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

Johan Vik Mathisen

May 31, 2024

Our work in this thesis builds upon the paper "Vector Quantized Time Series Generation with a Bidirectional Prior Model" [**TimeVQVAE**]. We simplify the model architecture by omitting the high-low frequency split, reducing the model to what they refer to as naive TimeVQVAE in their paper. We expand upon naive TimeVQVAE by integrating a self-supervised learning (SSL) extension.

A schematic figure of our proposed tokenization model is given in Figure **??**. To improve reconstruction, we add a regularizing term by reconstructing augmented views. We hypothesize that this approach enables the model to generalize better to unseen data by allowing the decoder to "see" the augmented views. Additionally, to improve class separation, we introduce a non-contrastive self-supervised loss. The intuition is that the SSL loss pushes the representations of original and augmented views closer together.

0.1 Proposed model: NC-VQVAE

Our model, termed NC-VQVAE, is a generative time series model that learns expressive discrete latent representations by combining VQVAE [**VQVAE**] with non-contrastive SSL algorithms. NC-VQVAE uses the two-stage modeling approach presented in [**TimeVQVAE**] and can be considered an extension of their "naive TimeVQVAE." Our model primarily extends the tokenization stage, incorporating Barlow Twins [**zbontar2021barlow**] and VIbCReg [**lee2024computer**] as our non-contrastive SSL methods, while the framework remains flexible. For the second stage, we model the prior using a bidirectional transformer similar to MaskGIT [**chang2022maskgit**].

0.1.1 Stage 1: Tokenization

The architecture of the tokenization model, shown in Figure **??**, consists of two branches: the original and augmented branch. The model takes a time series x as input and creates an augmented view x'. The original branch follows the naive TimeVQVAE architecture from [**TimeVQVAE**], while the augmented branch is an autoencoder, constructed by omitting the quantization layer. The views are passed through their respective branches, and we compute the SSL loss derived from the original discrete latent representation z_q and the augmented continuous latent z', before the decoder reconstructs each latent representation.

The SSL loss is calculated by concatenating the global average and max pool of both representations individually and passing the resulting vectors through the projector.

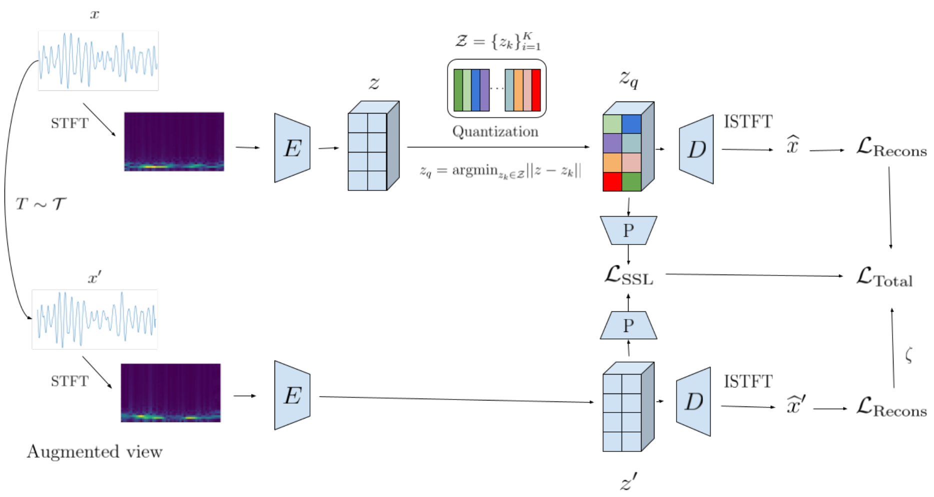
x
STFT
$T \sim \mathcal{T}$
x'
STFT
Augmented view
E
z
$\mathcal{Z} = \{z_k\}_{i=1}^K$
Quantization
$z_q = \text{argmin}_{z_k \in \mathcal{Z}} ||z - z_k||$
z_q
D
ISTFT
$\hat{x}$
$\mathcal{L}_{\text{Recons}}$
P
$\mathcal{L}_{\text{SSL}}$
$\mathcal{L}_{\text{Total}}$
P
ζ
E
D
ISTFT
$\hat{x}'$
$\mathcal{L}_{\text{Recons}}$
z'

Figure 1: Overview of proposed model. NC-VQVAE. Fix index on Z, add weight to arrow from ssl loss to total. fix argmin index thing.

Loss

Our training objective mirrors the training objective from TimeVQVAE in equation **??**, without the frequency split. Our contribution is the addition of an SLL loss together with a reconstruction loss on the augmented branch.

To refresh the reader's memory, the VQ loss consists of a reconstruction loss and a codebook loss, which is the MSE of continuous and discrete latent representations, along with a commitment loss to prevent the codewords from diverging. In our setup, the codebook loss reduces to

$$\begin{aligned}\mathcal{L}_{\text{codebook}} = &\, || \, \text{sg}[z_q] - z_q||_2^2 \\ &+ \beta ||z_q - \text{sg}[z_q]||_2^2,\end{aligned} \tag{1}$$

and the reconstruction loss to

$$\mathcal{L}_{\text{recons}} = ||x - \hat{x}||_2^2 + ||u - \hat{u}||_2^2. \tag{2}$$

Our VQ loss is then given by

$$\mathcal{L}_{\text{VQ}} = \mathcal{L}_{\text{codebook}} + \mathcal{L}_{\text{recons}}. \tag{3}$$

The SSL loss varies depending on the SSL method used. It is calculated on derived values from z_q and z'. We consider Barlow Twins **??** and VIbCReg **??**, both of which utilizes a projector. We apply a global average and max pool operation on both tensors and pass them through the projector before calculating the SSL loss.

The augmented reconstruction loss is simply given as

$$\mathcal{L}'_{\text{recons}} = ||x' - \widehat{x}'||_2^2 + ||u' - \widehat{u}'||_2^2. \tag{4}$$

This loss ensures that the encoder and decoder reconstruct the augmented view, which, in conjunction with the SSL loss, influences the codebook to encode information regarding the augmentations. Additionally, it helps prevent the encoder from ignoring reconstruction in favor of the SSL loss. Initial tests showed that omitting the augmentation reconstruction led to severe overfitting.

The total loss is given by

$$\mathcal{L}_{NC-VQVAE} = \mathcal{L}_{VQ} + \eta\mathcal{L}_{\text{SSL}} + \zeta\mathcal{L}'_{\text{recons}}, \tag{5}$$

where η and ζ are hyperparameters influencing the importance of each term in the total training objective.

0.1.2 Stage 2: Prior learning

In our model, the input embedding for the bidirectional transformer is initialized with the codebook, which has learned structure from both reconstruction and SSL loss. Instead of introducing an additional masking vector in the embedding matrix, we use the codebook directly and create a separate learnable masking vector to mask the embedded sequences. In order to separate this masking vector, we do the embedding stage before masking, effectively factoring the embedding out of the transformer. This approach ensures that the learning of the masked token embedding is independent of the other embeddings, further leveraging the learned codewords from stage 1 without unnecessary influence. Except for this adjustment, and the possibility of class conditional sampling from TimeVQVAE, our method is equivalent to MaskGITs.

The process for generating samples at inference time follows the same iterative steps as MaskGIT, ensuring robust and high-quality sample generation.