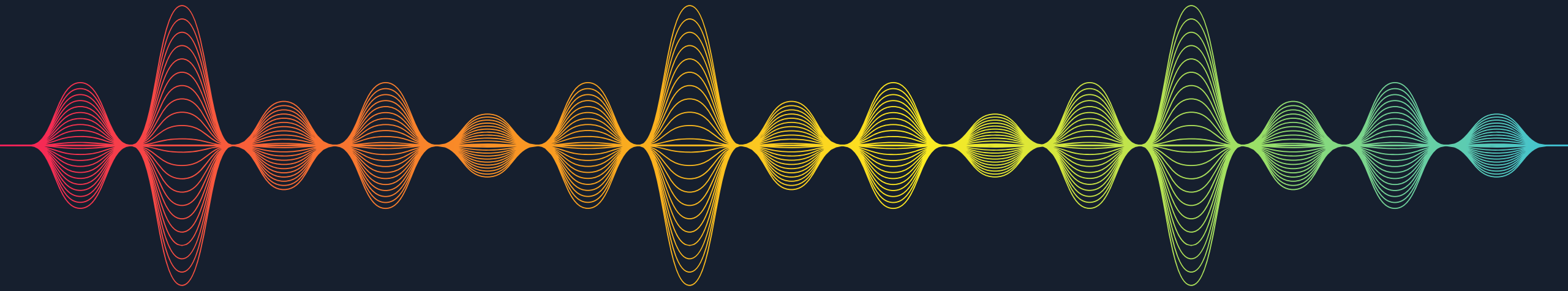
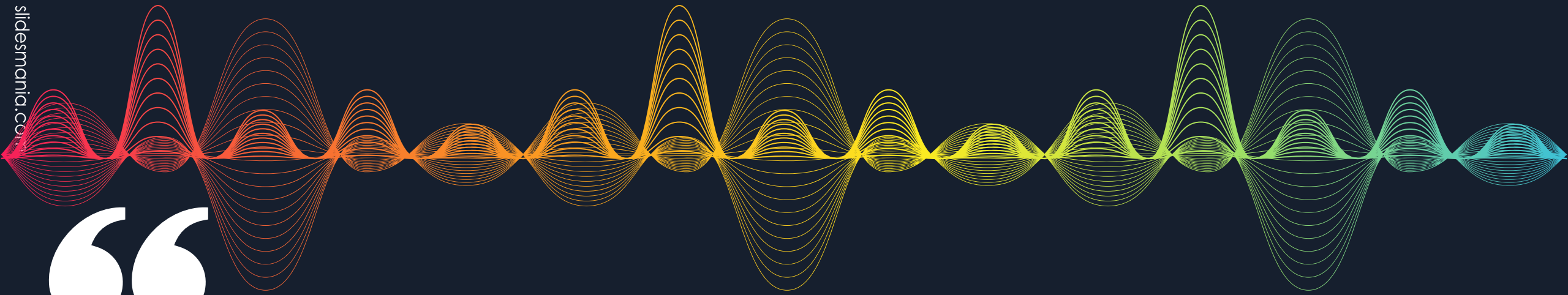
# Musical Style Transfer in Sparse Audio Representation Domains

Flor Sanders

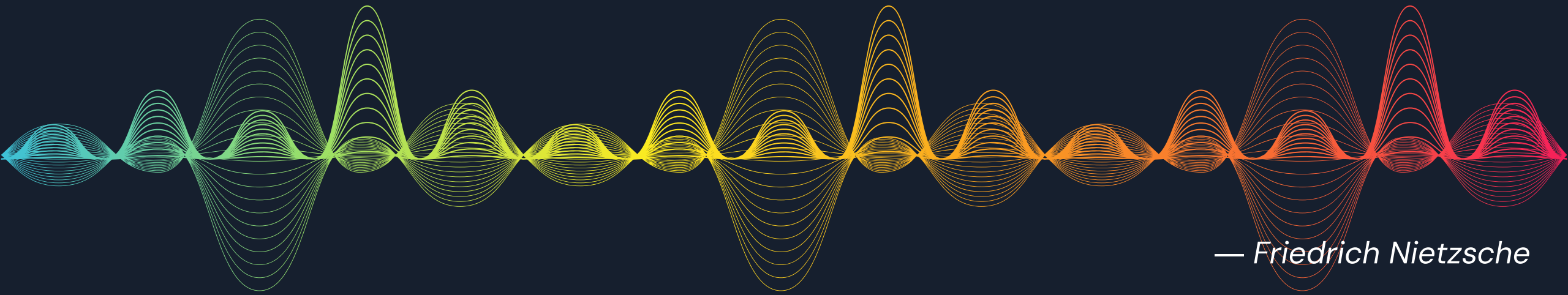
*Sparse and Low-Dimensional Models for High-Dimensional Data*





“

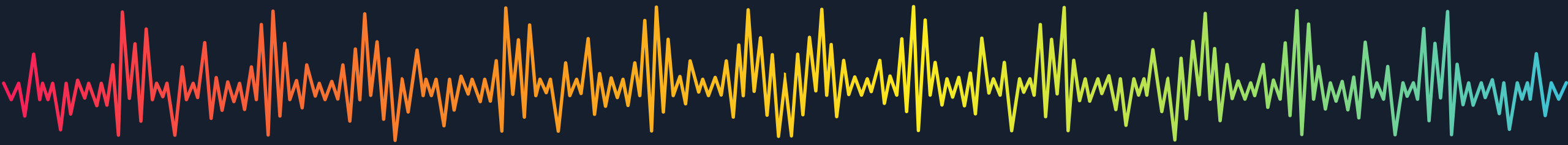
# Without music, life would be a mistake.



