Music Generation with Wasserstein Autoencoders

Giovanni Bindi

Università degli Studi di Firenze

September 8th, 2020

Introduction

Aim

Being able to generate music from scratch.

Methodology

Training deep Wasserstein Autoencoders (WAEs).

Data

Pianoroll samples derived from pop/rock songs.

Generative Models

- We have a set of *i.i.d.* data points $\{x^{(1)}, \dots, x^{(N)}\} \subseteq \mathcal{X}$.
- We do not know the true data distribution $P_X(X)$.
- We would like to find a distribution $P_G(X)$ over $\mathcal{X}$ that is *similar* to $P_X(X)$.

Latent Variable Model (LVM)

1. Introduce a latent space $\mathcal{Z}$.
2. Sample $P_Z(Z)$, a *prior* distribution on the latent space.
3. Map the sample to $\mathcal{X}$ through $P_G(X|Z)$, i.e. the *decoder*.

$$p_G(x) = \int_{\mathcal{Z}} p_G(x|z) p_z(z) dz \tag{1}$$

Variational Autoencoders (VAEs): Kingma & Welling, 2014 - [1]

Minimize the **KL-divergence** between P_X and P_G:

$$\inf_{P_G} \mathrm{KL}(P_X \| P_G) \iff \inf_{P_G} \mathbb{E}_{P_X}[-\log P_G] \tag{2}$$

Variational bound: for any conditional distribution $Q(Z|X)$ the following inequality holds

$$-\mathbb{E}_{P_X}[\log P_G(X)] \leq \mathbb{E}_{P_X}\left[\mathbb{E}_{Q(Z|X)}[-\log P_G(X|Z)] + \mathrm{KL}(Q(Z|X)\|P_Z(Z))\right] \tag{3}$$

- Decoders for which $\log P_G(X|Z)$ can be evaluated and differentiated.
- Often $P_Z(Z) = \mathcal{N}(Z; 0, I)$ and $Q(Z|X) = \mathcal{N}(Z; \mu(X), \Sigma(X))$.
- Everything is parametrized with Deep Neural Networks.
- The bound is minimized via Stochastic Gradient Descent.

Optimal Transport (OT)

Given a cost function $c : \mathbb{X} \times \mathbb{X} \to \mathbb{R}^+$ the OT problem is

$$W_c(P_X, P_G) = \inf_{\Gamma \in \mathcal{P}(X \sim P_X, Y \sim P_G)} \mathbb{E}_\Gamma[c(X, Y)] \tag{4}$$

For deterministic decoders, i.e. $P_G(X|Z) = \delta_{G(Z)}$, for any $G : \mathcal{Z} \to \mathcal{X}$ then

$$W_c(P_X, P_G) = \inf_{Q : Q_z = P_z} \mathbb{E}_{P_X}\left[\mathbb{E}_{Q(Z|X)}[c(X, G(Z))]\right] \tag{5}$$

where $Q_z = \mathbb{E}_{P_X}[Q(Z|X)]$ is the *aggregated posterior*.

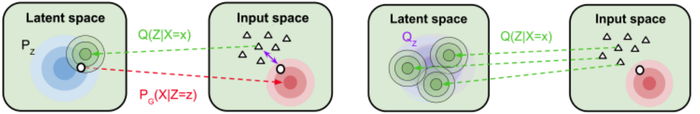


Figure 1: VAE approach (left) vs WAE approach (right)

Wasserstein Autoencoders (WAEs): Tolstikhin & al., 2018 - [2]

Minimizing the optimal transport:

$$\inf_{P_G} W_c(P_X, P_G) \iff \inf_{P_G} \inf_{Q: Q_z = P_z} \mathbb{E}_{P_X}\left[\mathbb{E}_{Q(Z|X)}[c(X, G(Z))]\right] \quad (6)$$

Relaxation:

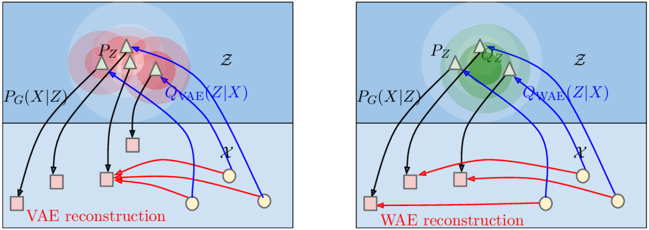
$$\inf_{P_G} \inf_{Q(Z|X) \in \mathcal{Q}} \mathbb{E}_{P_X}\left[\mathbb{E}_{Q(Z|X)}[c(X, G(Z))]\right] + \lambda \cdot \mathcal{D}_z(Q_z, P_z) \quad (7)$$

for any non-parametric set of encoders $\mathcal{Q}$ and divergence $\mathcal{D}_z$.

