XIII JORNADA DE JÓVENES INVESTIGADORES/AS DEL I3A

ROBOTICS, COMPUTER VISION & ARTIFICIAL INTELLIGENCE GROUP

# GPianoroll: a Deep Learning System with Human Feedback for Music Generation

Miguel Marcos  Lorenzo Mur  Rubén Martinez-Cantin

## Motivation and challenges

Could we use deep learning generative models to generate our perfect song? Yes! If we could solve these issues:

- **No eye for detail**

Deep learning models are trained to generalize, but at the cost of tiny details. **Generating one specific sample is tough.**

*"It's not right, but it's okay"* - Whitney Houston

- **Not easy to scale**

Music generation is a **high-dimensional scenario.** Several instruments can play several notes simultaneously.

*"Why'd you have to go and make things so complicated?"* - Avril Lavigne

- **No taste**

The quality of a song depends on the listener, so **we cannot know how "good" a song is** before playing it.

*"How bad can I possibly be?"* - from Dr Seuss' *The Lorax*

- **No time to waste**

**Evaluation is sequential** (you can't listen to two songs at once) and a person's **taste is fleeting**, so it should be time efficient.

*"Don't keep me waitin' when I'm in the mood"* - Glenn Miller & his orchestra

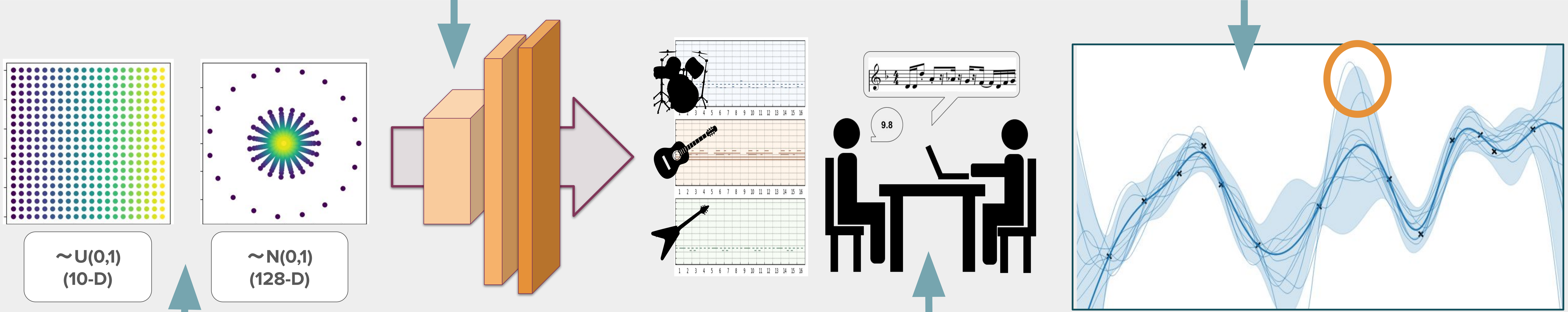
## Our proposal

### Music Composition model

We train a custom version of the MuseGAN[1] neural network. Gaussian noise comes in, Drums, Guitar and Bass come out! The output is a piano roll: a MIDI-like 3D matrix of instrument, time and pitch. We can then convert it to audio and play it using MIDI libraries.

### Bayesian Optimization

Global, sample efficient, function agnostic and accounting for noise. We build a surrogate model for our user's taste as a Gaussian Process, using the BayesOpt[2] library. We compute an acquisition function over it to find where a potential maximum is, query there next and add it to the model. We repeat until we run out of iterations, and get the historic best.




### Dimensionality reduction

We apply Random Embeddings[3] and the Box-Muller transform to a Uniform input. We only need to optimize 10 dimensions, instead of our model's 128. We can also map a subspace of the model's input to a single point, saving time.