— *Friedrich Nietzsche*

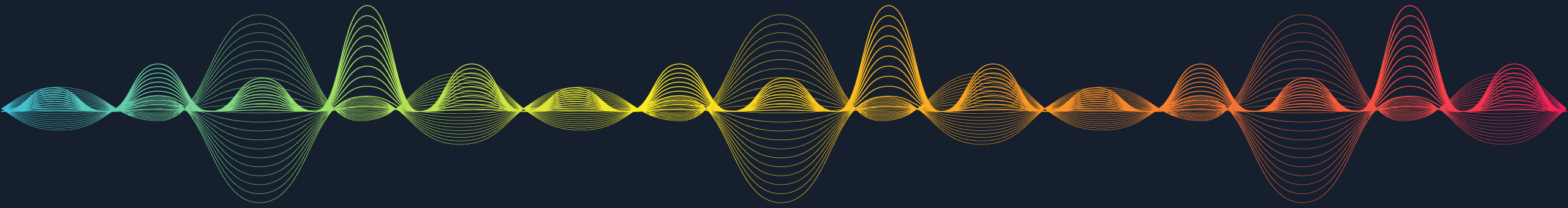
# Table of Contents





# 1

# Musical Style Transfer



# Musical Style Transfer

ASSUMPTION

**Music** = content & style

**Content** = domain-variant

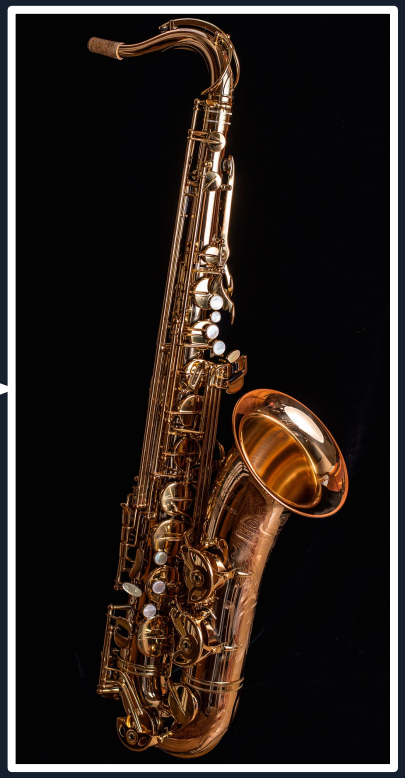
**Style** = domain-invariant

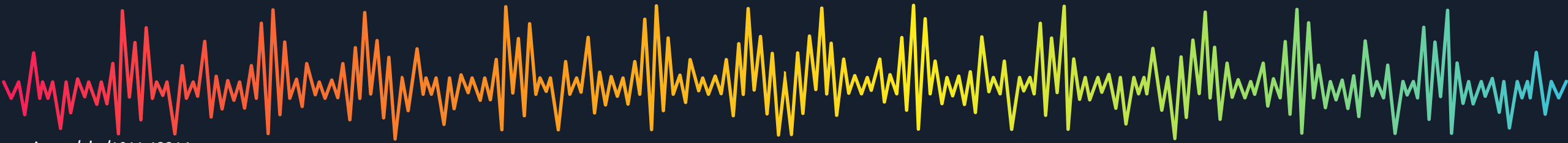
TASK

Modify the style while preserving the content

ISSUE

These concepts are only vaguely defined

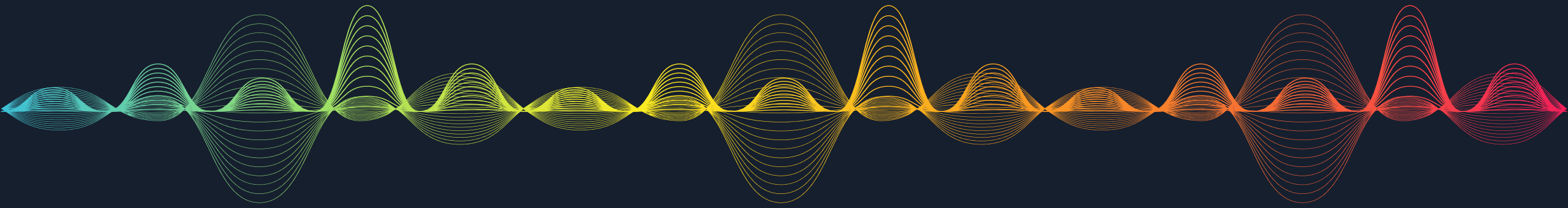




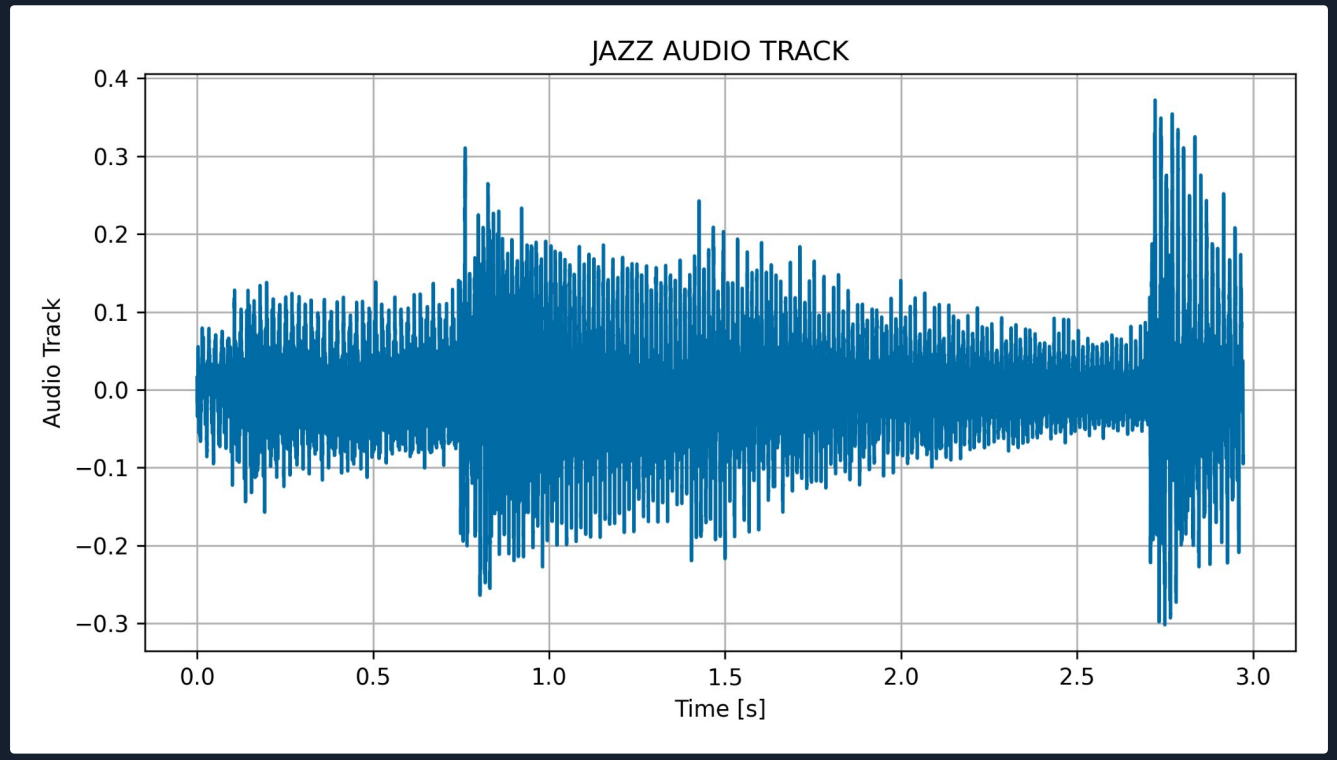
arxiv.org/abs/1811.12214

# 2

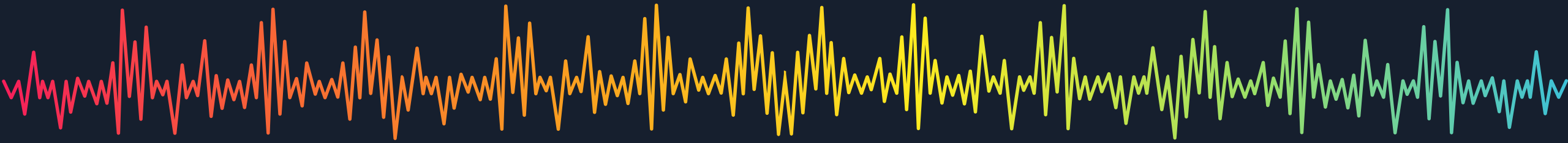
# Music & Signal Processing



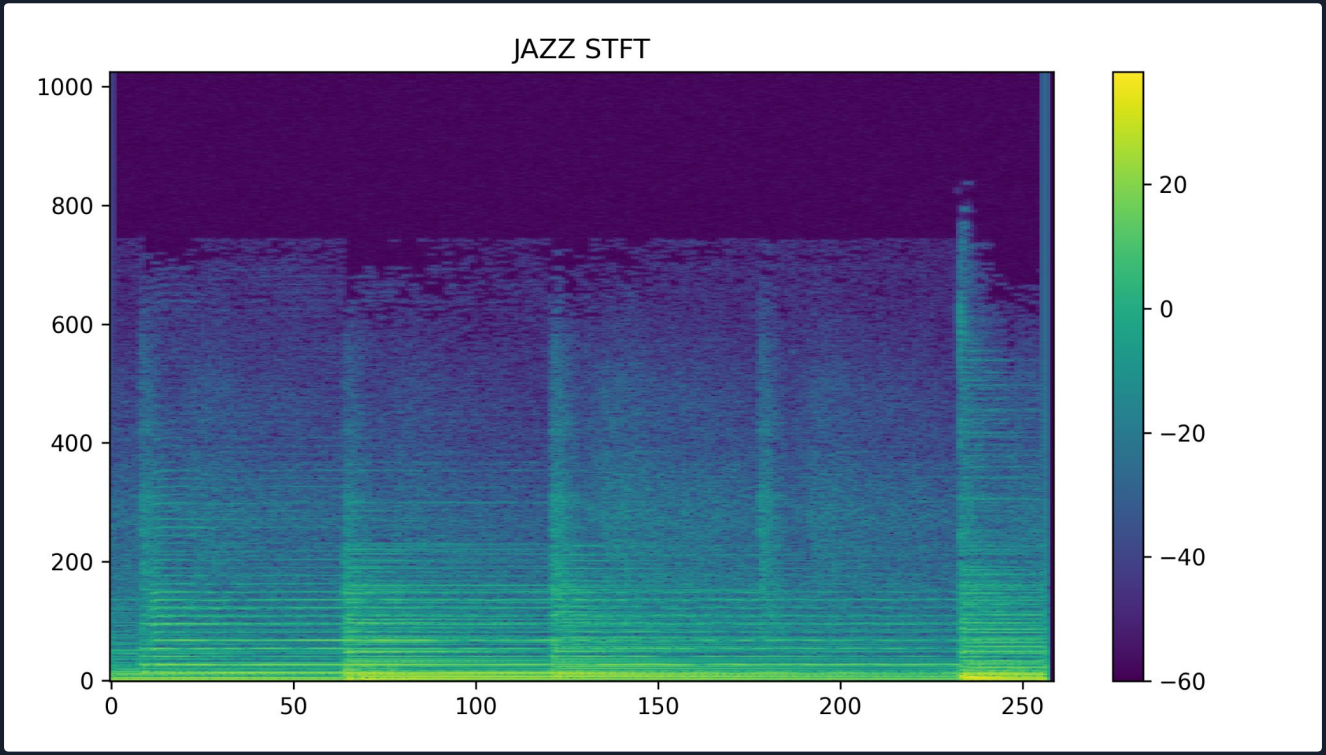
# Music as a Signal



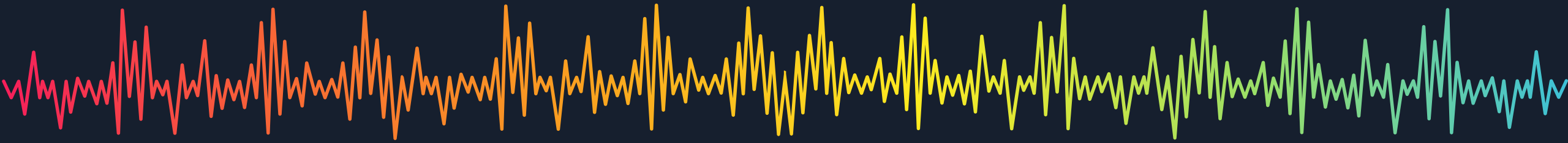

Where is the structure?



