VADA: a Data-Driven Simulator for Nanopore Sequencing

Jonas Niederle$^{1[0009-0009-9854-0489]}$, Simon Koop$^{1[0009-0003-2673-6104]}$, Marc Pagès-Gallego$^{23[0000-0001-8888-5699]}$, and Vlado Menkovski$^{1[0000-0001-5262-0605]}$

^{1} Eindhoven University of Technology, Eindhoven, The Netherlands
^{2} Oncode Institute, Utrecht, The Netherlands
^{3} Center for Molecular Medicine, UMC Utrecht, Utrecht, The Netherlands

Abstract. Nanopore sequencing offers the ability for real-time analysis of long DNA sequences at a low cost, enabling new applications such as early detection of cancer. Due to the complex nature of nanopore measurements and the high cost of obtaining ground truth datasets, there is a need for nanopore simulators. Existing simulators rely on handcrafted rules and parameters and do not learn an internal representation that would allow for analysing underlying biological factors of interest. Instead, we propose VADA, a purely data-driven method for simulating nanopores based on an autoregressive latent variable model. We embed subsequences of DNA and introduce a conditional prior to address the challenge of a collapsing conditioning. We introduce an auxiliary regressor on the latent variable to encourage our model to learn an informative latent representation. We empirically demonstrate that our model achieves competitive simulation performance on experimental nanopore data. Moreover, we show we have learned an informative latent representation that is predictive of the DNA labels. We hypothesize that other biological factors of interest, beyond the DNA labels, can potentially be extracted from such a learned latent representation.

Keywords: nanopore sequencing · generative AI · computer simulation · autoregressive models · latent variable models

1 Introduction

DNA contains the genetic instructions needed for all living organisms to grow, reproduce, and function. Nanopore sequencing is an emerging DNA sequencing technique, which allows real-time analysis of long sequences of DNA, has low costs, is portable, and requires little preparation time, in steep contrast to traditional DNA sequencing approaches, which are costly, can only process short sequences of DNA and require much more preparation and processing time. These advantages make nanopore sequencing suitable to, for example, be used for early detection and treatment of cancer [13,12]. Furthermore, during a pandemic, nanopore sequencing can be used for rapid detection of virus mutations [12].

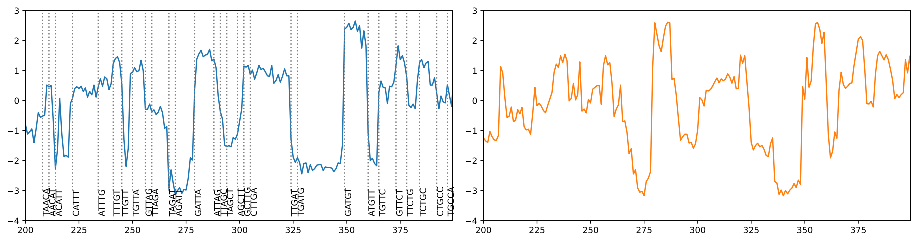

Fig. 1: An example of experimental nanopore signal (left) with aligned k-mers and generated signal for the same k-mer alignment (right).

Nanopore sequencing works by applying current to a tiny hole, a nanopore, passing a sequence of DNA through it, and measuring the resulting change in current. DNA *bases*, A, C, G, and T, make up the individual elements of a DNA sequence, and, a *k-mer* refers to a subsequence of length *k*. Each DNA base affects the current signal differently, thus capturing the change in electrical current allows the identification of the DNA sequence.

Determining the DNA sequence from the current measurements is challenging because of several reasons. Firstly, multiple bases are in the nanopore simultaneously. Therefore, the change in observed electrical current depends on multiple bases, i.e. a k-mer [14]. Secondly, the DNA sequence moves through the nanopore at a non-constant speed [14]. This causes variability in the number of nanopore measurements that corresponds to one k-mer. Therefore, multiple *timewarped* versions of a current sequence can occur.

