalanced distribution between *jazz* and *folk*, which implies its bias towards the jazz genre. On the other hand, **RM** is the best system based on the balance between *jazz* and *folk*, but the high percentage of *neither* suggests the confusion of the listeners. It is clear that a single criterion is insufficient for determining the best system. Therefore, we design a metric to summarize the results and represent the fusion level (**FL**) of the evaluated systems with the following equation:

$$\text{maximize} \quad \mathbf{FL} = 1 - \frac{\|C_{jazz} - C_{folk}\| + C_{neither}}{C_{jazz} + C_{folk} + C_{mixture} + C_{neither}},$$

where C_i indicates the count of i. $\|C_{jazz} - C_{folk}\|$ means the unbalanced error, and $C_{neither}$ can be treated as another type of error. A higher **FL** value implies better fusion.
In the *Musicality* test, **MLE** (45.5%), **GAN** (42.0%), and **Fusion** (43.8%) were voted by the majority as *expert*. Overall, **Fusion** shows a balanced performance in both *Fusion recognition* and *Musicality* tests, which demonstrates the effectiveness of our proposed method.

Table II: Listening test results

Fusion recognition					
	jazz	*folk*	*mixture*	*neither*	**FL**
RM	25.0%	22.5%	12.5%	40%	57.5%
MLE	43.6%	9.1%	30.9%	16.4%	49.1%
GAN	34.0%	17.0%	26.0%	14%	69%
RL	20.1%	28.3%	20.8%	30.8%	61%
MC	32.0%	2.0%	14.0%	52%	16%
Fusion	35.9%	25.0%	20.0%	19.1%	70%
Musicality					
	expert		*newbie*		*robot*
RM	22.5%		50.0%		27.5%
MLE	45.5%		21.8%		32.7%
GAN	42.0%		32.0%		26.0%
RL	32.1%		37.7%		30.2%
MC	30.0%		36.0%		34.0%
Fusion	43.8%		28.1%		28.1%

V. Conclusion

In this paper, we proposed a three-way GAN-based learning framework to integrate multiple domains. A Wasserstein distance based metric is introduced to indicate the blending progress. The evaluation investigated objective metrics looking at the pitch and note duration distributions of the generated data. A user study explicitly illustrates the validity of the proposed method as compared to the baselines, as the melodies generated by our model were preferred by the majority of users. While the listening test results are encouraging, future work will benefit from removing key dependence of the training data, a careful look into perceptually meaningful evaluation metrics that take into account the sequential nature of music, and a careful listening test design, including human-composed melodies for comparison.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv:1701.07875*, 2017.
[2] S. Colton, R. L. de Mántaras, and O. Stock, "Computational creativity: Coming of age," *AI Magazine*, vol. 30, no. 3, p. 11, 2009.
[3] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural audio synthesis of musical notes with wavenet autoencoders," *arXiv:1704.01279*, 2017.
[4] L. A. Gatys, A. S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *arXiv preprint*, pp. 3–7, 2015.
[5] J. Gerald David, *Encyclopedia of African American Society*. London: SAGE Publications, Inc., 2005.
[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.
[7] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
[8] T. Kim, M. Cha, H. Kim, J. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," *arXiv:1703.05192*, 2017.
[9] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *NIPS*, 2016, pp. 469–477.
[10] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv preprint arXiv:1411.1784*, 2014.
[11] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv:1609.03499*, 2016.
[12] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *German Conference on Pattern Recognition*. Springer, 2016, pp. 26–36.
[13] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008, vol. 338.
[14] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," *arXiv preprint arXiv:1704.02510*, 2017.
[15] X. Zhang and Y. LeCun, "Text understanding from scratch," *arXiv preprint arXiv:1502.01710*, 2015.
[16] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv:1703.10593*, 2017.