

# You're your own best teacher: A Self-Supervised Learning Approach For Expressive Representations

Johan Vik Mathisen

May 19, 2024

## 0.1 Implementation details

We follow [TimeVQVAE] closely in the encoder/decoder/codebook implementation, and

### Time Frequency Modelling

$n_{fft} = 8$ , other params are default parameters of pytorch implementation of STFT.

### Encoder and decoder

Appendix C.3 from [TimeVQVAE]

Encoder block: Conv2d + Batchnorm + LeakyReLU

Convlayer has kernel size (3,4), stride (1,2) and padding (1,1) (padding mode = "replicate").

Residual block: LeakyReLU + Conv2d + LeakyReLU + Conv2d

First conv: kernel size=3, stride=1, padding=1

Second conv: kernel size=1, stride=1, padding=0

### VQ

Implementation from lucidrains/vector-quantize-pytorch. <https://github.com/lucidrains/vector-quantize-pytorch>

codebook: size = 32, dim = 64

### Projector

We follow the implementation from [lee2024computer] for both Barlow Twins and VIBCReg.

## 0.2 Initial Experimentation and Model Development

The overarching objective in creating our model is to learn more expressive latent representations for better time series generation. We want to improve the reconstruction capabilities of the tokenization model. The rationality is that if the tokenization model reconstructs well the latent representations contains all relevant information of the input. We simultaneously want enforce better class separability in the latent representations, as we hypothesize that such additional structure eases/improved learning of the generative model.

During development we encountered several problems:

When we attempted a siamese architecture, with quantization in the augmented branch, and to derive the SSL loss from the discrete representations there were a correlation problem. The codewords were very highly correlated, which resulted from the passing both views through the VQ.  $SSL(z_q, z'_q)$

In an attempt to solve this we attempted to derive the SSL loss from the continuous latent representations, but the resulting discrete latent representations performed poorly on the downstream classification task. Separability problem:  $SSL(z, z')$

The solution was to remove the VQ in the augmented branch and rather derive the SSL loss from  $z_q$  and  $z'$ . Solution:  $SSL(z_q, z')$

Overfitting problem: Using  $SG()$  on augmented branch / Not using augRecons

### 0.3 Main Experiments

We are primarily interested in two things. For stage 1, if NC-VQVAE learns more expressive representations, i.e are we able to reconstruct on par with VQVAE while simultaneously improve on downstream classification. For stage 2 we are interested in the effect NC-VQVAE has on prior learning and time series generation.

We evaluate our model NC-VQVAE against the naive VQVAE as described in [TimeVQVAE]. Firstly we look at the tokenization models, evaluating the reconstruction capability and performance on downstream classification. Then we train a prior model on top of the different tokenization models and evaluate the performance of the generative models by IS, FID, CAS and visual inspection.

Additionally we provide ablations investigating effect of different augmentations and the effect of augmentations reconstruction weight.

### 0.4 Stage 1

Naive VQVAE as described in [TimeVQVAE] is baseline.

BarlowTwins and VlbCReg as SSL method

Trained using 1000 epochs in both stages.

Batch size:

Nr of runs and Seeds:

Augmentations used:

### 0.4.1 Augmentations

Through initial testing we settled on consequently using the augmentations amplitude resizing + window warp for the augmented branch, while leaving the original branch unaltered. Leaving the original branch unaltered improved reconstruction error. Our final choice in augmentations is biased by the results on datasets we experimented on initially, which were FordA, UWavesGestureLibraryAll, TwoPatterns and ShapesAll. Of those amplitude resizing + window warp was a top performer, except for FordA where slice and shuffle did well. When amplitude resizing + window warp produced quite good results, while simultaneously not altering the semantics of the data or introducing high frequency components, we settled on that for ease of experimentation. We provide a small ablation on the effect of augmentations in section ??.

- Amplitude Resizing + Window Warp
- Slice and Shuffle
- Gaussian noise

### 0.4.2 Evaluation

Reconstruction, Classification and Visual inspection

## 0.5 Stage 2

In order to make use of the additional information in the latent representations from stage 1 our MaskGIT models the probabilities of codebook indices conditional on the codewords.

### 0.5.1 Evaluation metrics

- IS:
- FID:
- CAS:
- Visual inspection:
- Token usage:

## 0.6 UCR Time Series Classification Archive

The evaluation of our model NC-VQVAE is done on a subset of the UCR Time Series Archive [UCRArchive2018]. The UCR archive is a collection of 128 datasets of univariate time series for time series classification. The different datasets

in the archive span a wide range characteristics, sensor, device, image-derived, simulated, . Each dataset has a predefined training and test split.

Our subset of the UCR archive is

**TODO:** Table of datasets