Generating new music with deep probabilistic models

Valentin Vignal

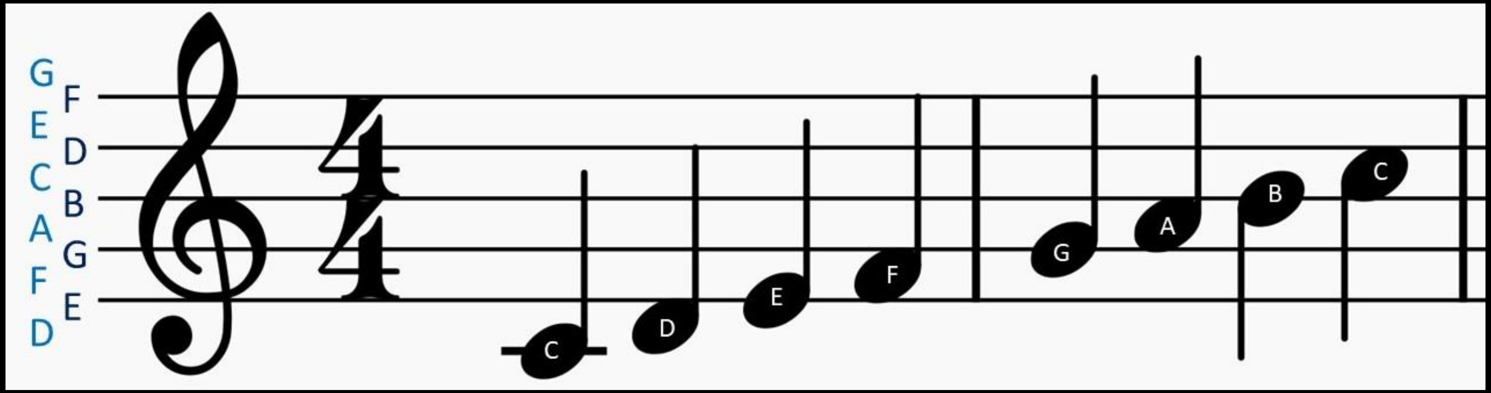
List of contents

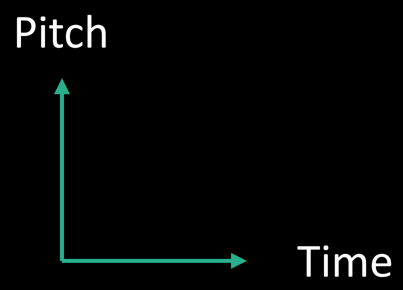
- Background knowledge
- Related works
- Contribution
- Results
- Band Player
- Questions

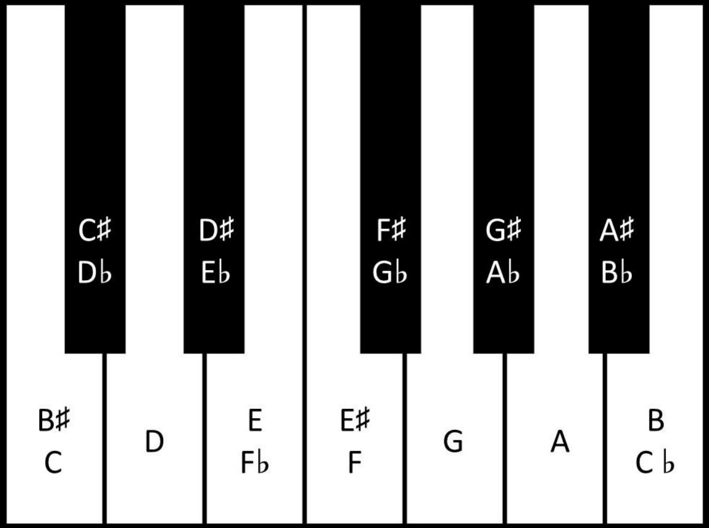
Background knowledge

- MIDI format
- Music theory
- Physical properties

Musical Stave





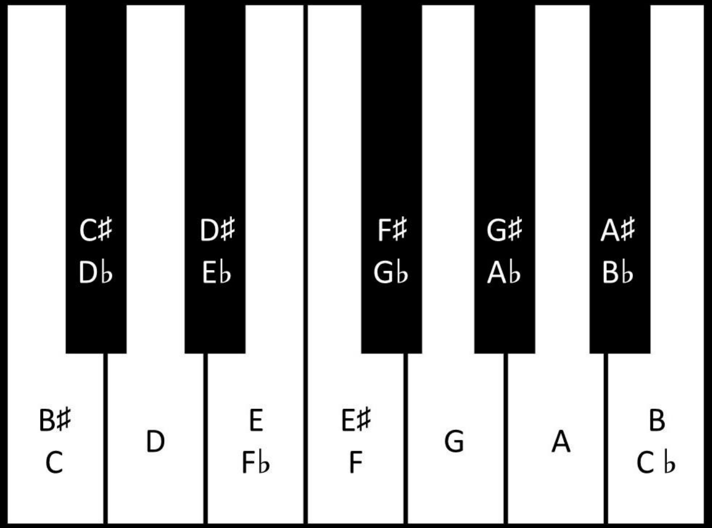


Note shape	Name	Note length
𝅝	Whole note	4 beats
𝅗𝅥	Half note	2 beats
♩	Quarter note	1 beat
♪	Eighth note	½ beat
𝅘𝅥𝅯	Sixteenth note	¼ beat

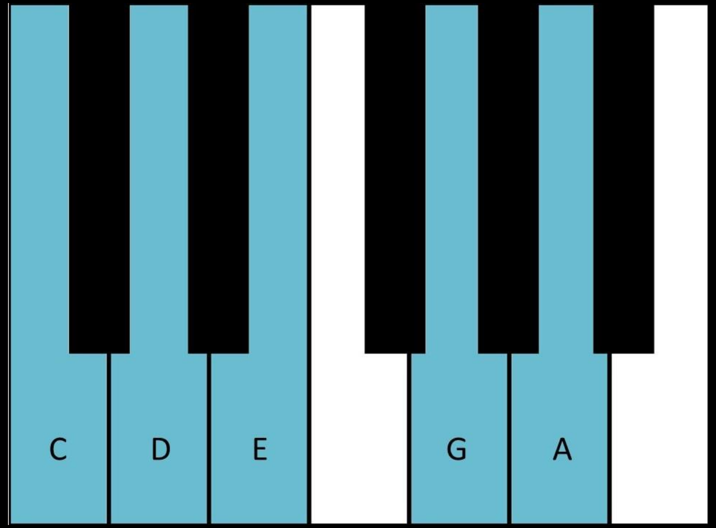
Musical Scale

A scale is a set of notes.

The white keys of a piano form the C major scale



C major pentatonic



MIDI Format

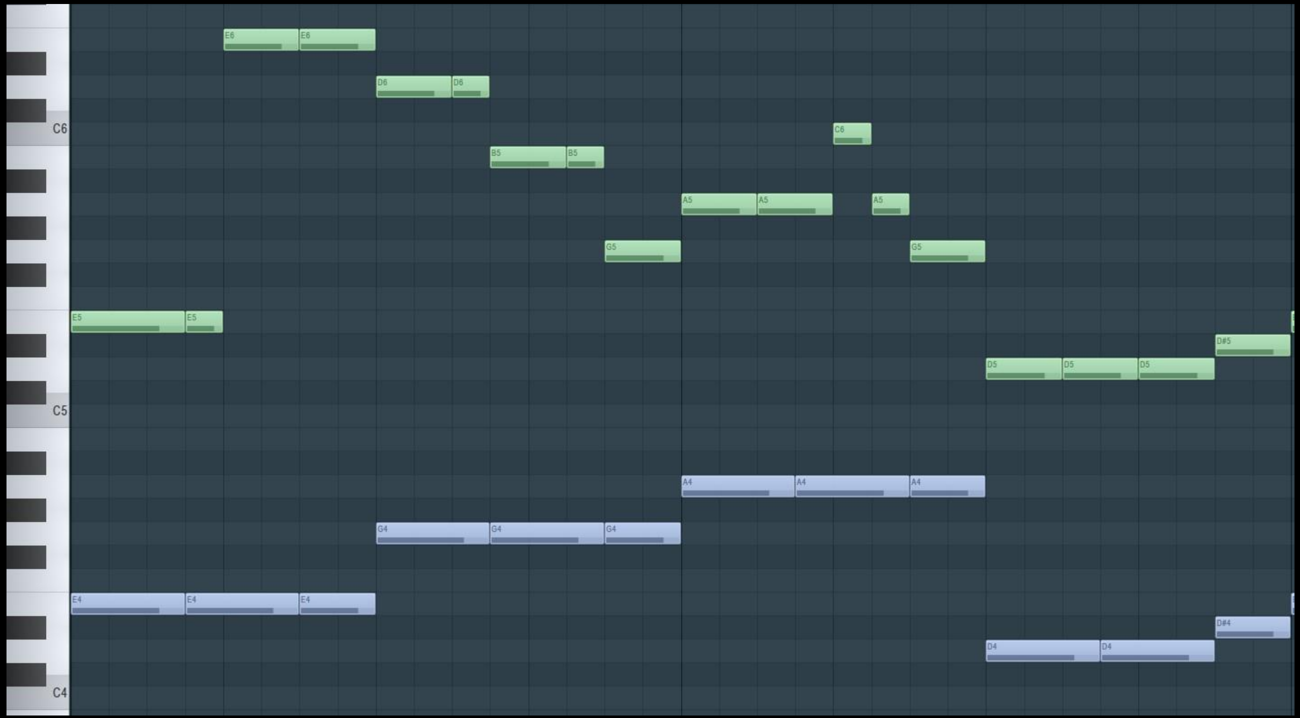
It is a protocol to carry musical events between musical devices or software

- *"Note on"* event:
  $< NoteOn, channel, pitch, velocity >$
- *"Note off"* event:
  $< NoteOff, channel, pitch, velocity >$

- $channel \in [0, 15]$
- $pitch \in [0, 127]$
- $velocity \in [0, 127]$

MIDI format

Pianoroll



Text

E5, _ , _ , E5, E6, _ , E6, _ , D6, _ , D6, B5, _ , B5, G5, _ , …
E4, _ , _ , E4, _ , _ , E4, _ , G4, _ , _ , G4, _ , _ , G4, _ , …

Harmonics

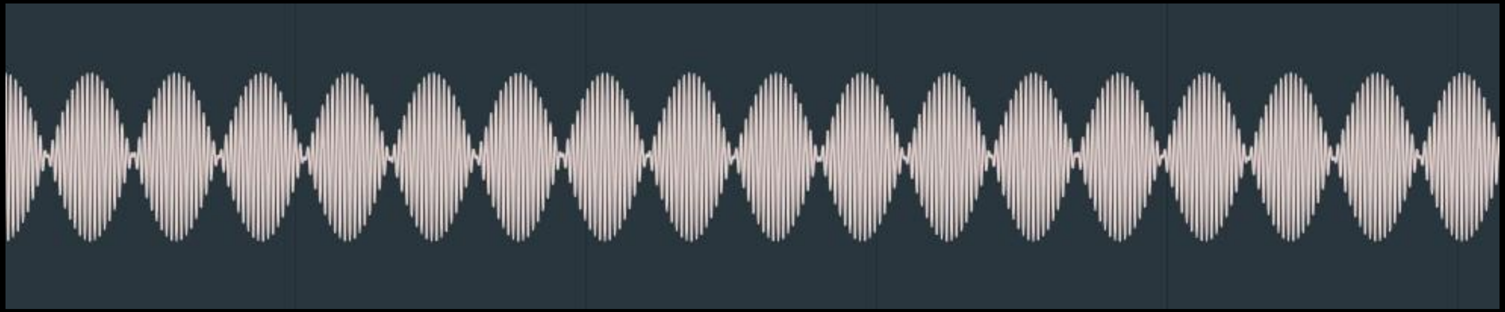
A musical sound is composed of harmonics

$$s(t) = \sum_{n=1}^{\infty} \alpha_n \sin(nft + \phi_n)$$

Harmonic number	Frequency (Hz)	Note Name	Musical Interval
1	440	A4	Unison
2	880	A5	Octave
3	1320	E5	Fifth
4	1760	A6	Octave
5	2200	C#6	Major Third
6	2640	E6	Fifth

Resonance phenomena

Two sounds with a close frequency will generate a resonance phenomena



Resonance phenomena from a B4 and a C5

$$\cos(f) + \cos(f + \delta f) = 2\cos\left(\frac{2f+\delta f}{2}\right)\cos\left(\frac{\delta f}{2}\right)$$

Intervals

"Acceptable" intervals

Interval	Reason	Example with A4
Octave	Contained in tonic's Harmonics	A5
Minor Third	Share harmonics (A and C share E in their harmonics)	C6
Major Third	Contained in tonic's harmonics	C#6
Fifth	Contained in tonic's harmonics	E6

"Not acceptable" intervals

Interval	Reason	Example with A4
Semitone	Resonance phenomena between fundamental frequencies	A#5
Tone	Resonance phenomena between fundamental frequencies	B5
Tritone	Resonance phenomena between fundamental frequency and third harmonic (D and D#)	D#6

Related works

- Objective
  - Generate musical parts (DeepBot)
  - Harmonize (DeepBach, DoodleBach)
- Data Representation
  - MIDI format translated to text
  - Songs transposed in C major
- Architectures Users
  - NADE (BachDoodle)
  - AutoEncoder (most common) (DeepBach, BachBot)
  - VAE
  - GAN
  - Transformers

Contribution

- Objectives
- Data representation
- RMVAE (Recurrent Multimodal Variational AutoEncoder)
  - Global architecture
  - RPoE
  - Encoder/Decoder Architecture
  - Activation function
- Custom loss functions
  - Scale
  - Rhythm
  - Harmony

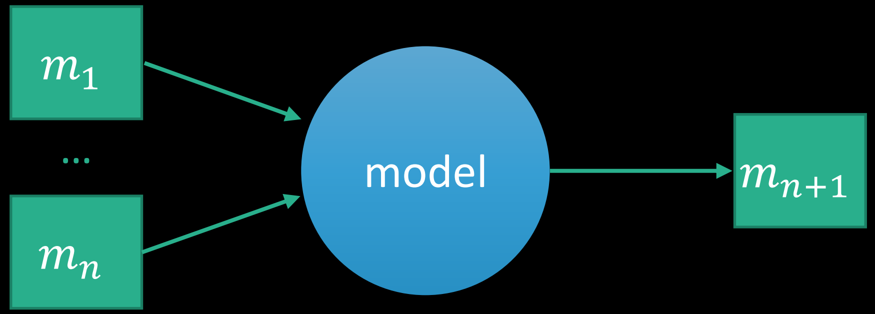
Objectives

- Generate a music with several musical parts
- Create an accompaniment from a melody
- Create a melody from an accompaniment
- Create musical parts from other musical parts

Data representation

- I consider binary rhythm only (1 beat is divided in 2 equal beats)
- The smallest notes I consider are the Sixteenth notes

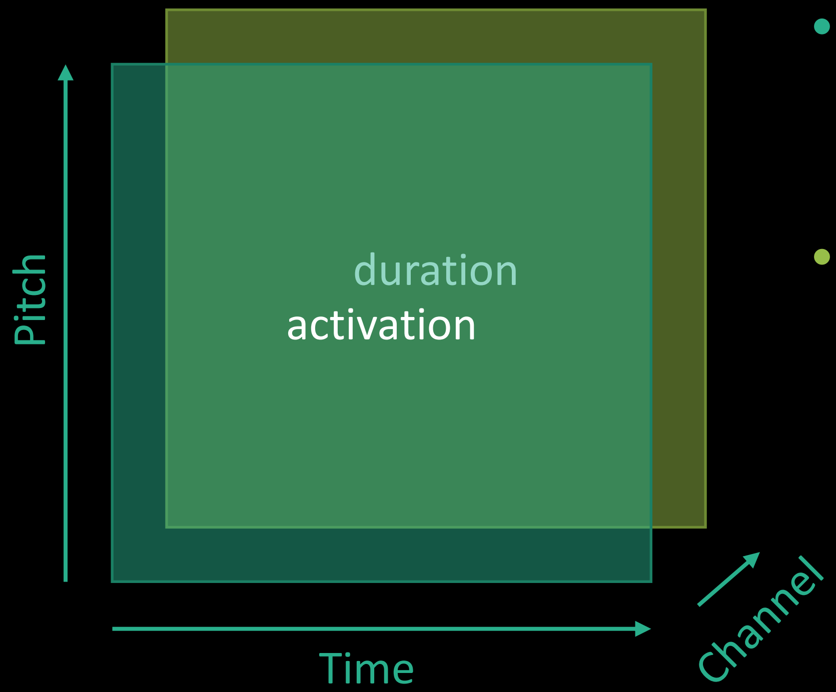
- A *“step”* is a measure
  - There are 16 Sixteenth notes division in a measure
  - From a fixed number of measures, the model will predict the next one



- 128 different pitches
- A tensor representing a measure: `(16, 128, channels)`

Data Representation – Polyphonic Music

`(16, 128, channels=2)`



- Activation channel:
  - Sigmoid activation function
  - Binary cross-entropy loss
- Duration channel:
  - ReLU activation function
  - Mean squared error loss

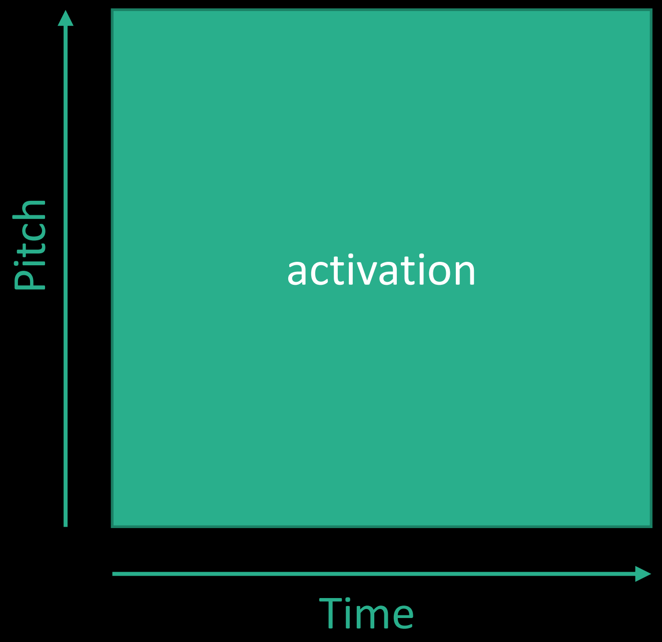
$activation \geq 0.5 \Rightarrow$ a note is played
$activation < 0.5 \Rightarrow$ no note is played

$duration = length_{note}$
in number of sixteenth notes

$duration = 1 \Rightarrow$
$duration = 2 \Rightarrow$
$duration = 4 \Rightarrow$
$duration = 8 \Rightarrow$
$duration = 16 \Rightarrow$

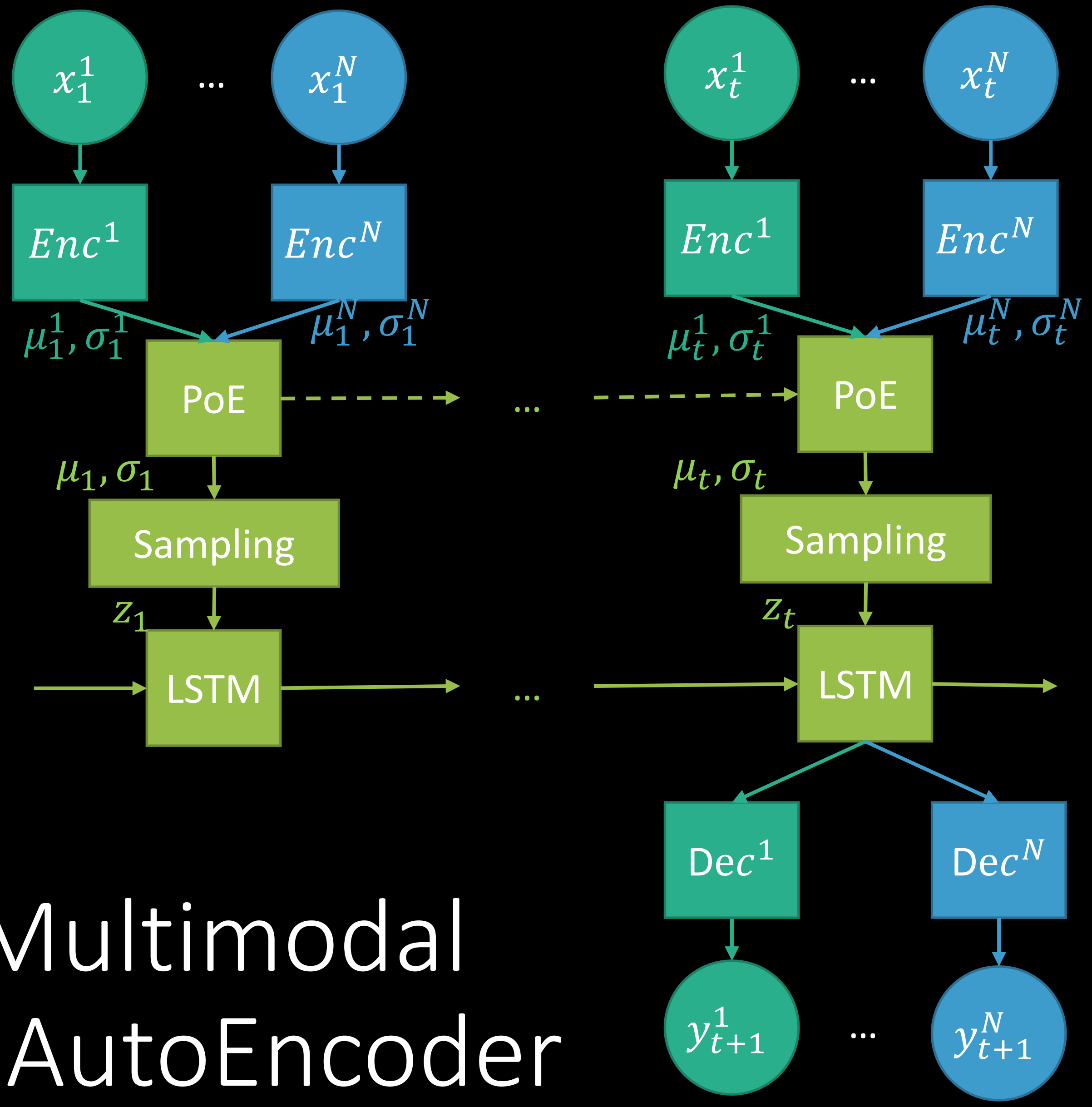
Data Representation –Monophonic Music

`(16, 128 + 1, channels=1)`



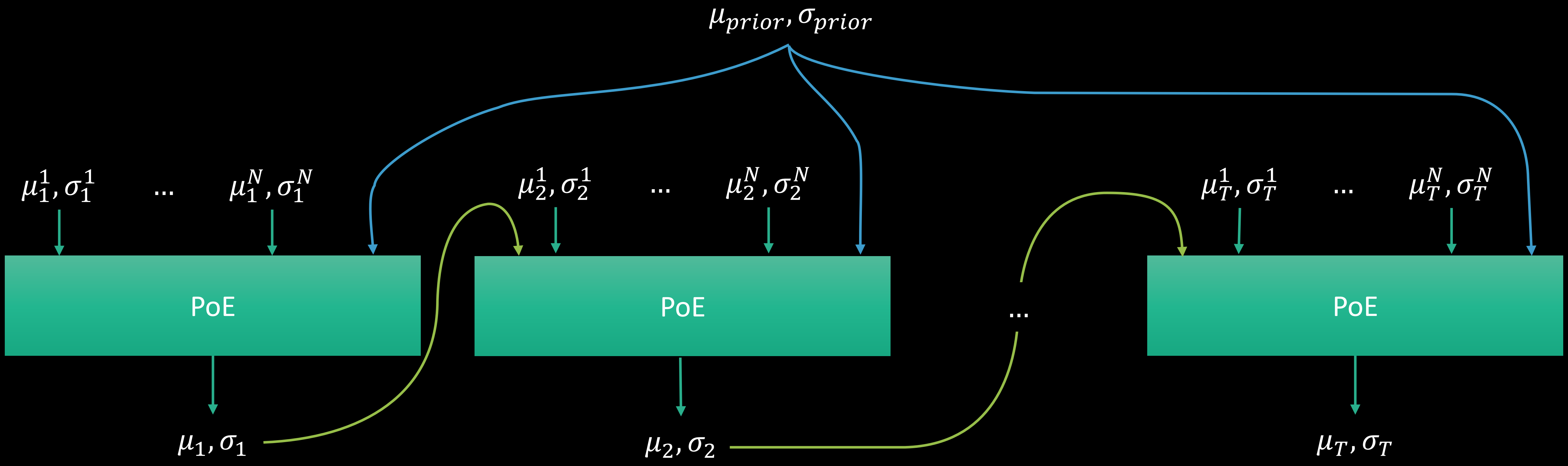
- Activation channel:
  - Softmax activation function
  - Categorical cross-entropy loss
  - Extra $note_{continue}$ $\Rightarrow$ continue the previous note

