A Neuromorphic Inspired VAE: To Latent Space & Beyond

Sonal Bihani, 21032559 [1] **Callum Takasaka, 21000024** [1]

Abstract

This project proposes a modification to the variational autoencoder (VAE) framework by implementing it as a fully-spiking neural network using the unified gated leaky integrate-and-fire (GLIF) neurons (Yao et al., 2022). GLIF neurons introduce other learnable parameters to better represent complex processes occurring within each neuron and allow for the connection of neuronal activity throughout the encoding and decoding portions of the VAE model. We also introduce the use of a Bernoulli distribution to represent the latent space, allowing for autoregressive Bernoulli spike sampling, a method taken from the fully-spiking VAE proposed by (Kamata et al., 2021). The modification builds on previous research that has used the VAE to model the visual cortex, suggesting potential similarities between the VAE and the underlying biological processes of vision in the brain.

1. Introduction

The variational autoencoder (VAE) framework (Kingma & Welling, 2022) has been regarded as a stable generative model, with many proposed modifications, most typically to the loss function of the model itself. Variational autoencoders encode images into latent spaces, typically in the form of a Gaussian distribution. (Higgins et al., 2017) postulate that the disentanglement of this latent space can be explored to provide interpretable latent factors. Their β-VAE model aims to further optimize the disentanglement of these latent factors, separating the dimensions of the latent space, and ensuring that an encoding distributes information in a more uniform manner. Many papers have followed, such as (Burgess et al., 2018) and (Chen et al., 2019). These papers explore different possible loss functions which may result in greater disentanglement. While models of this kind often do show improvements over the classical VAE, they do not modify the architecture of the VAE in a novel way.

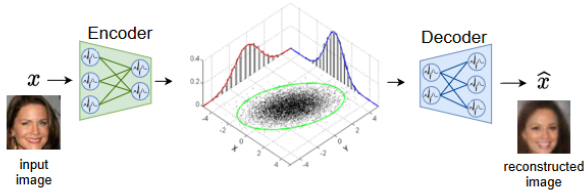

Figure 1. Simple outline of a typical VAE structure

The VAE has previously been used to model the visual cortex (Han et al., 2017), introducing some idea that the VAE may have similarities with the underlying biological processes of vision in the brain. To follow this idea, we implement the VAE as a fully-spiking neural network, first proposed by (Kamata et al., 2021), using unified gated leaky integrate-and-fire (GLIF) neurons (Yao et al., 2022). While (Kamata et al., 2021) implemented a fully-spiking VAE, the neuron used lacks many biological processes observed in actual neurons. GLIF neurons introduce other learnable parameters which better represent the complex processes occurring within each neuron and allow for the connection of neuronal activity throughout the encoding and decoding portions of the VAE model, respectively. We propose that allowing
various different ways will increase the overall density of information that can be stored within the model.

Similar to (Kamata et al., 2021), we represent the latent space as a Bernoulli distribution, to allow for sampling even when our GLIF neurons can only output binary values. This introduces other benefits, as noted by (Kamata et al., 2021), such as sampling without requiring random float value generation. This process, autoregressive Bernoulli spike sampling, is further explained by (Kamata et al., 2021).

[*]Equal contribution [1]Department of Mathematics, University of Waterloo, Waterloo ON, Canada. Correspondence to: Sonal Bihani <sbihani@uwaterloo.ca>, Callum Takasaka <ctakasa@uwaterloo.ca>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

2. Related Works

2.1. A Rundown on SNNs

In a typical artificial neural network (ANN) model, non-linear activation functions such as the ReLU, and sigmoid functions, are used to allow the model to discover various features within data.

SNN models use neurons that are modeled according to the processes observed in real neurons. (Maass, 1997) defines the SNN as a type of neural network which uses spiking neurons, a simplified model of real neurons, which provide output spike-trains. These spike trains are records of when a neuron's membrane potential has exceeded some threshold and represent when a signal is sent to further connected neurons. The accumulation of these signals within a network of neurons is shown to have similar computational power to ANNs of similar sizes, albeit with more complex learning functions.