- Hyperparameter $\lambda \in \mathbb{R}^+$.
- Parametrize G (*decoder*) and Q (*encoder*) with DNNs.
- Run SGD on (7).

VAEs vs WAEs [2]

(a) VAE (b) WAE



VAE:

$$\inf_{P_G} \inf_{Q \in \mathcal{Q}} \mathbb{E}_{P_X} \left[\mathbb{E}_{Q(Z|X)} [- \log P_G(X|Z)] \right] + \mathbb{E}_{P_X} [\mathrm{KL}(Q(Z|X) \| P_Z(Z))]$$

WAE:

$$\inf_{P_G} \inf_{Q \in \mathcal{Q}} \mathbb{E}_{P_X} \left[\mathbb{E}_{Q(Z|X)} [c(X, G(Z))] \right] + \lambda \cdot \mathcal{D}_z(\mathbb{E}_{P_X}[Q(Z|X)], P_Z(Z))$$

WAE-GAN: adversarial penalty [2]

$$\mathcal{D}_z(Q_z, P_z) = D_{JS}(Q_z, P_z) = \frac{1}{2}\mathrm{KL}(P_Z \| M) + \frac{1}{2}\mathrm{KL}(Q_Z \| M)$$

Where $M = \frac{1}{2}(P_Z + Q_Z)$. A discriminator D is introduced in the latent space.

- Using the adversarial training:

$$D_{JS}(Q_Z, P_Z) \approx \sup_D \mathbb{E}_{Z \sim P_Z}[\log D(Z)] + \mathbb{E}_{Z^* \sim Q_Z}[\log(1 - D(Z^*))]$$

- **Con**: Leads to the $\min - \max$ problem.
- Pro: Should be easier than typical GANs (usually $\mathcal{Z}$ is low dimensional and P_Z uni-modal).

WAE-MMD: non-adversarial penalty [2] and [3]

$$\mathcal{D}_z(Q_Z, P_Z) = MMD_k(Q_Z, P_Z) = MMD_k(P_Z, Q_Z)$$

With

$$MMD_k(P_z, Q_z) = \left\| \int_{\mathcal{Z}} k(z, \cdot) dP_Z(z) - \int_{\mathcal{Z}} k(z, \cdot) dQ_Z(z) \right\|_{\mathcal{H}_k} \quad (8)$$

Where $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is any p.d. reproducing kernel and $\mathcal{H}_k$ its RKHS. Given $\{z_i\}_{i=1}^N \sim P_Z$ and $\{z_i^*\}_{i=1}^N \sim Q_Z$, if k is characteristic then an unbiased estimate of the MMD is:

$$\frac{1}{N(N-1)} \left[\sum_{i=1}^{N} \sum_{j \neq i} k(z_i, z_j) + \sum_{i=1}^{N} \sum_{j \neq i} k(z_i^*, z_j^*) \right] - \frac{2}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} k(z_i, z_j^*) \quad (9)$$

Pianoroll

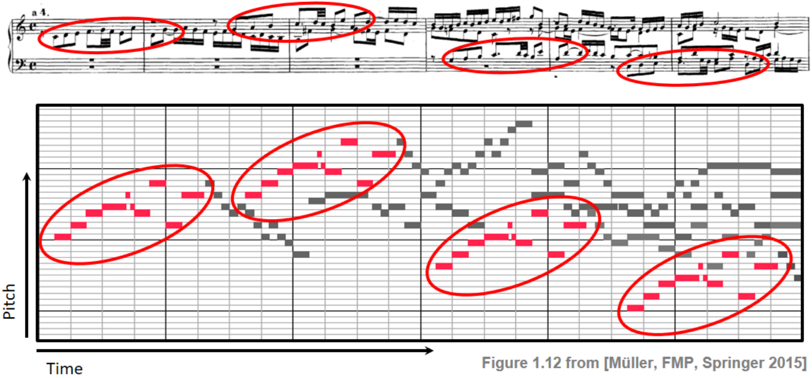


Figure 2: Six bars from Fugue BWV 846 in C major by J.S.Bach

Data Description: Structure - Dong & al., 2017 - [4]