# STFT Spectrogram Feature Space

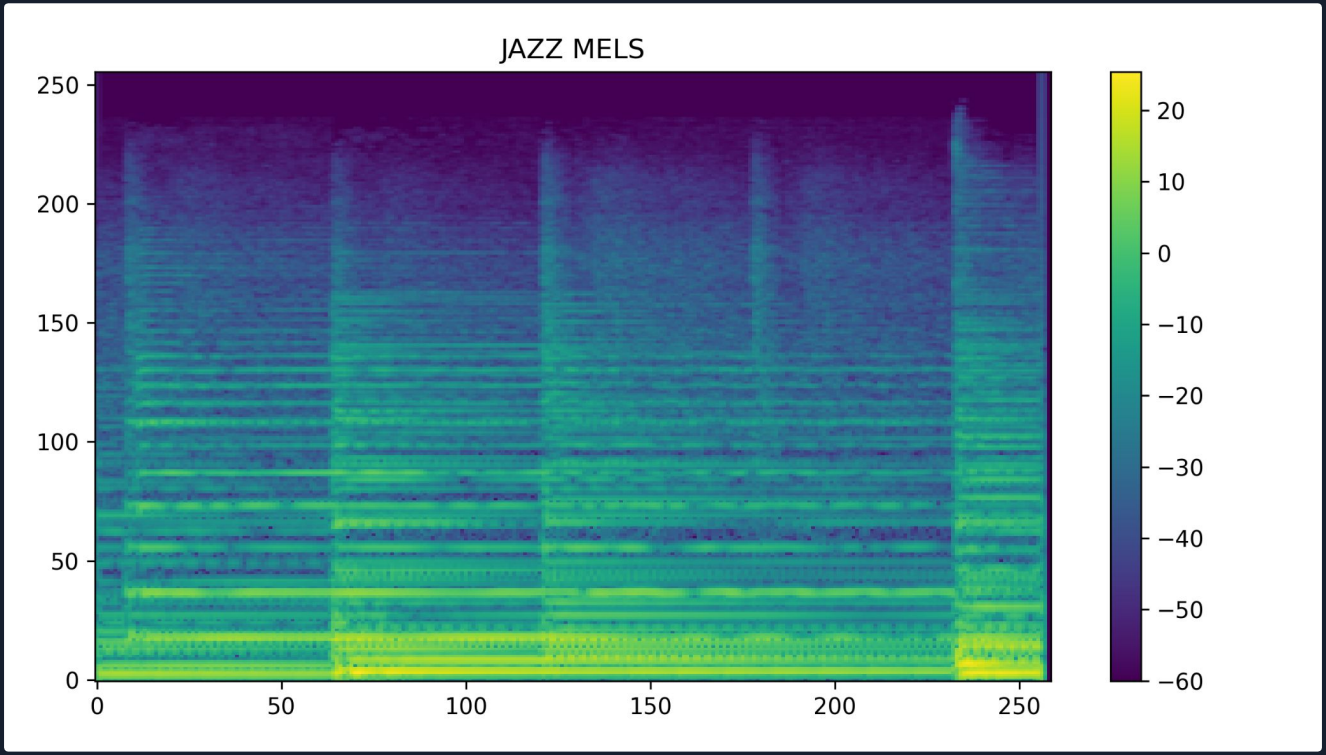



Poor use of space…

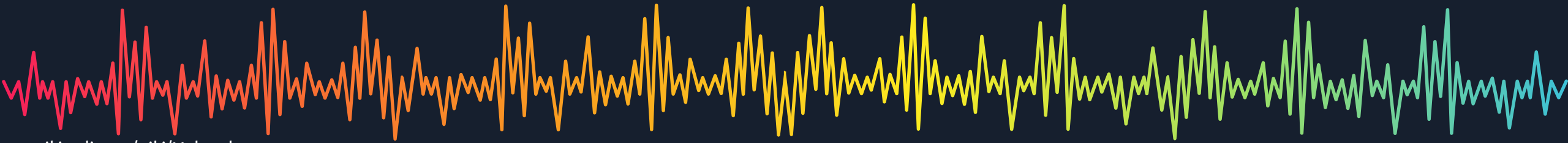


$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

# Mel Spectrogram Feature Space

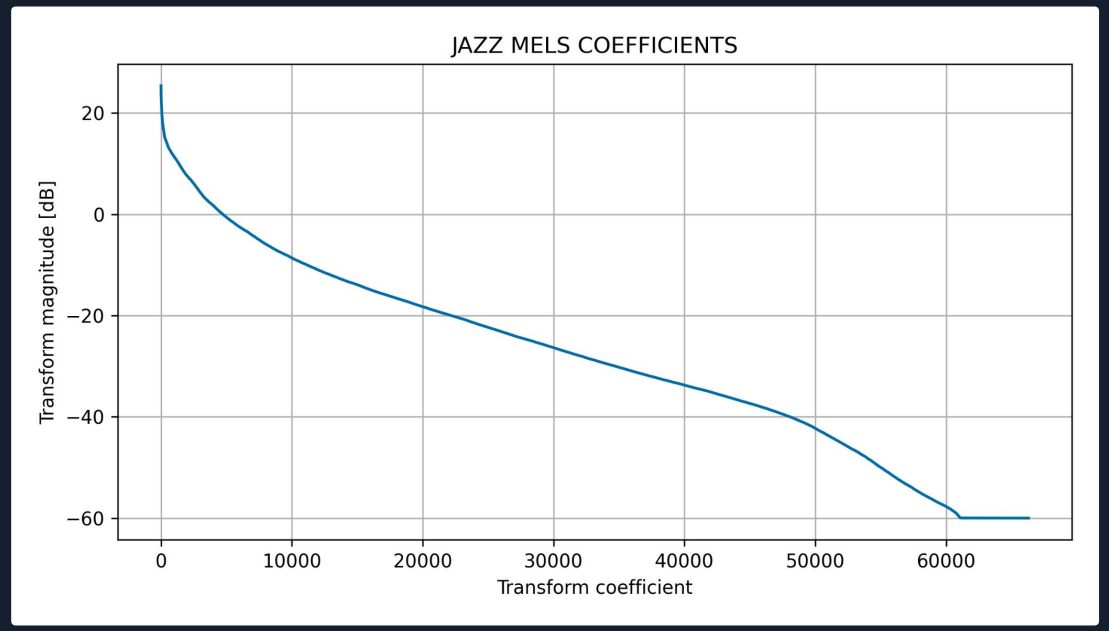


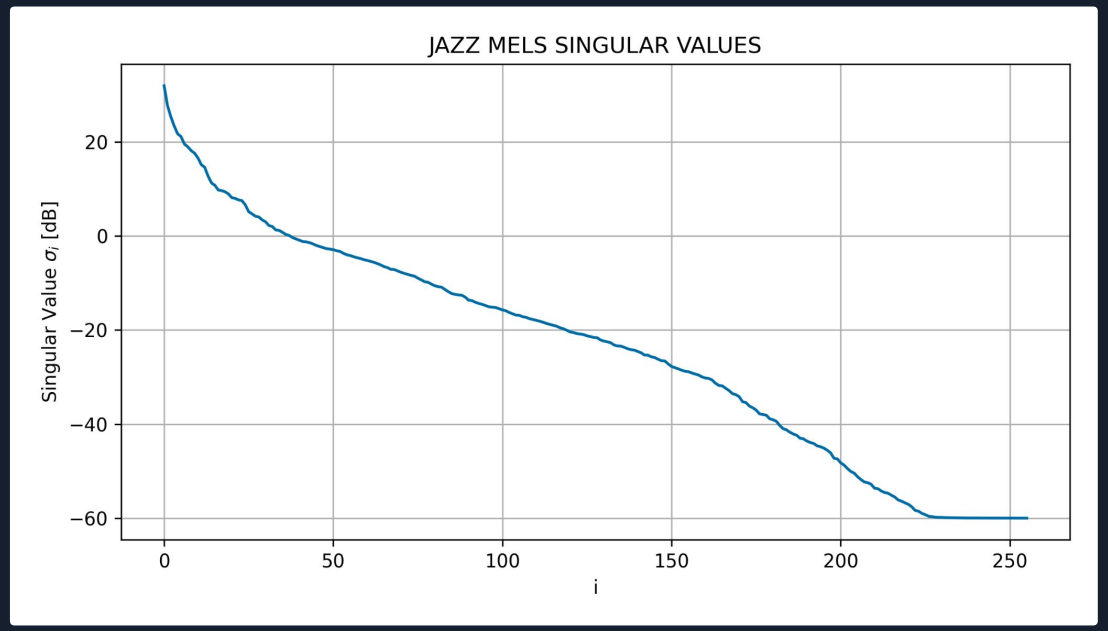

Equidistant frequencies for human hearing



en.wikipedia.org/wiki/Mel_scale

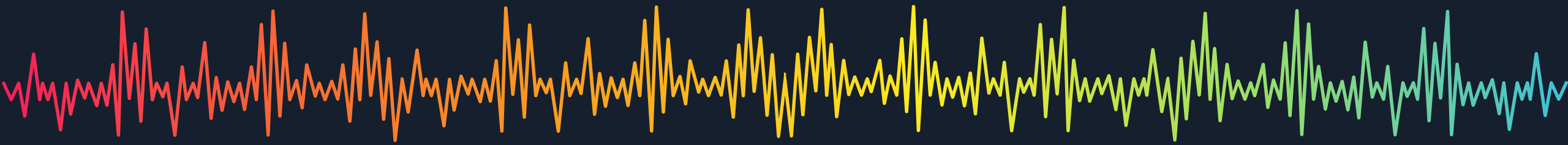
# Mel Spectrogram Feature Space



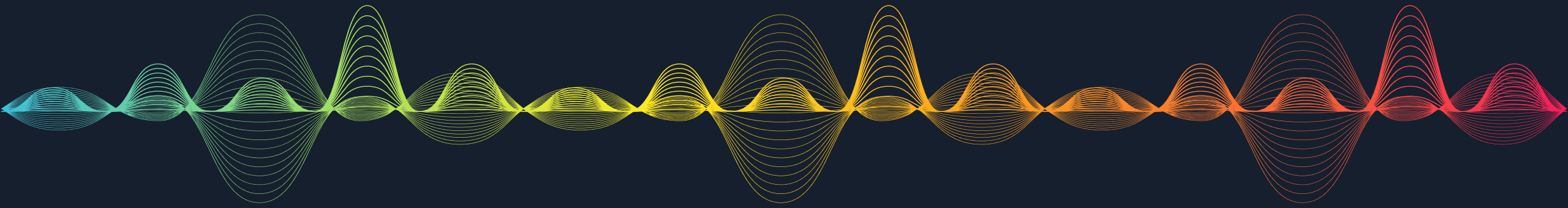




Sparse

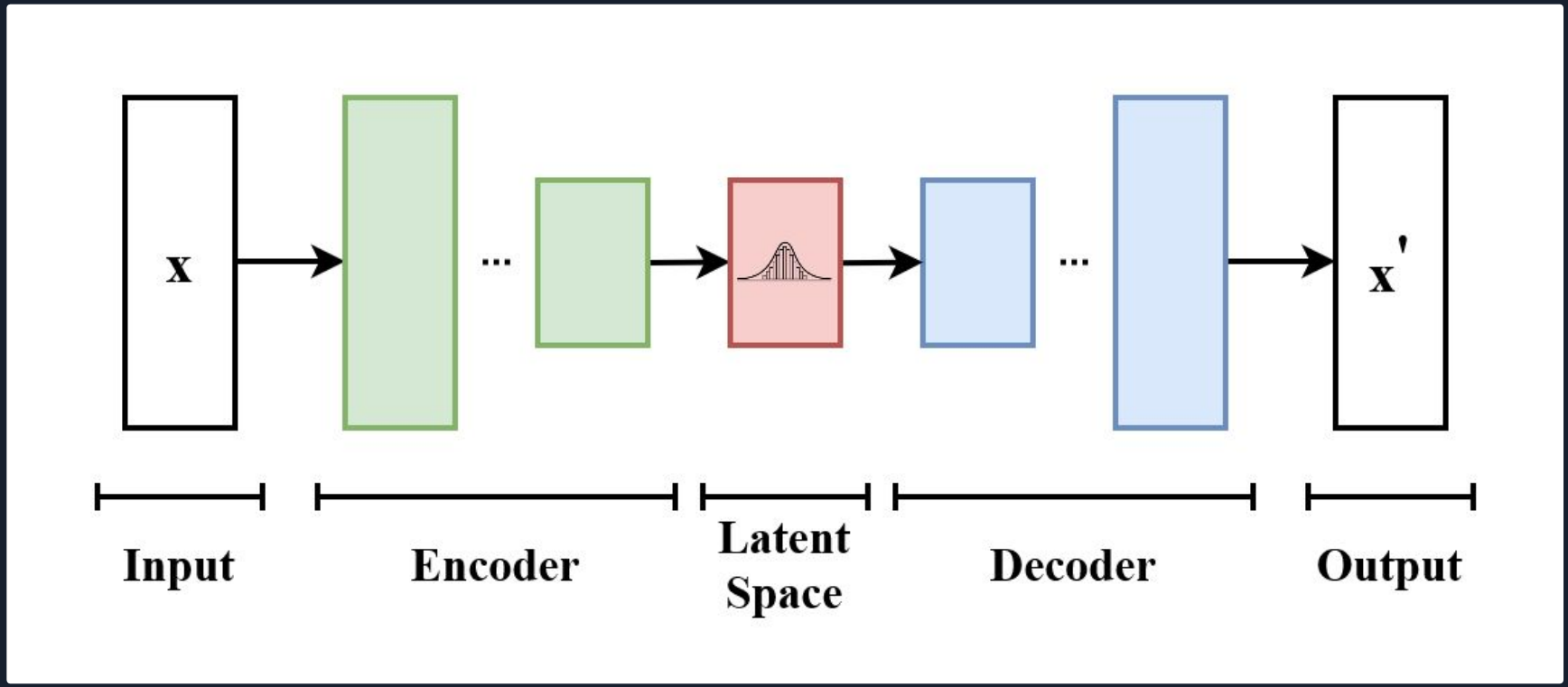
Low Rank

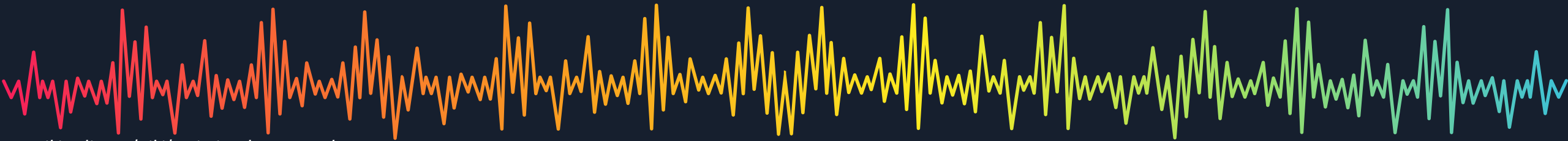


# 3 Autoencoders



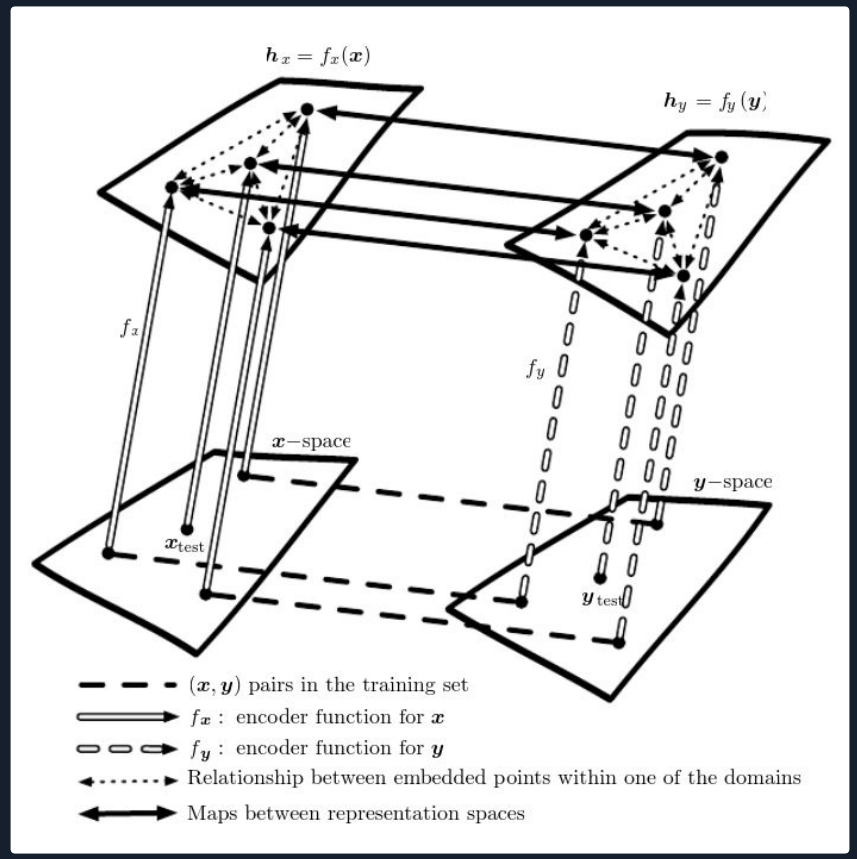
# Autoencoders

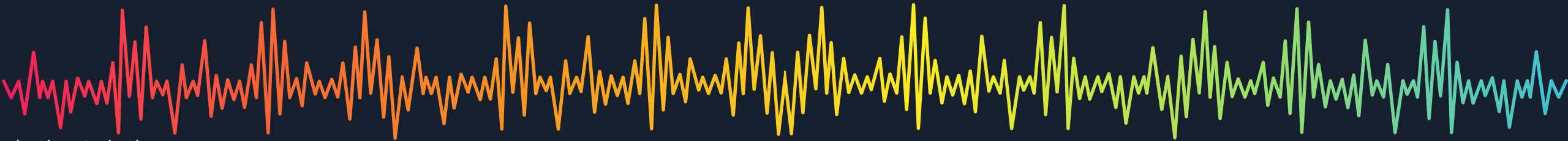


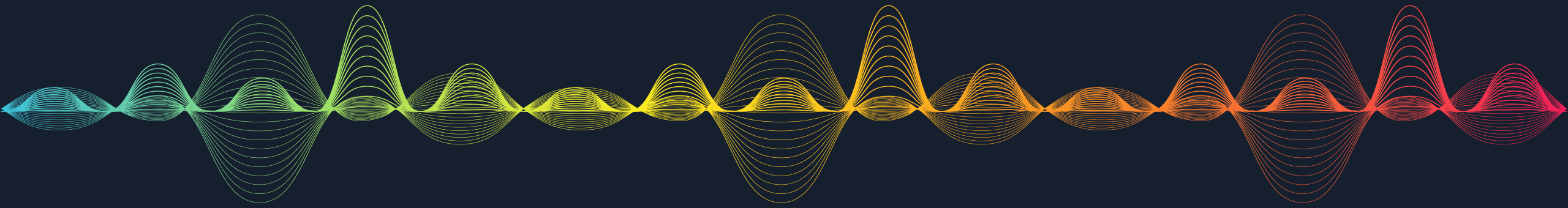
en.wikipedia.org/wiki/Variational_autoencoder

# Representation Learning

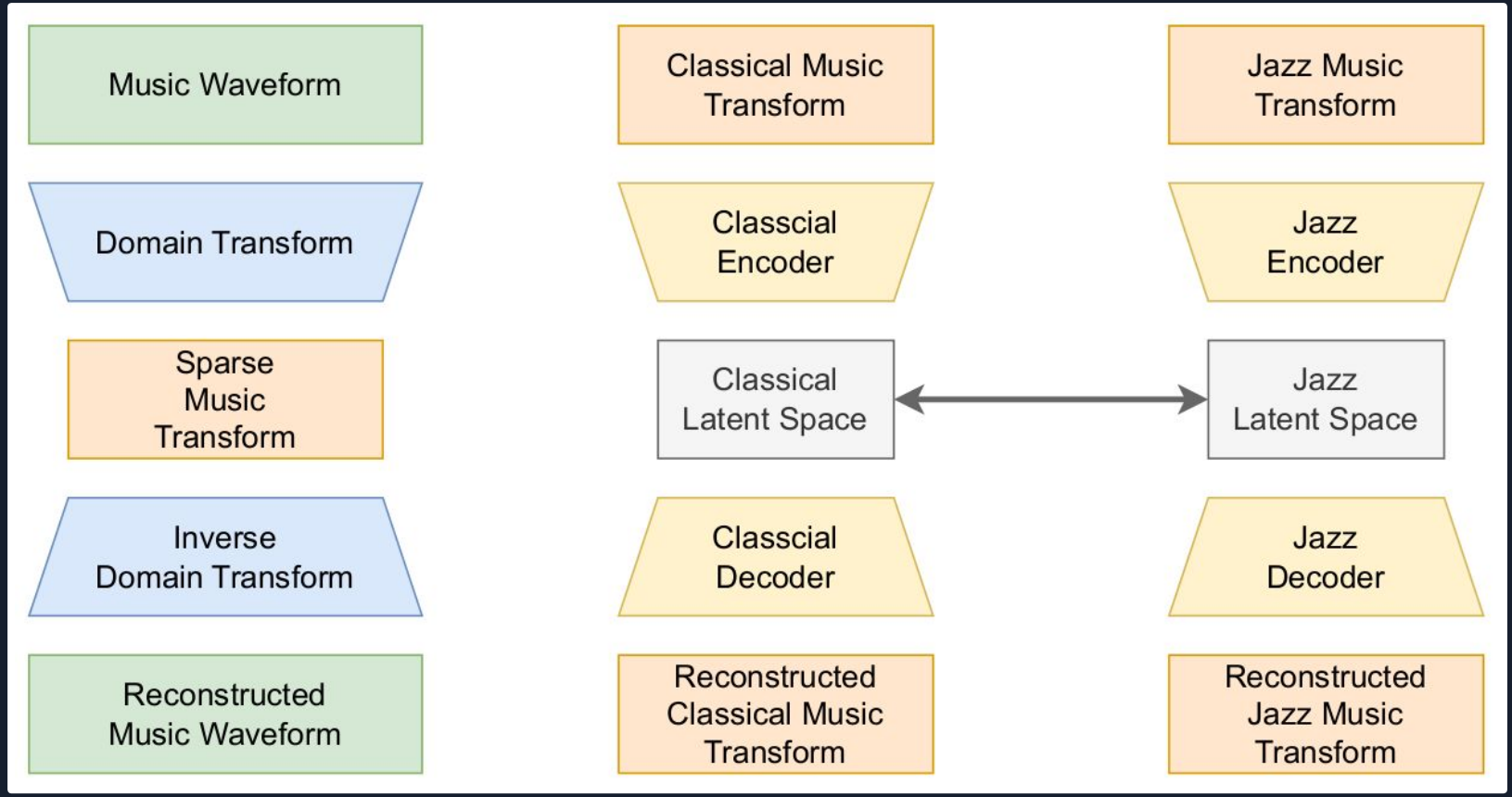


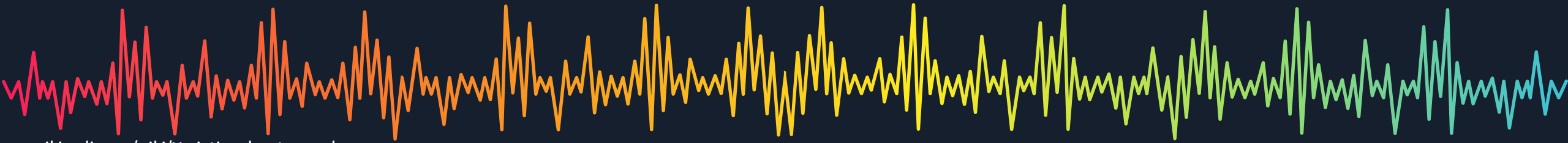



# 4 Experiments & Results



# The Idea



en.wikipedia.org/wiki/Variational_autoencoder

# The Data

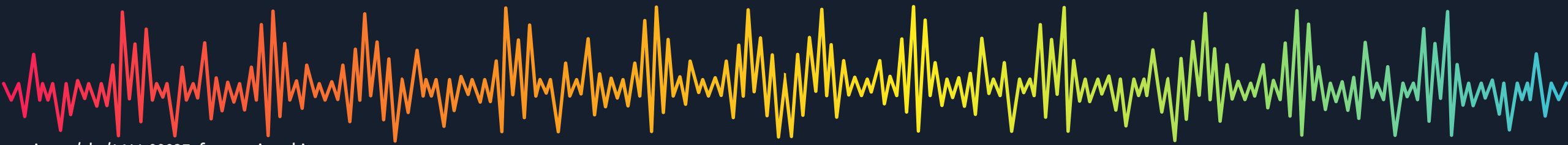
## 2 MusicNet

| Start | End | Instrument | Note | Measure | Beat | Note Value |
|---|---|---|---|---|---|---|
| 45.29 | 45.49 | Violin | G5 | 21 | 3 | Eighth |
| 48.99 | 50.13 | Cello | A#3 | 24 | 2 | Dotted Half |
| 82.91 | 83.12 | Viola | C5 | 51 | 2.5 | Eighth |