Training is typically achieved by way of back-propagation through time, where each spike within a spike train is unrolled and changes are systematically propagated through them. We utilize this training method, along with some approximations of the Heaviside function, provided by (Yao et al., 2022).

2.2. The Gated Leaky Integrate-and-Fire Neuron

(Yao et al., 2022) describe the GLIF neuron, as a more dynamic LIF model which includes multiple bio-features (bio standing for biologically plausible). This is accomplished through the addition of multiple gates within each neuron, whose gating factors are learned, and are computed by sigmoid to allow derivation. These gating factors control things like membrane potential leakage (both linear and exponential), integration accumulation, and spike initiation. All gating factors can be set to a fixed state, to also allow the testing of individual features. This is explored in our ablation study.

While these neurons allow for greater neuronal dynamics, we suspect that they may result in less stability throughout the training process. (Yao et al., 2022) provide some features which may aid this process, including ways to share parameters across neurons that fire together, either layer-wise or channel-wise.

2.3. SNN Based VAE

The fully-spiking VAE proposed by (Kamata et al., 2021) provides a strong framework on which we can build our model. The way in which they defined sampling neuronal spike-trains as a Bernoulli distribution proved to be a good method.

Table 1. Hidden dimensions in each layer of our encoder.

LAYER #	HIDDEN DIMENSIONS
1	32
2	64
3	128
4	256
5	512

We explored various different VAEs, however in the interest of comparability, we provide example reconstructions and samples from a β-VAE (Higgins et al., 2017) model. The DIP-VAE proposed by (Kumar et al., 2018) shows better results than the β-VAE, but this model uses a complex loss function which would make it difficult to understand what effect the GLIF neuron is having on the model.

3. Model Architecture

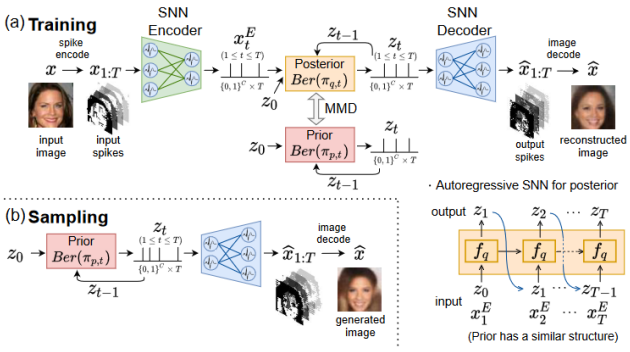

Figure 2. Image from (Kamata et al., 2021) This figure represents the general blocks within our model, save the use of MMD.

3.1. Input

Our models take in a 64x64x3 (RGB) image. In the case of the CelebA dataset, we follow the common practice - taking a 64x64 crop from the center of the image. This input is passed to the encoder as T timesteps, where T is the length of each spike-train generated by our neurons. For the majority of our models, T is set to 5. This is a significantly lower value than used by (Kamata et al., 2021), which was necessary for training feasibility.

3.2. Encoder

The encoder in each model consists of a 5-layer CNN structure. Each layer uses a 3x3 kernel, with padding to maintain image size throughout. A table of layer dimensions is included (Table 1). All activations are GLIF neurons. Batch normalization is applied after each layer.

Table 2. Hidden dimensions in each layer of our decoder.

LAYER #	HIDDEN DIMENSIONS
1	512
2	256
3	128
4	64
5	32

3.3. Latent Space & Sampling

In order to transform our encoder output into linear spike trains for our latent space, we apply a linear layer with GLIF neurons as activation. This fully connected layer reduces the hidden dimensions of our spike trains to 128.