- **Dataset**: Lakh Pianoroll Dataset (LPD)-5, *cleansed* version.
  - 21425 elements.
  - Every element is a phrase of 4 bars for 5 instruments.
  - Instruments are: *Drums*, *Guitar*, *Bass*, *Piano* and *Strings*.
  - Every element has a time signature of 4/4.
  - Quantization is set to 1/48th.
  - Spans from C1 to B7 (84 notes).
  - Majors to CMaj and minors to AMin.
  - Binary valued (*note-on* information).
- **Pro**: Simple, image-like, representation.
- **Con**: Partial information.

Data Description: Visualization

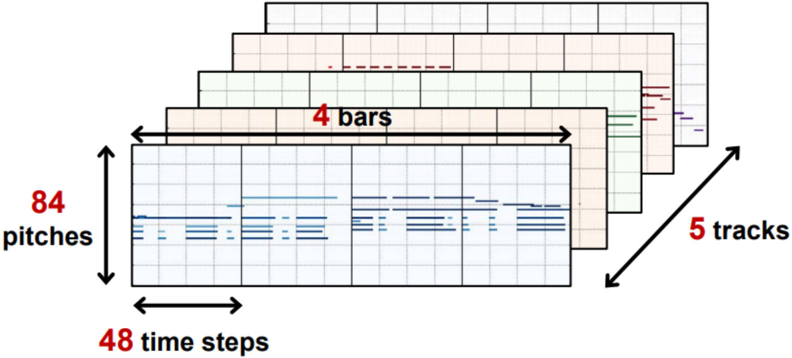


Figure 3: Visualization of a data point (image from dataset authors)

This results in a dataset whose elements live in $\mathbb{D} = \{0, 1\}^{84\times 192\times 5}$

Experiment details

Choices:

1. Euclidean latent space: $\mathcal{Z} = \mathbb{R}^{d_z}$.
2. Gaussian prior: $P_Z(Z) = \mathcal{N}(Z; 0, \sigma_z^2 \cdot I_{d_z})$.
3. Squared L2 norm as cost function: $c(x, y) = \|x - y\|_2^2$.

Implementation:

- Encoder Q_ϕ and Decoder G_θ as deep CNNs in a DCGAN-style (*Conv* / *Deconv* $\rightarrow$ *Batch Norm* $\rightarrow$ *ReLU*).
- Discriminator D_γ (only in the WAE-GAN algorithm) as a deep NN (*Dense* $\rightarrow$ *ReLU*).
- Adam optimizer.

Training Phase

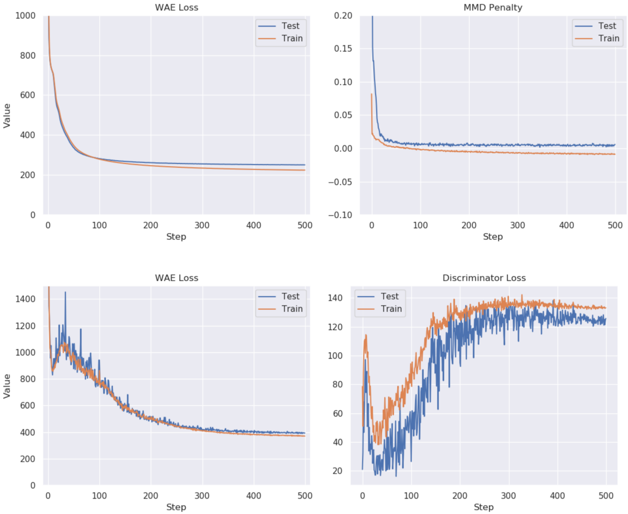


Figure 4: WAE-MMD (top) WAE-GAN (bottom)

- *Latent space dimension*: $d_z = 128$
- *Weight*: $\lambda = 100$
- *Convolutional filter size*: 6×6
- *Epochs*: 500
- *Number of filters*: $[8, 16, 32, 64]$
- *Learning rates*: $10^{-4} \rightarrow 10^{-5}$
- *Samples*: 8192 (train), 2048 (test)

Metrics

Lack of an objective measure of performance

- Reconstruction error: Mean Squared Error (**MSE**).