Argmax of $activation$ is played



Recurrent Multimodal Variational AutoEncoder

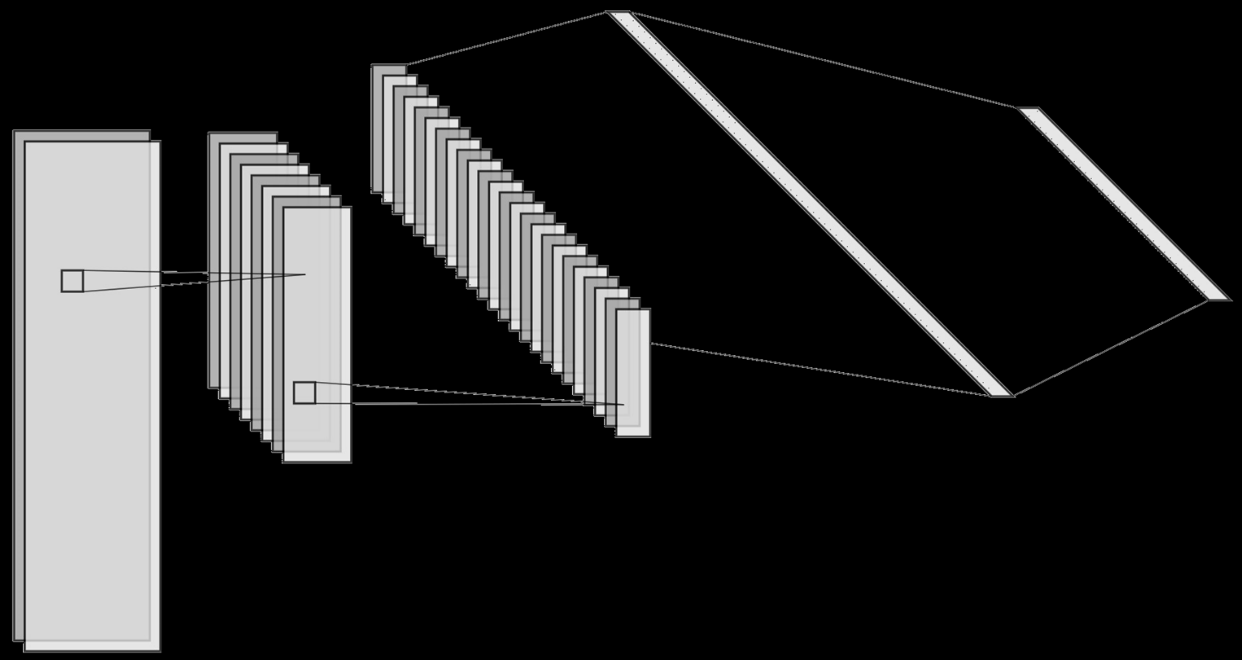
Recurrent Product of Experts



Convolutional Encoder/Decoder

Input tensor : `(16, 128, channels)`

- Encoder
  - Convolutional layers
  - Fully Connected layers
- Decoder
  - Fully Connected layers
  - Transposed Convolutional layers
  - 1 Fully Connected layer

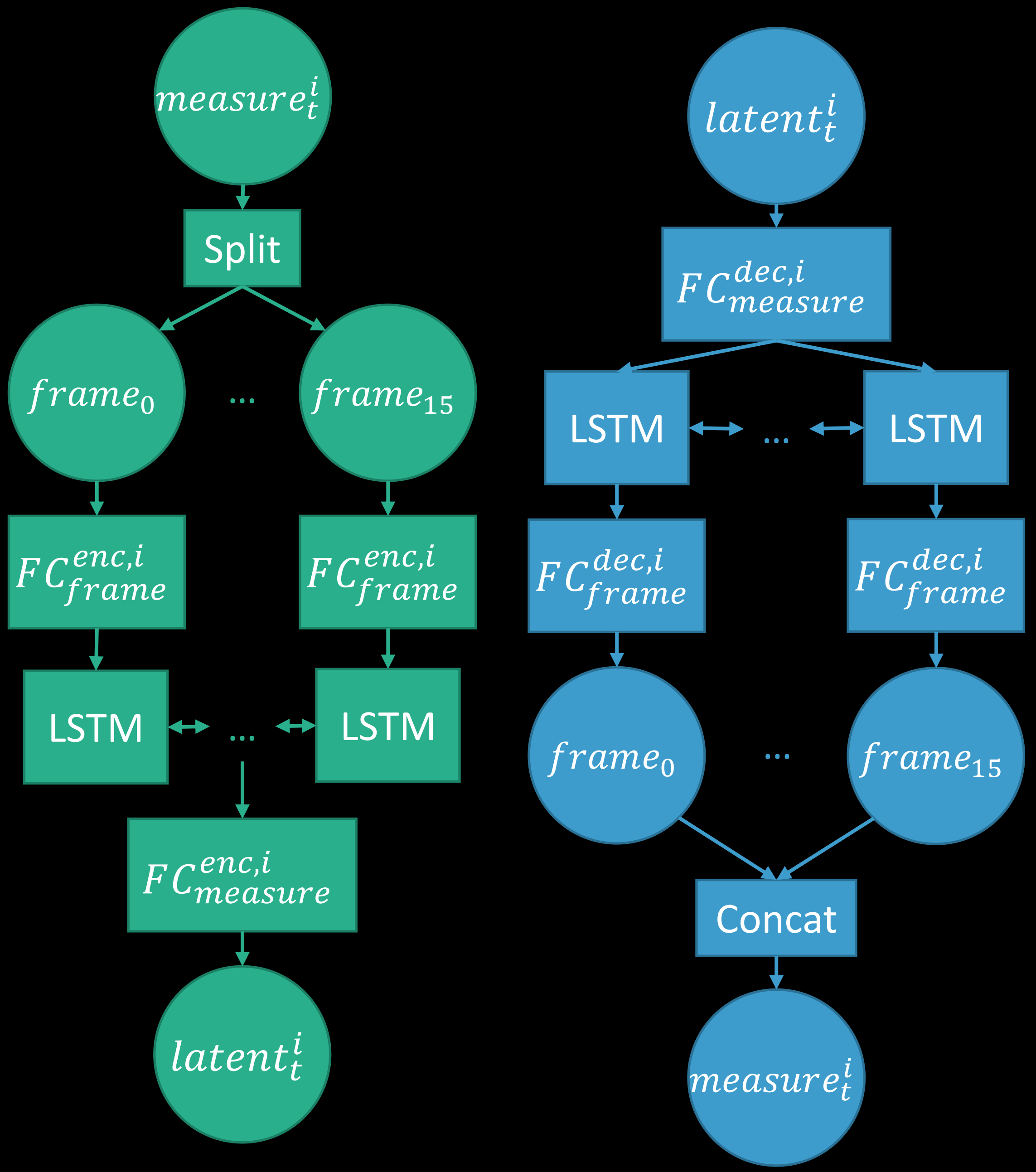


Convolutional filter : `(5, 5)`

Recurrent Encoder/Decoder

Input tensor : `(16, 128)`

- Encoder
  1. Splits the measure into several frames
  2. Encodes the frames with fully connected layers
  3. Extracts the latent space with bidirectional LSTMs
  4. Encodes the latent space with fully connected layers
- Decoder
  1. Decodes the latent space with fully connected layers
  2. Generates the encoded frames with bidirectional LSTMs
  3. Decodes the frames with fully connected layers
  4. Concatenates the frames to create the measure



Activation and Loss function

- Softmax activation
- Categorical crossentropy loss

Activation and Loss function

- $note_{continue}$
  - Sigmoid activation
  - Binary crossentropy
- *“real”* notes
  - Softmax activation
  - Categorical crossentropy

Prior knowledge with loss function

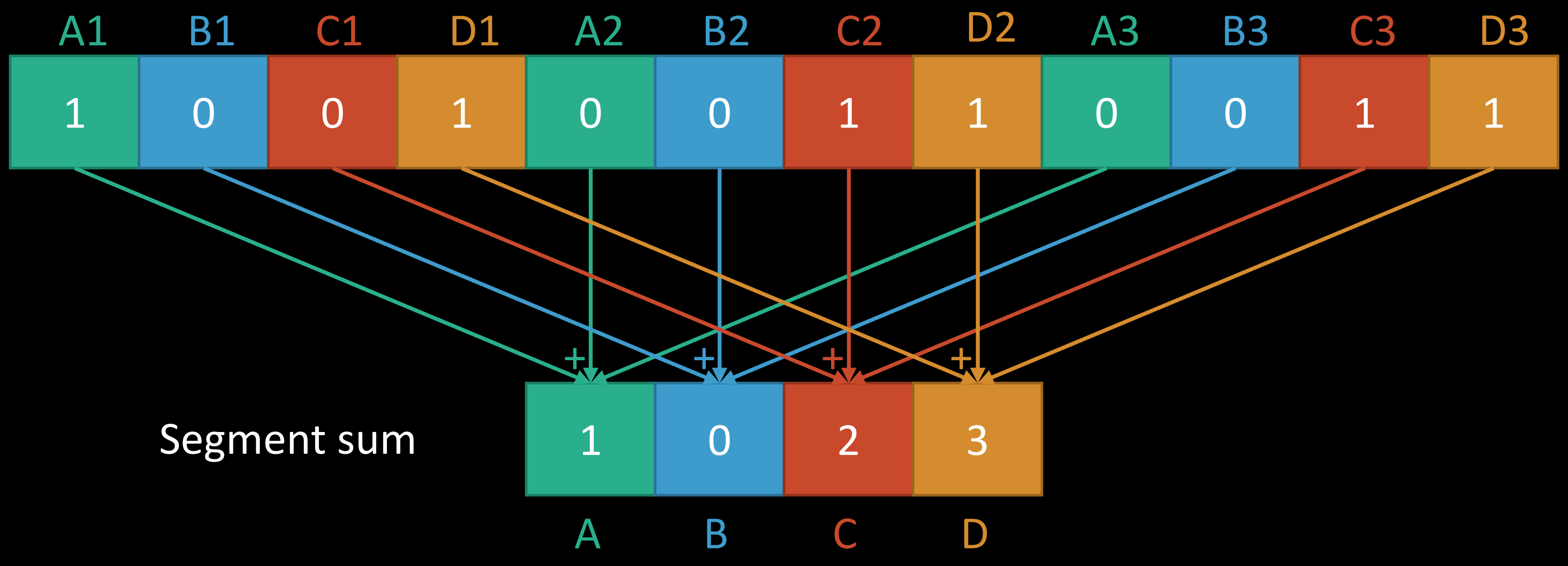
- Helps the model to do *“acceptable mistakes”*
  - Gives a reward (decrease the loss)
- Prevents the model to do *“unacceptable mistakes”*
  - Gives a penalty (increase the loss)

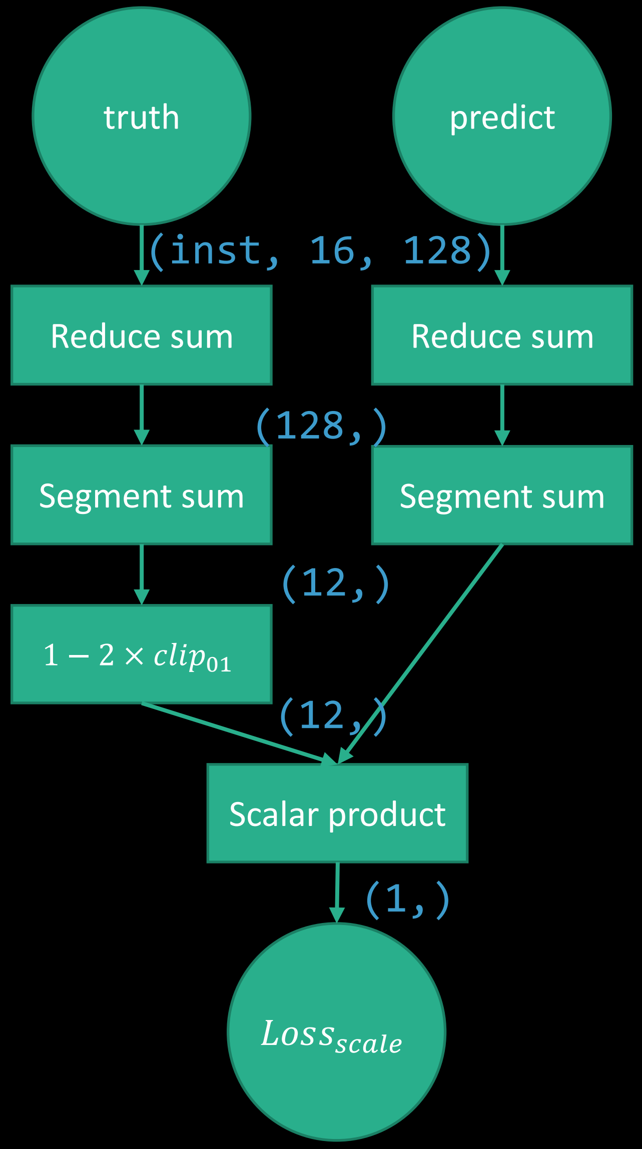
```
var loss;
var reward > 0;
var penalty > 0;
if (soundsGood(note_predicted)) {
    loss -= reward;
} else {
    loss += penalty;
}
```

Scale loss

Reconstructs the local scale with the notes present in the *truth* tensor

Incites the model to generate note present in the truth tensor

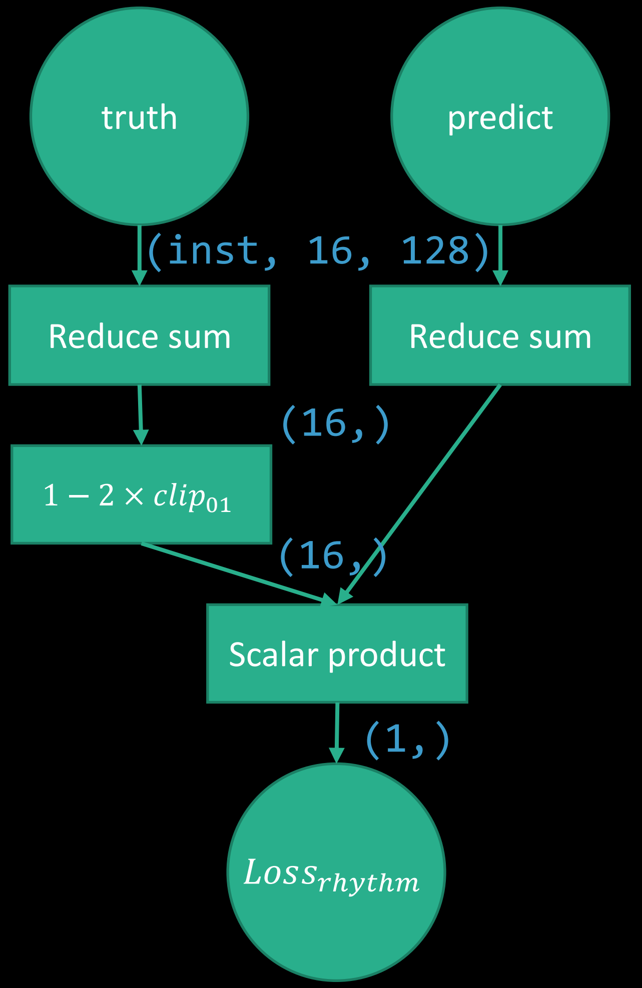




Rhythm loss

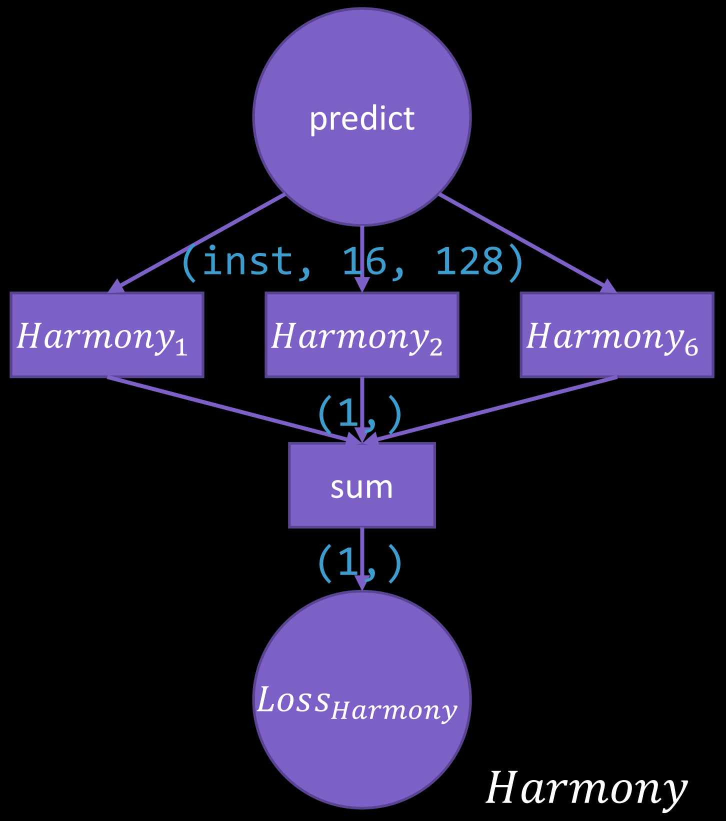
Reconstructs the local rhythm with the notes present in the *truth* tensor

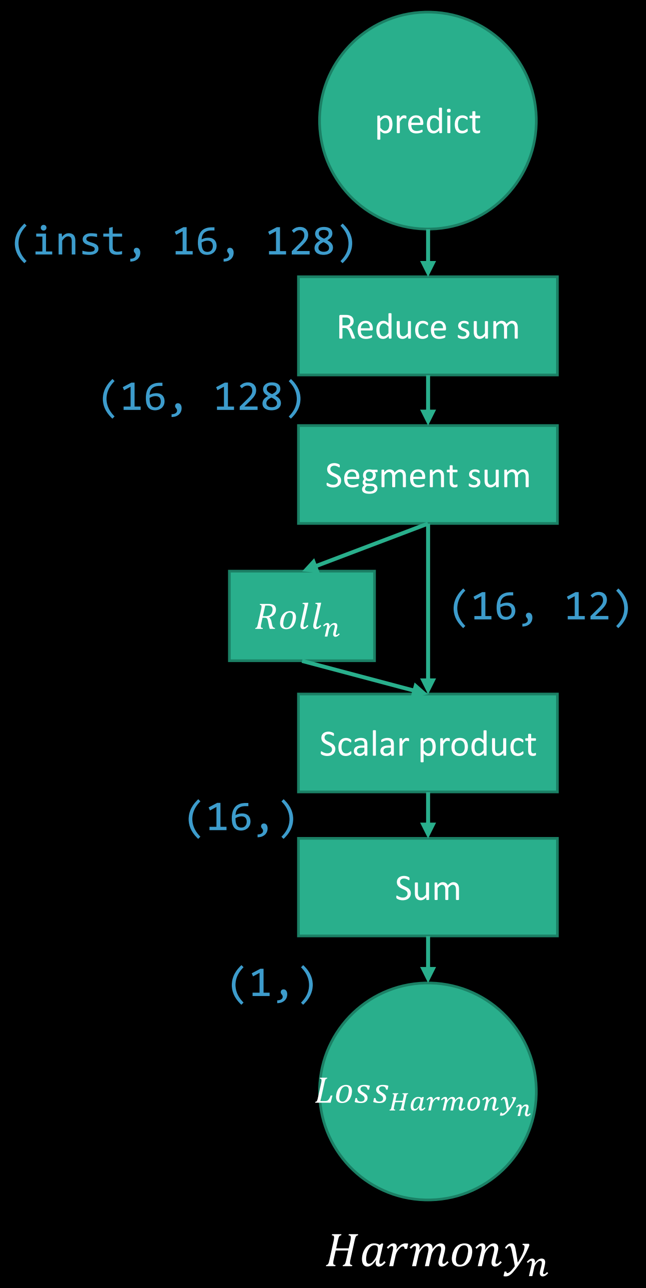
Incites the model to generate notes when a note is played in the *truth* tensor