Lastly, besides the sequence of bases, other exogenous variables influence the observed current. For example, additional chemical modifications, such as methylation, can occur on top of the four canonical bases, influencing the biological function of DNA and the resulting current.

The left plot in Figure 1 shows an example of Nanopore signal with an aligned sequence of k-mers. Here one can clearly see the variability in nanopore current and speed of DNA moving through the nanopore.

As a result of this complexity, Machine Learning-based predictive models are used to determine the sequence of bases from the raw current measurements. This process is referred to as *basecalling* and is an essential step for most downstream applications [18].

Besides basecalling, new methods are rapidly being developed to analyse nanopore data for downstream tasks [2,10]. Such new methods require labeled data for benchmarking and potentially also for training. Although evaluation on empirical data is important to guarantee the quality of any method, such empirical data is costly and ground truth data is hard to come by: obtaining ground truth labels involves sequencing the same DNA using an orthogonal sequencing method. Simulating nanopore signals and the resulting base calls allows for cheap supplementation of empirical data for more extensive benchmarking.

To address this need, multiple simulators have been developed [10,11,1,15]. These approaches are typically implemented in two separate steps. First, a *deterministic* estimation of the expected nanopore current is produced for each k-mer. Then, Gaussian noise is added to the expected current to effectively produce a sample from a probability distribution that governs the simulation process. Importantly, the standard deviation of the noise term is often *user-defined* and *constant* for all k-mer [1,11] or estimated by a Gaussian distribution [15], and always *independent* of the number of consecutive k-mer measurements and any other context.

These assumptions about the variance in the observations do not match what we observe in experimental data. The distribution of nanopore current measurements varies greatly per k-mer, it is not normally distributed, and it depends on the number of consecutive measurements of the k-mer.

This is further illustrated by the fact that errors of discriminative basecalling models occur more frequently for some k-mers than for others [14,3]. So, existing approaches are fundamentally incapable of modeling the variability that is observed in the data.

Moreover, capturing all the sources of variation in the data is not only essential for effectively simulating nanopore sequencing, but a model that does so, can also be of use to biologists for analysing these sources of variation. For example, nanopore sequencing has been used for the detection of DNA methylation [18]. Existing approaches do not offer the ability for further analysis of underlying and potentially unknown patterns in the DNA.

To address these limitations, we develop a *data-driven nanopore simulator* based on a deep generative model. The main goal of our model is to capture the variability in nanopore current measurements that correspond to the DNA bases. Therefore, we aim to model the *distribution over nanopore current sequences*, conditioned on a given DNA sequence. As we aim to efficiently simulate the nanopores, our model needs to allow for efficient sampling from this distribution. Importantly, in contrast to current approaches, a data-driven simulator must learn to model the stochastic process of nanopore sequencing exclusively from data, and thus cannot rely on the estimation of deterministic values or make assumptions on the shape of the distribution of nanopore currents.

Accordingly, we propose a latent variable model similar to a Variational Autoencoder [9] and DIVA [7]. By introducing a latent variable, we can model high dimensional, complex distributions of arbitrary shape, while enabling us to efficiently sample multiple nanopore observations by sampling from the latent space.

To condition our model in accordance with the physical properties of nanopore sequencing, we represent the DNA sequence by embedding k-mers of DNA, as done in other machine learning tasks in this domain [16]. Initial empirical results showed that a straightforward approach to conditioning the latent variable model results in a *conditioning collapse*, where the model ignores the conditioning and produces bad samples unrelated to the DNA sequence. To overcome this problem, and effectively condition our model on a sequence of DNA, we introduce a conditional prior distribution on the latent space. Moreover, we introduce an auxiliary regressor on the latent space to encourage the model to learn a latent representation that contains information about the DNA sequence.