To model the posterior and prior probability distributions, we will use SNNs and define them as Bernoulli distributions which take binary values. This is necessary because all SNN features must be binary time-series data. In contrast to the conventional VAE, we cannot use the reparameterization trick to sample from the normal distribution.

$$q(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} q(z_t|x_{\leq t}, z_{<t}) \tag{1}$$

$$p(z_{1:T}) = \prod_{t=1}^{T} p(z_t|z_{<t}) \tag{2}$$

To generate z_t sequentially from $z < t$, we use an autoregressive SNN model. By randomly selecting one by one from its output, we can sample from a Bernoulli distribution. A more detailed explanation for the sampling process can be found in (Kamata et al., 2021).

While the MMD loss function has been shown to be effective in learning a more diverse and representative latent space, its higher computational complexity makes it less practical for a small-scale project. As a result, we have used the widely-used KLD loss function, which efficiently learns a compressed and meaningful representation of the data in the learned latent space.

3.4. Decoder

The decoder takes sampled spike-trains from the posterior, passing them through a fully-connected layer that expands our hidden dimensions from 128, back to 512, with GLIF activations. The structure is similar to that of the encoder, consisting of a 5-layer CNN, with batch normalization. The same sized 3x3 kernel is used. For clarity, a table is provided (Table 2) showing the hidden dimensions of each layer.

3.5. Output

The final output is reconstructed from the output of the decoder. The decoder outputs a transposed result of the final layer, making it easier to access timesteps. These timesteps are passed through a "membrane output layer", where the membrane potential of the final neurons is calculated after the spike-trains provided by the decoder are passed through. This allows us to effectively transform the (binary) spike-trains into float values which will then be interpreted as our image.

3.6. Additional Information

Throughout our model, we only use GLIF neurons, with all parameters set to be learned. This ensures that our model is fully-spiking, and can make full use of all neuronal complexity afforded by the GLIF neuron.

In training the model, we use back-propagation through time similarly to (Yao et al., 2022), using two different gradient approximations for the Heaviside step function. The first is a linear approximation (eq. 3), and the other is shown as eq. 4, which will be called the triangle approximation. We also use channel-wise parameter sharing between neurons in each portion of the model, which reduces parameter count, and also allows for intra-channel relationships. We propose that this may improve disentanglement by enforcing stricter relationships while producing encoded embeddings within the encoder portion of our model.

$$\frac{d\mathcal{H}(x)}{dx} = \mathcal{H}(0.5 - \mid x \mid) \tag{3}$$

$$\frac{d\mathcal{H}(x)}{dx} = \mathcal{H}(\max(1 - \mid x \mid, 0)) \tag{4}$$

While (Kamata et al., 2021) use a spike-train length (T) of 16 timesteps in their fully-spiking VAE model, we reduced the length of the spike-train to 5 timesteps in the interest of training feasibility. We will briefly talk about the ramifications of this modification in the experimentation & ablation portions of this report.

3.7. Hybrid ANN & SNN Model

While attempting to implement the autoregressive Bernoulli spike sampling technique from (Kamata et al., 2021), we began to implement a variant of our model, which utilized a hybrid SNN & SNN architecture (SNN Encoder, ANN Decoder). This model suffered from sampling issues, as we had not properly implemented the previously mentioned sampling technique correctly at the time, however its ability to reconstruct images given low training time was very impressive. Due to the pervasiveness of bugs within the hybrid model, we decided to exclude it from our report.

4. Experimentation

We trained various models on the CelebA dataset. Initially, we had planned to also train on the Animal Faces, however, due to a lack of training resources, and time constraints, we were unable to do so.

The models trained were given 6 GPU hours each unless otherwise specified. This is much more than required for an equivalent ANN VAE, which is expected due to the fact that SNN models must be trained via back-propagation through time, and must perform significantly more forward passes. 20 epochs were allotted for each model, which is a far lower amount than (Kamata et al., 2021) use. The fact that we are using GLIF neurons with many learned parameters, compared to classical LIF model-based neurons, also exacerbates the effects of this lack of training, since each neuron contributes to a less (initially) stable learning environment.