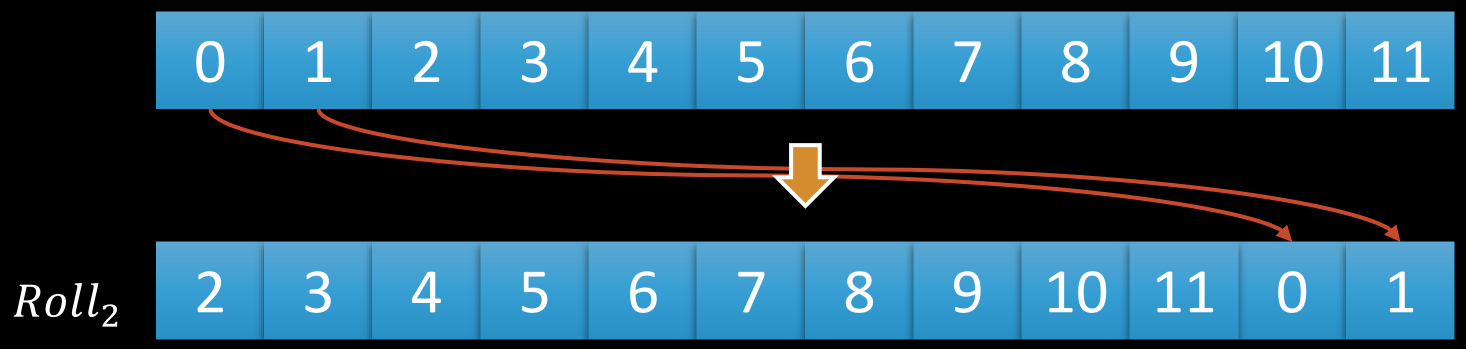


Harmony loss

- 3 intervals are dissonant on their own:
    - Semitone
    - Tone
    - Tritone
- Harmony prevents the model to play these intervals







Results

- Implemented tasks
- Experiments
  - Transposed data
  - Model size
  - Scale and Rhythm losses
  - Harmony loss
  - RPoE layer

Generate

The generate task takes several measures and generate the next one

The accuracy of the generate task is 0.8

```
var seed; // list of measures
var n; // number of measures to generate
function generate(seed, n) {
    var seedLength = seed.length;
    var generated = seed;
    for (k=1; k<=n; k++) {
        var input = genetared[-seedLength:];
        var output = model.predict(input);
        generated.push(output);
    }
    return generated;
}
```

Generate

The generate task takes several measures and generate the next one

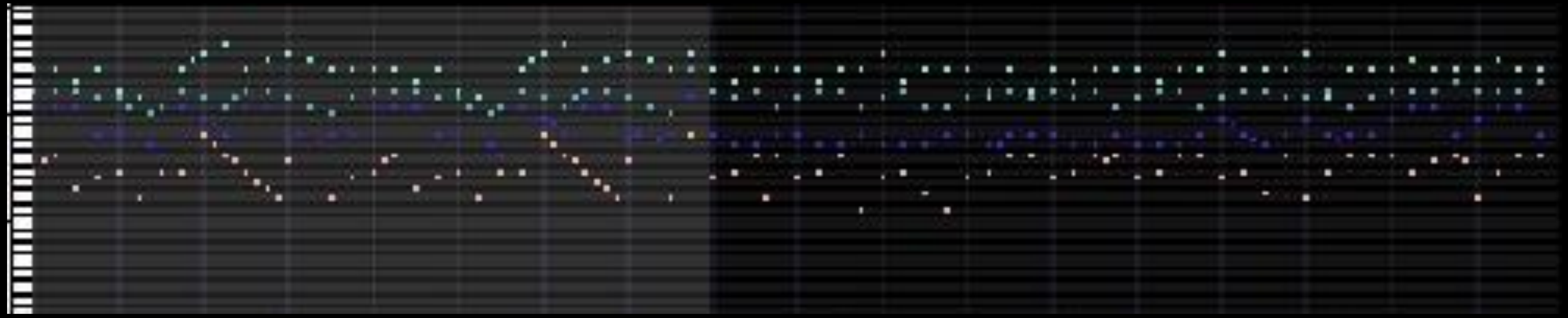
The accuracy of the generate task is 0.8

```
var seed; // list of measures
var n; // number of measures to generate
function generate(seed, n) {
    var seedLength = seed.length;
    var generated = seed;
    for (k=1; k<=n; k++) {
        var input = genetared[-seedLength:];
        var output = model.predict(input);
        generated.push(output);
    }
    return generated;
}
```

Generate – Convolutional and Recurrent encoder/decoder

Convolutional

Recurrent



Fill

Re-generates a voice of a song

The accuracy of the fill task is 0.7

```
var song; // list of measures
var instrument; // Instrument to replace
function fill(song, instrument) {
    song.deleteInstrument(instrument);
    var filled = song;
    for (k=0; k < song.length – seedLength; k++) {
        var input = genetared[k:k + seedLength];
        var output = model.predict(input);
        filled[k + seedLength, instrument] = output[instrument];
    }
    return filled;
}
```

Fill

Original

Replace first voice

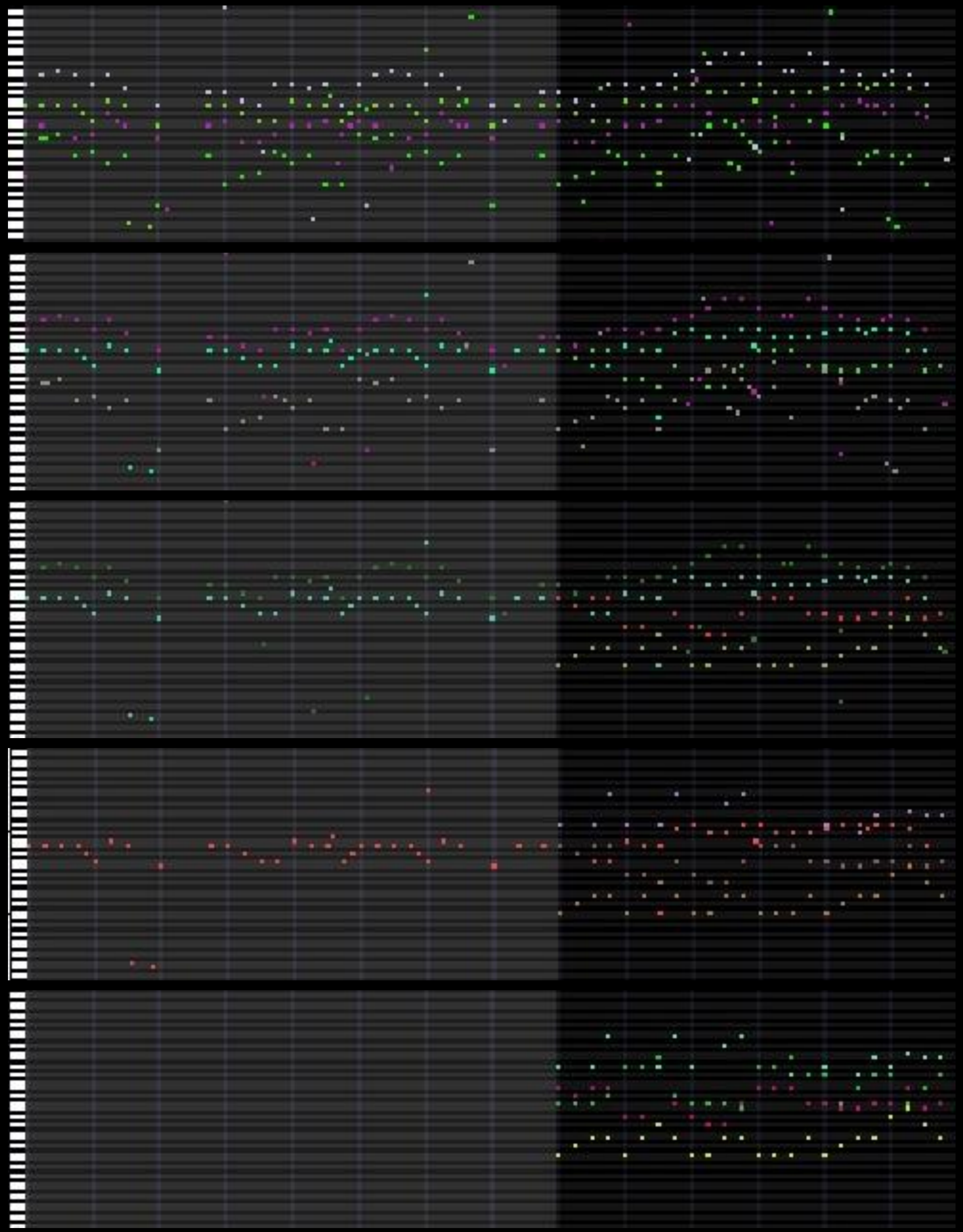
Redo

- Recreates a song by replacing one by one every voices

The accuracy of the redo task is 0.6

```
var song; // list of measures
function redo(song) {
    var redone = song;
    for (k=0; k < nbInstruments; k++) {
        redone = fill(redone, k);
    }
    return redone;
}
```

Redo



Original

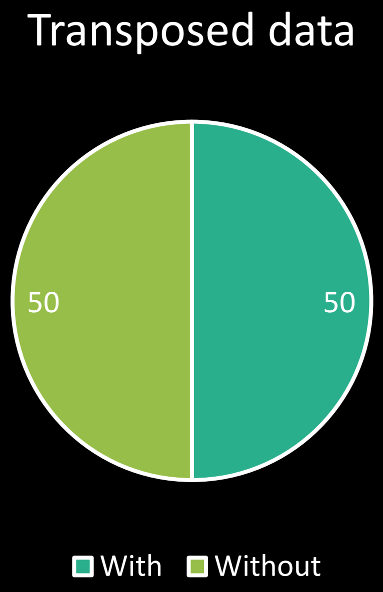
Step 1
Replace instrument 3

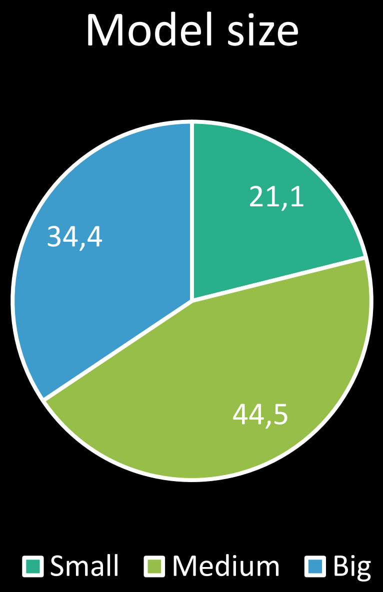
Step 2
Replace instrument 4

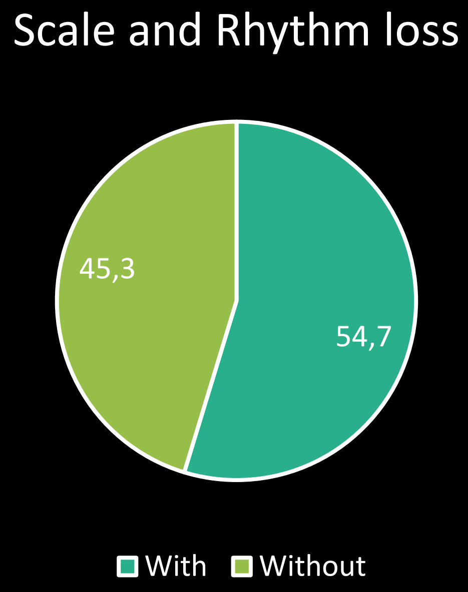
Step 3
Replace instrument 1

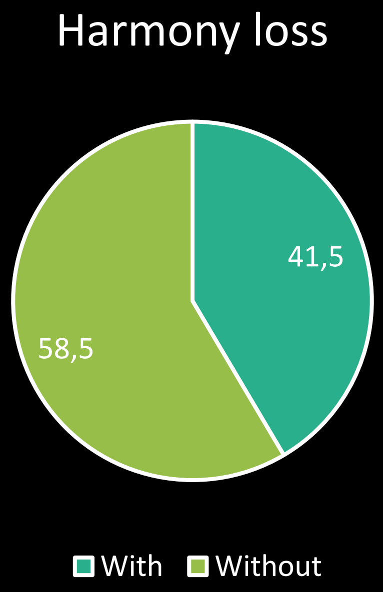
Step 4
Replace instrument 2

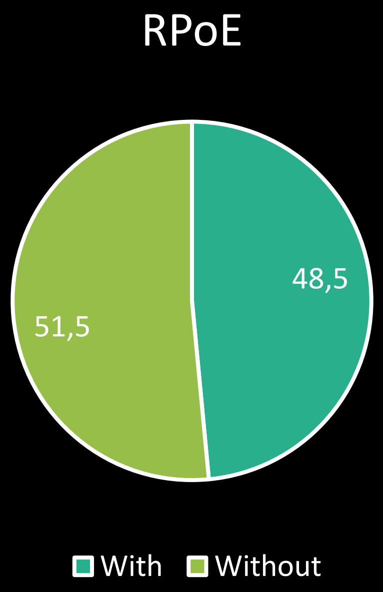
Experiments











Band Player

- Takes a trained model and plays with the user
  - The model had to be trained to predict the second next measure (for real time purpose)
  - It is able to be a real application because the neural network is only called once every measures

Conclusion

- The RMVAE is an architecture able to handle several tasks
- The results are poor in complexity and variations
- The different loss functions and the RPoE layer don’t help the training.

Future works:

- An exploration on most of the hyper parameter could help the model to perform better
- The scale loss can be improved by including some scale templates

References

- C. Doersch, “Tutorial on Variational Autoencoders,” arXiv:1606.05908 [cs, stat], Jun. 2016. [Online]. Available: http://arxiv.org/abs/1606.05908
- M. Wu and N. Goodman, “Multimodal Generative Models for Scalable Weakly-Supervised Learning,” in Advances in Neural Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 5575–5585. [Online]. Available: http://papers.nips.cc/paper/ 7801-multimodal-generative-models-for-scalable-weakly-supervised-learning. Pdf
- G. Hadjeres, F. Pachet, and F. Nielsen, “DeepBach: a Steerable Model for Bach Chorales Generation,” arXiv:1612.01010 [cs], Dec. 2016. [Online]. Available: http://arxiv.org/abs/1612.01010
- F. T. Liang, M. Gotham, M. Johnson, and J. Shotton, “Automatic Stylistic Composition of Bach Chorales with Deep LSTM,” in ISMIR, 2017.
- C.-Z. A. Huang, C. Hawthorne, A. Roberts, M. Dinculescu, J. Wexler, L. Hong, and J. Howcroft, “The Bach Doodle: Approachable music composition with machine learning at scale,” arXiv:1907.06637 [cs, eess, stat], Jul. 2019, arXiv: 1907.06637. [Online]. Available: http://arxiv.org/abs/1907.06637

Questions

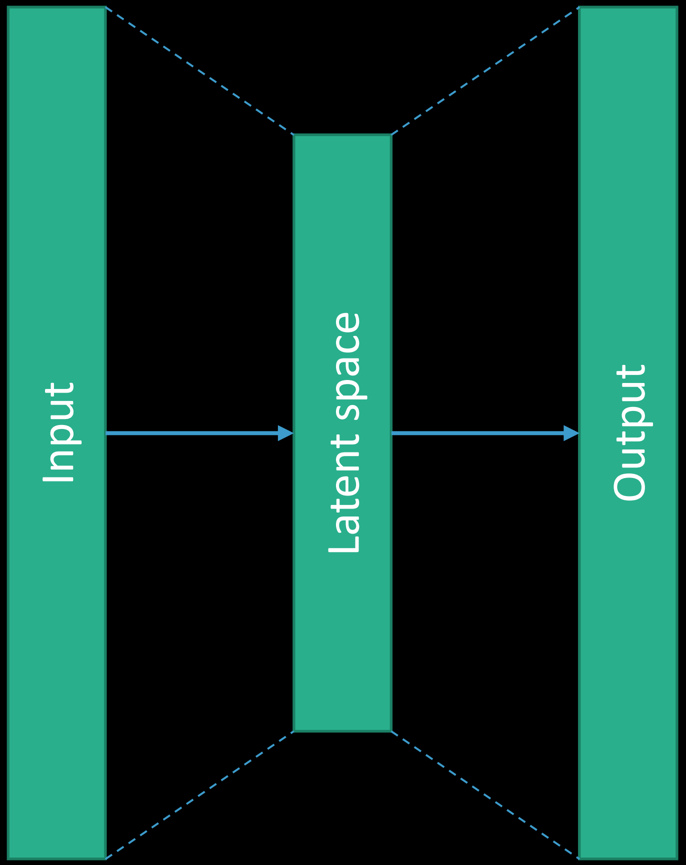
Thank you for your attention.

AutoEncoder

- Encoder
  - Takes the input
  - Encodes it in the latent space
- Decoder
  - Takes a point from the latent space
  - Reconstructs the output

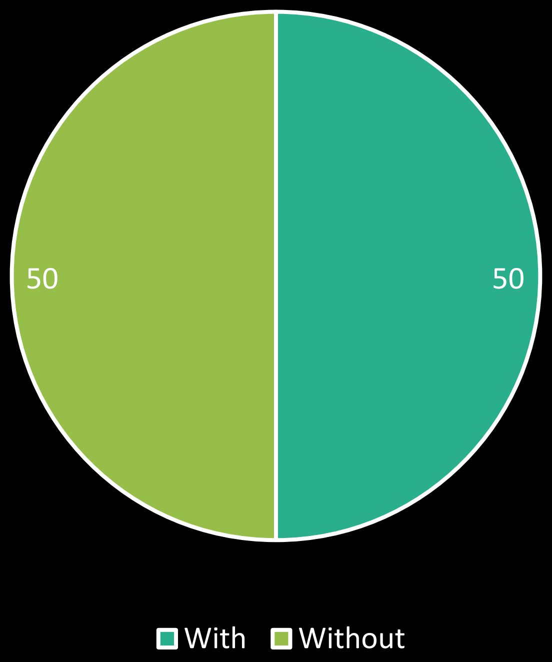
The AutoEncoder tries to reconstruct the input