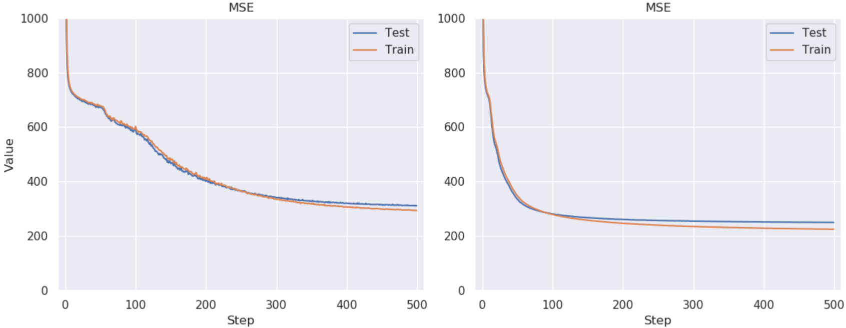
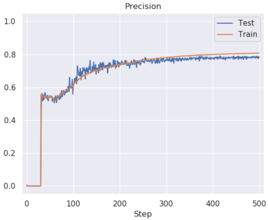
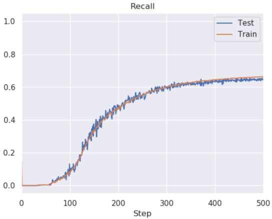


Figure 5: MSE: WAE-GAN (left) vs WAE-MMD (right)

Lack of an objective measure of performance

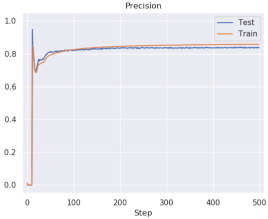
- Precision (**P**) and Recall (**R**) - Top: WAE-GAN, bottom: WAE-MMD.

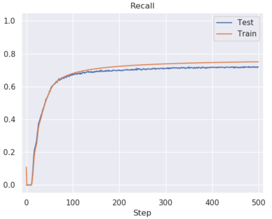




ALG	P	R
WAE-GAN	0.83	0.7
WAE-MMD	0.87	0.78

Table 1: Train Set





ALG	P	R
WAE-GAN	0.78	0.63
WAE-MMD	0.83	0.70

Table 2: Test Set

Metrics [4]

Data[1]	**EBR**	**QNR**	**POLY**
Test	0.05	0.88	0.51
Reconstruction - MMD	0.15	0.80	0.50
Reconstruction - GAN	0.14	0.76	0.49
Random Uniform on $\{0, 1\}$	0.0	0.24	1.0
Random Samples - MMD	0.13	0.65	0.29
Random Samples - GAN	0.16	0.64	0.38

Where these metrics are

$$\begin{cases} \text{Empty Bars Rate } (\textbf{EBR}) = \frac{\#\ [\text{empty bars}]}{\#\ [\text{bars}]} \\ \text{Qualified Notes Rate } (\textbf{QNR}) = \frac{\#\ [\text{notes no shorter than } 1/16th]}{\#\ [\text{notes}]} \\ \text{Polyphonicity } (\textbf{POLY}) = \frac{\#\ [\text{time steps where} \geq 2 \text{ pitches}]}{\#\ [\text{time steps}]} \end{cases}$$

[1] Every set in this column is composed of 2048 samples.

Conclusions

- WAEs are a promising learning framework.
  - Flexible and theoretically grounded.
  - Stable training.
- The lack of a “musical performance” metric is an obstacle
  - Qualitative analysis.
  - Hyperparameter optimization is difficult.

Listen to the results

`https://w00zie.github.io/post/wae`

Look at the code

`https://github.com/w00zie/wae_music`


Thank you for your
attention!