4.1. Results

We provide plots of the loss values during the training of our models, as well as several reconstructions for each model. While it is difficult to quantify the differences between the tested models, it is clear that several are capable of producing better samples than others.

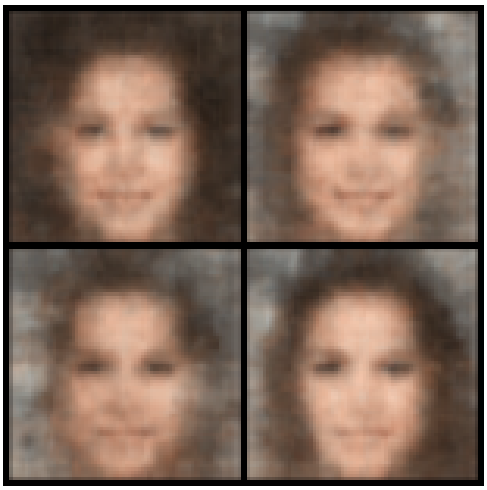

Figure 4. Samples generated from our GLIF SNN VAE T = 5, linear gate gradient approximations

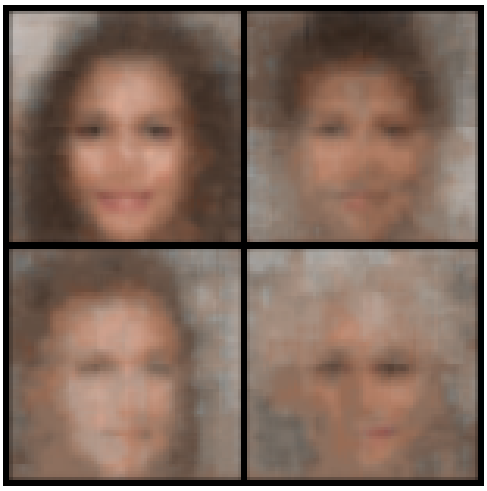

Figure 3. Samples generated from our GLIF SNN VAE T = 3, linear gate gradient approximations

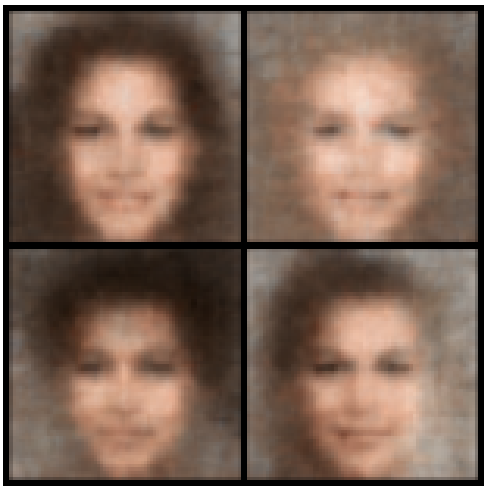

Figure 5. Samples generated from our GLIF SNN VAE T = 5, triangle gate gradient approximations

It is important to note that the absolute performance of these models should not be assumed from this report, as the models were not provided with sufficient training epochs to perform optimally. Not only this, but the low T (spike-train length) value, as stated in the "Model Architecture" section, has a profound effect on model performance as well. Due to these factors, we only will make observations about each model tested in relation to each other.