$$|space_{latent}| \ll |space_{input}|$$



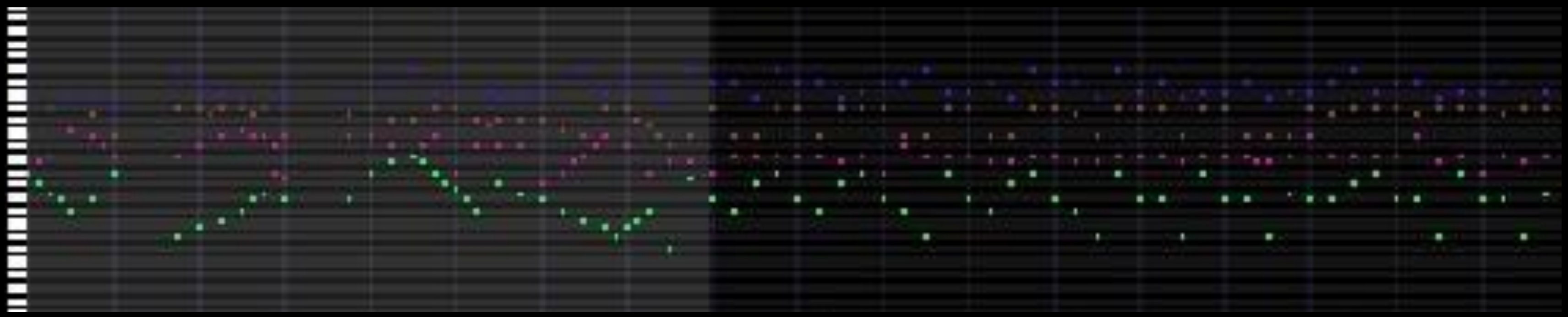
Transposed data

Transposed data



14 people answered the survey

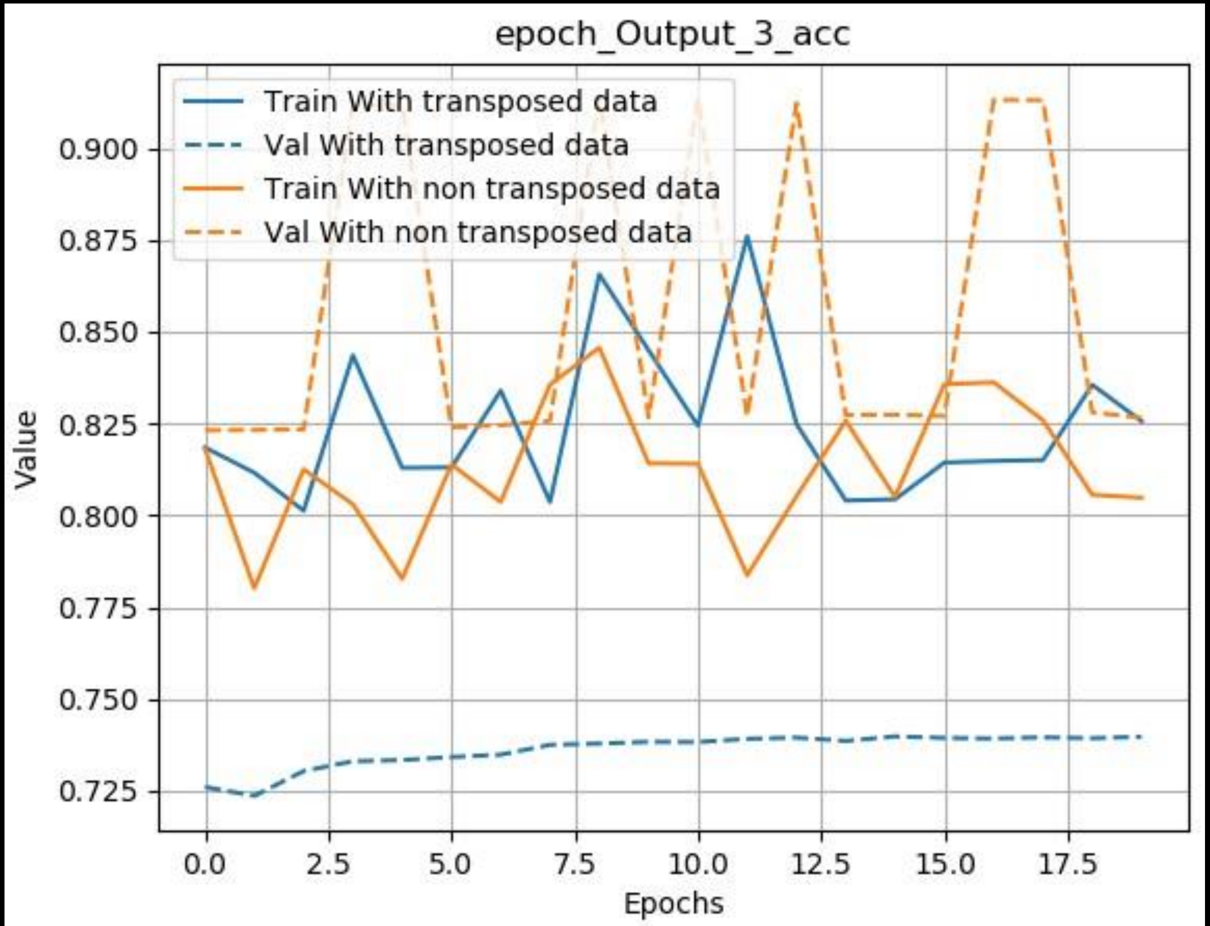
Without transposed data



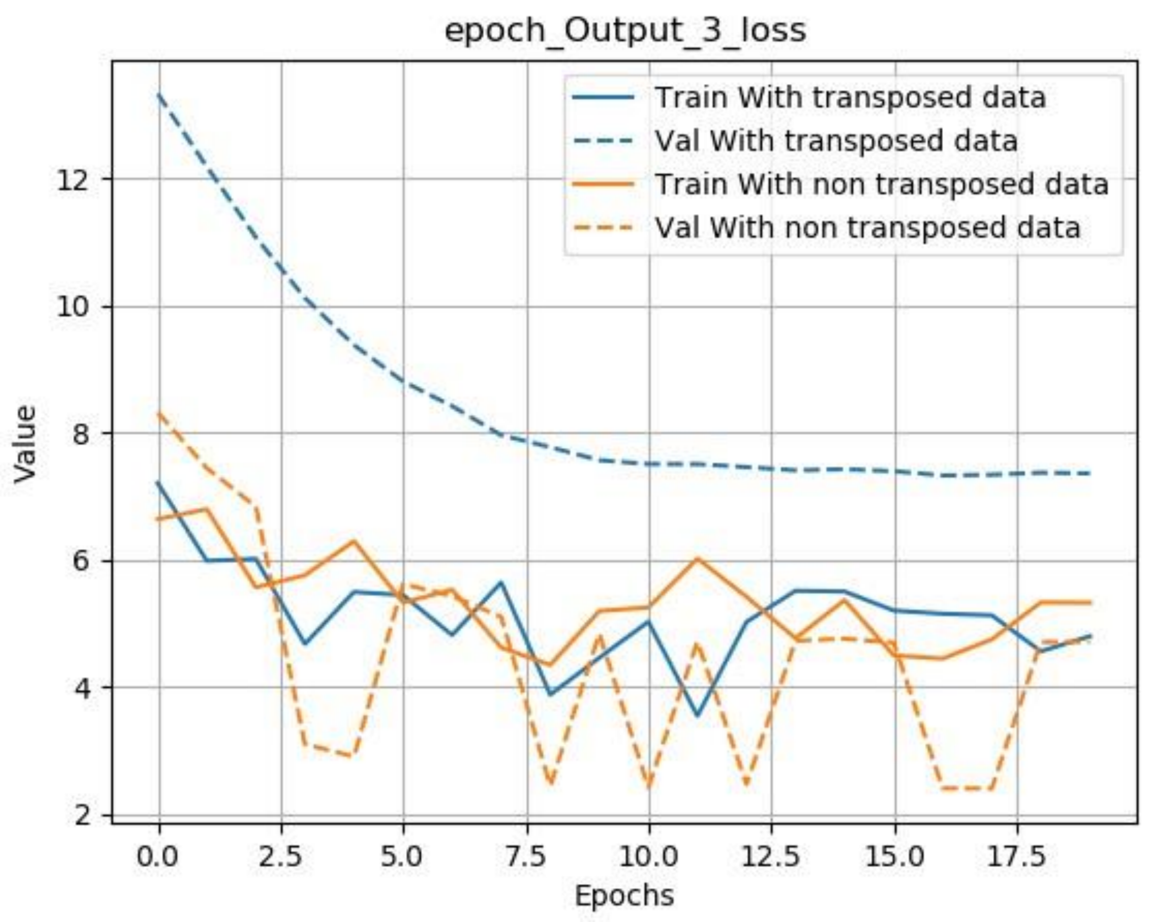
With transposed data

Transposed data

Output accuracy

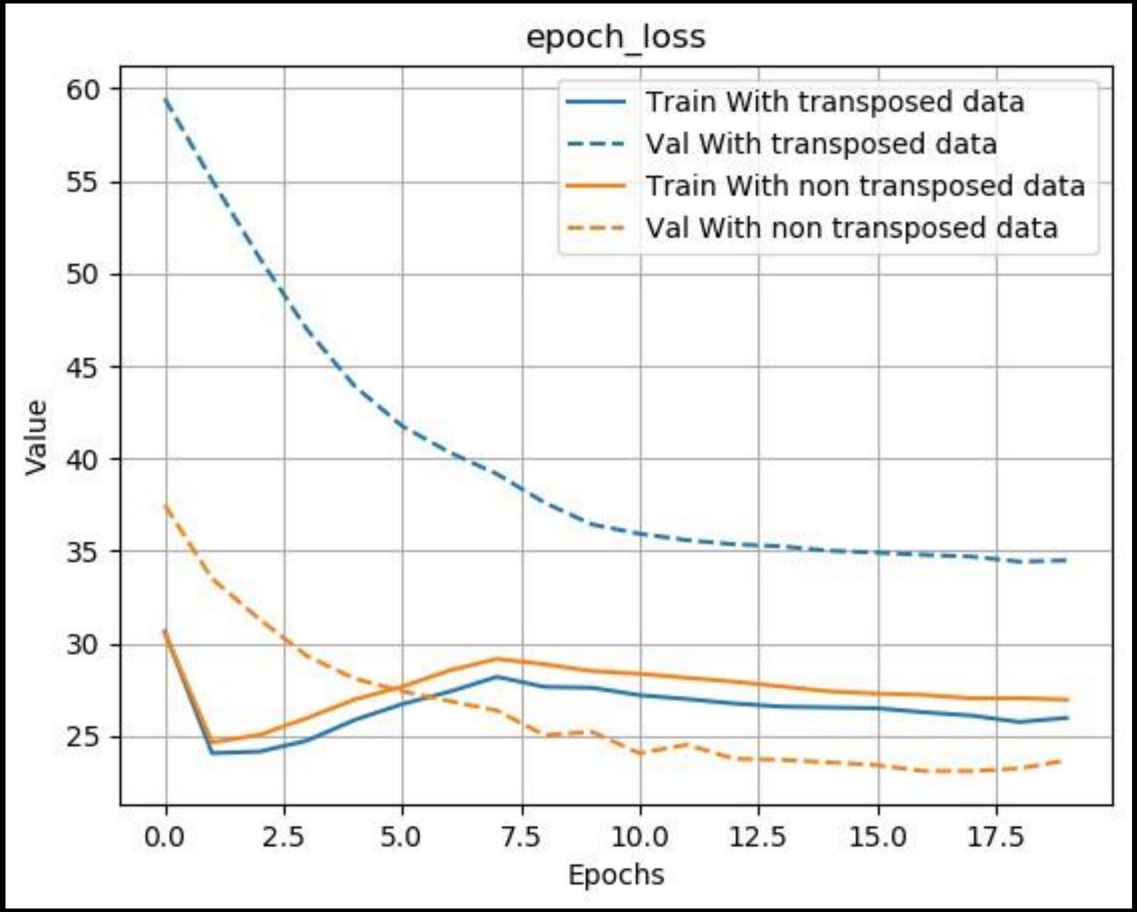


Output loss

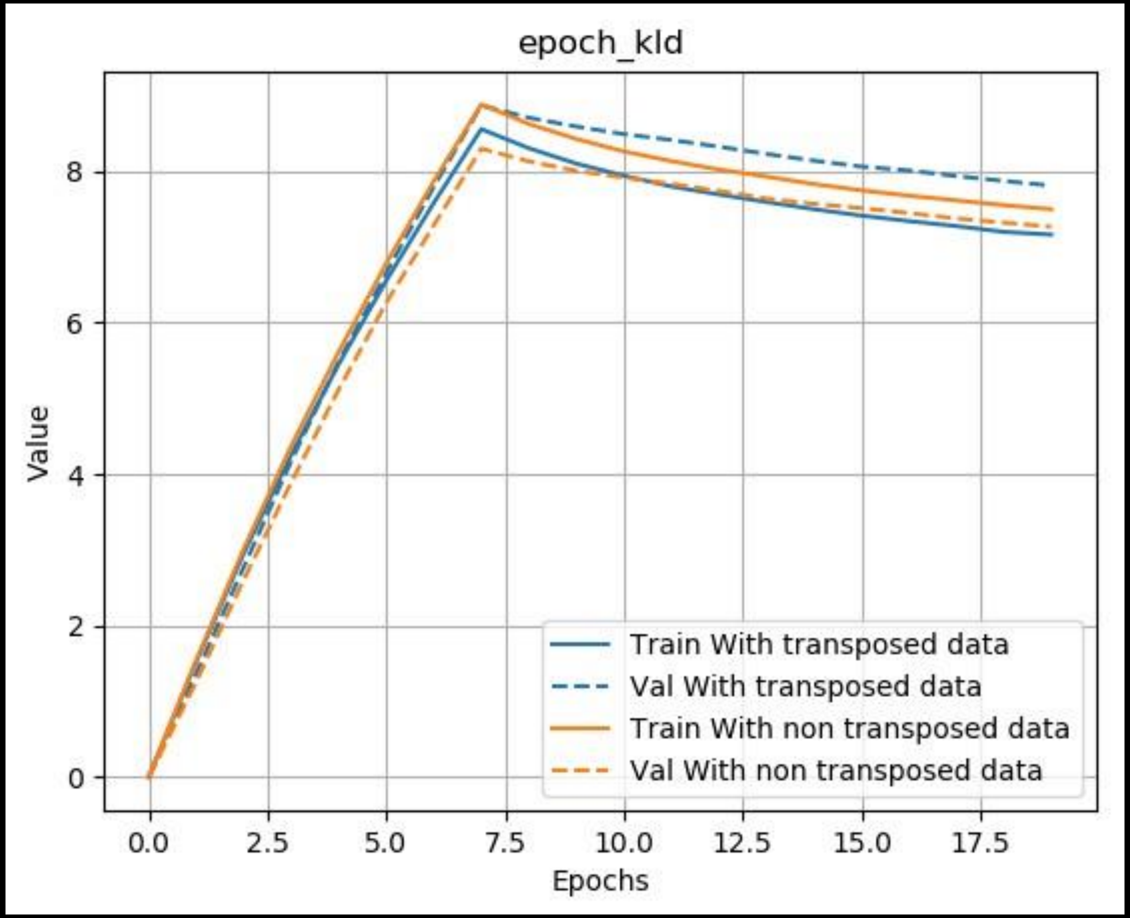


Transposed data

Global loss

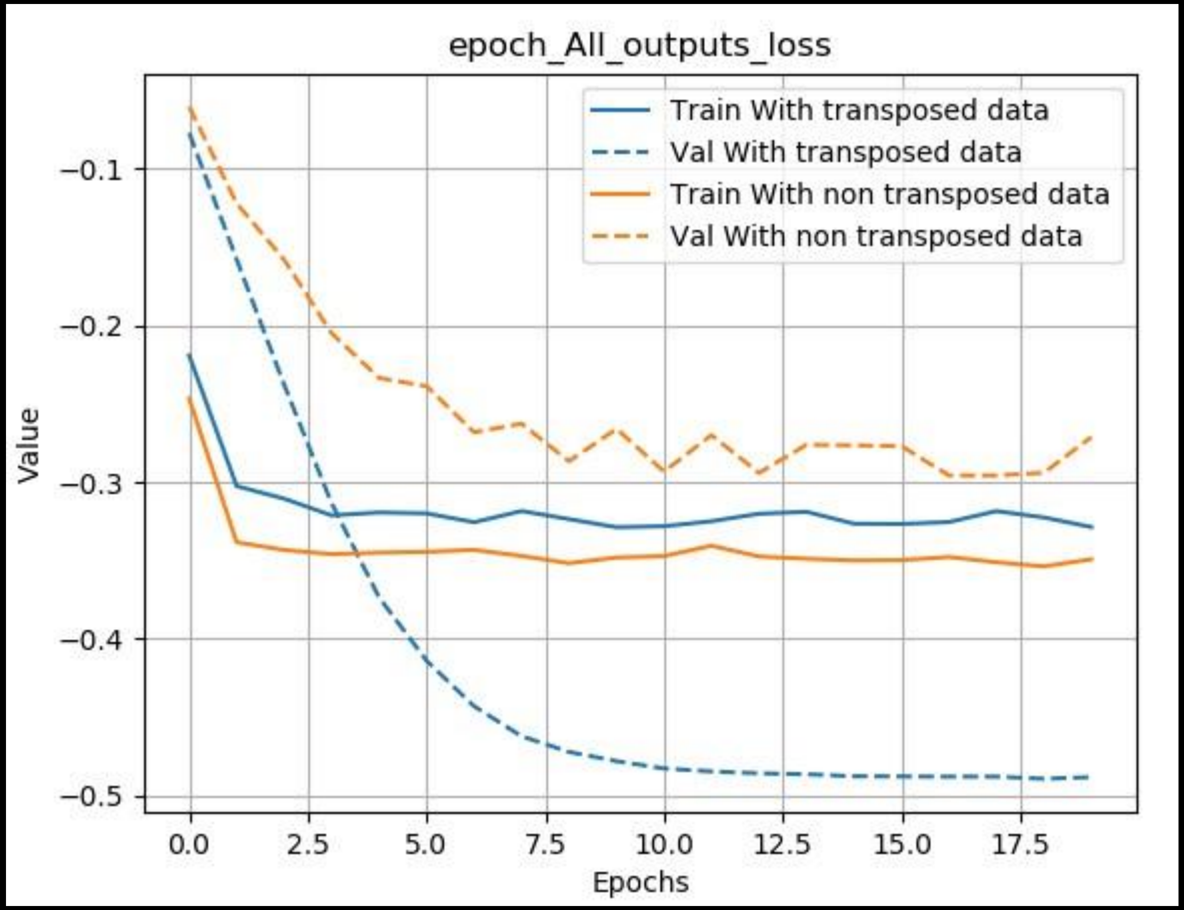


KL Divergence

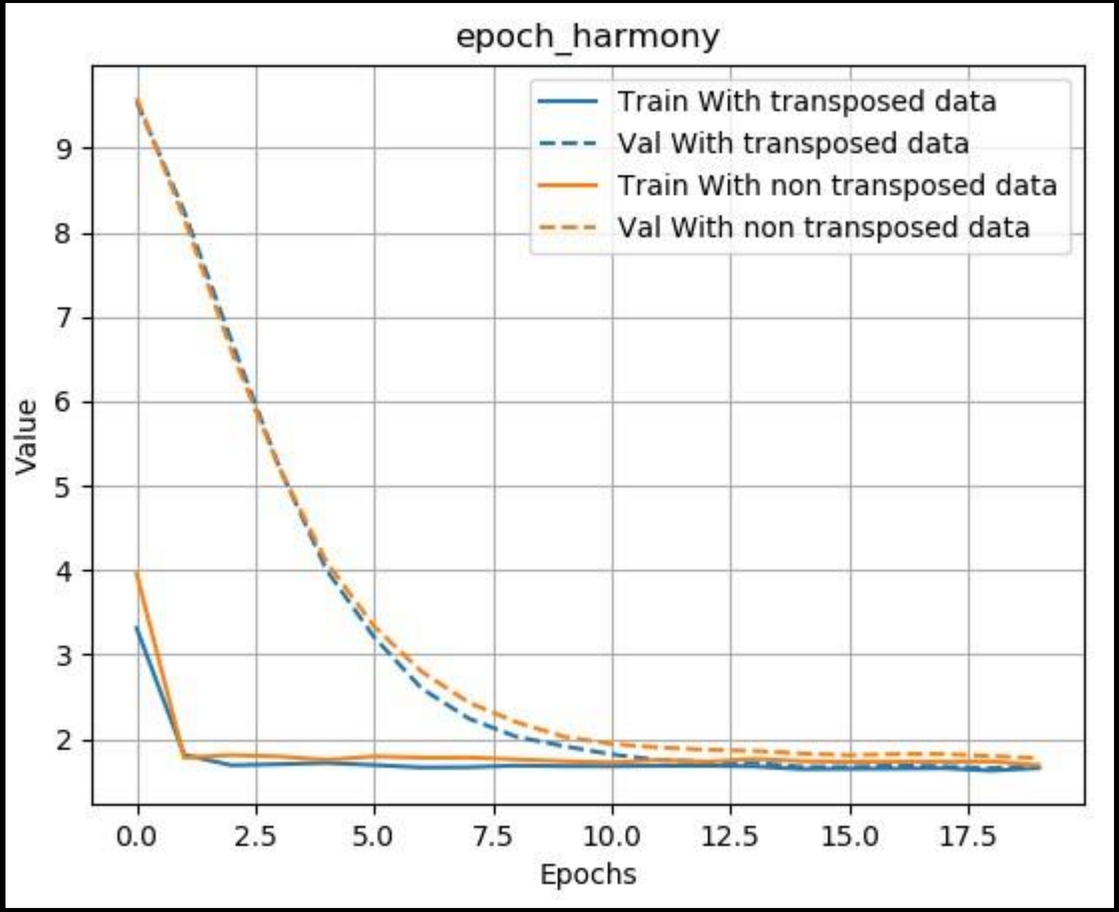


Transposed data

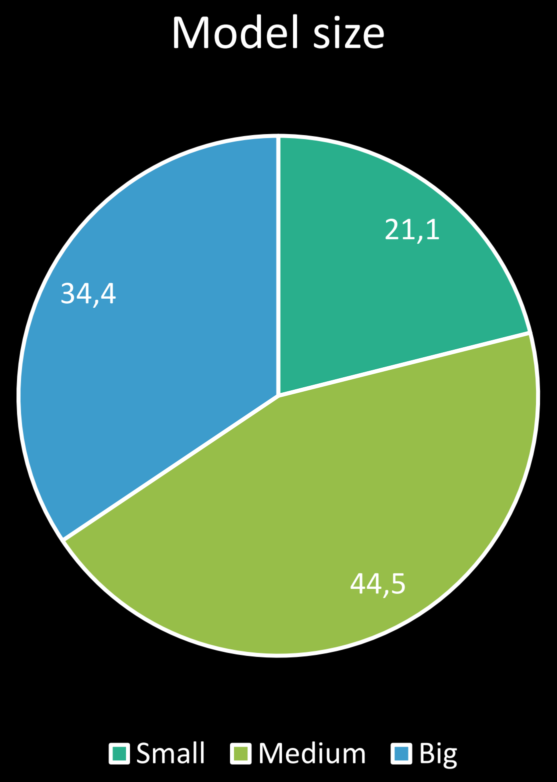
Scale and Rhythm losses



Harmony loss



Model size



32 people answered the survey

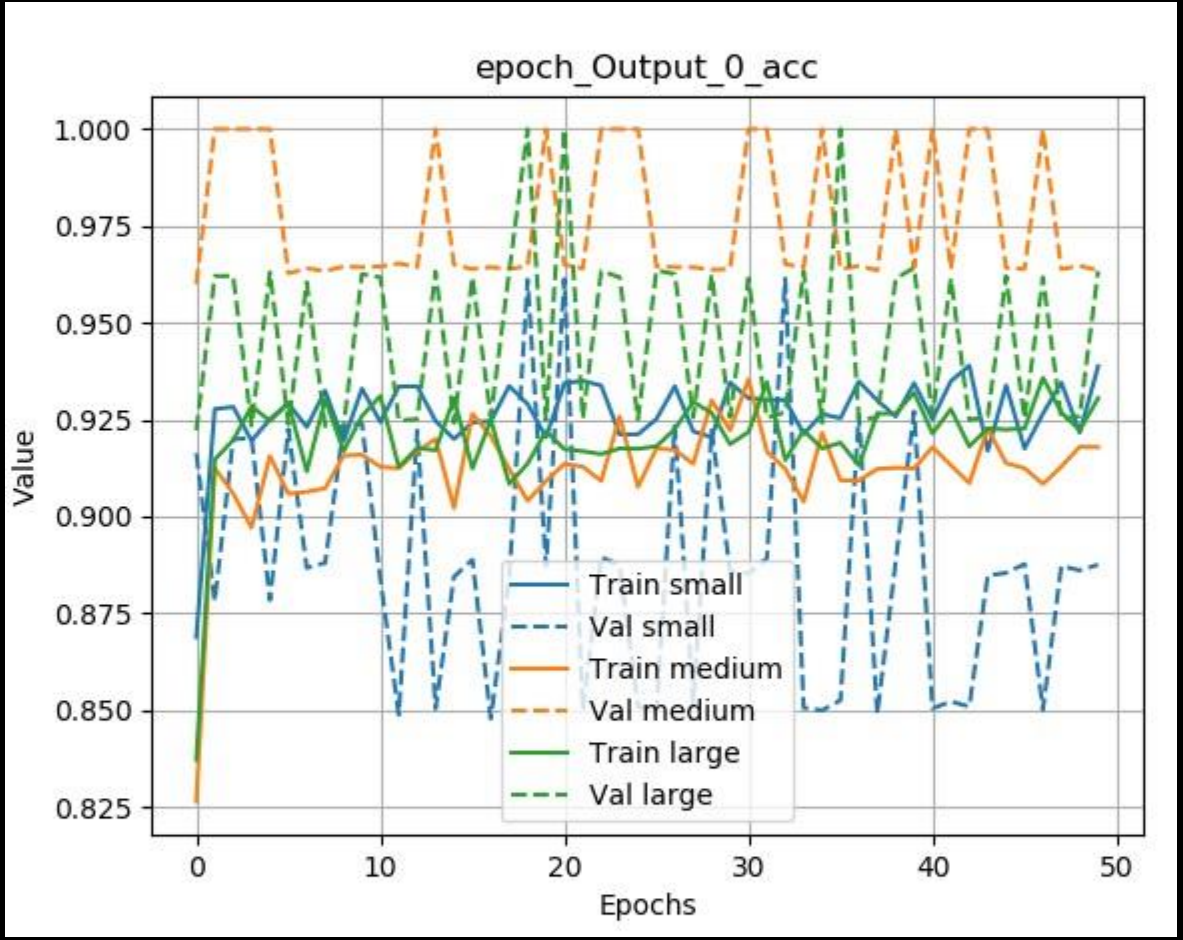
Small

Medium

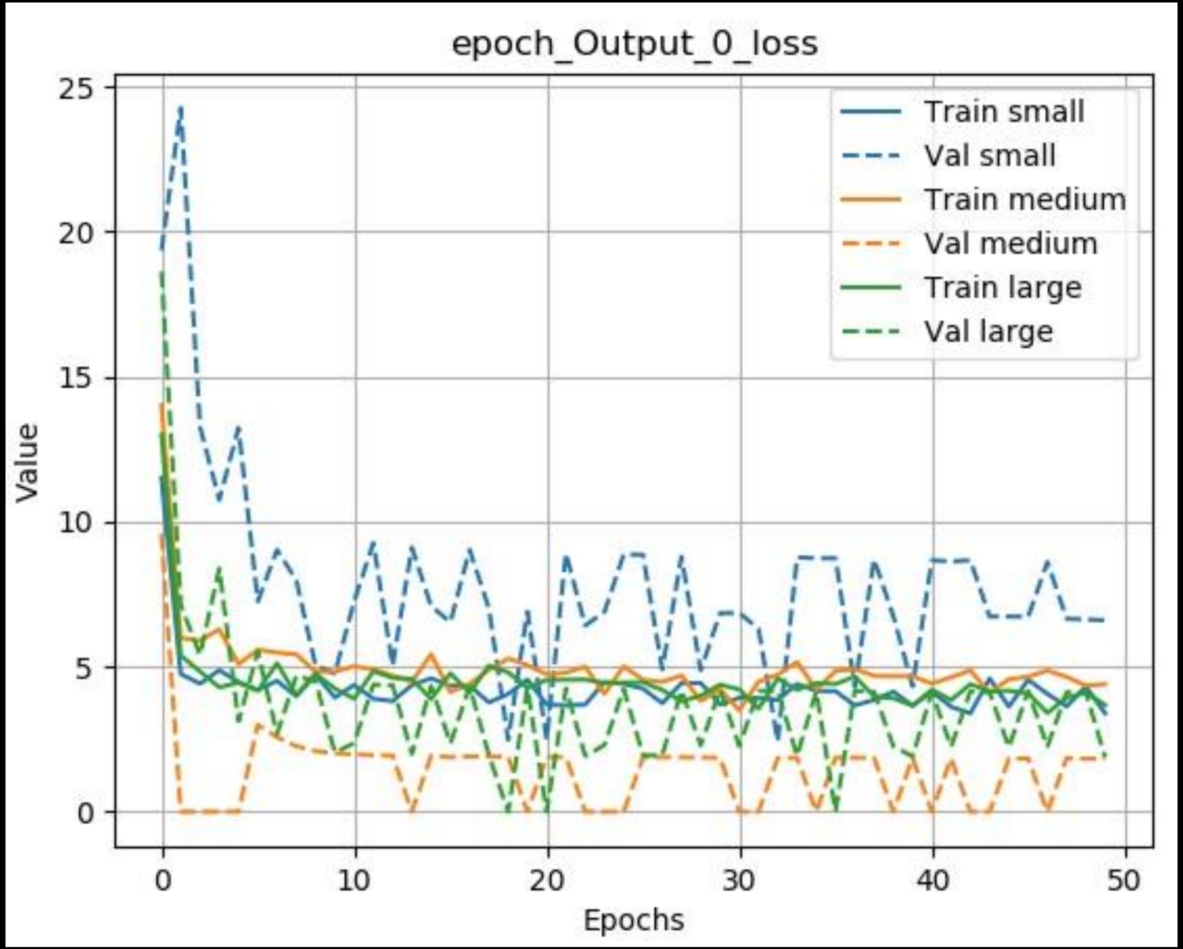
Large

Model size

Output accuracy

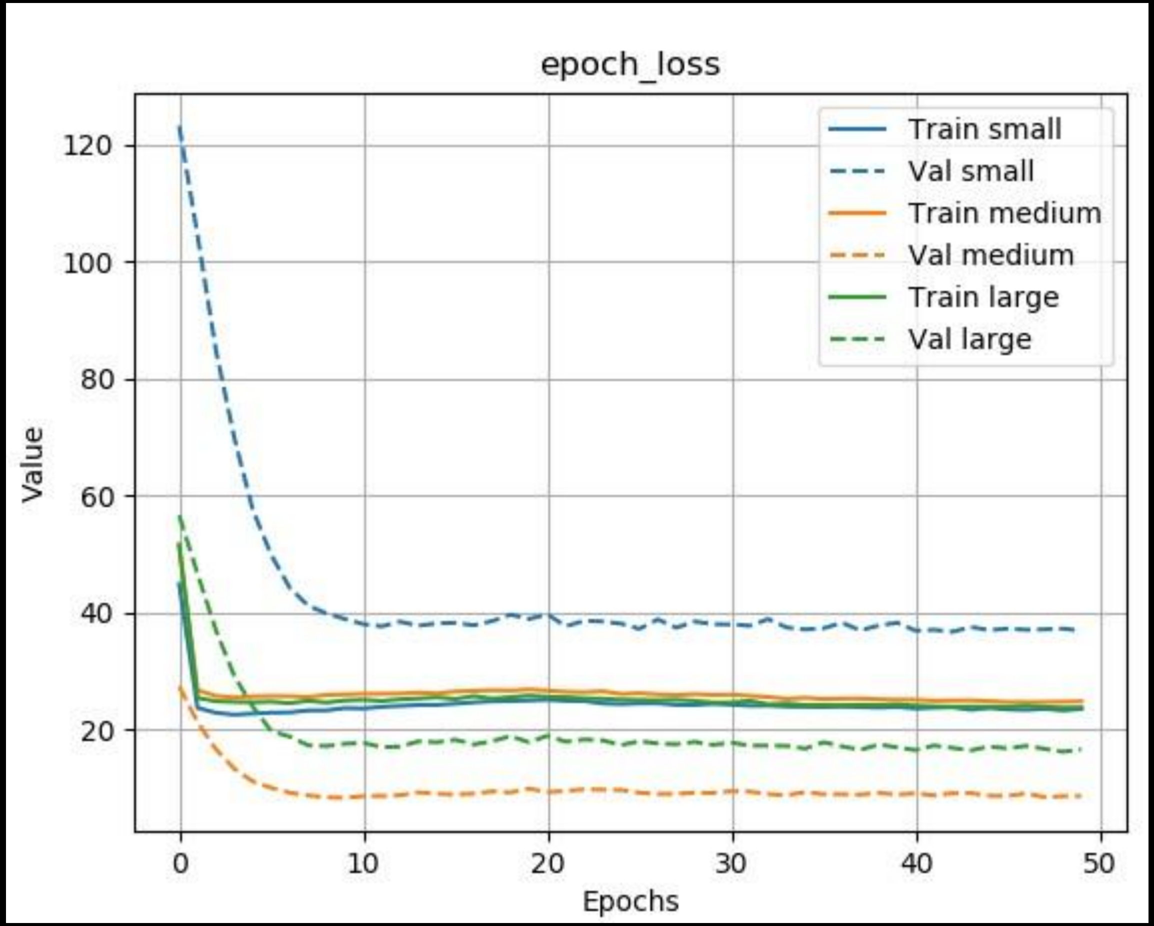


Output loss

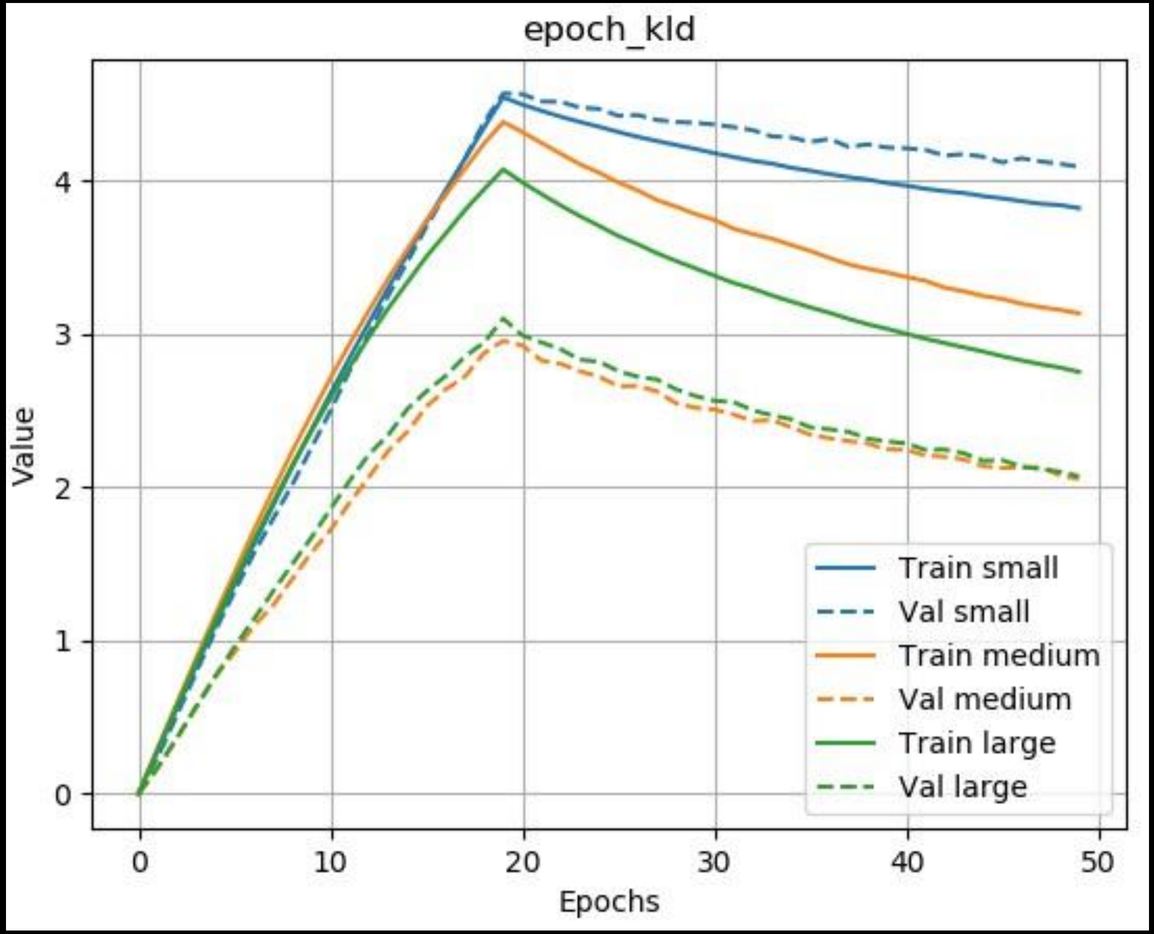


Model size

Global loss

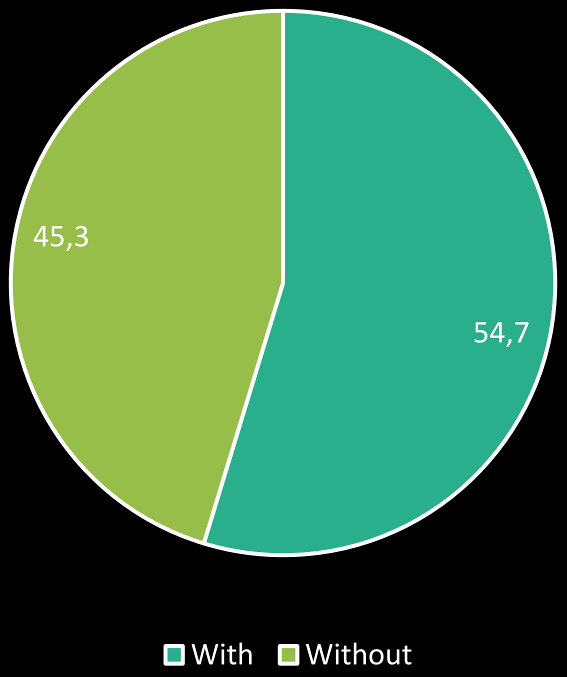


KL Divergence



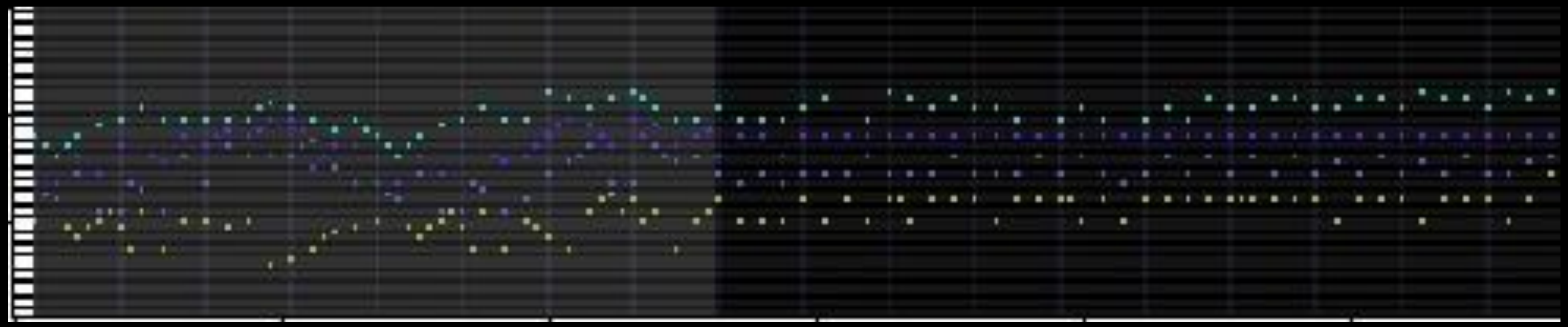
Scale and Rhythm losses

Scale and Rhythm loss



17 people answered the survey

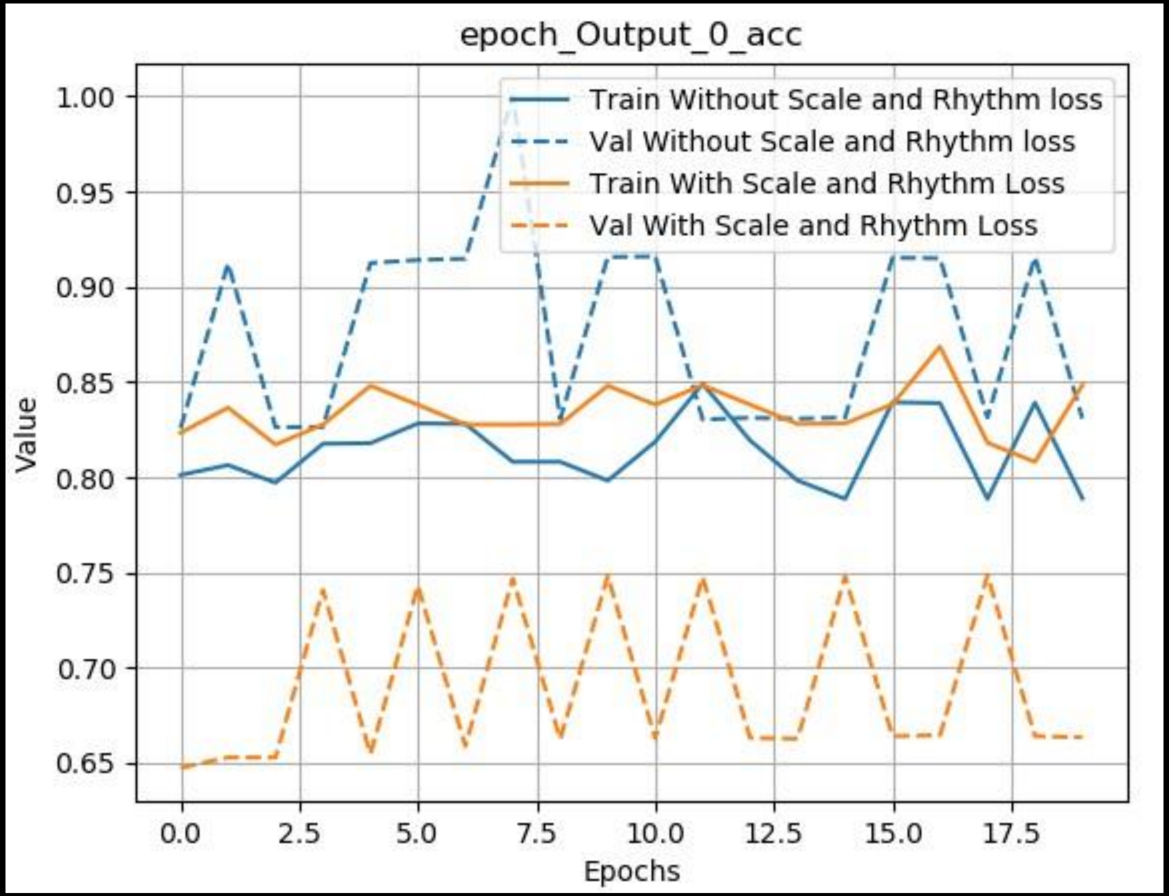