In this work, we propose a data-driven nanopore simulator based on a deep generative model. We summarize our contributions as follows:

- We propose the Variational Autoregressive DNA-conditioned Autoencoder (VADA), an autoregressive probabilistic model for data-driven simulation of nanopore sequencing.
- We show VADA can effectively model DNA-conditioned probability distributions over nanopore current sequences to produce varying current observations, and does so by exclusively learning from data.
- We evaluate VADA on publicly available experimental nanopore data, which was obtained by sequencing human DNA. We show our results are competitive to a non-data driven approach.
- We show VADA learns a meaningful representation of nanopore current sequences that can be used for analysis, by training a classifier on samples from the approximate posterior on the latent space, and demonstrating that we can accurately determine the DNA bases that produced the nanopore current sequence.

2 Methods

The simulation of nanopore sequencing can be described in terms a sequence of nanopore current measurements $x^0, \ldots, x^T$ and an aligned sequence of DNA k-mers $y^0, \ldots, y^T$ as sampling from a distribution $p(x^0, \ldots, x^T \mid y^0, \ldots, y^T)$ of current measurements conditioned on the aligned sequence of k-mers. Here, y^t denotes the k-mer that was (in the center of) the nanopore at time t and corresponds to nanopore measurement x^t. Our goal is to learn this distribution from data. The DNA sequence is represented as a sequence of 5-mers, as we know that approximately 5 bases are in the nanopore simultaneously and because pragmatically, there is a center DNA base in the k-mer. As an example, a single sample at time t might be described by $x^t = -0.4$ and $y^t = CATCG$.

2.1 VADA: Variational Autoregressive DNA-conditioned Autoencoder

We are interested in modeling a distribution over nanopore observations for a given DNA sequence. We approach this task by modeling windows of nanopore current sequences of length δ. We know that nanopore measurements are predominantly influenced by the k-mer currently in the pore. Additionally, because nanopore sequencing is a continuous process and DNA bases at the edge of the k-mer might be only partly in the pore, the previous window of nanopore measurements $x^{t-\delta:t}$ will affect the current window of measurements $x^{t:t+\delta}$. Therefore, we model the distribution $p(x^{0:T} \mid y^{0:T})$ autoregressively as a product of

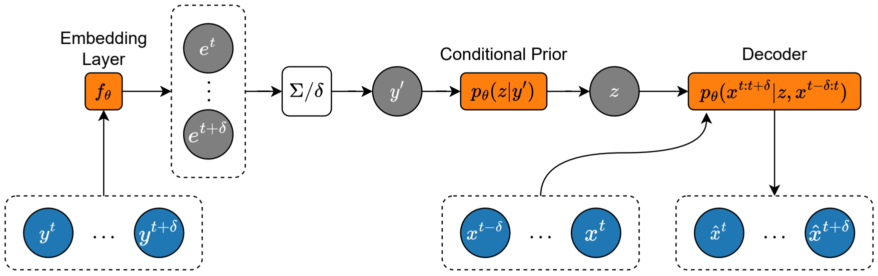

Fig. 2: VADA sampling overview.

distributions over windows:

$$p_\theta(x^{t:t+\delta}|y^{t:t+\delta}, x^{t-\delta:t}) \tag{1}$$

Where θ are the parameters of the model. For the initial window, we learn a separate distribution as $p_\theta(x^{0:\delta} \mid y^{0:\delta})$.

Latent Variable Model Our goal is to model a high-dimensional, complex distribution over windows of nanopore current sequences, without making any assumptions about the shape of the distribution. As we know the same sequence of DNA can result in a diverse set of nanopore current observations, and the model needs to capture this variability. At the same time, we want to efficiently sample multiple observations from this distribution.

Therefore, we model the problem using a latent variable model [9]. This type of model can represent a complex distribution in a high dimensional space while allowing for efficient sampling of nanopore observations, by sampling from the latent space. Specifically we model Equation 1 using latent variable z as $\int p_\theta(x^{t:t+\delta}|z, x^{t-\delta:t})p_\theta(z|y^{t:t+\delta})dz$, where $p_\theta(x^{t:t+\delta}|z, x^{t-\delta:t})$ and $p_\theta(z|y^{t:t+\delta})$ are parameterized by neural networks.