### Human interaction

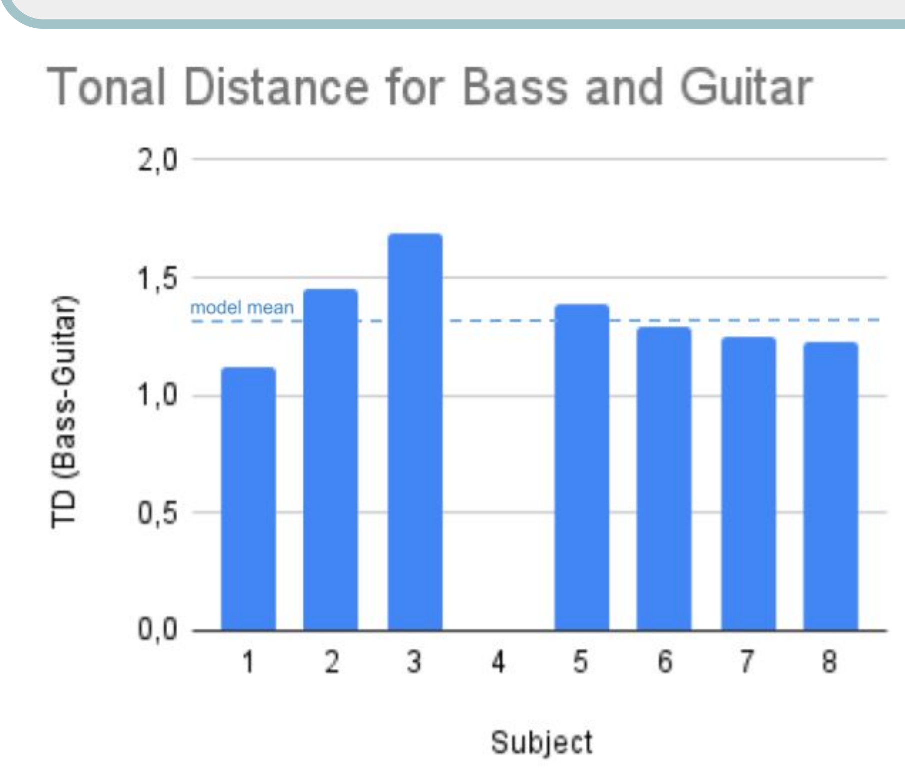
We play the song for the user and ask them to grade the generated pieces, from 0 to 10. Instead of having an explicit function for human taste, we will model it implicitly with samples. And we can do so without retraining the composition model!