References I

[1] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, 2013. arXiv: 1312.6114 [stat.ML].

[2] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf, *Wasserstein auto-encoders*, 2017. arXiv: 1711.01558 [stat.ML].

[3] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf, "Kernel mean embedding of distributions: A review and beyond", *Foundations and Trends® in Machine Learning*, 2017. [Online]. Available: http://dx.doi.org/10.1561/2200000060.

References II

[4] 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*, 2017. arXiv: 1709.06298 [eess.AS].

Deriving the ELBO in 1

$$\log p_\theta(x) = \int_{\mathbb{Z}} \log p_\theta(x) q_\phi(z|x) dz = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x)] = \tag{10}$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{p_\theta(z|x)}\right] = \tag{11}$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log \left(\frac{p_\theta(x,z)}{q_\phi(z|x)} \frac{q_\phi(z|x)}{p_\theta(z|x)}\right)\right] = \tag{12}$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right] = \tag{13}$$

$$= ELBO(\theta, \phi) + KL(q_\phi(z|x) \| p_\theta(z|x)) \tag{14}$$

Then

$$ELBO(\theta, \phi) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z) p_\theta(z)}{q_\phi(z|x)}\right] = \tag{15}$$

$$= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(z)}\right] = \tag{16}$$

$$= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x) \| p_\theta(z)) \tag{17}$$

WAE-GAN

Init $G_\theta, Q_\phi, D_\gamma$
while (θ, ϕ, γ) not converged **do**
Sample $\{x^{(1)}, \dots, x^{(n)}\}$ from D.
Sample $\{z^{(1)}, \dots, z^{(n)}\}$ from $\mathcal{Z}$.
Sample $\tilde{z}^{(i)}$ from $Q(Z|X = x^{(i)})$ for $i = 1, \dots, n$.
Update D_γ by ascending:

$$\frac{\lambda}{n} \sum_{i=1}^{n} \log D(z^{(i)}) + \log(1 - D(\tilde{z}^{(i)})$$

Update (G_θ, Q_ϕ) by descending:

$$\frac{1}{n} \sum_{i=1}^{n} c(x^{(i)}, G_\theta(\tilde{z}^{(i)})) - \lambda \cdot \log D(\tilde{z}^{(i)})$$

end while

WAE-MMD

Init G_θ, Q_ϕ
while (θ, ϕ) not converged **do**
Sample $\{x^{(1)}, \dots, x^{(n)}\}$ from D.
Sample $\{z^{(1)}, \dots, z^{(n)}\}$ from $\mathcal{Z}$.
Sample $\tilde{z}^{(i)}$ from $Q(Z|X = x^{(i)})$ for $i = 1, \dots, n$.
Update (G_θ, Q_ϕ) by descending:

$$\frac{1}{n} \sum_{i=1}^{n} c(x^{(i)}, G_\theta(\tilde{z}^{(i)}))+$$

$$+ \frac{\lambda}{n(n-1)} \left[\sum_{i=1}^{n} \sum_{j \neq i} k(z^{(i)}, z^{(j)})+ \right.$$

$$\left. + \sum_{i=1}^{n} \sum_{j \neq i} k(\tilde{z}^{(i)}, \tilde{z}^{(j)}) \right] -$$

$$- \frac{2}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(z^{(i)}, \tilde{z}^{(j)})$$

end while

Kernels

Def (Positive definite kernel): Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel if it is symmetric and the Gram matrix is positive definite:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$$

for any $n \in \mathbb{N}$, any $x_1, \ldots, x_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$. A p.d. kernel defines a space of functions from $\mathcal{X}$ to $\mathbb{R}$ called a Reproducing Kernel Hilbert Space (**RKHS**) $\mathcal{H}_k$, hence is also called a reproducing kernel. In particular the reproducing property says that for each $x \in \mathcal{X}$ there exists a function $k_x = k(x, \cdot) \in \mathcal{H}_k$ such that

$$\langle k_x, f \rangle_{\mathcal{H}_k} = \langle k(x, \cdot), f \rangle = f(x), \qquad \forall f \in \mathcal{H}_k$$