DNA representation Multiple DNA bases influence each nanopore current measurement. Consequently, as mentioned earlier, we represent y_t as a k-mer containing $k = 5$ DNA bases. The embedding layer f_θ processes each y_t in the window independently, producing embeddings $e^{t:t+\delta} = f_\theta(y^{t:t+\theta})$.

Conditioning To simulate nanopore current sequences specific to a DNA sequence, we need to condition the model. But, a straight-forward approach where the model is conditioned via the approximate posterior, i.e. the encoder, and the decoder [4] results in a *conditioning collapse*. Meaning, the model largely ignores the DNA sequence and only learns to reconstruct from a latent sample of the approximate posterior, but cannot produce realistic samples for a *given* DNA sequence by sampling from the prior distribution.

Additionally, we know the same sequence of DNA bases can result in strongly varying nanopore currents. Therefore, to enable our model to simulate this behavior and to overcome the problem of conditioning collapse, we condition our model via the prior distribution [7]. Specifically, the embedded k-mers are summed and averaged over the window producing $y' = \sum_{k=t}^{t+\delta-1} e^k/\delta$. Subsequently, y' is used to condition the prior distribution $p_\theta(z|y')$. We learn the prior distribution, which is modeled as $p_\theta(z|y') = \mathcal{N}(\mu^{\text{prior}}(y'), \sigma^{\text{prior}}(y'))$, where μ^{prior} and σ^{prior} are neural networks.

Experimental nanopore sequencing data contains an alignment of each nanopore current measurement to a specific k-mer. However, this alignment is created by using an optimization algorithm [14]. Moreover, perfectly aligning nanopore currents measured at fixed time intervals is not possible due to the continuous nature of nanopore sequencing. Thus, despite the alignment, there is variability in the correspondence of k-mer with the nanopore current sequence. Accordingly, by aggregating the conditioning we enable our model to simulate nanopore current observations with variability in time alignment of the k-mer sequence.

After conditioning, the latent sample $z \sim p_\theta(z|y')$ is processed by the decoder together with the previous window of observations $x^{t-\delta:t}$, to produce a distribution $p_\theta(x^{t:t+\delta}|z, x^{t-\delta:t})$ over the next window of nanopore observations. The entire sampling process is visualized in Figure 2.

Informative latent space Our model should not only be able to simulate nanopore current observations, but should also allow for analysis of the underlying sources of variation.

Our model inherently learns a latent representation of the nanopore currents via the latent variable z and is further encouraged to encode information about the DNA sequence into z through the use of the conditional prior [7]. Nevertheless, there is no guarantee that z captures the true underlying sources of variation, such as the DNA sequence that was in the pore. Therefore, to encourage our model to learn a representation that contains information about the DNA sequence, we make use of an auxiliary regressor during training, as done in other VAE's that aim to learn an informative latent space, such as DIVA [7]. Specifically, during training, this regressor processes a latent sample z to predict the aggregated embedding, y', that was used to condition the prior, as $\hat{y}' = r_\theta(z)$. The auxiliary regressor is included in the visualization of the training procedure (Fig. 3).

Training Following the VAE framework [9] we utilize an approximate posterior $q_\theta(z|x^{t:t+\delta})$ during training. We optimize our model using a modified β-VAE loss [6,7], where Mean Squared Error (MSE) is used to optimize the auxiliary regressor. The contribution of the MSE to the overall loss is scaled using a hyperparameter β_{aux}. The complete loss term is given by Equation 2.