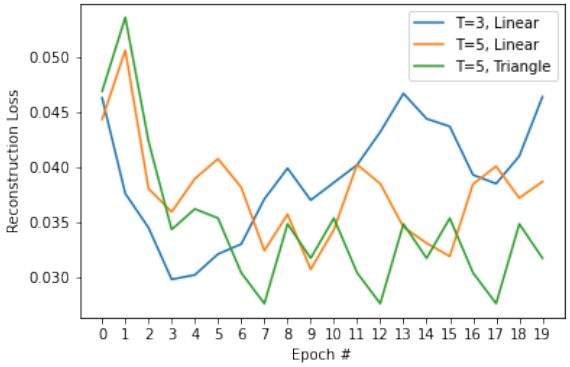

Figure 6. Image Reconstruction loss across epochs, for our models

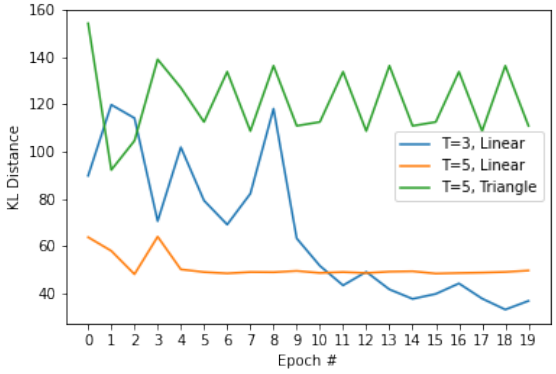

Figure 7. KL Divergence across epochs, for our models

When comparing models by number of timesteps, it can be noticed that the samples produced by the model using 3 timesteps (Fig. 3) contain less detail than the models using 5 timesteps. This is expected, as fewer timesteps means shorter spike-trains, and therefore less time for neurons to express information.

There seems to be less pixelation in the samples produced under the model which was trained using the triangle gradient approximation for the heaviside function (Fig. 5), when compared to the model which used a linear approximation (Fig. 4). These models were trained over the same number of epochs, and with the same number of GPU hours (6).

Loss graphs show some interesting behaviour among the models. We can see that a higher timestep value results in lower reconstruction loss (Fig. 6), which agrees with our earlier statement that smaller spike-trains are unable to sufficiently learn the reconstruction task. KL divergence however, is minimized by the lower timestep model (Fig. 7), implying a clear tradeoff between the two factors within our VAE loss function. This is a well-known problem with using KL divergence within VAE loss, and is to be expected. This can be aleviated by using the MMD loss which (Kamata et al., 2021) use. Again, this behaviour may not remain constant given sufficient training time, and when the timestep values are scaled up.

4.2. Ablation Study

To better understand the effects of the learned parameters in the GLIF neurons, we train a VAE following the same architecture outlined in the "Model Architecture" section, with neurons which have static parameters. Any models trained by (Kamata et al., 2021) would not work in this case, since they have approximately 650 GPU hours of training, and are trained with far more resources than our models. This would make the comparison uninformative.

We fix any learnable parameters within the GLIF neurons, ensuring that no learning occurs within. This effectively turns each neuron into an LIF neuron. We provide some samples from this model.

It is important to note that we may have introduced a bug while implementing this static model. However, the provided graphics may still be informative.