FREE MUSIC ARCHIVE

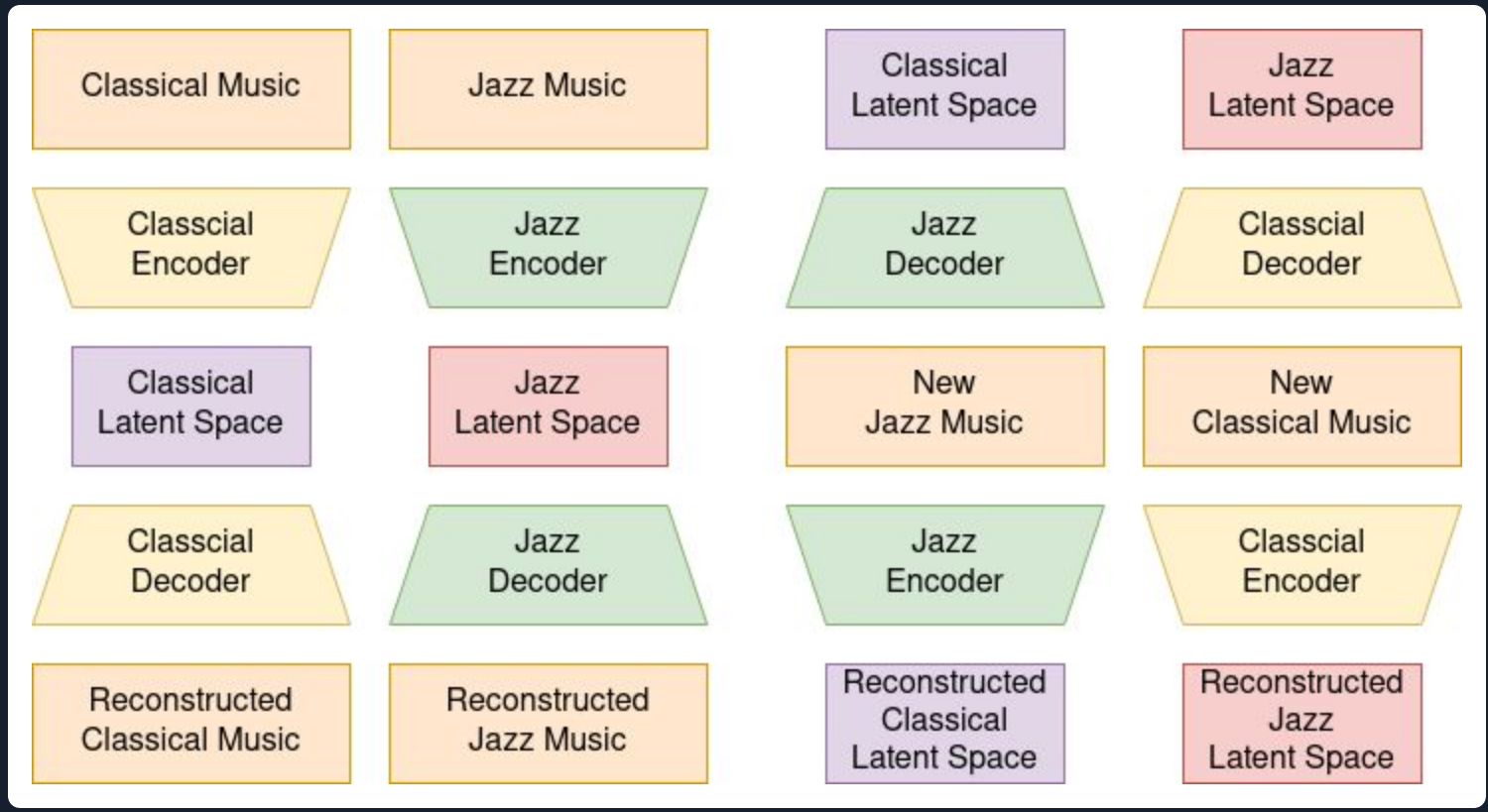
Classical Music

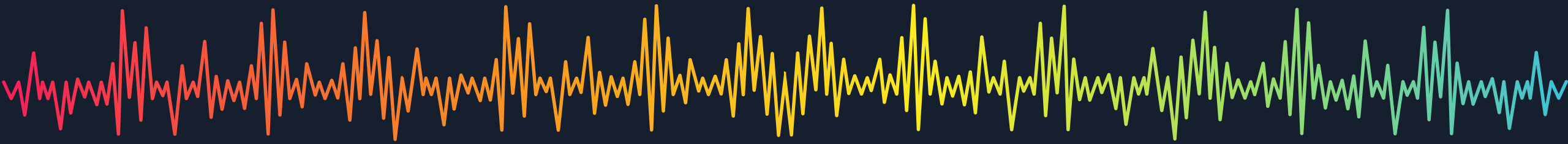
Jazz Music



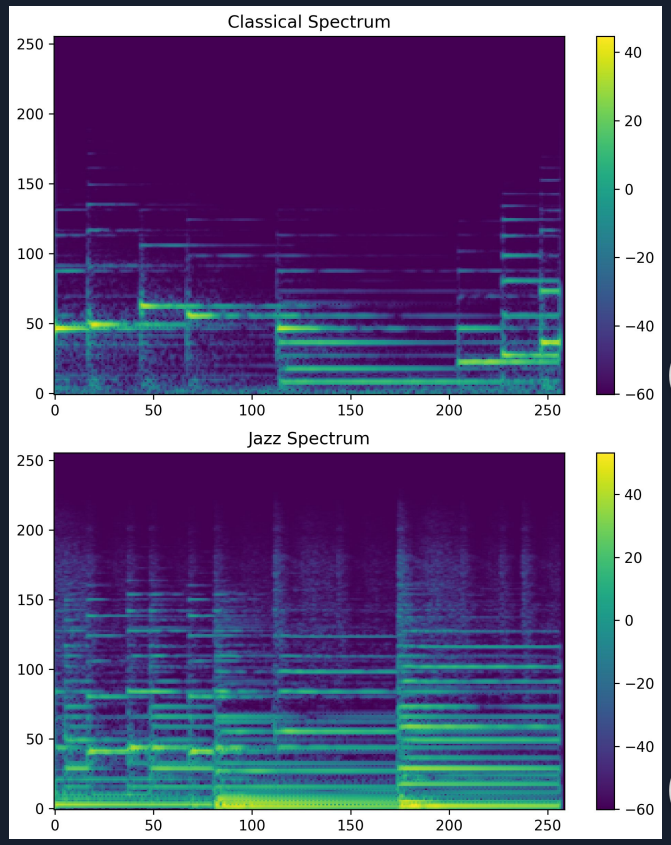
arxiv.org/abs/1611.09827, freemusicarchive.org