## Experiments and Results

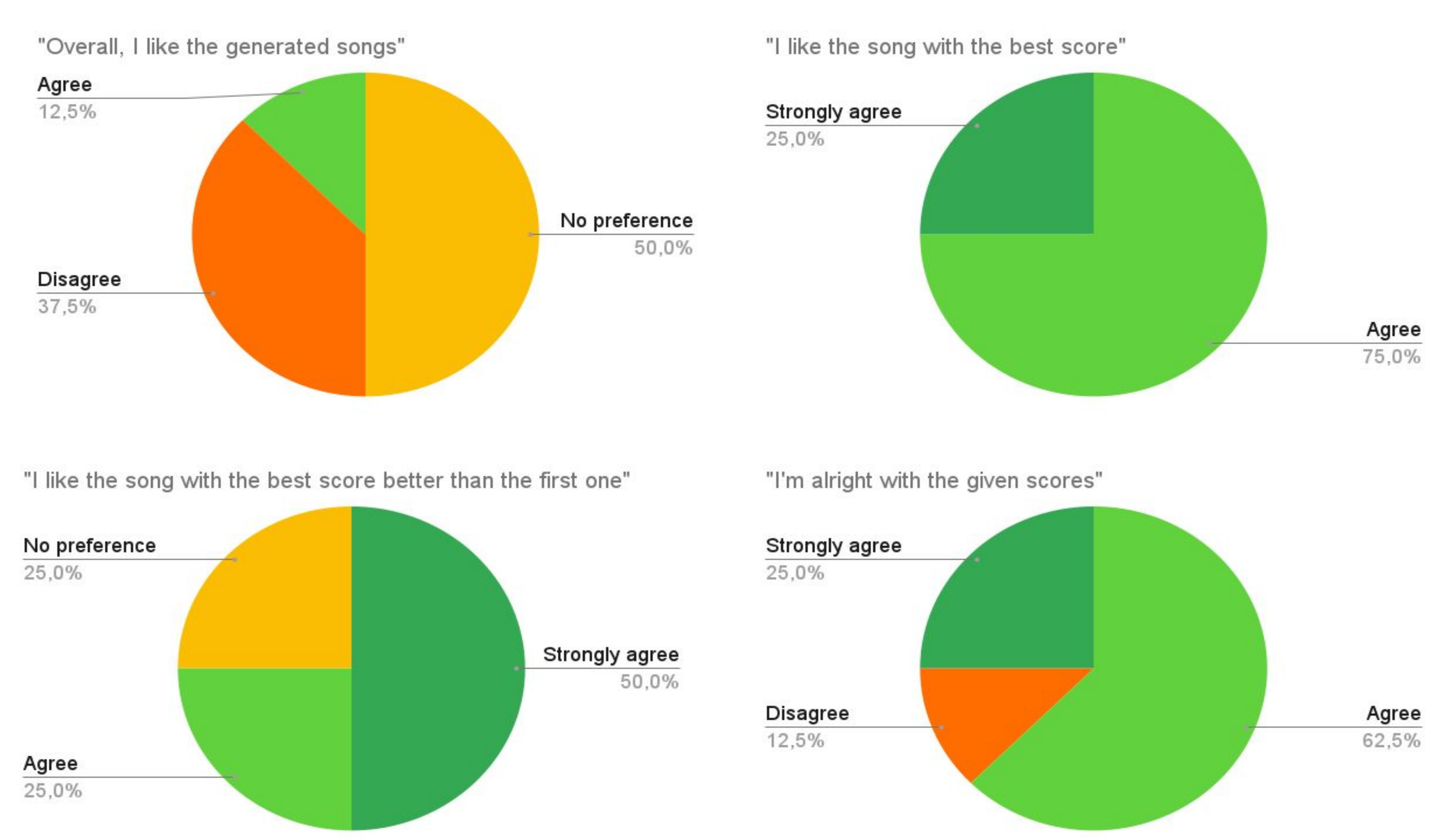
- **8 volunteers:** 64 tries to find their favorite track among the possible generations, for a total duration of 30 minutes each.
- We gathered metrics for the generated pieces. **The final pieces showed great variation among different users.**
- We surveyed the participants about their experience and opinion on the compositions. **All volunteers agreed the selected song was good**. Answers were, overall, positive.

## Conclusions

- **Model customization is a necessity**. Metrics show people like very different music, so each person needs generative models to behave differently according to their taste.
- **Bayesian Optimization is effective for music generation**. We were able to find a good song for every volunteer, efficiently and with a single generative model for all of them, saving time and resources.








**REFERENCES**

1. DONG, H.W. and HSIAO, W.Y. and YANG, L.C. and YANG Y.H. MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. *In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*. New Orleans, USA. 2018.
2. MARTÍNEZ-CANTÍN, R. BayesOpt: A Bayesian Optimization Library for Nonlinear Optimization, Experimental Design and Bandits. *Journal of Machine Learning Research*, 2014, 15(11, pp. 3735-3739.
3. WANG, Z. and ZOGHI, M. and HUTTER, F. and MATHESON, D. and FREITAS, N. Bayesian optimization in a billion dimensions via random embeddings. *IJCAI International Joint Conference on Artificial Intelligence*, 2013.

**Listen to the samples at our GitHub page!**
https://mikceroese.github.io/GPianoroll/