Without scale and rhythm losses



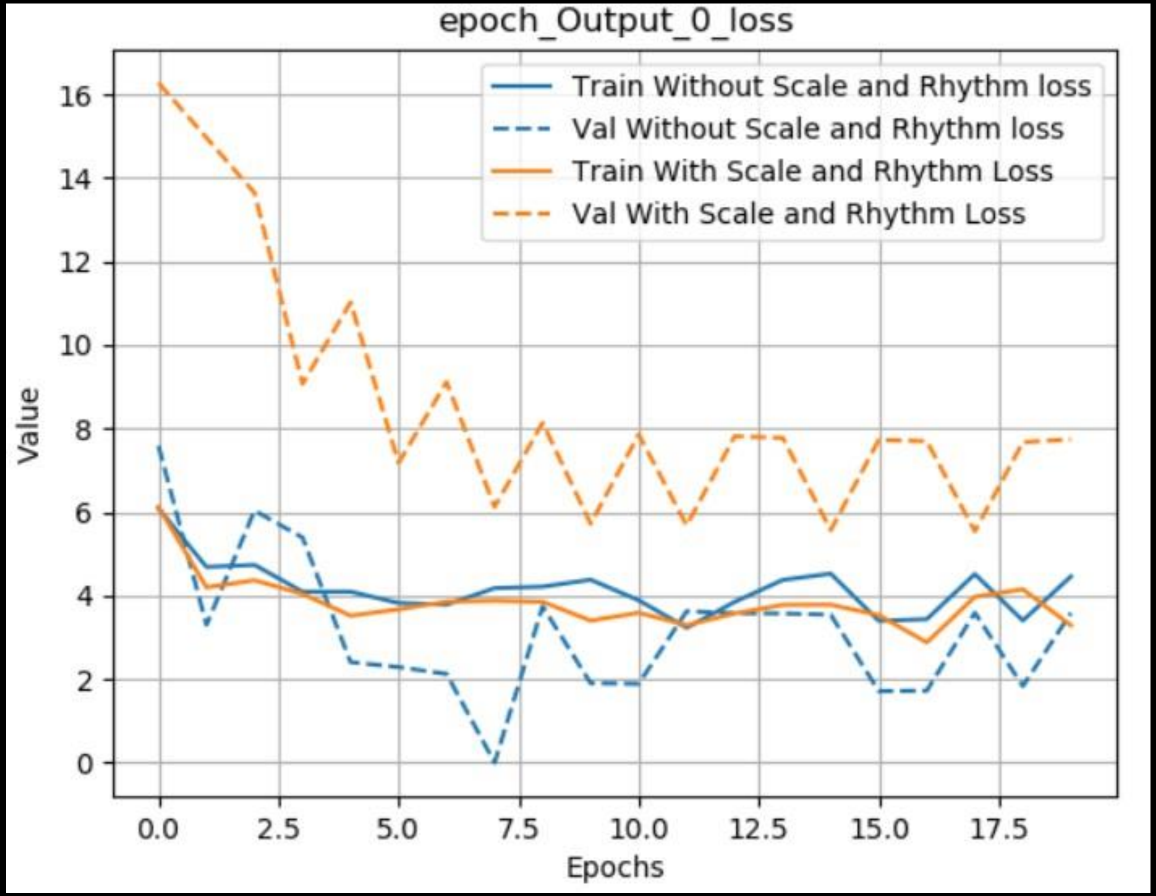
With scale and rhythm losses

Scale and Rhythm losses

Output accuracy

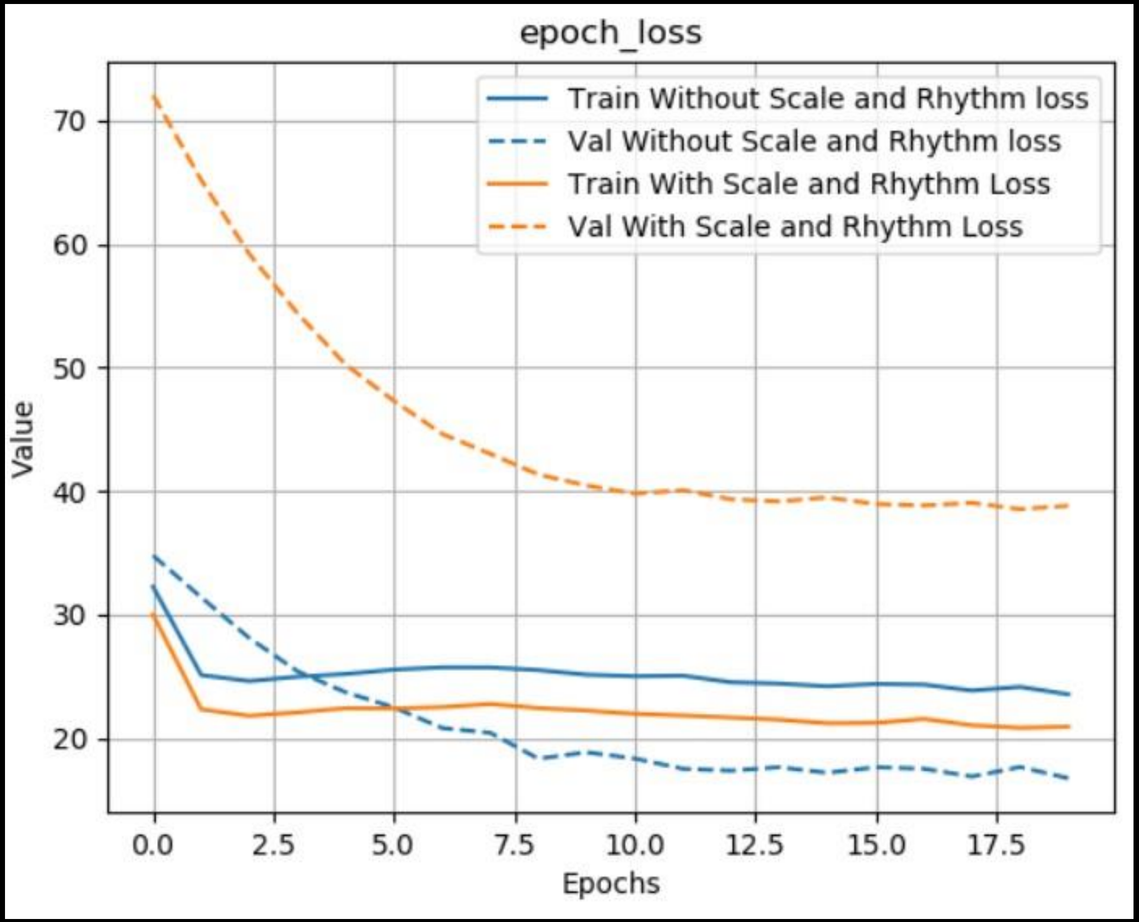


Output loss

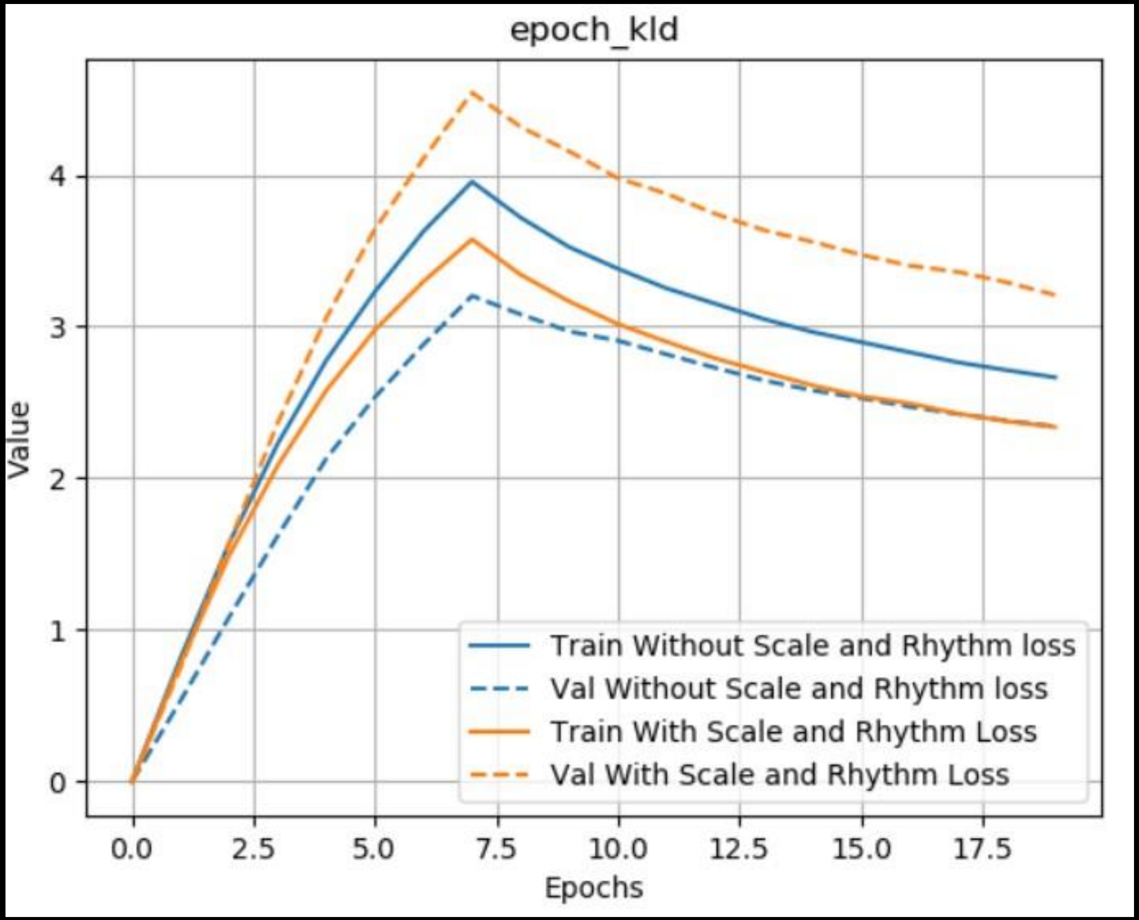


Scale and Rhythm losses

Global loss

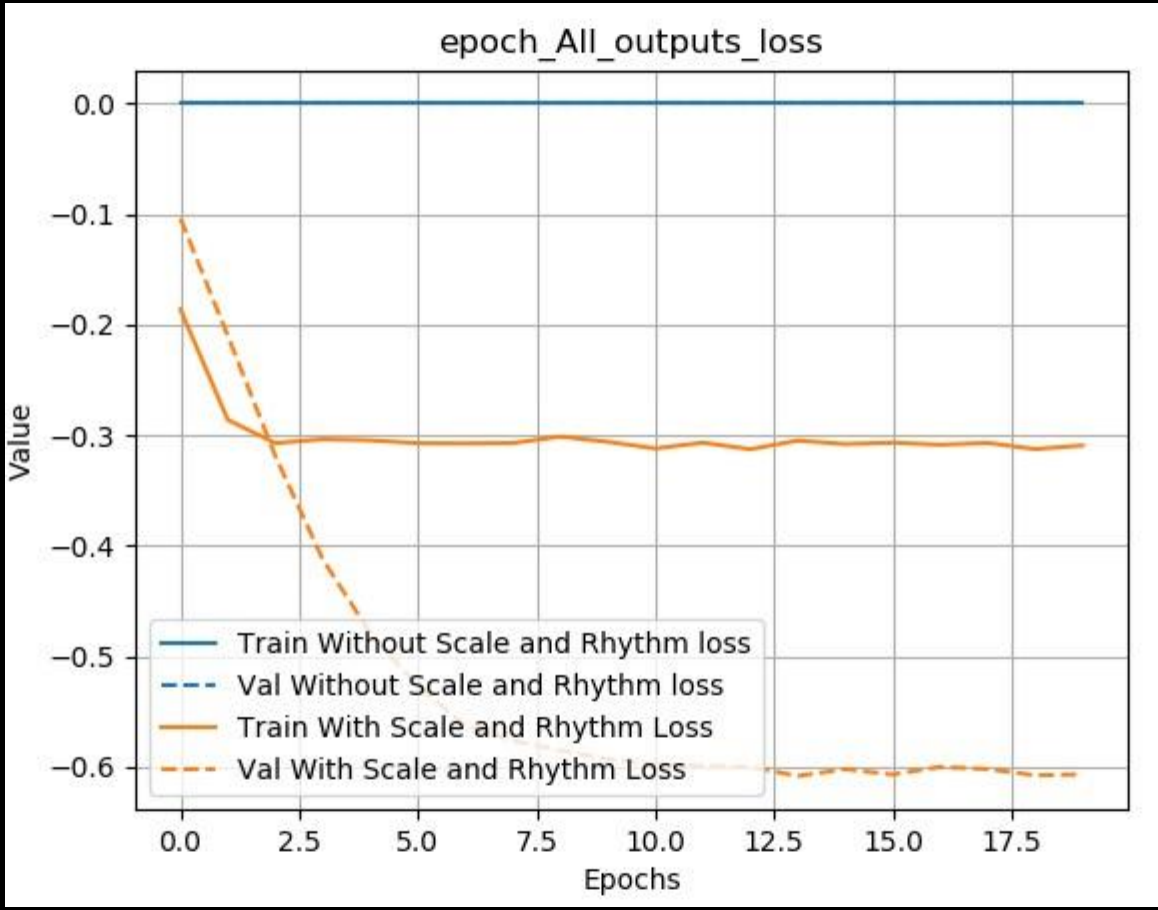


KL Divergence

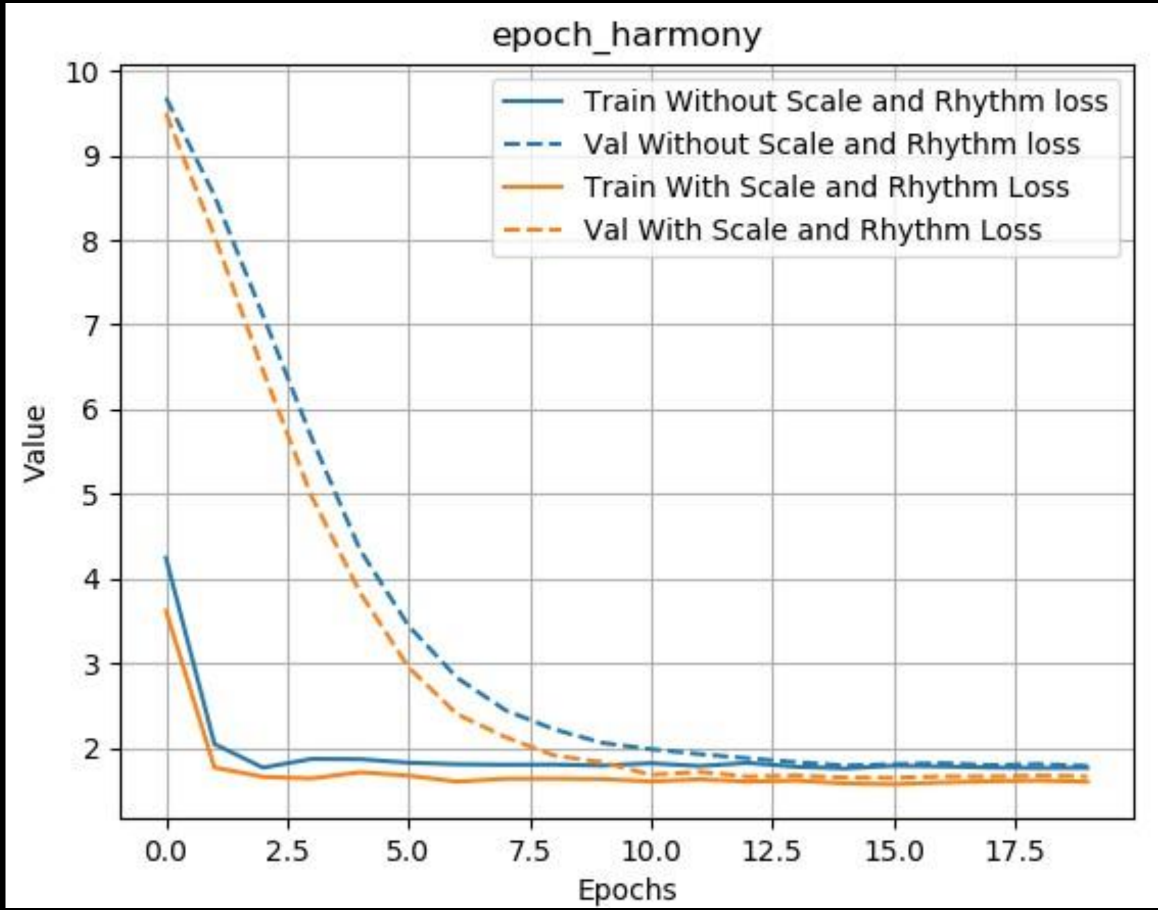


Scale and Rhythm losses

Scale and Rhythm losses

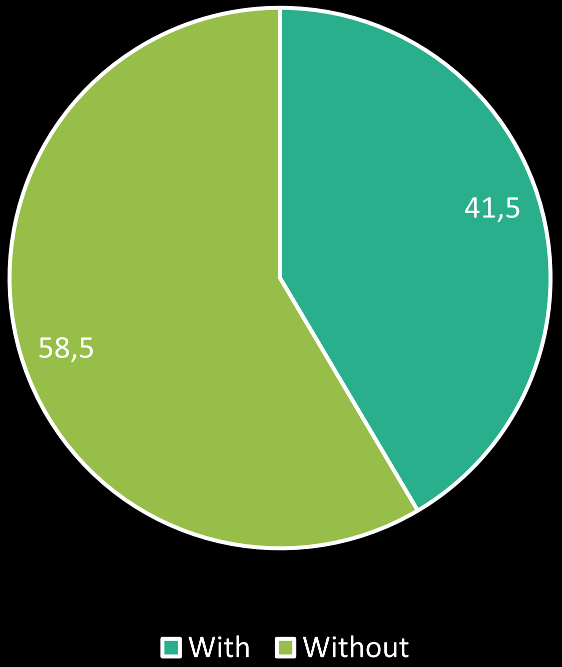


Harmony loss



Harmony

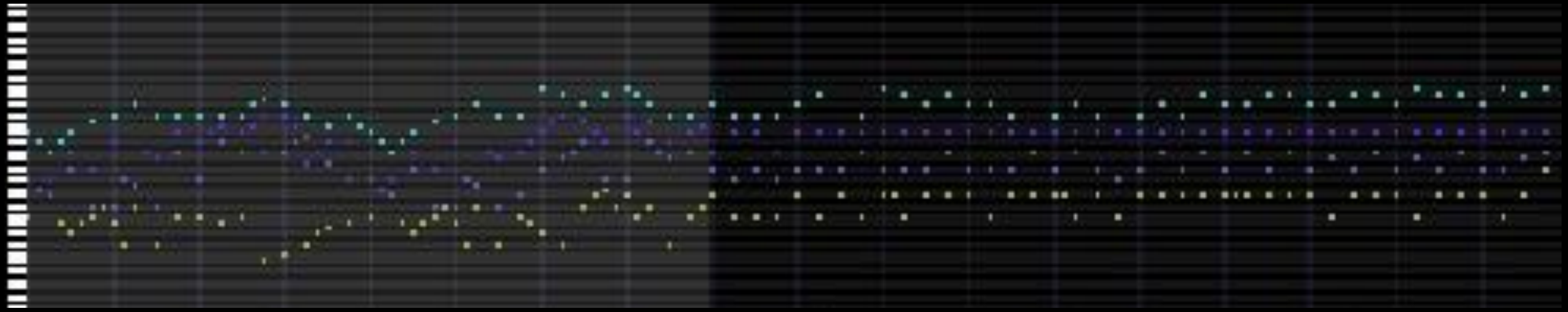
Harmony loss



22 people answered the survey

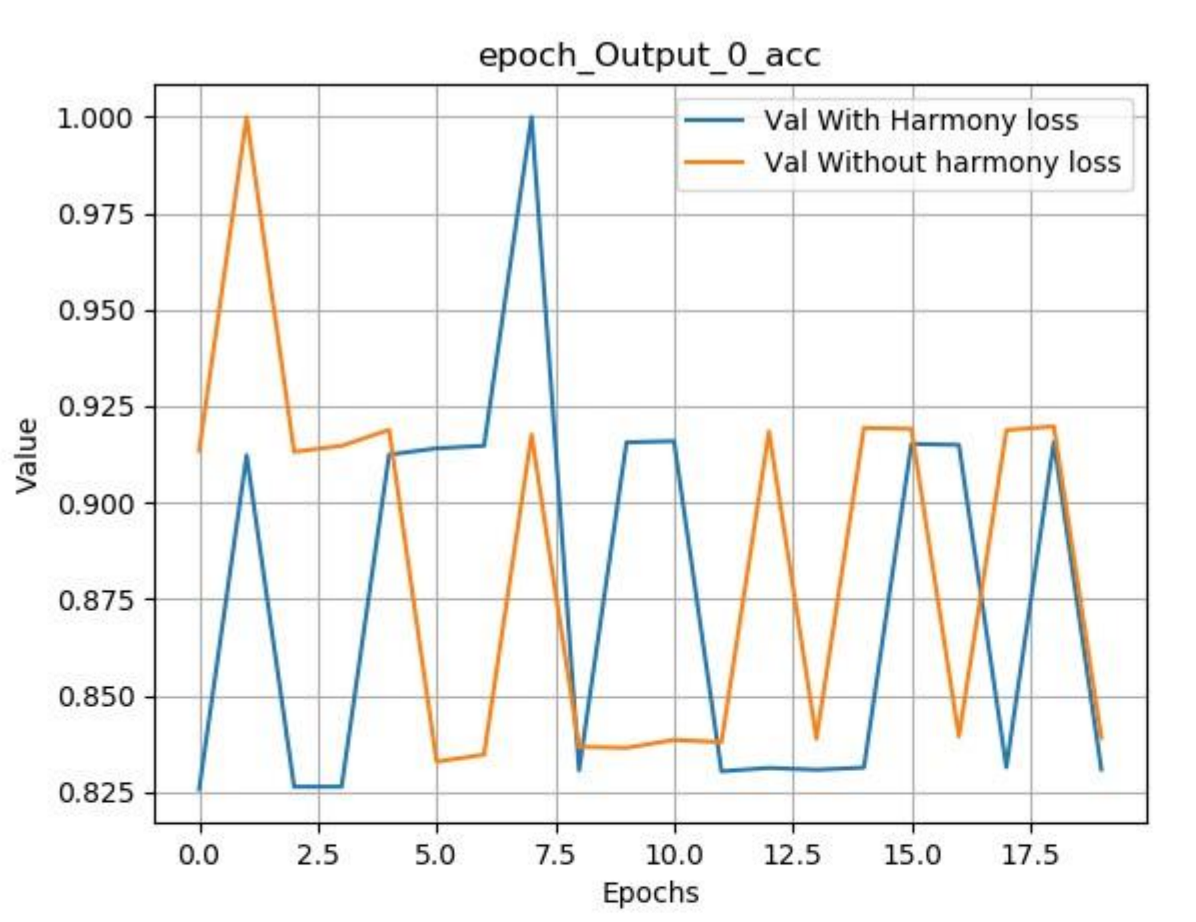
Without harmony loss

With harmony loss

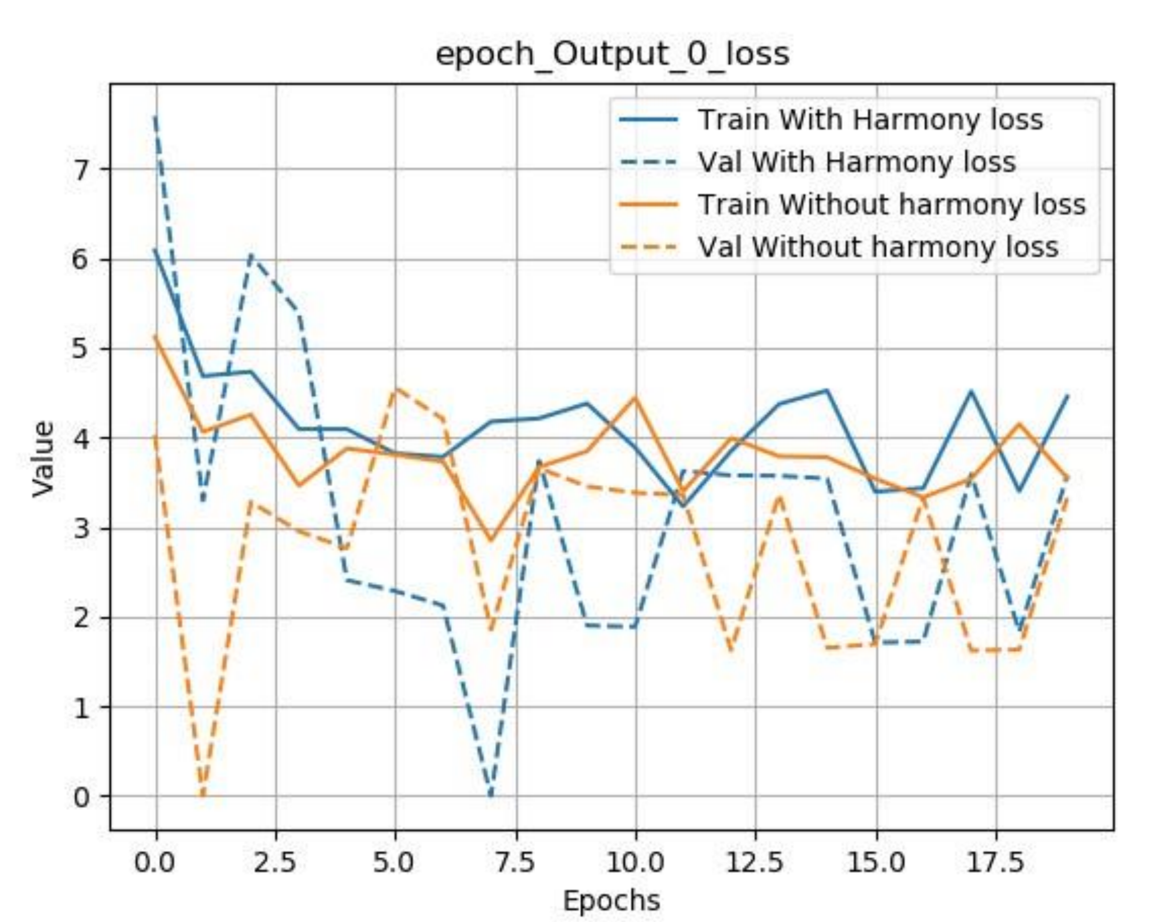


Harmony

Output loss

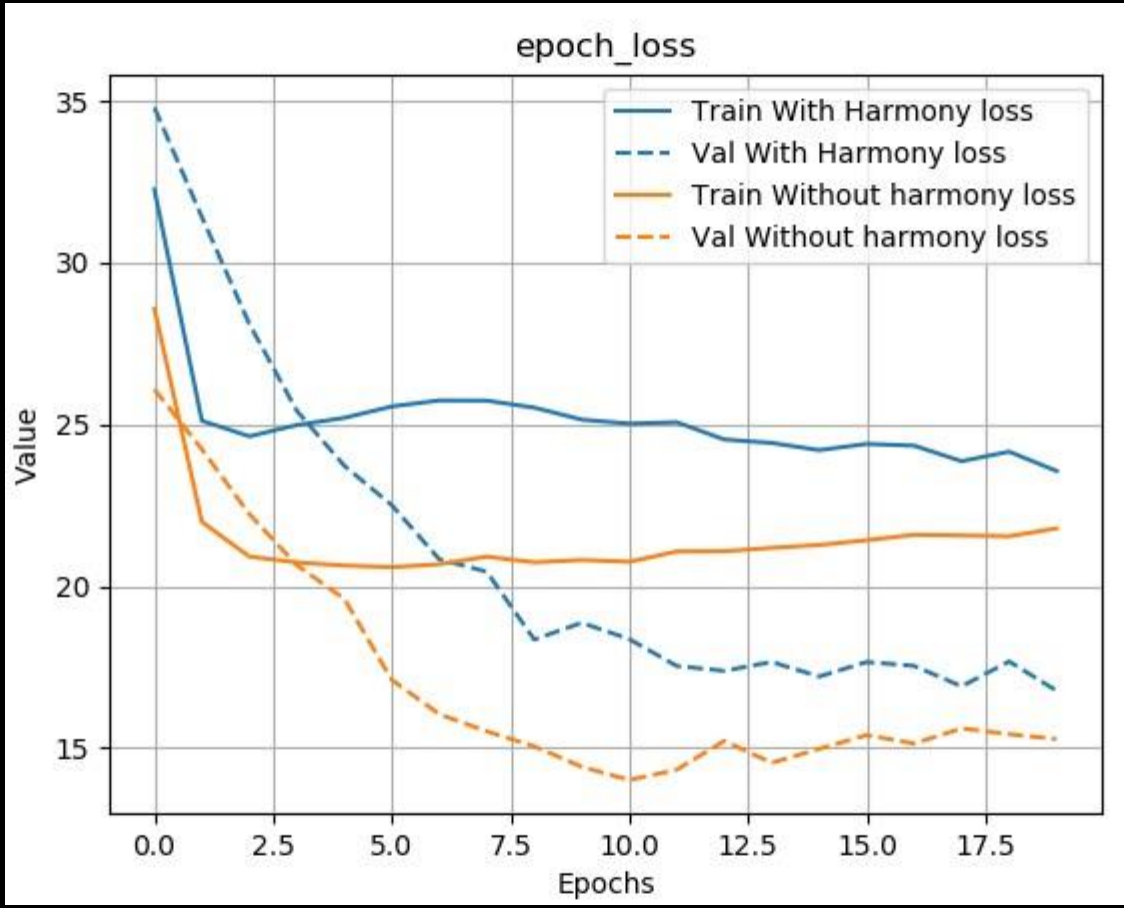


Output accuracy

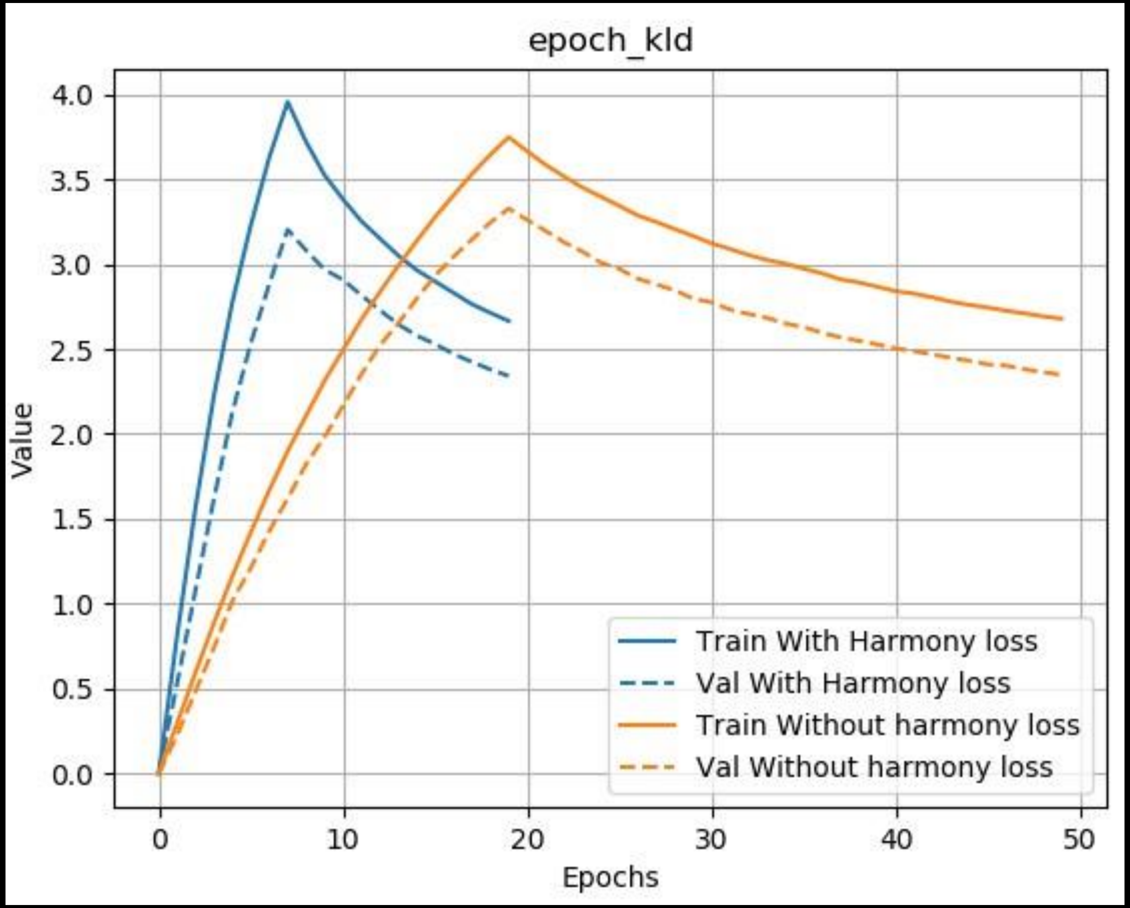


Harmony

Global loss

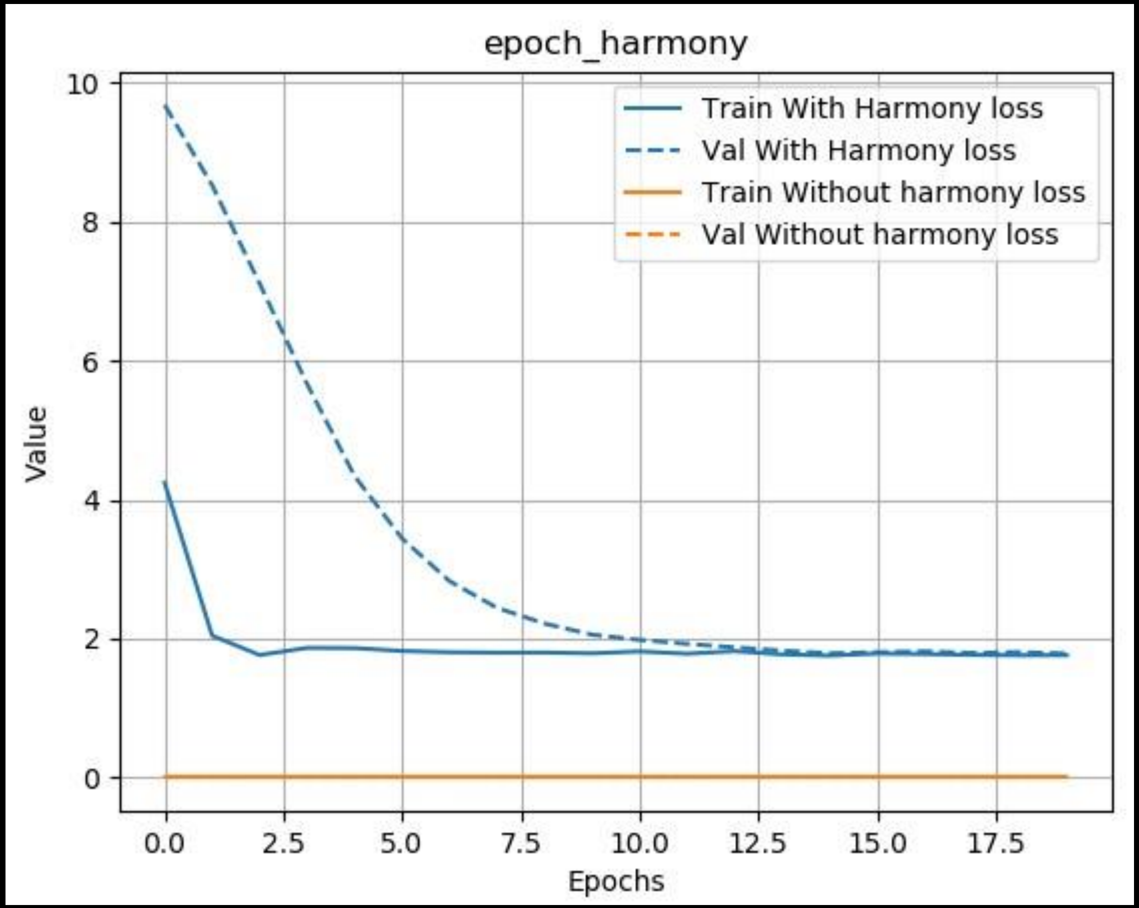


KL Divergence

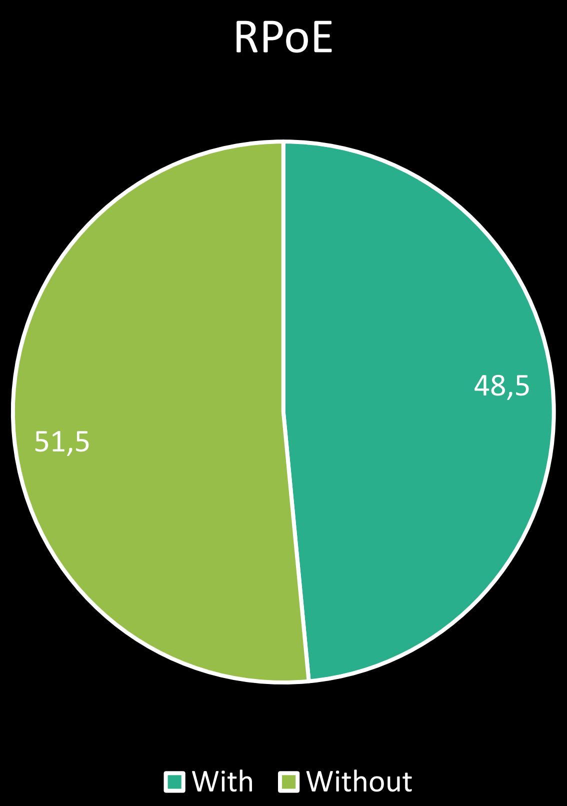


Harmony

Harmony loss



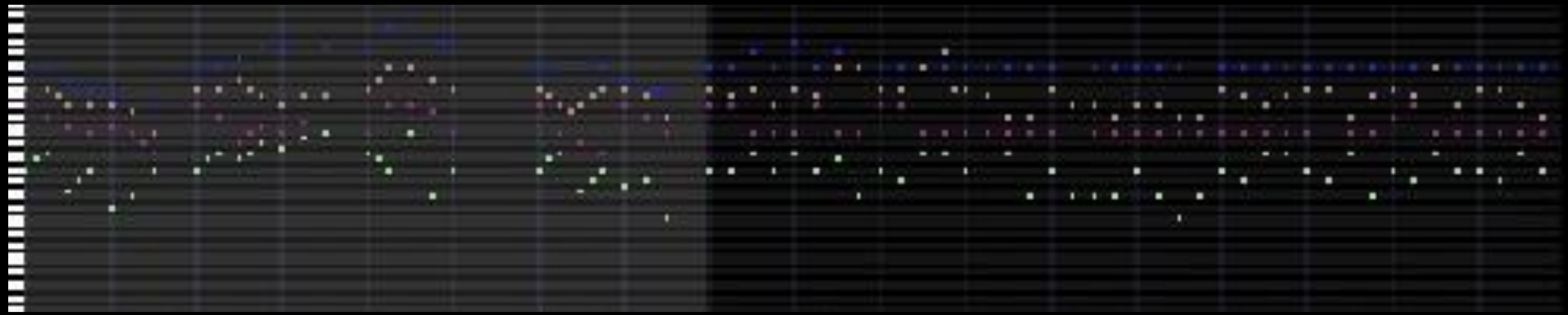
RPoE



13 people answered the survey

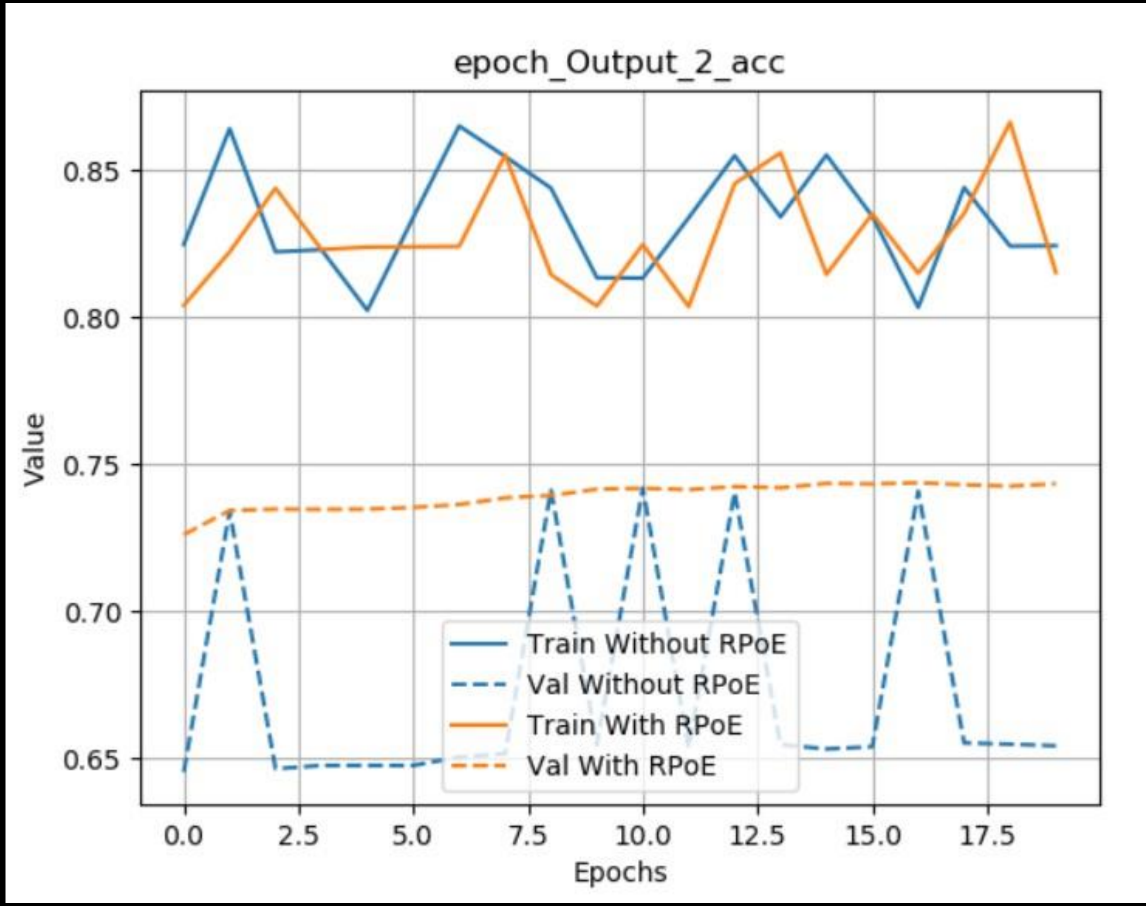
Without RPoE

With RPoE