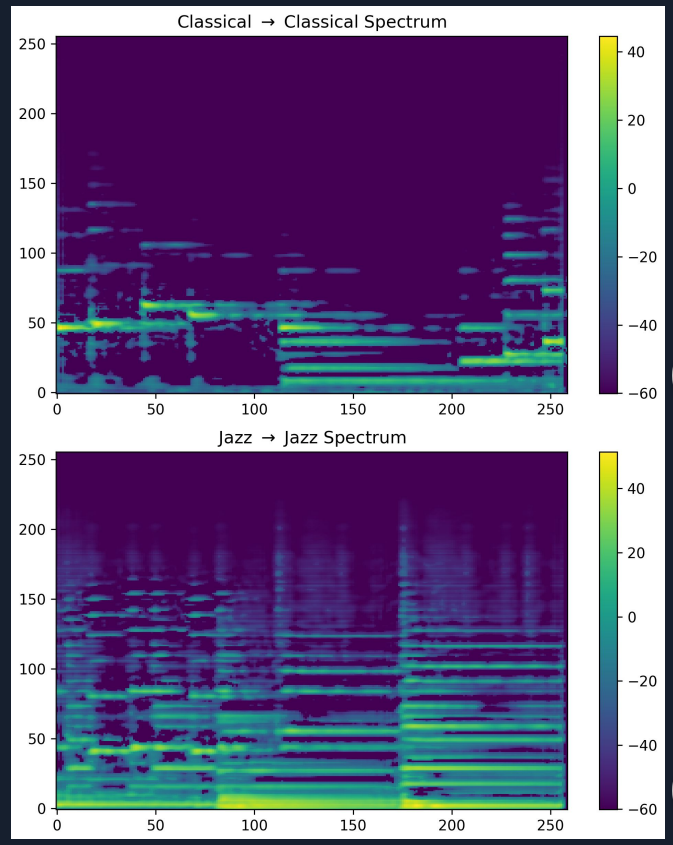
# 1 - Naive Twin Autoencoders

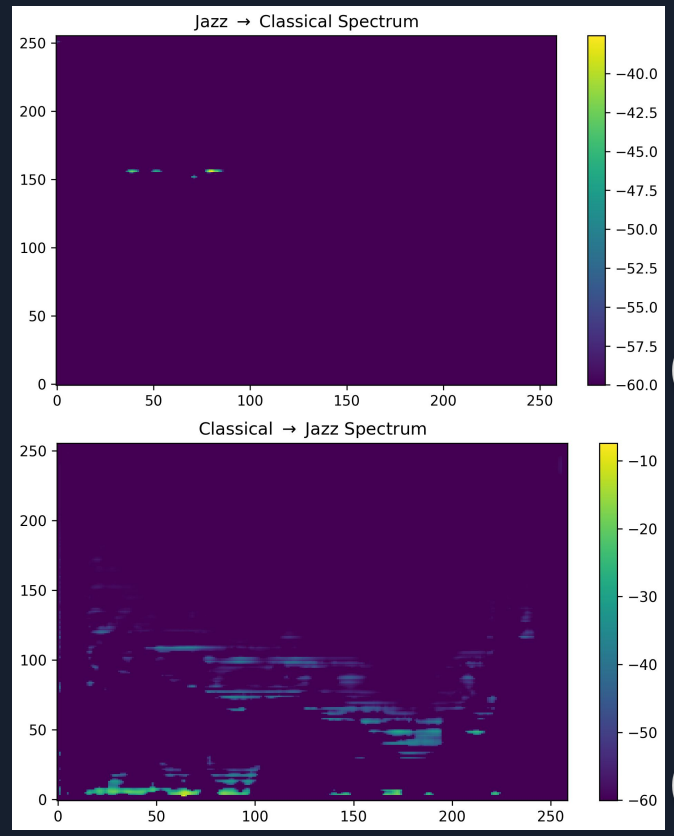


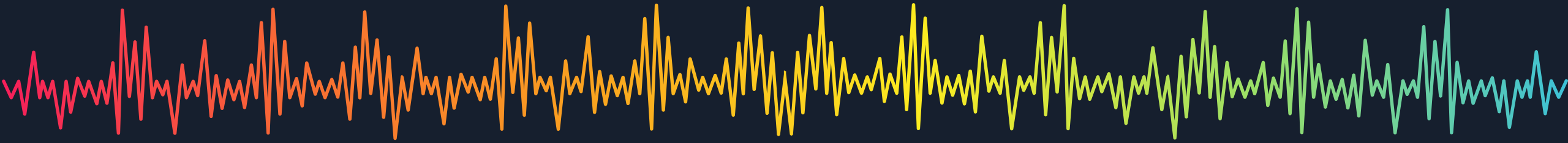



# 1 - Naive Twin Autoencoders

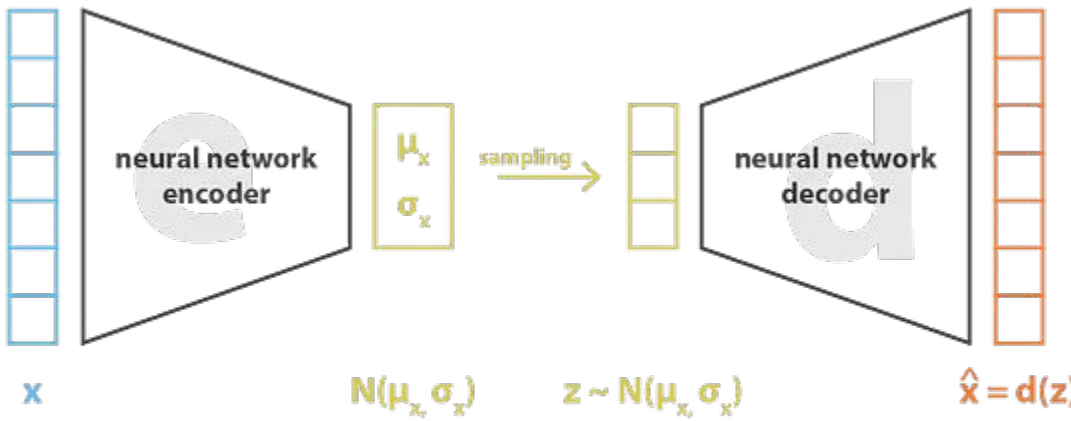


# 2 - Variational Twin Autoencoders

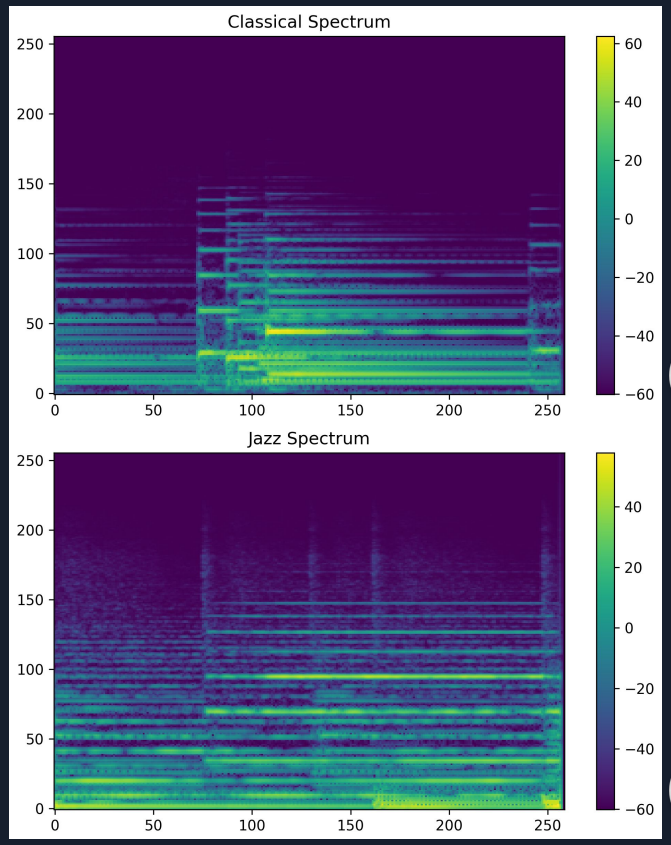


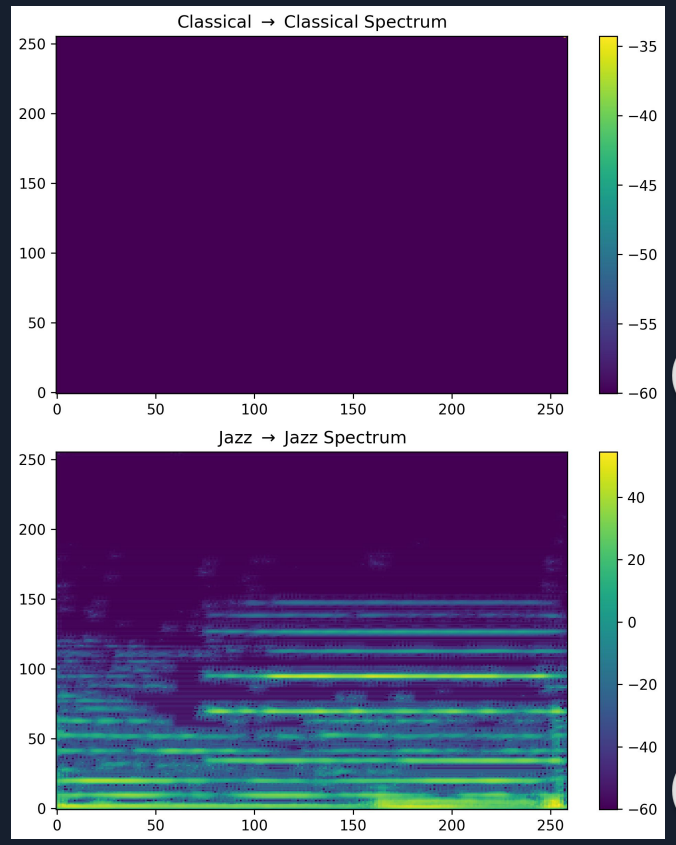

loss = $\| x - \hat{x} \|^2 + KL[ N(\mu_x, \sigma_x), N(0, I) ] = \| x - d(z) \|^2 + KL[ N(\mu_x, \sigma_x), N(0, I) ]$

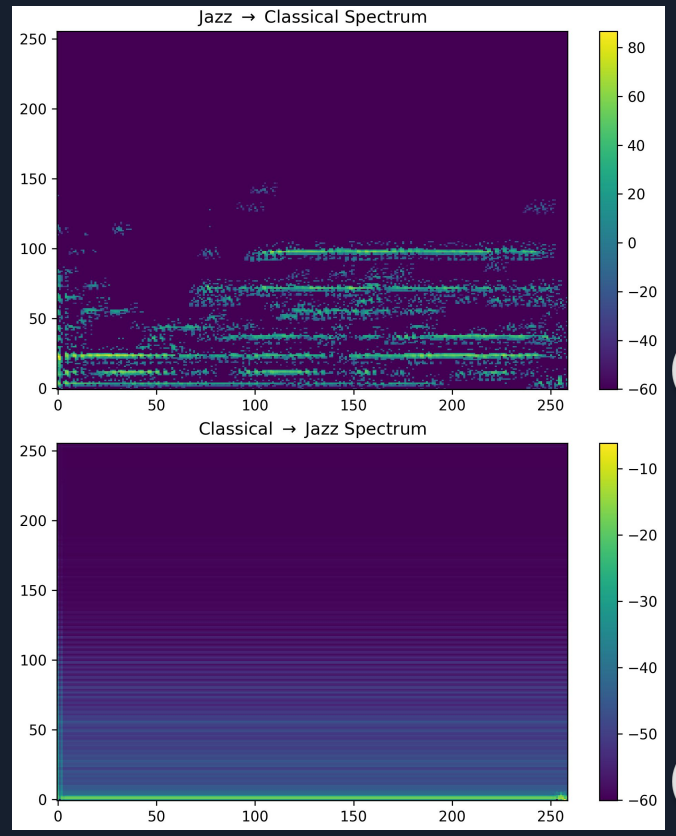
$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right),$$

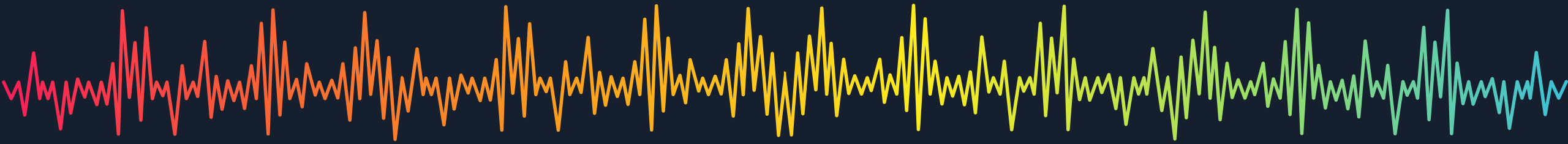
towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73, en.wikipedia.org/wiki/Kullback–Leibler_divergence