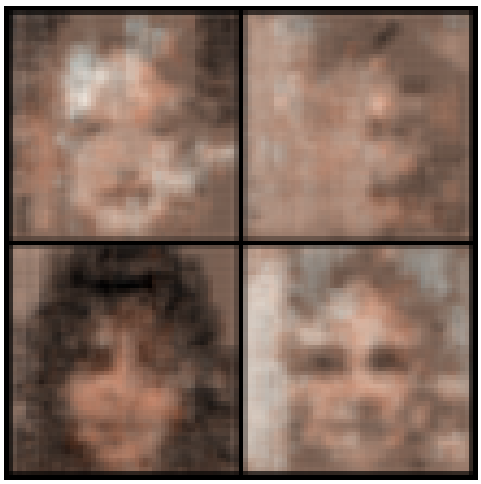

Figure 8. Static Model Samples

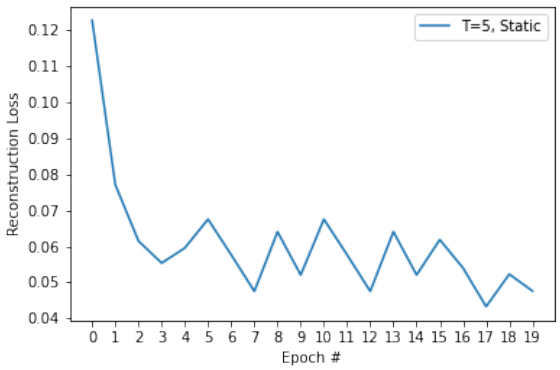

Figure 9. Static Model Image Reconstruction loss across epochs

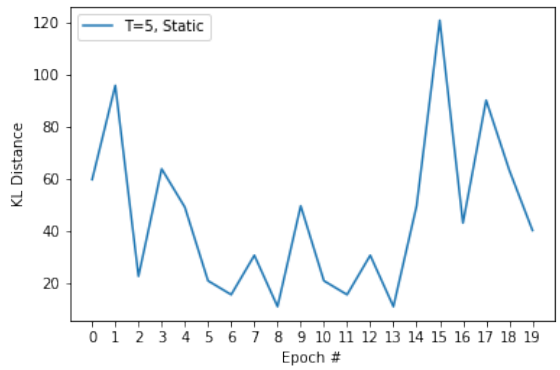

Figure 10. Static Model KL Divergence across epochs

5. Conclusion

We have presented a modification to the VAE framework by implementing it as a fully-spiking neural network using GLIF neurons and autoregressive Bernoulli spike sampling. The proposed modification builds on previous research that has used the VAE to model the visual cortex, suggesting potential similarities between the VAE and the underlying biological processes of vision in the brain.

We trained various models on the CelebA dataset and demonstrated that our proposed modifications can produce high-quality images if trained for a significant number of epochs, and with longer spike-trains. While our models require more training time than an equivalent ANN, they provide the advantage of learning a more diverse and representative latent space. The code for the project can be found on Github.

6. Further Work

Given more resources, it would be interesting to see the capabilities of a GLIF driven SNN VAE when trained for a sufficient number of epochs. The multitude of learnable dynamics within the model may give way to a VAE which outperforms other ANN VAE models.

The hybrid ANN & SNN architecture mentioned earlier in the report may be useful as an encoding model when used concurrently with other models which would benefit from such behaviour.

For all of the GLIF models showcased in our report, a potential use-case would be using them as part of an image recognition or segmentation model. By combining a GLIF VAE with a linear head, it may be possible to use computed encodings provided by a trained (or semi-trained) GLIF encoder to attempt computer vision tasks. These encodings are unique in that they are generated by a large variety of learned dynamics, and may provide more information than is typically available. The fact that these encodings can be used as spike-trains, and therefore can be interpreted as series, means that the encodings may provide layer-wise information which could help in the segmentation of objects within an image based on latent features.

References

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in β-vae, 2018.

Chen, R. T. Q., Li, X., Grosse, R., and Duvenaud, D. Isolating sources of disentanglement in variational autoencoders, 2019.

Han, K., Wen, H., Shi, J., Lu, K.-H., Zhang, Y., and Liu, Z. Variational autoencoder: An unsupervised model for modeling and decoding fmri activity in visual cortex. *bioRxiv*, 2017. doi: 10.1101/214247. URL https://www.biorxiv.org/content/early/2017/11/05/214247.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Kamata, H., Mukuta, Y., and Harada, T. Fully spiking variational autoencoder, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes, 2022.

Kumar, A., Sattigeri, P., and Balakrishnan, A. VARIATIONAL INFERENCE OF DISENTANGLED LATENT CONCEPTS FROM UNLABELED OBSERVATIONS. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1kG7GZAW.

Maass, W. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(97)00011-7. URL https://www.sciencedirect.com/science/article/pii/S0893608097000117.

Yao, X., Li, F., Mo, Z., and Cheng, J. GLIF: A unified gated leaky integrate-and-fire neuron for spiking neural networks. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=UmFSx2c4ubT.