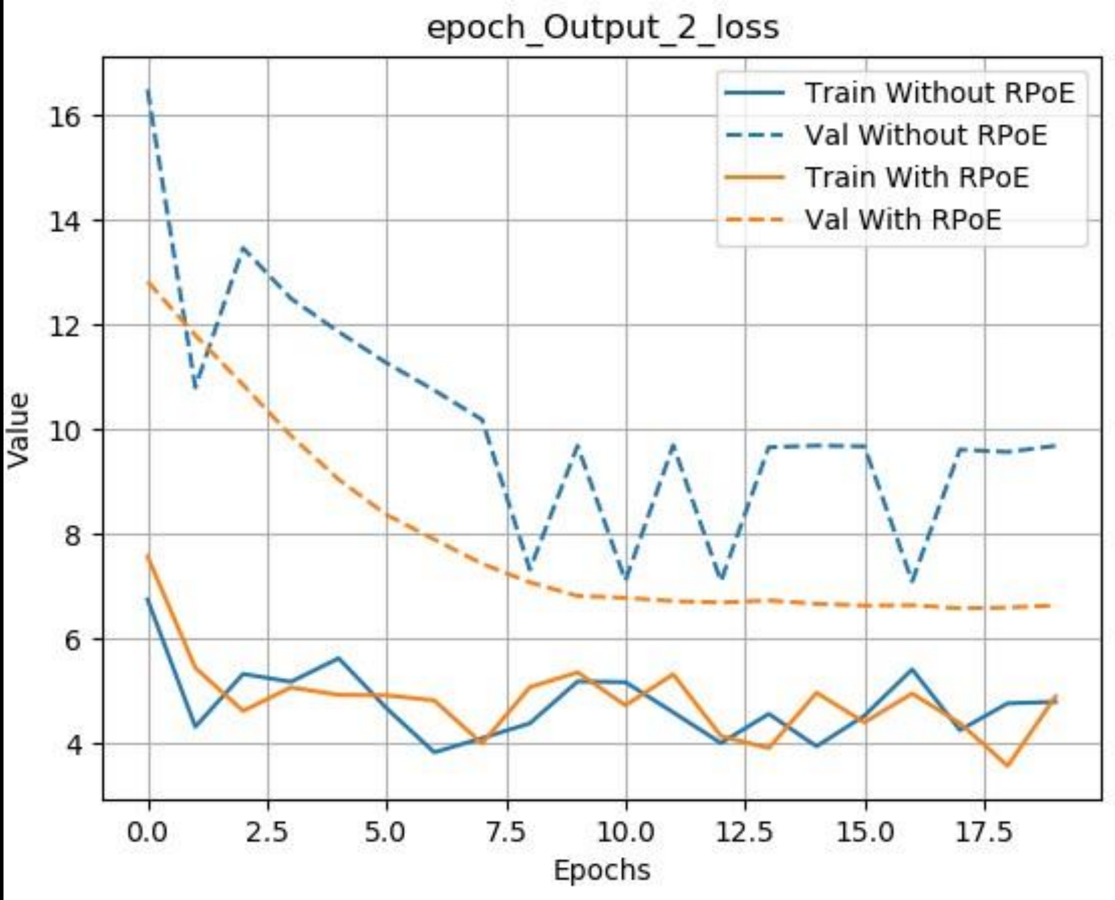


RPoE

Output accuracy

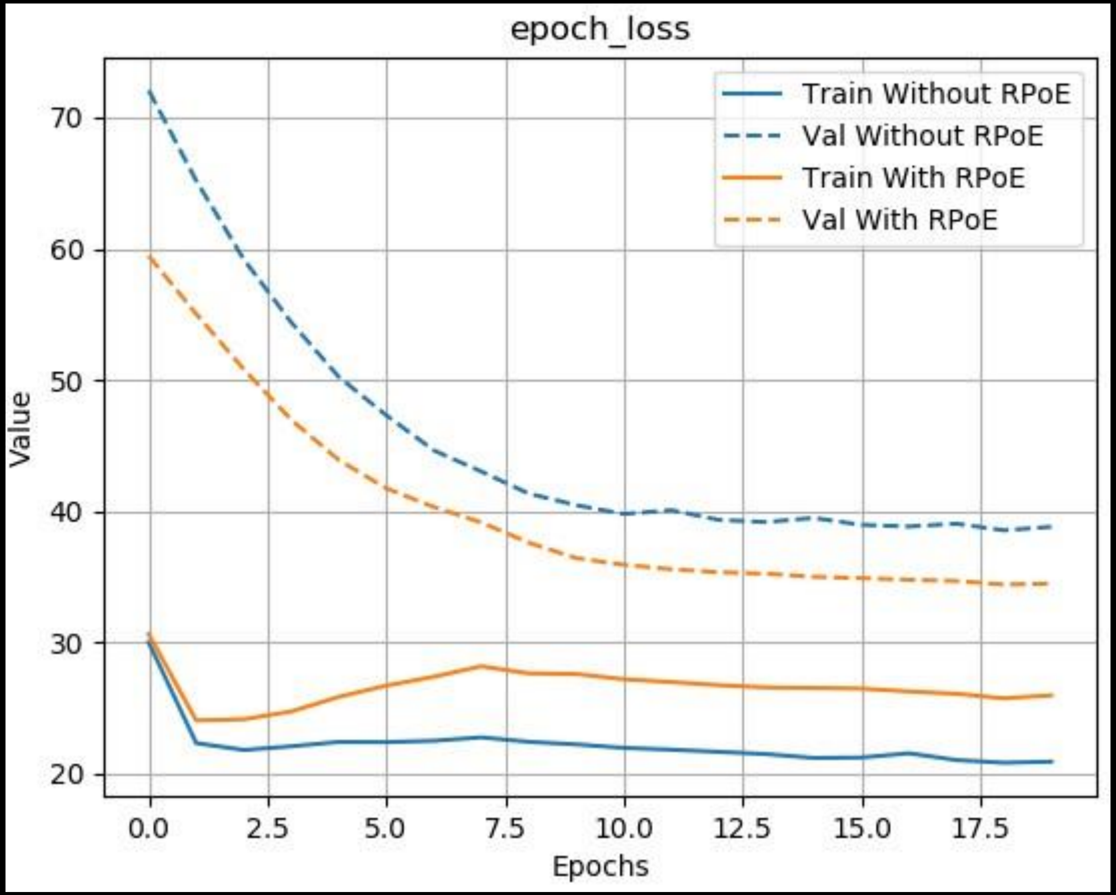


Output loss

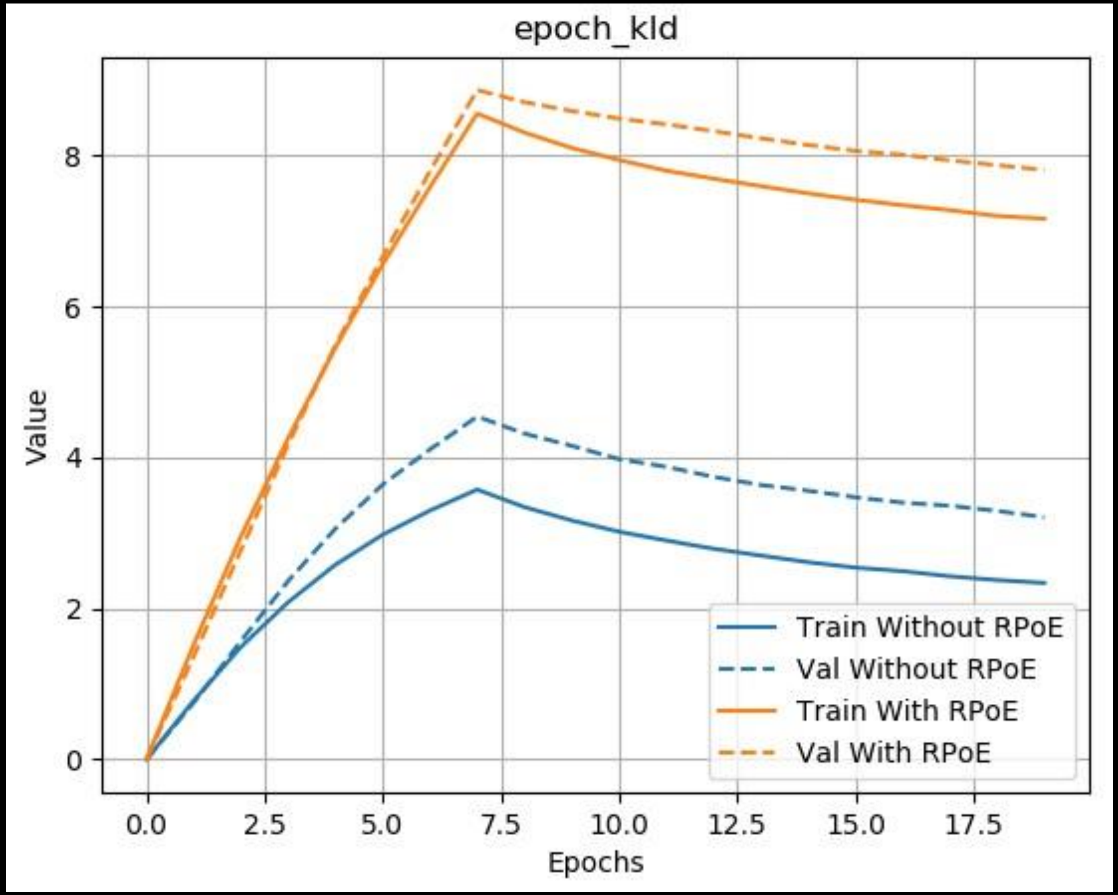


RPoE

Global loss

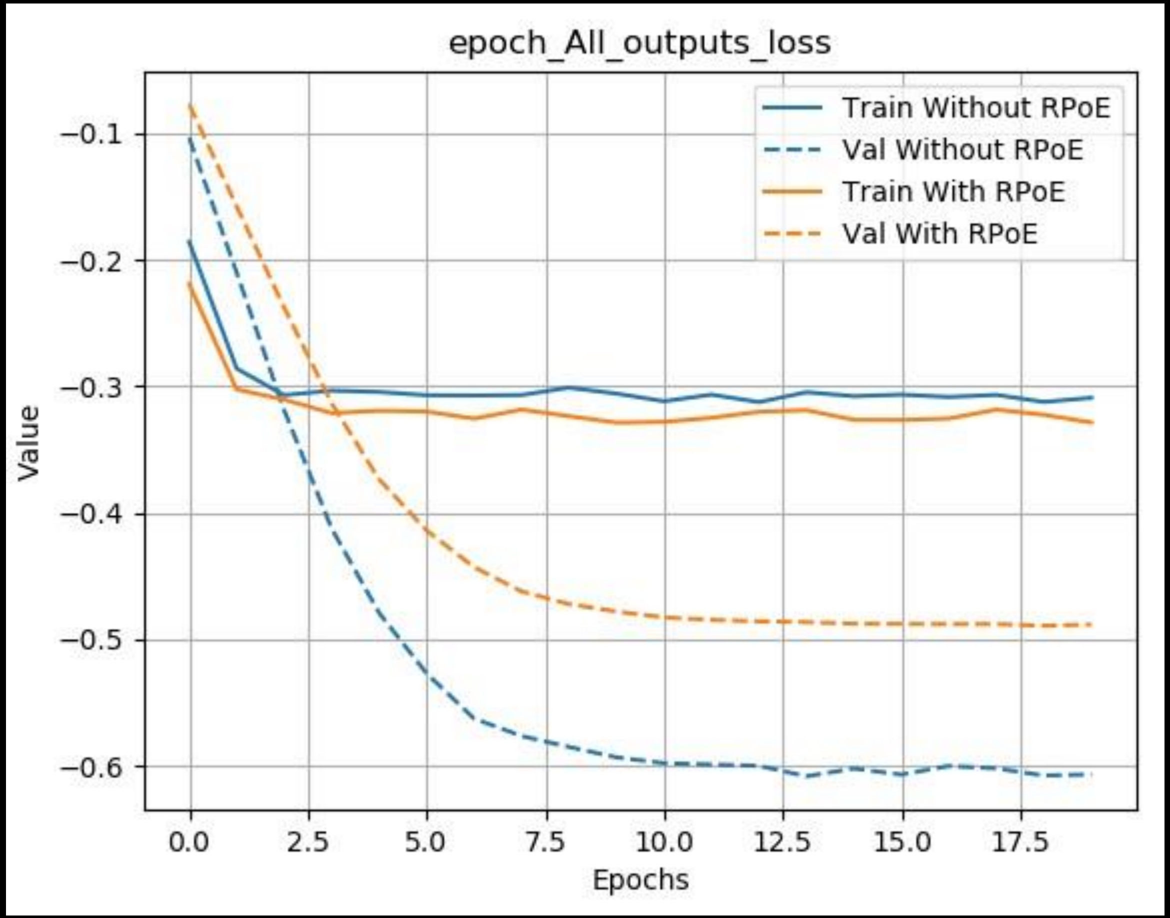


KL Divergence

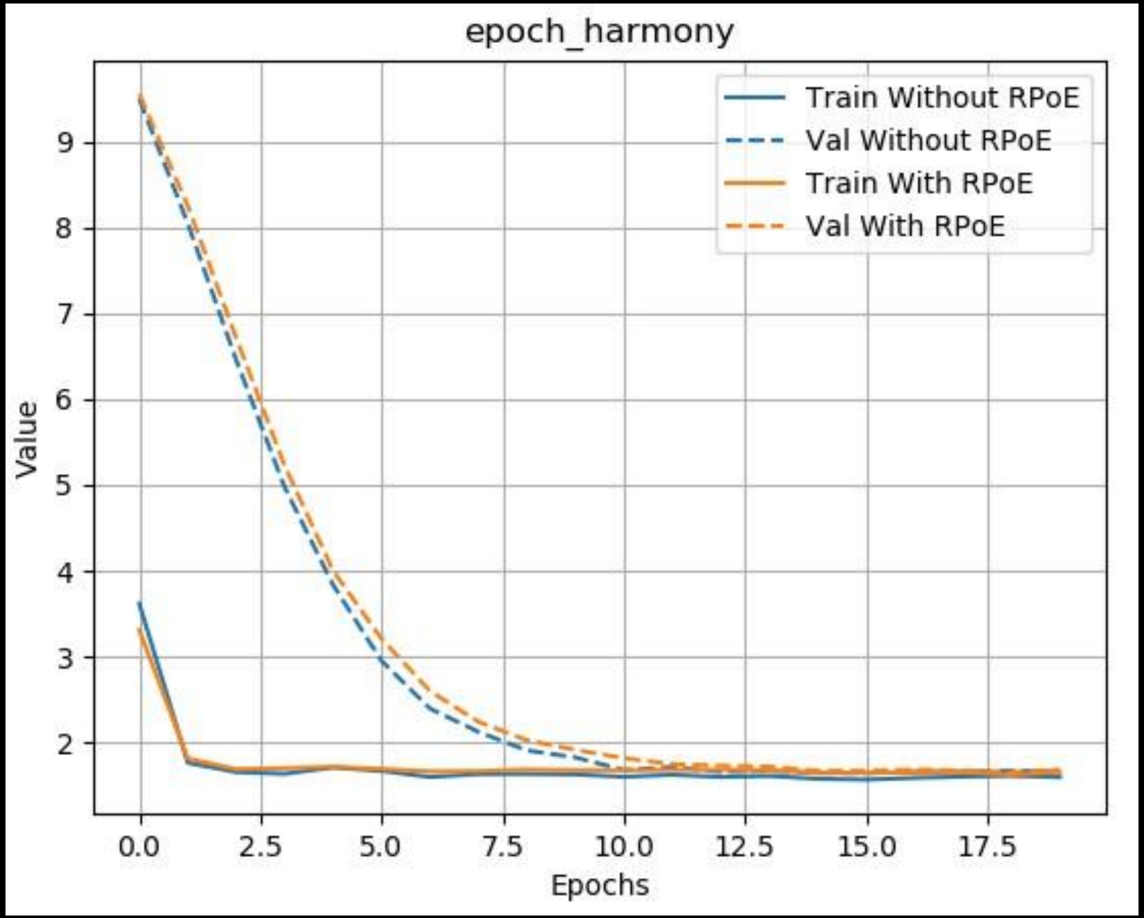


RPoE

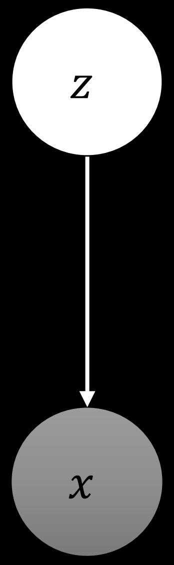
Scale and Rhythm losses



Harmony loss



Variational AutoEncoder



- Tries to maximize the marginal likelihood

$$argmax_{\theta}[\log(p_{\theta}(x))], x \in data$$

$$\log(p_{\theta}(x)) = \log\left(\int_{z} p_{\theta}(x|z)p(z)dz\right)$$

- Maximizes the ELBO

$$ELBO(x) = \mathbb{E}_{q_{\phi}(z|x)}\left(\log\left(p_{\theta}(x|z)\right)\right) - \mathbb{D}_{KL}\left(q_{\phi}(z|x), p(z)\right)$$

$$p(z) \sim \mathcal{N}(0, 1)$$

Reparameterization trick

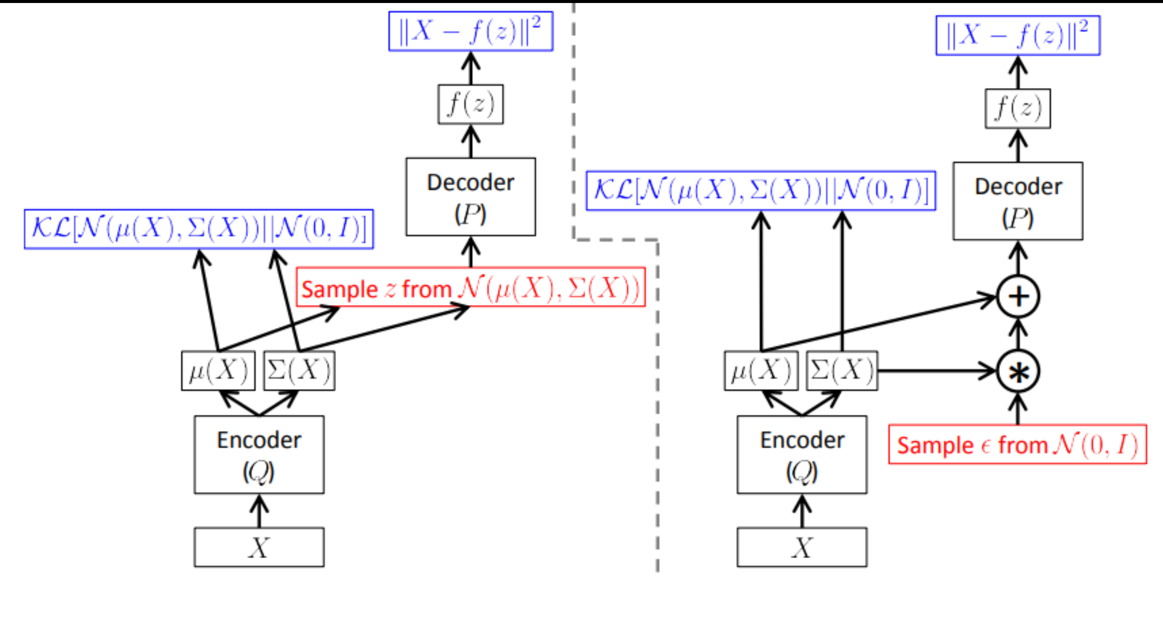


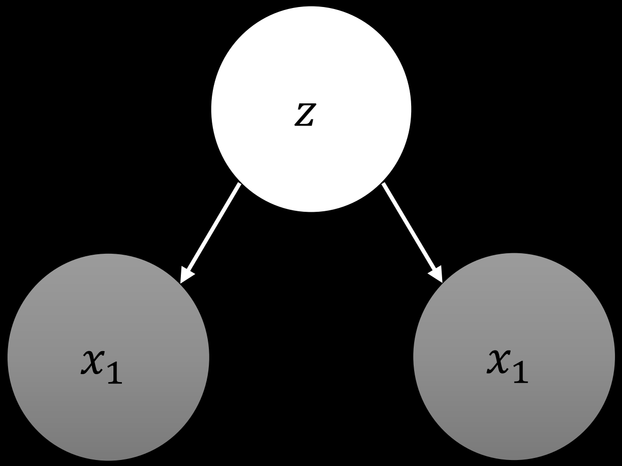
Figure 4: A training-time variational autoencoder implemented as a feed-forward neural network, where $P(X|z)$ is Gaussian. Left is without the "reparameterization trick", and right is with it. Red shows sampling operations that are non-differentiable. Blue shows loss layers. The feedforward behavior of these networks is identical, but backpropagation can be applied only to the right network.