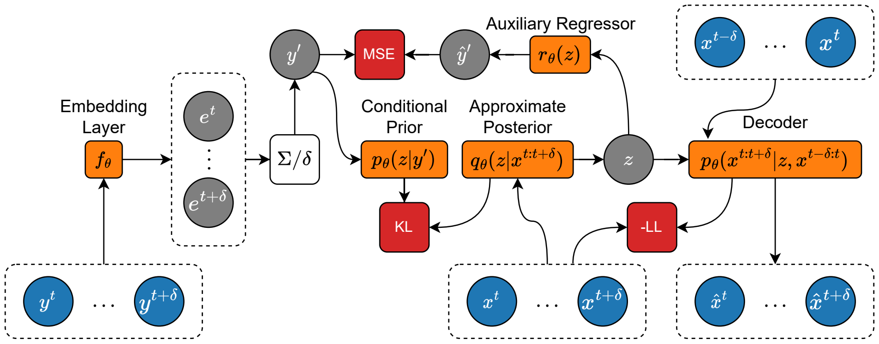

Fig. 3: VADA training overview.

$$\mathcal{L}(x^{t:t+\delta}, x^{t-\delta:t}, y^{t:t+\delta}) = \mathbb{E}_{q_\theta(z|x^{t:t+\delta})}\left[p_\theta(x^{t:t+\delta}|z, x^{t-\delta:t}) - \beta_{aux}(y' - \hat{y}')^2\right] \qquad (2)$$
$$- \beta_{KL} KL\left[q_\theta(z|x^{t:t+\delta})||p_\theta(z|y')\right],$$

where $y' = \sum_t f_\theta(y^{t:t+\delta})/\delta$ and $\hat{y}' = r_\theta(z)$.

3 Experiments

The goals of this empirical analysis are to **1)** asses VADA's effectiveness in modeling the distribution over nanopore currents, **2)** compare our data-driven approach to existing methods for nanopore simulation, **3)** determine whether the learned latent space can be used for the analysis of underlying sources of variation in nanopore current measurements.

A dataset provided by Oxford Nanopore Technologies is used for training VADA and performing the experiments. The dataset consists of sequenced human DNA and is publicly available for download on GitHub[4]. The nanopore current sequences are split into sequences of length 1000, resulting in a total of 1089009 sequences. A test set containing 5% of the available sequences is used for assessing the performance of VADA and a separate training set is used for training all models. The network architectures can be found in Appendix A and code is available on GitHub[5].

3.1 Simulation Evaluation

We simulate nanopore currents for all sequences in our test set and visualize the true and simulated current distribution for different k-mers. Specifically, Figure 4 shows simulation results for several k-mers where VADA produces current

[4] https://github.com/nanoporetech/bonito
[5] https://github.com/jmniederle/VADA

distributions that accurately match the distribution in experimental data (top row Fig. 4) and samples with worse simulation results (bottom row Fig. 4). From these qualitative results, we conclude that indeed the variability and distribution of nanopore current measurements differ between k-mers. Note that several k-mers on the bottom row contain the bases CG, a combination of bases for which methylation commonly occurs [14]. Methylation influences the resulting current measurements and can potentially explain the skewed distribution shape. On the other hand, VADA produces a distribution that closely matches the distribution of each k-mer. However, the results are not yet perfect, and for most k-mers, VADA underestimates the mode of the distribution, and for some k-mers, the tails of the distribution are not precisely modeled.

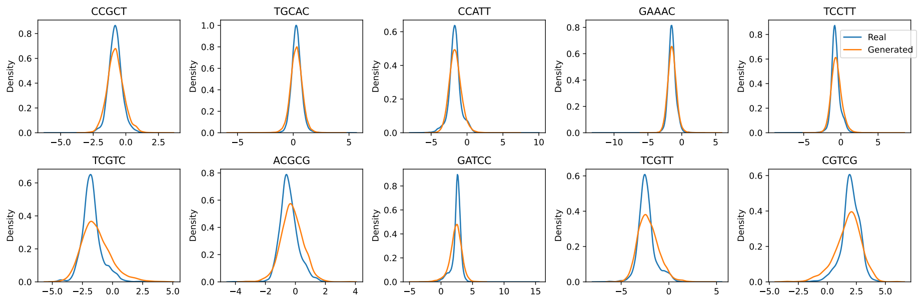

Fig. 4: Qualitative results for well-performing k-mers (top) and under-performing k-mers (bottom).