# 2 - Variational Twin Autoencoders (5 Epochs)

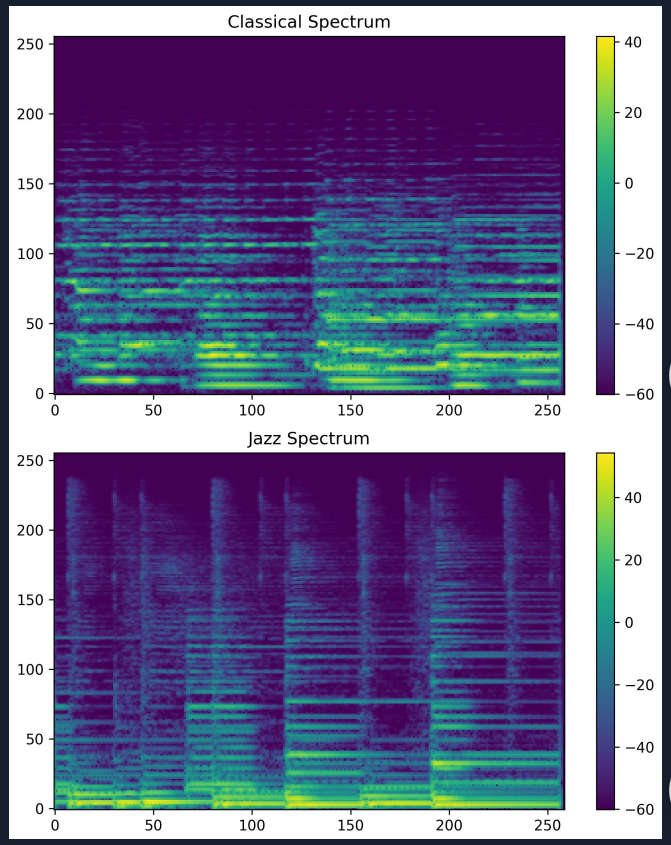


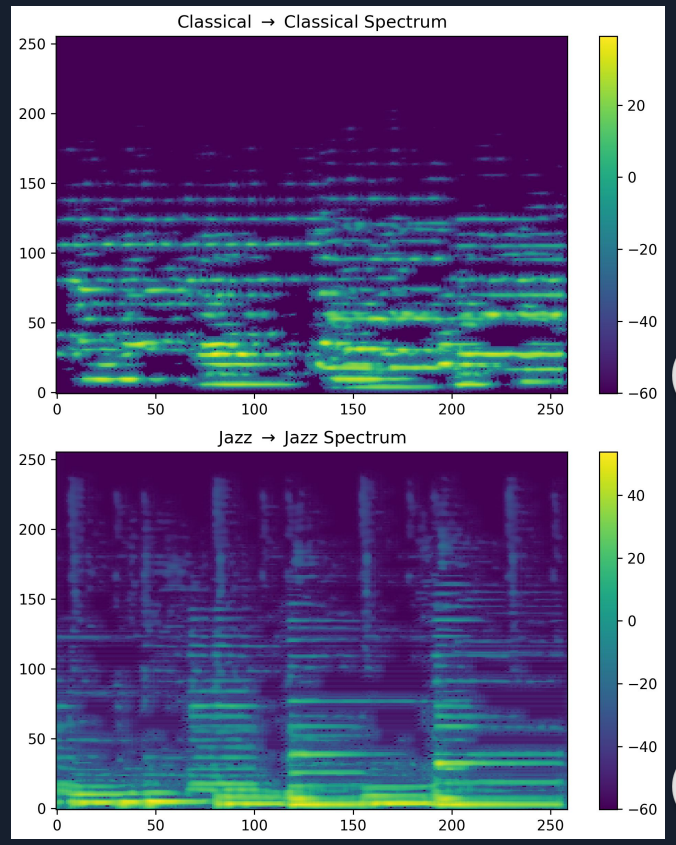



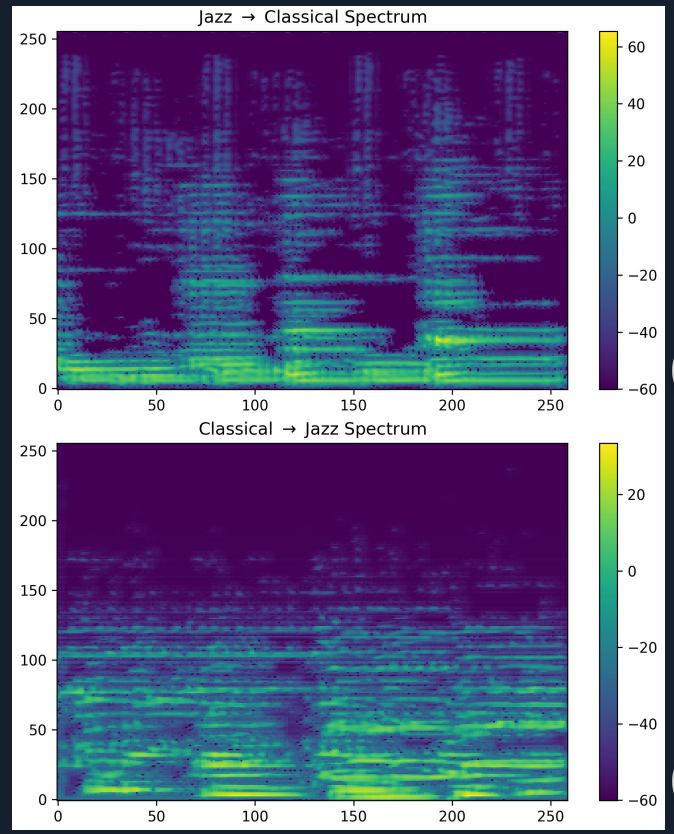



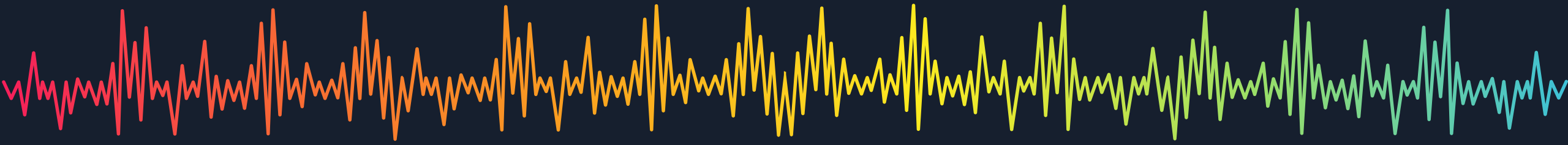



# 2 - Variational Twin Autoencoders

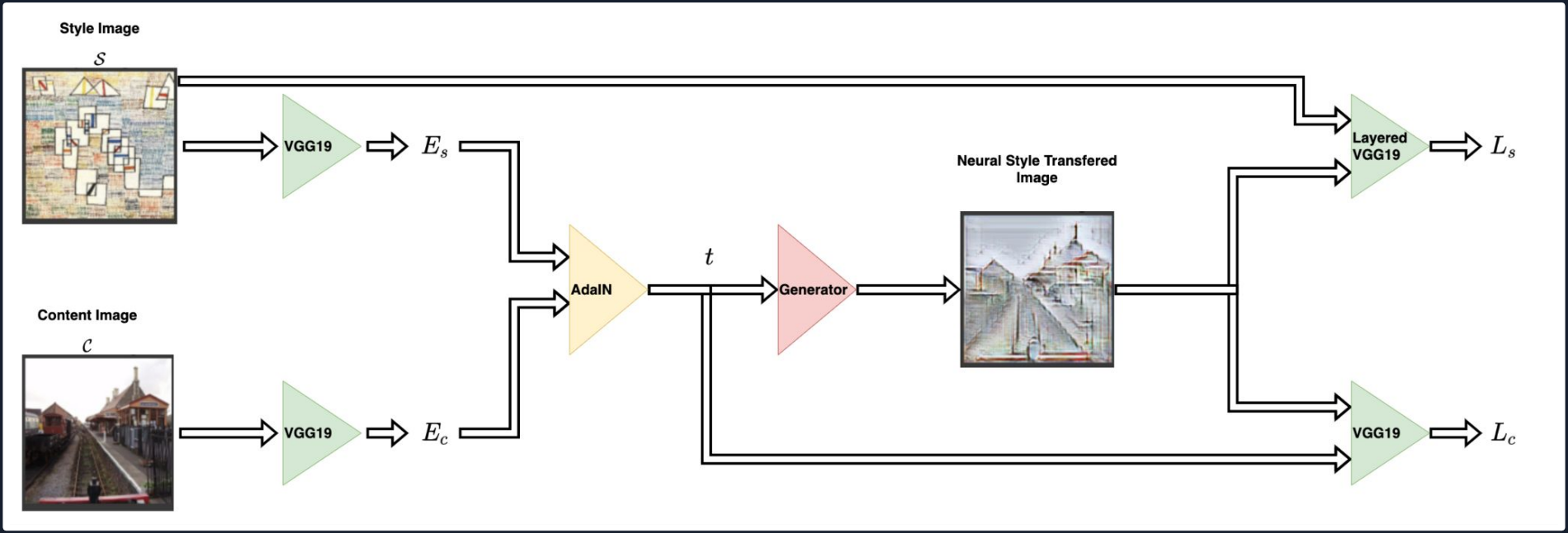


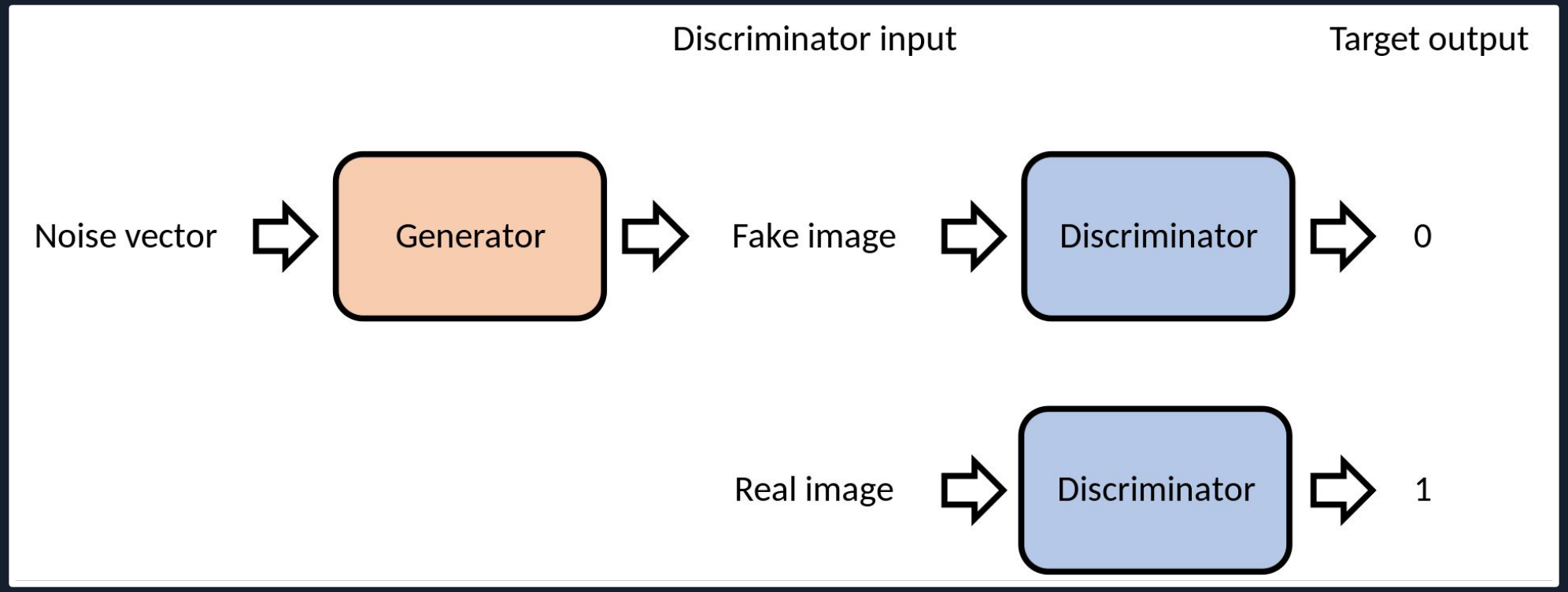
# 3 - GAN + AdaIN Twin Autoencoders




$$AdaIn(x, y) = \sigma(y)(\frac{x - \mu(x)}{\sigma(x)}) + \mu(y)$$

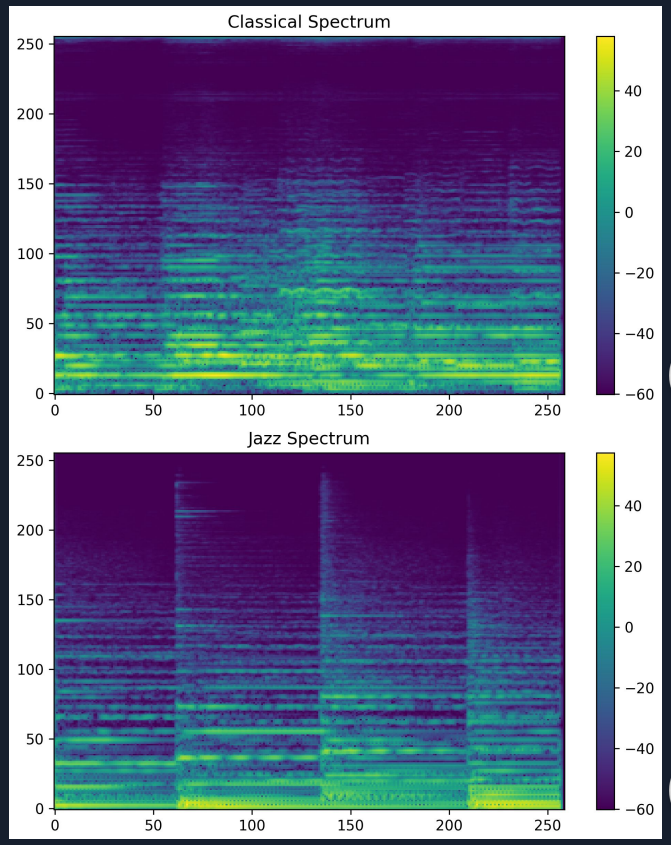
keras.io/examples/generative/adain

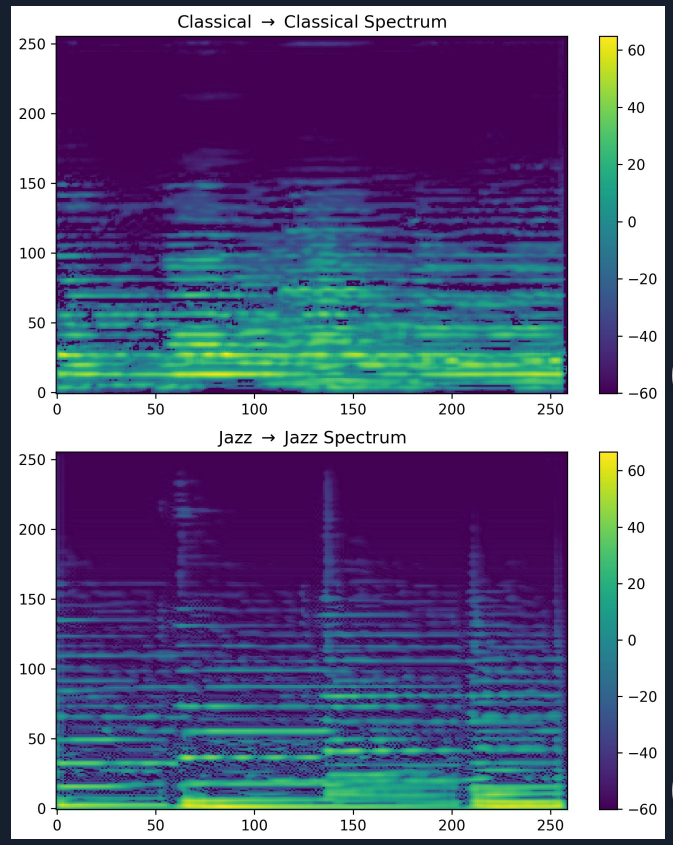
# 3 - GAN + AdaIN Twin Autoencoders

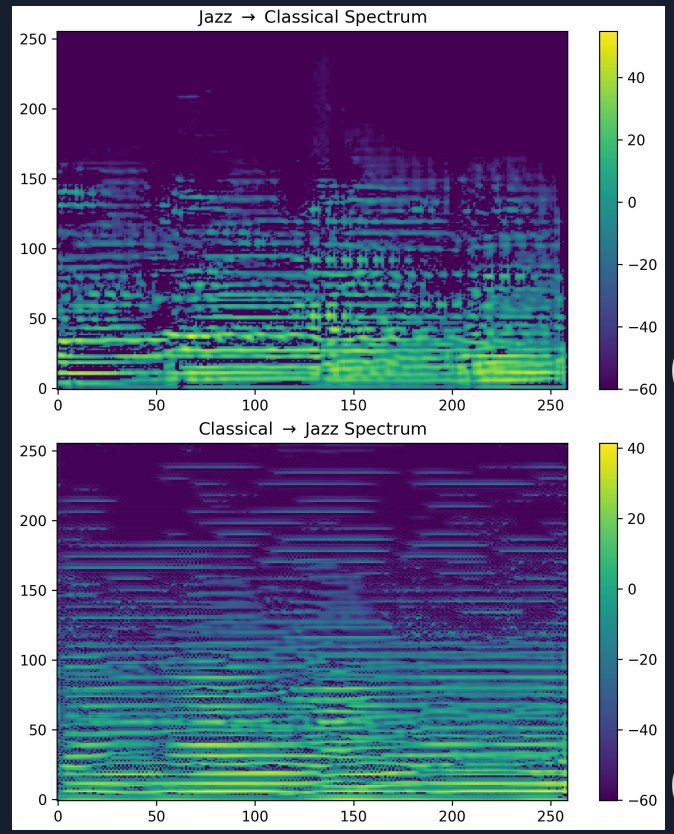


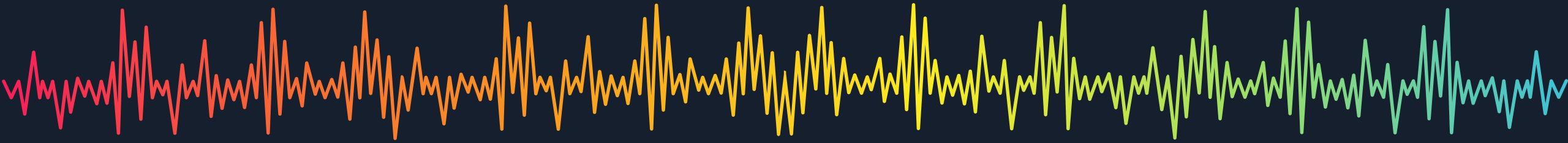

en.wikipedia.org/wiki/Generative_adversarial_network