Reparameterization trick from C. Doersch "Tutorial on Variational Autoencoders"

Multimodal Variational AutoEncoder

- Introduced by Mike Wu to solve the multi-model inference problem
- Miximizes the ELBO

$$ELBO(x) = \mathbb{E}_{q_\phi(z|x)}\left(\sum_{x_i \in X} \lambda_i \log\big(p_\theta(x|z)\big)\right) - \beta\ \mathbb{D}_{KL}\big(q_\phi(z|x), p(x)\big)$$

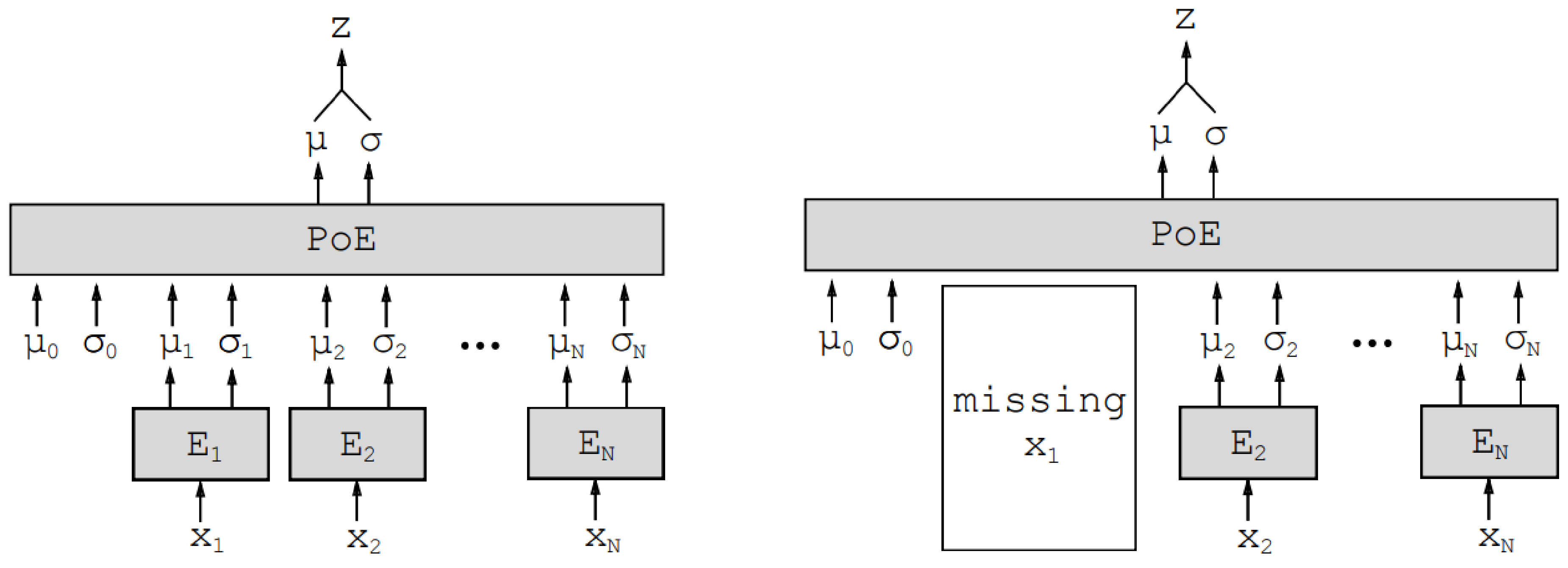


Product of Experts

Approximates the joint posterior

$$p(z|x_1, \dots, x_N) \propto p(z) \prod_{i=1}^{N} q(z|x_i)$$

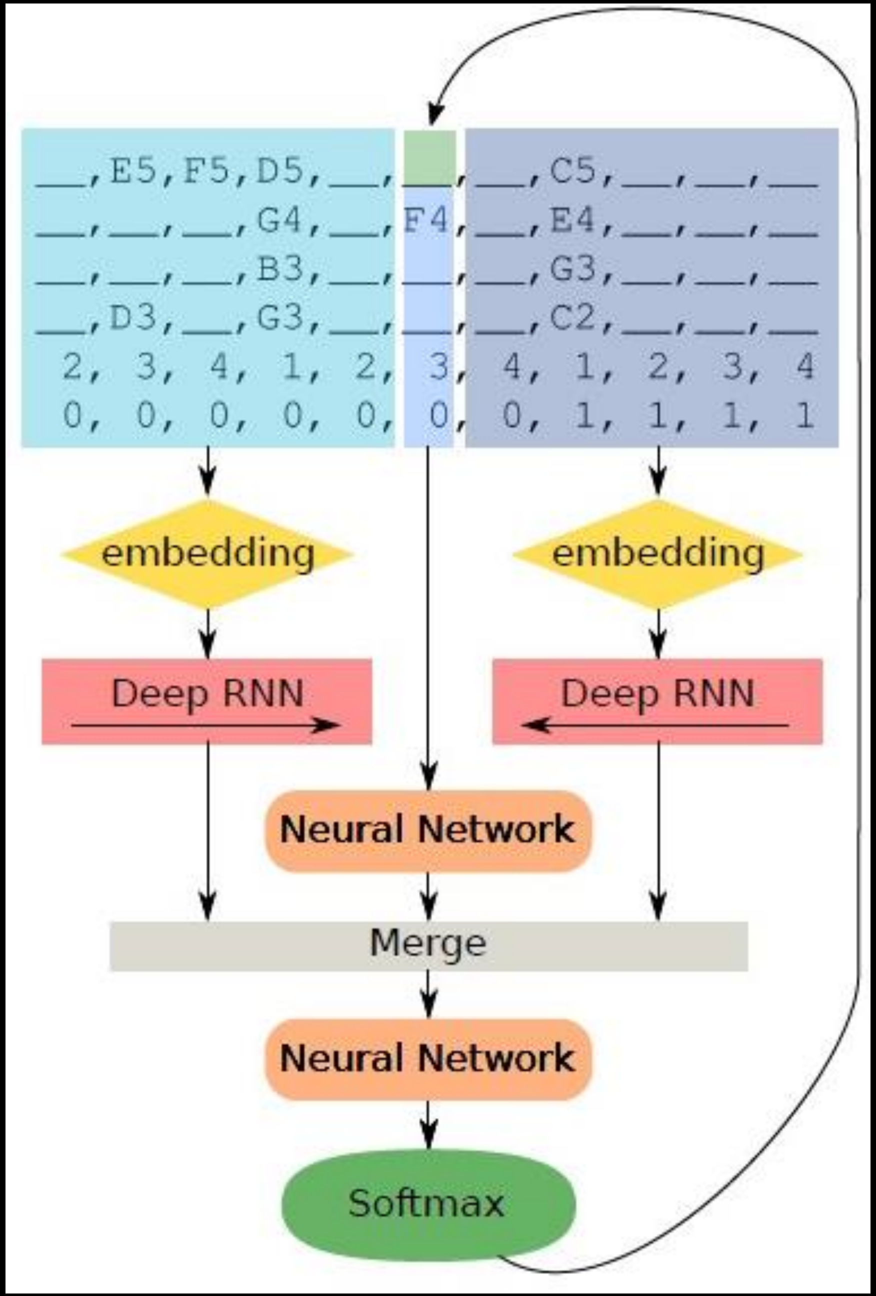
With $q(z|x_i)$ an estimator of $\frac{p(z|x_i)}{p(z)}$



Source: Mike Wu’s paper

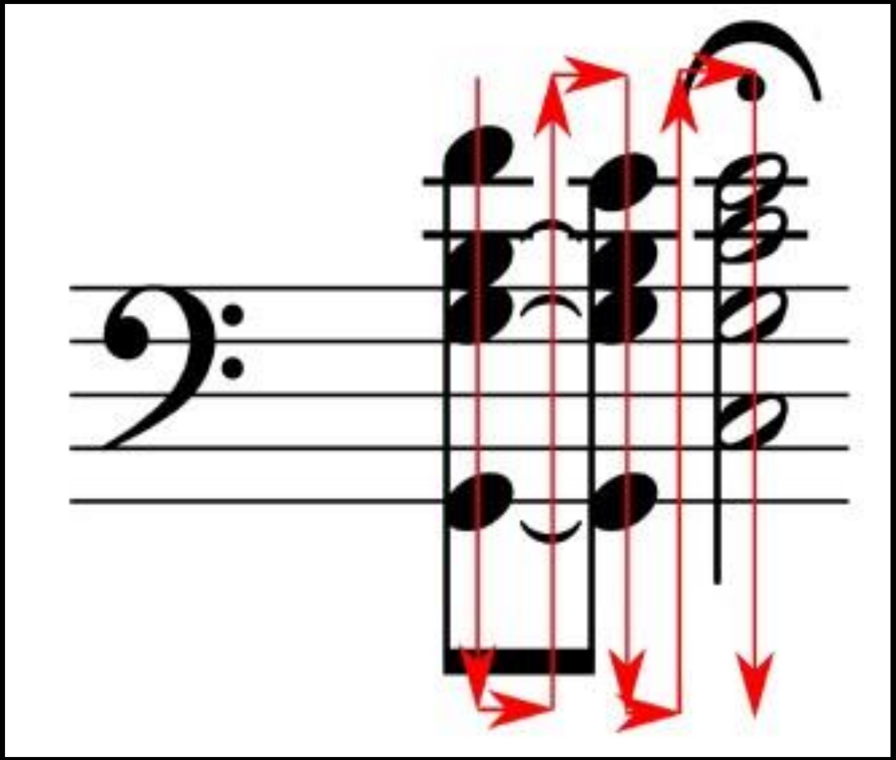
DeepBach

DeepBach architecture:



Source: Gaetan Hadjeres' paper: DeepBach: a Steerable Model for Bach Chorales Generation

BachBot



```
START            (60, False)
(65, False)      (55, False)
(59, False)      (48, False)
(55, False)      |||
(43, False)      END
|||
(64, False)
(59, True)
(55, True)
(43, True)
|||
(.)
(64, False)
```