One may alternatively consider $k(x, \cdot)$ an implicit feature mapping $\varphi(x)$ from $\mathcal{X}$ to $\mathcal{H}_k$ (which is therefore also called the feature space), so that $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}_k}$ can be viewed as a measure of similarity between points $x, x' \in \mathcal{X}$.
Def (characteristic kernel): A kernel k is said to be characteristic if, named $\mathbb{P}$ the space of all distributions over $\mathcal{X}$, the mapping $\mu : P \to \mu_P$ is injective, for all $P \in \mathbb{P}$. The quantity μ_P is defined as

$$\mu_P = \mathbb{E}_P[k(X, \cdot)] = \mathbb{E}_P[\varphi(X)] = \int_{\mathcal{X}} \varphi(x) dP(x)$$

Kernel Mean Embeddings: MMD

Def (MMD): The maximum mean discrepancy is a distance-measure between distributions $P(X)$ and $Q(Y)$ (taking values on $\mathcal{X}$) which is defined as the squared distance between their embeddings in the RKHS $\mathcal{H}_k$:

$$
\begin{aligned}
MMD[\mathcal{H}_k, P, Q] &= \sup_{\|f\|_{\mathcal{H}_k} \leq 1} \left\{ \int_{\mathcal{X}} f(x) dP(x) - \int_{\mathcal{X}} f(y) dQ(y) \right\} = \\
&= \sup_{\|f\|_{\mathcal{H}_k} \leq 1} \left\{ \langle f, \int_{\mathcal{X}} k(x, \cdot) dP(x) \rangle - \langle f, \int_{\mathcal{X}} k(y, \cdot) dQ(y) \rangle \right\} = \\
&= \sup_{\|f\|_{\mathcal{H}_k} \leq 1} \{ \langle f, \mu_P - \mu_Q \rangle \} = \\
&= \|\mu_P - \mu_Q\|_{\mathcal{H}_k}
\end{aligned}
$$

Def (U-Statistic): Let $X_1, X_2, \ldots, X_m$ be be independent observations on a distribution $P(X)$. Let $\theta(P)$ be the functional defined by

$$
\theta(P) = \mathbb{E}_P[h(X_1, \ldots, X_m) = \int \cdots \int h(x_1, \ldots, x_m) dP(x_1) \ldots dP(x_m)
$$

for some real-valued measurable function h called a kernel. The corresponding U-statistic of order m for estimation of θ on the basis of a sample $x_1, \ldots, x_n$ of size $n \geq m$ is obtained by averaging the kernel h symmetrically over the observations:

$$
U_n = U(x_1, \ldots, x_m) = \frac{1}{\binom{n}{m}} \sum_c h(x_{i_1}, \ldots, x_{i_n})
$$

where $\sum_c$ denotes the sum over the $\binom{n}{m}$ combinations of m distinct elements $\{i_1, \ldots, i_m\}$ from $\{1, \ldots, n\}$.

Kernel used

The kernel used is the sum of inverse multiquadratics kernels at various scales:

$$k(x, y) = \sum_{s \in S} \frac{2sd_z\sigma_z^2}{2sd_z\sigma_z^2 + \|x - y\|_2^2}$$

where $S = \{0.1, 0.2, 0.5, 1, 2, 5, 10\}$.

#	Layer	Input	Ouput	Params
1	`Conv2D(32)`	$(B, 192, 84, 5)$	$(B, 96, 84, 32)$	5760
2	`BatchNorm`	$(B, 96, 84, 32)$	$(B, 96, 84, 32)$	128
3	`Conv2D(64)`	$(B, 96, 84, 32)$	$(B, 48, 42, 64)$	73728
4	`BatchNorm`	$(B, 48, 42, 64)$	$(B, 48, 42, 64)$	256
5	`Conv2D(128)`	$(B, 48, 42, 64)$	$(B, 24, 21, 128)$	294912
6	`BatchNorm`	$(B, 24, 21, 128)$	$(B, 24, 21, 128)$	512
7	`Conv2D(256)`	$(B, 24, 21, 128)$	$(B, 8, 7, 256)$	1179648
8	`BatchNorm`	$(B, 8, 7, 256)$	$(B, 8, 7, 256)$	1024
9	`Flatten`	$(B, 8, 7, 256)$	$(B, 14336)$	0
10	`Dropout(0.2)`	$(B, 14336)$	$(B, 14336)$	0
11	`Dense(128)`	$(B, 14336)$	$(B, 128)$	1835136