# 3 - GAN + AdaIN Twin Autoencoders

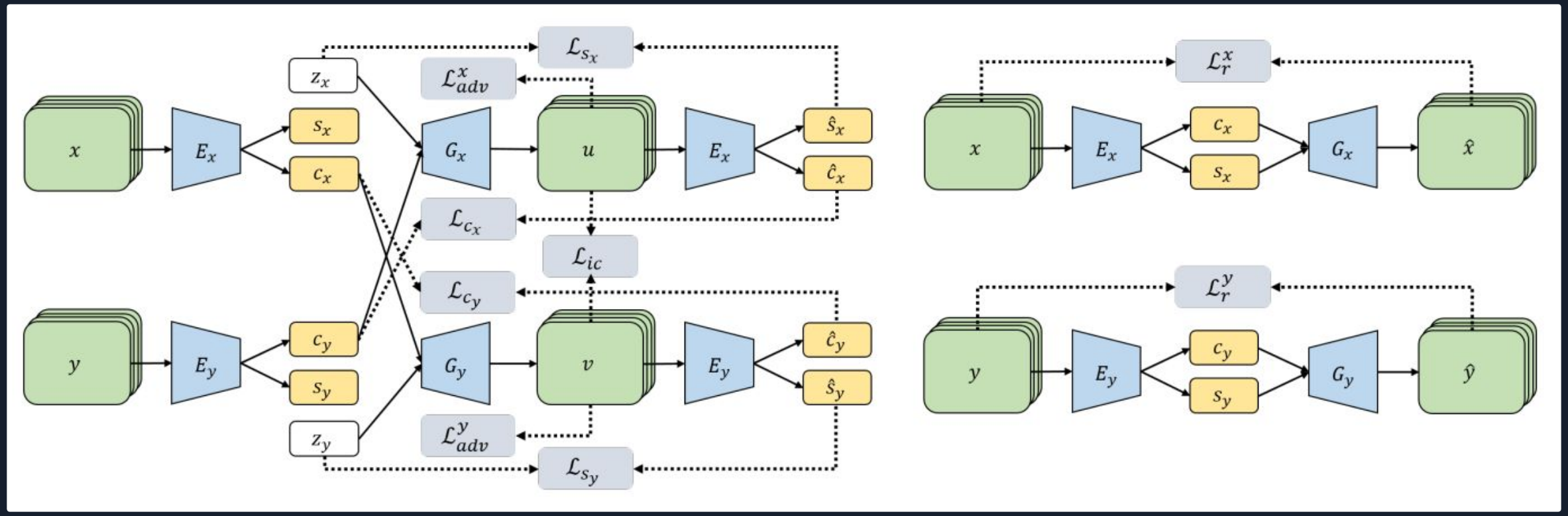


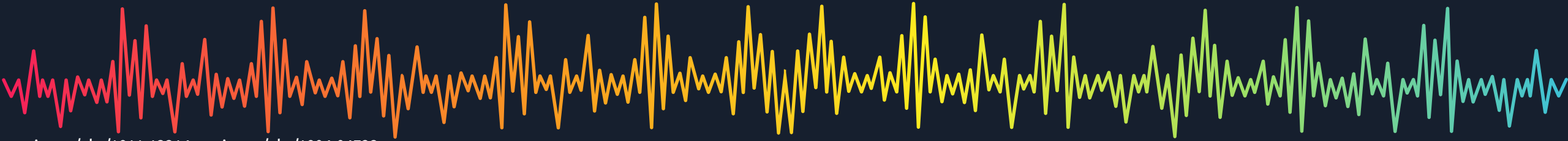



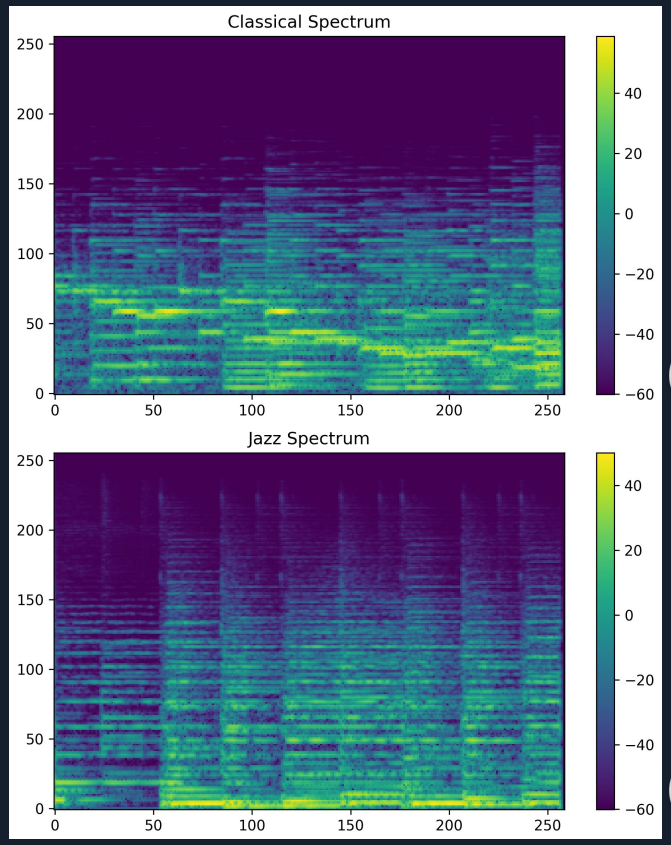
# 4 - Variational + GAN + Adain Twin Autoencoders

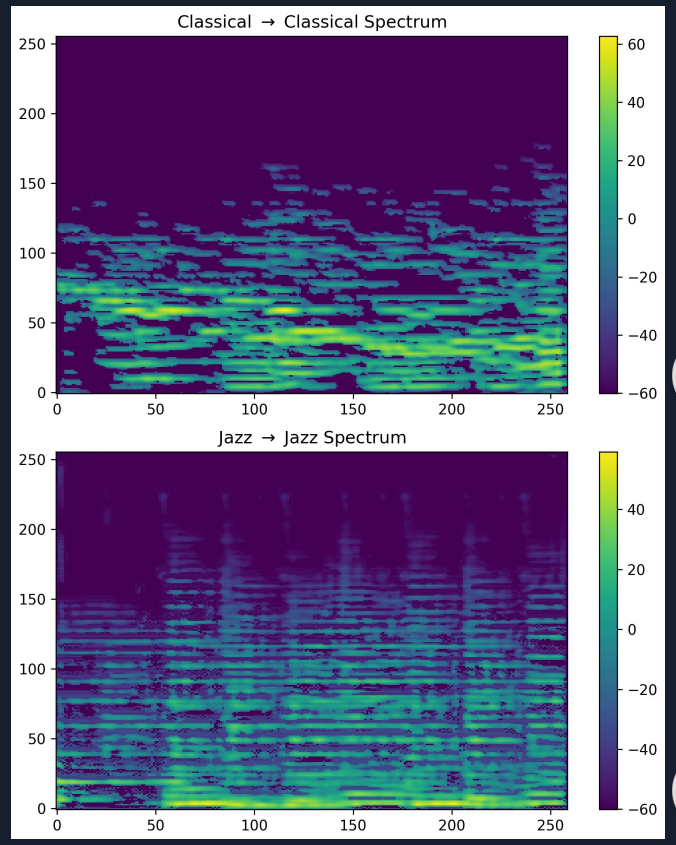


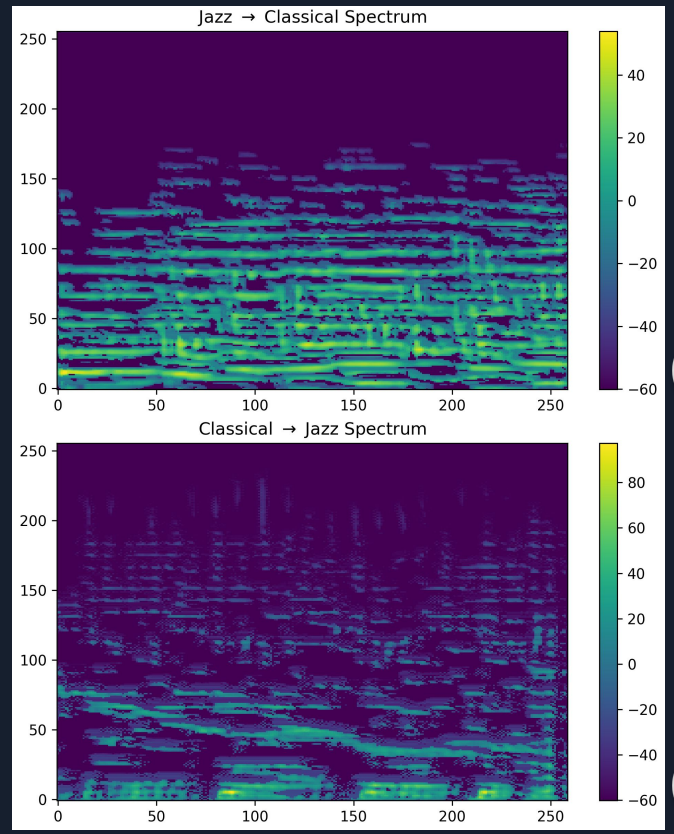



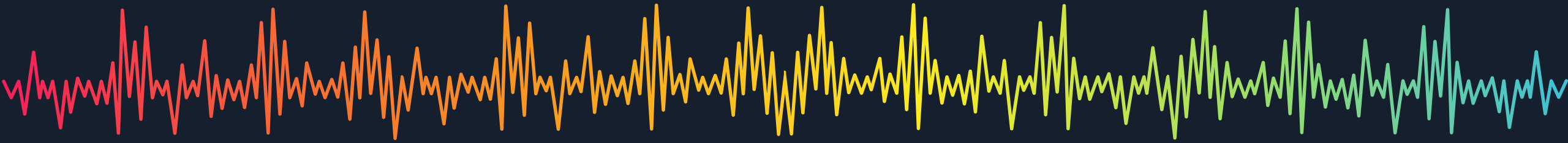
arxiv.org/abs/1811.12214, arxiv.org/abs/1804.04732

# 4 - Variational + GAN + Adain Twin Autoencoders

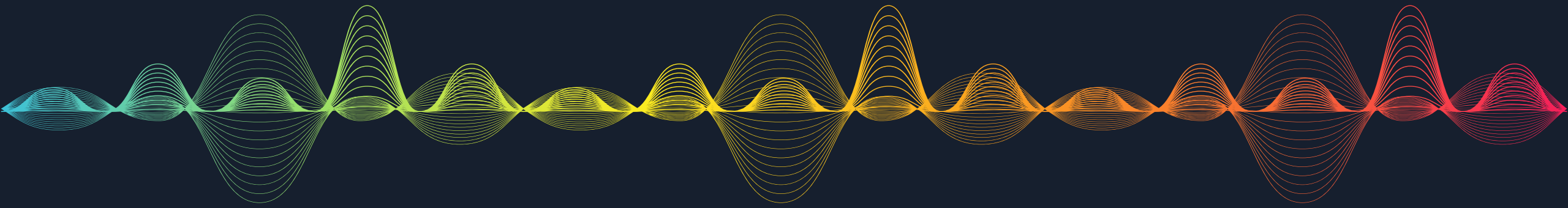


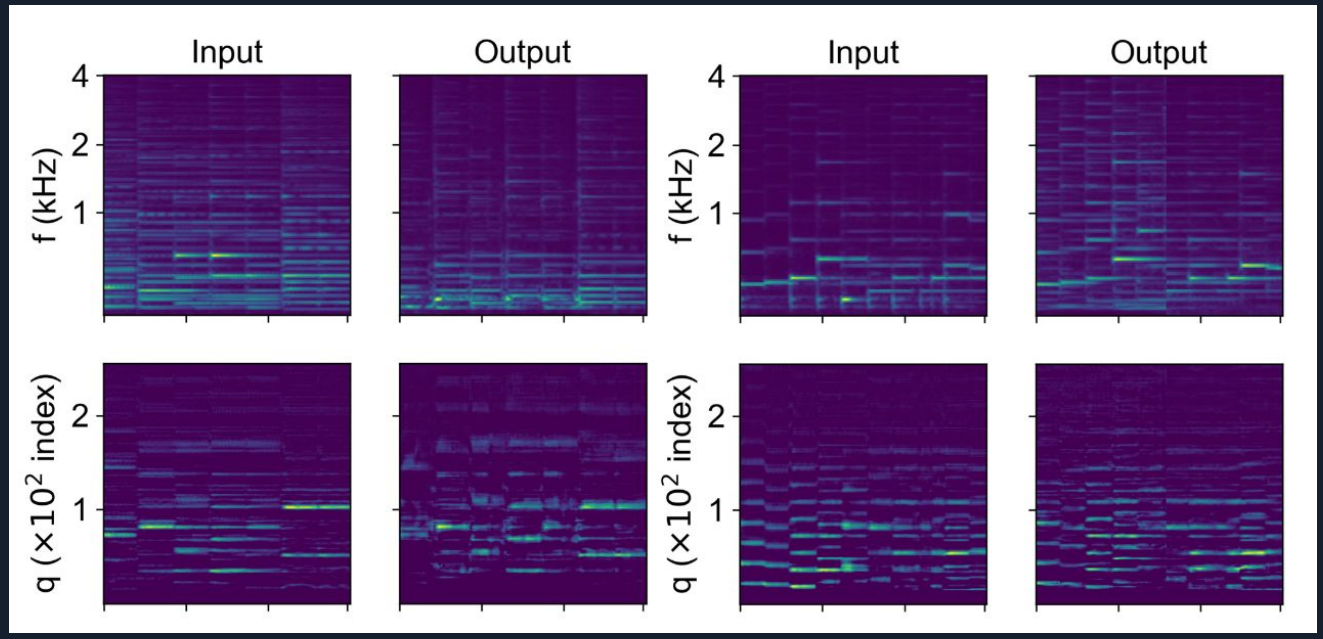



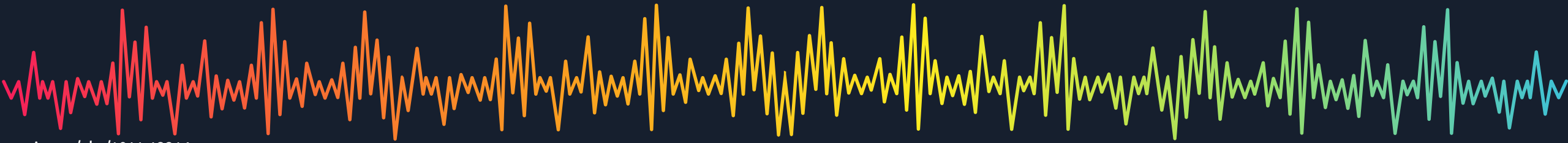



# 5 State of the Art



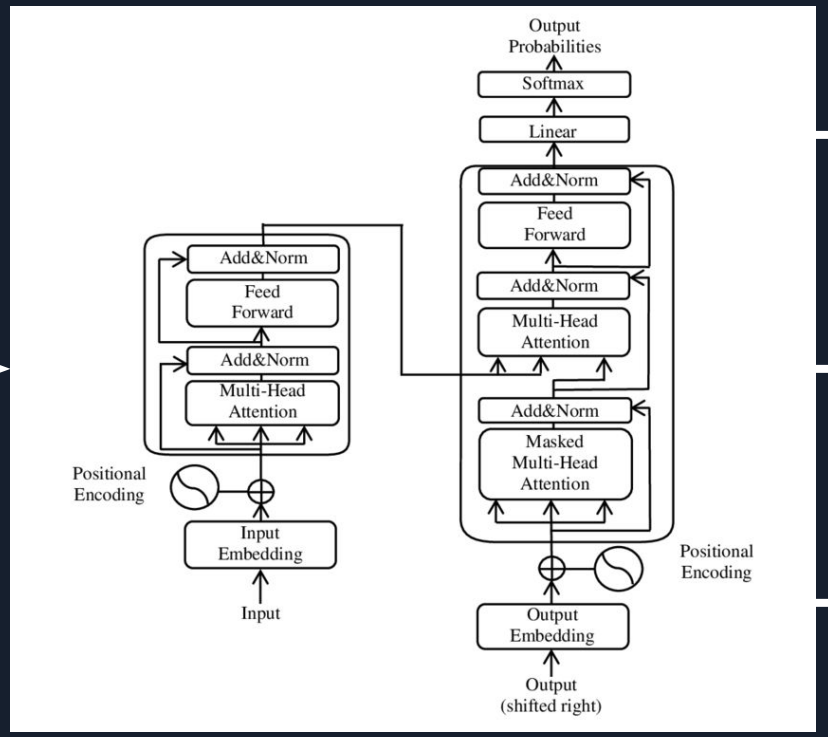
# (2018) Play as You Like: Timbre-enhanced Multi-modal Music Style Transfer