Source: Feynman Liang's paper: AUTOMATIC STYLISTIC COMPOSITION OF BACH CHORALES WITH DEEP LSTM

Product of Experts details

$$p(z|x_1,\dots,x_N) = \frac{p(x_1,\dots,x_N|z)p(z)}{p(x_1,\dots,x_N)} = \frac{p(z)}{p(x_1,\dots,x_N)}\prod_{i=1}^{N} p(x_i|z)$$

$$= \frac{p(z)}{p(x_1,\dots,x_N)}\prod_{i=1}^{N}\frac{p(z|x_i)p(x_i)}{p(z)} = \frac{\left(\prod_{i=1}^{N} p(z|x_i)\right)}{\prod_{i=1}^{N-1} p(z)}\frac{\left(\prod_{i=1}^{N} p(x_i)\right)}{p(x_1,\dots,x_N)}$$

$$\propto \frac{\prod_{i=1}^{N} p(z|x_i)}{\prod_{i-1}^{N-1} p(z)} = p(z)\prod_{i=1}^{N} q(z|x_i)$$

With $q(z|x_i)$ an estimator of $\frac{p(z|x_i)}{p(z)}$

ELBO

$$\mathcal{L}\left(q_\phi(z|x)\right) + \mathbb{D}_{KL}\left(q_\phi(z|x), p(z)\right)$$
$$= \int_z q_\phi(z|x)\log\big(p_\theta(x|z)\big) - \int_z q_\phi(z|x)\log\left(\frac{p(z)}{q_\phi(z|x)}\right)$$
$$= \int_z q_\phi(z|x)\log\left(\frac{p_\theta(x,z)}{q_\phi(z)}\right) - \int_z q_\phi(z|x)\log\left(\frac{p(z)}{q_\phi(z|x)}\right)$$
$$= \int_z q_\phi(z)\left(\log\big(p_\theta(x,z)\big) - \log\left(q_\phi(z)\right) - \log\big(p(z|x)\big)\right.$$
$$\left.+ \log\left(q_\phi(z)\right)\right) = \int_z q_\phi(z)\log\big(p_\theta(x)\big) = \log\big(p_\theta(x)\big) = ELBO(x)$$