Table 3: Encoder

Divergence Plots

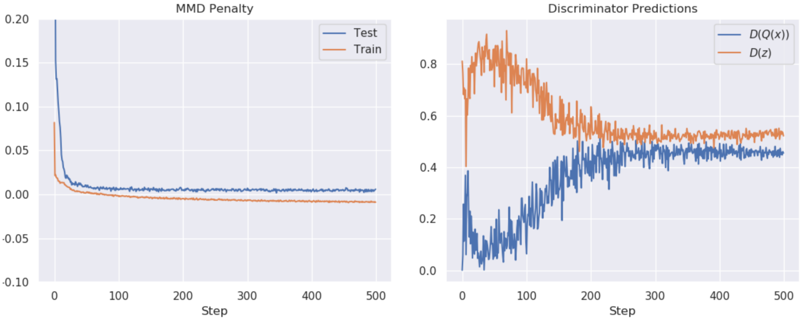


Figure 6: Plots for MMD (left) and discriminator values (right). Plot on the right is made on the test set only

Precision and Recall

Let

$$\hat{x}^{(i)} = G_\theta(Q_\phi(x^{(i)}))$$

where $x^{(i)} \in D$. Let the true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) defined as

$$\begin{cases} TP = \#[x_{jkl}^{(i)} = \hat{x}_{jkl}^{(i)} = 1] \\ TN = \#[x_{jkl}^{(i)} = \hat{x}_{jkl}^{(i)} = 0] \end{cases} \qquad \begin{cases} FP = \#[x_{jkl}^{(i)} = 0 \wedge \hat{x}_{jkl}^{(i)} = 1] \\ FN = \#[x_{jkl}^{(i)} = 1 \wedge \hat{x}_{jkl}^{(i)} = 0] \end{cases}$$

Then precision (**P**) and recall (**R**):

$$\begin{cases} \mathbf{P} & = \frac{TP}{TP+FP} \\ \mathbf{R} & = \frac{TP}{TP+FN} \end{cases}$$

Metrics from test set and relative reconstructions

Instrument	**EBR**	**QNR**	**POLY**	**ISR**
Drums	0.0471	-	-	-
Piano	0.0498	0.9076	0.6945	0.7144
Guitar	0.0426	0.8588	0.5822	0.7081
Bass	0.0173	0.8986	0.0229	0.4666
String	0.0753	0.8665	0.7453	0.7068

Table 4: Test set

Instrument	**EBR**	**QNR**	**POLY**	**ISR**
Drums	0.1831	-	-	-
Piano	0.1324	0.7305	0.6742	0.7395
Guitar	0.1870	0.6633	0.5653	0.7229
Bass	0.0610	0.7885	0.0138	0.4408
String	0.1172	0.8574	0.7195	0.7192

Table 5: Test set reconstructions with WAE-GAN

Instrument	**EBR**	**QNR**	**POLY**	**ISR**
Drums	0.3467	-	-	-
Piano	0.1031	0.7677	0.6747	0.7255
Guitar	0.1488	0.6966	0.5644	0.7055
Bass	0.0433	0.8269	0.0164	0.4565
String	0.1075	0.8951	0.7272	0.7117

Table 6: Test set reconstructions with WAE-MMD

Metrics from decoded samples

Instrument	**EBR**	**QNR**	**POLY**	**ISR**
Drums	0.1812	-	-	-
Piano	0.1187	0.6375	0.5139	0.7285
Guitar	0.2067	0.5759	0.4406	0.7047
Bass	0.1940	0.5949	0.0151	0.4507
String	0.0951	0.7384	0.5683	0.7177

Table 7: Decoded with WAE-GAN

Instrument	**EBR**	**QNR**	**POLY**	**ISR**
Drums	0.2841	-	-	-
Piano	0.0720	0.6316	0.3865	0.7413
Guitar	0.1128	0.5727	0.2920	0.7104
Bass	0.1000	0.6268	0.0234	0.4205
String	0.0643	0.7557	0.4601	0.7522

Table 8: Decoded with WAE-MMD