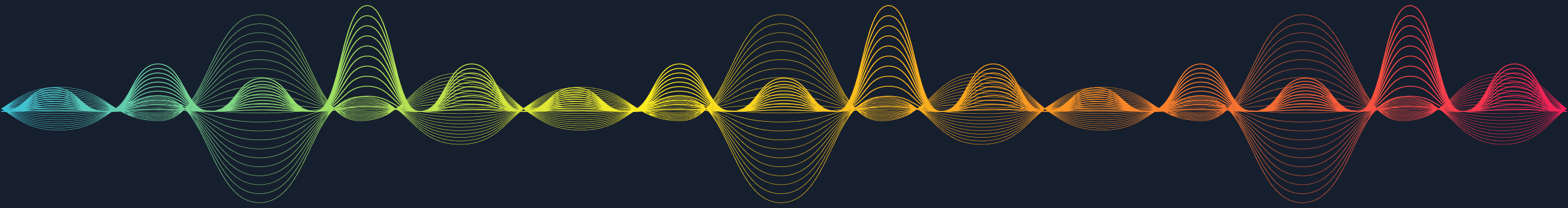
arxiv.org/abs/1811.12214

# (2023) Meta MusicGen: Simple and Controllable Music Generation

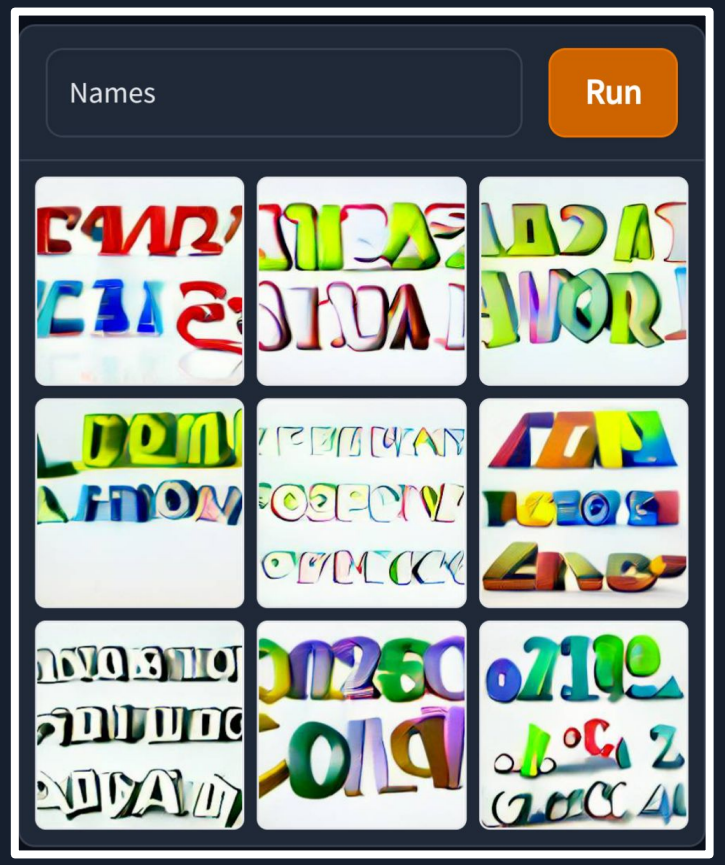



arxiv.org/abs/2306.05284, en.wikipedia.org/wiki/Transformer_(machine_learning_model)

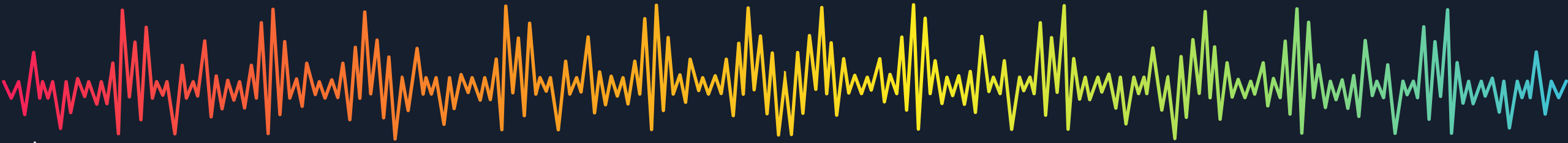
# 6 Conclusion



# Style Transfer / Generative AI is a Difficult Problem

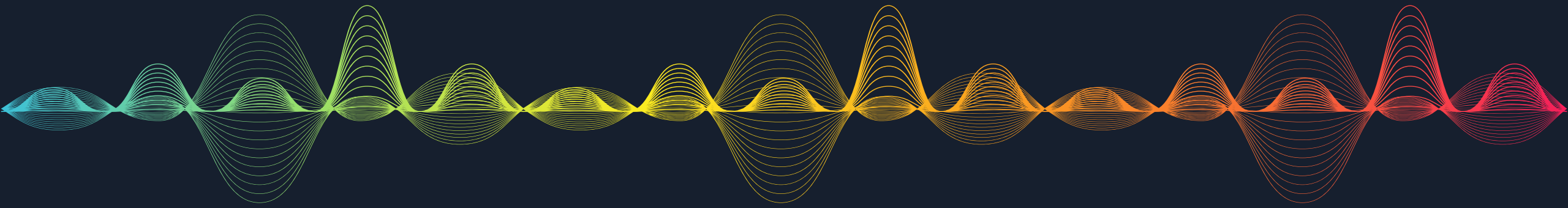




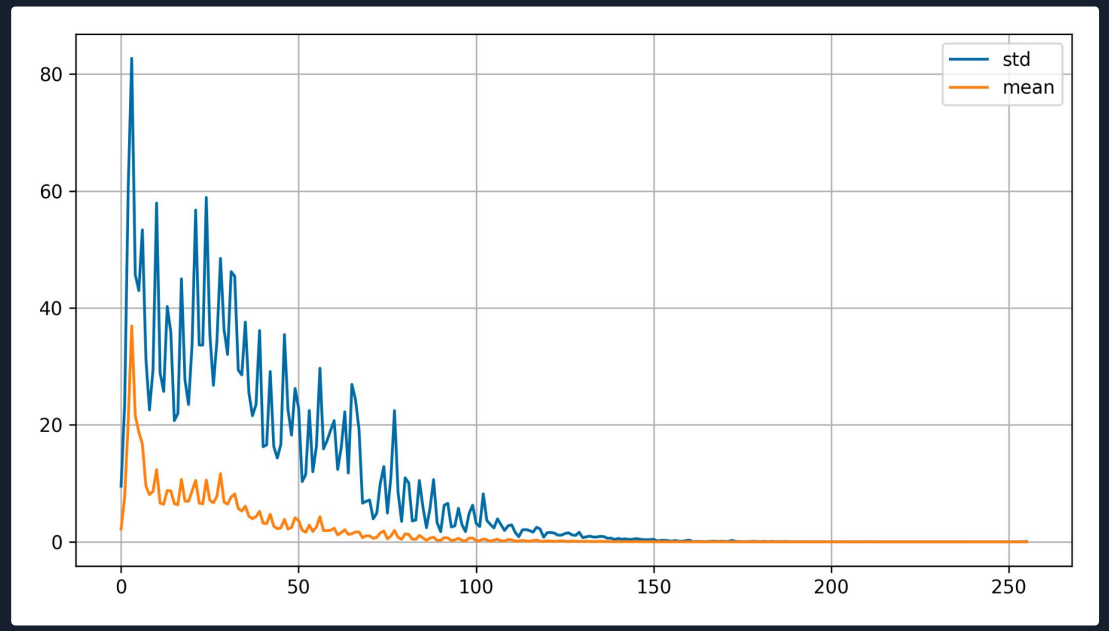


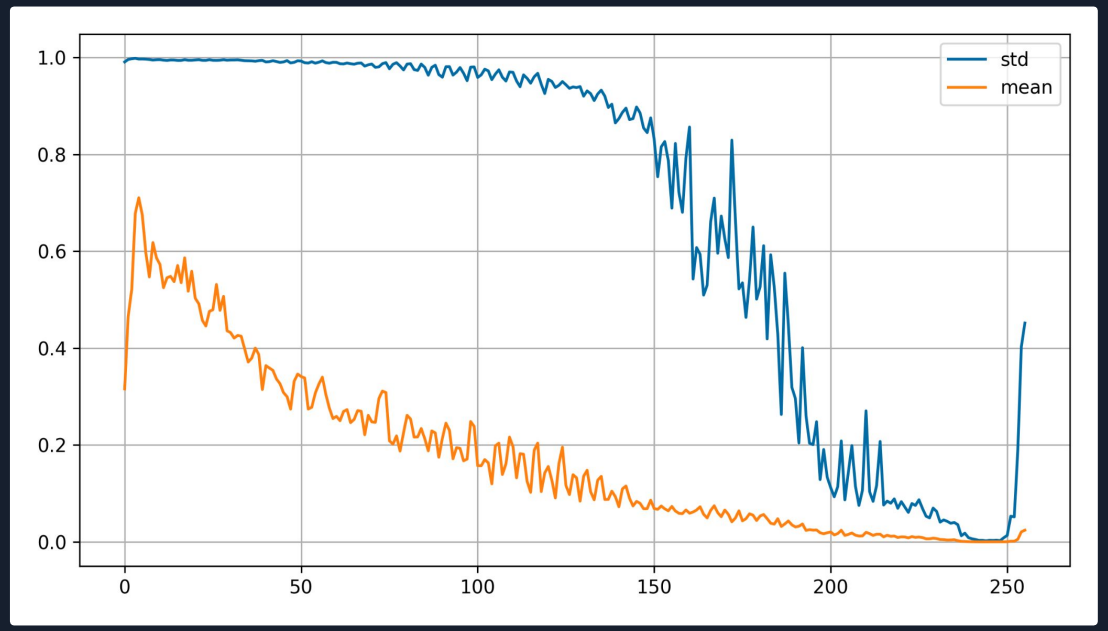
# 7

# Appendices



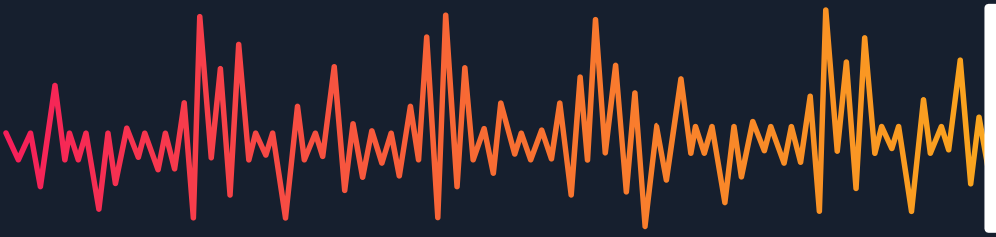
# Mel Spectrogram Normalization



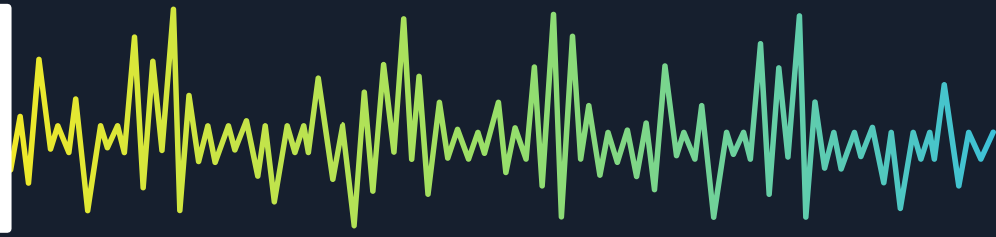




Before Normalization

After Normalization



$$\frac{\log_{10}(1 + X/\sigma_X)}{\sigma_{\log_{10}(f)}}$$

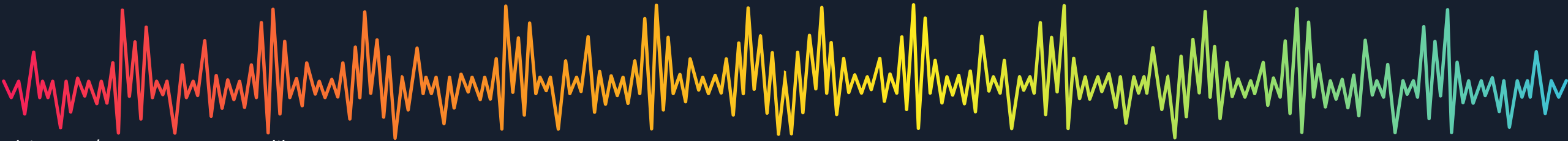


# Audio Reconstruction

**Algorithm 1** Griffin-Lim algorithm (GLA)

**Fix** the initial phase $\angle c_0$
**Initialize** $c_0 = s \cdot e^{i \angle c_0}$
**Iterate** for $n = 1, 2, \ldots$
$c_n = P_{c_1}\left(P_{c_2}(c_{n-1})\right)$
**Until convergence**
$x^* = \mathbf{G}^\dagger c_n$

$$P_{c_1}(c) = \mathbf{G}\mathbf{G}^\dagger c$$
$$P_{c_2}(c) = s \cdot e^{i \angle c}.$$



doi:10.1109/WASPAA.2013.6701851, librosa.org