Next to investigating the results qualitatively, we quantitatively measure the performance of VADA's simulations and compare results to a state-of-the-art nanopore simulation approach that is not fully data-driven. We wish to compare simulated distribution to experimental nanopore current data, to quantify VADA's effectiveness in capturing k-mer-specific variability. We utilize the Kolmogorov-Smirnov (KS) test statistic D_{KS} as a metric to compute the similarity between the distribution of measurement values conditioned on the k-mer for simulated samples versus the corresponding distribution of experimental samples.

D_{KS} measures the largest absolute difference between the cumulative distribution functions, as such $D_{KS} = 0$ means the distributions correspond perfectly and $D_{KS} = 1$ means the distributions do not overlap at all. It makes no assumptions on either distribution being tested, allows for a two-sample test, and is widely used accross many disciplines.

We measure D_{KS} for each individual k-mer using VADA and the existing non-data-driven DeepSimulator [11] and report the results in the first column of Table 1. VADA performs competitively to DeepSimulator in terms of D_{KS}, thus simulations sampled from VADA match the distribution of nanopore currents of experimental data as well as simulations sampled from DeepSimulator. Furthermore, the standard deviation in terms of D_{KS} is much lower for VADA compared to DeepSimulator. From this, we conclude that VADA's ability to produce an accurate distribution is more stable across k-mers compared to DeepSimulator.

Ablation Analysis We further investigate the importance of the individual parts of architecture for good simulation performance by performing an ablation analysis. First, we evaluate an alternative to VADA that does not use a conditional prior. Instead, we use an unconditional gaussian $\mathcal{N}(0, 1)$ and condition the decoder $p_\theta(x^{t:t+\delta}|z, x^{t-\delta:t}, y')$ and approximate posterior $q_\theta(z|x^{t:t+\delta}, y')$. Second, we evaluate VADA without the auxiliary regressor. Details on the training procedure can be found in Appendix A. From the results in Table 1, we conclude that indeed conditioning via the prior is essential for achieving simulation performance competitive to the non-data-driven DeepSimulator. Furthermore, we conclude that including an auxiliary regressor does not clearly improve simulation performance and its use is only motivated by the requirement of learning an informative latent space.

3.2 Analysis of Latent Space

Furthermore, to investigate if the learned latent space can be used for the analysis of underlying sources of variation, we investigate if we can perform multi-label classification on the latent space. Specifically, we first sample a latent sample z by using the approximate posterior, i.e. the encoder, where intuitively z might contain a description of the sources of variation that resulted in nanopore current measurements $x^{t:t+\delta}$. Together with the embedding matrix θ_f of the embedding layer, z is processed by a classifier $g(z, \theta_f)$ to produce a binary class prediction for every possible k-mer. Specifically, there are $1025 = 4^5 + 1$ classes in total, representing all possible k-mer combinations of length 5 and a class for incomplete k-mers, which can occur at the start or end of a nanopore current sequence. We measure performance by computing the Area Under the ROC (AUC) on the test set and report a resulting AUC of **0.91**.We conclude that VADA has learned an informative latent representation that can be used for determining the k-mers that correspond to a window of nanopore measurements, thereby, successfully extracting one of the underlying sources of variation.

Table 1: Evaluation of VADA and DeepSimulator in terms of simulation performance, measured by mean ± standard deviation of D_{KS} (lower is better) over all k-mers.

Model	Simulation Performance (D_{KS})
DeepSimulator [11]	$\mathbf{0.11} \pm 0.06$
VADA (unconditional prior)	0.32 ± 0.10
VADA (no regressor)	$\mathbf{0.13} \pm 0.04$
VADA (ours)	$\mathbf{0.13} \pm 0.03$

4 Conclusions

In this work, we proposed VADA, a fully data-driven approach for probabilistic simulation of nanopores. We evaluate VADA on experimental nanopore sequencing data and demonstrate VADA performs competitively with existing non-data-driven approaches. Furthermore, we show VADA has learned a meaningful latent representation by demonstrating we can accurately classify k-mers from this representation. To conclude, we demonstrate the value of deep generative models for simulating nanopore sequencing and capturing the underlying sources of variability for nanopore currents.

4.1 Future Research

Although VADA does not outperform non-data-driven approaches in simulation quality outright, we do manage to match their performance without exploring the many techniques that have been developed over the years for improving the performance of VAE-like models [17]. Extending VADA with such techniques could be a fruitful direction of further research into nanopore sequencing simulation.

As we have demonstrated, the learned latent representation can capture useful information about the DNA being sequenced in experimental data. Further investigations into what sources of variability are captured could lead to new analytic methods for nanopore signals. Such investigations may enable easier detection of chemical modifications of DNA bases, leading to better understanding of their role in DNA function.

Various methods have been developed for separating sources of variability in latent space models [7,17]. Such techniques might improve the simulation capability of VADA, and enable the use of the encoder for more downstream tasks through added structure in the latent space.

References

1. Chen, W., Zhang, P., Song, L., Yang, J., Han, C.: Simulation of Nanopore Sequencing Signals Based on BiGRU. Sensors 2020, Vol. 20, Page 7244 **20**(24), 7244 (12 2020). https://doi.org/10.3390/S20247244
2. Deamer, D., Akeson, M., Branton, D.: Three decades of nanopore sequencing. NATURE BIOTECHNOLOGY **518** (2016). https://doi.org/10.1038/nbt.3423
3. Delahayeid, C., Nicolas, J.: Sequencing DNA with nanopores: Troubles and biases (2021). https://doi.org/10.1371/journal.pone.0257521
4. Doersch, C.: Tutorial on Variational Autoencoders (2016)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
6. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: International conference on learning representations (2016)

7. Ilse, M., Tomczak, J.M., Louizos, C., Welling, M., Nl, M.W.: DIVA: Domain Invariant Variational Autoencoders. Proceedings of Machine Learning Research **121**, 322–348 (2020)
8. Kingma, D.P., Ba, J.L.: Adam: A Method for Stochastic Optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (12 2014)
9. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings (12 2013)
10. Li, Y., Han, R., Bi, C., Li, M., Wang, S., Gao, X.: DeepSimulator: a deep simulator for Nanopore sequencing. Bioinformatics **34**(17), 2899–2908 (9 2018). https://doi.org/10.1093/BIOINFORMATICS/BTY223
11. Li, Y., Wang, S., Wang, S., Bi, C., Qiu, Z., Li, M., Gao, X.: DeepSimulator1.5: a more powerful, quicker and lighter simulator for Nanopore sequencing. Bioinformatics **36**(8), 2578–2580 (4 2020). https://doi.org/10.1093/BIOINFORMATICS/BTZ963
12. Lin, B., Hui, J., Mao, H.: Nanopore Technology and Its Applications in Gene Sequencing. Biosensors **11**(7) (2021). https://doi.org/10.3390/bios11070214
13. Norris, A.L., Workman, R.E., Fan, Y., Eshleman, J.R., Timp, W.: Nanopore sequencing detects structural variants in cancer. Cancer Biology & Therapy **17**(3), 246–253 (3 2016). https://doi.org/10.1080/15384047.2016.1139236
14. Pagès-Gallego, M., de Ridder, J.: Comprehensive benchmark and architectural analysis of deep learning models for nanopore sequencing basecalling. Genome Biology **24**(1), 1–18 (12 2023). https://doi.org/10.1186/S13059-023-02903-2/FIGURES/4
15. Rohrandt, C., Kraft, N., Gießelmann, P., Brändl, B., Schuldt, B.M., Jetzek, U., Müller, F.J.: Nanopore SimulatION - A raw data simulator for Nanopore Sequencing. Proceedings - 2018 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018 pp. 1536–1543 (1 2019). https://doi.org/10.1109/BIBM.2018.8621253
16. Trabelsi, A., Chaabane, M., Ben-Hur, A.: Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. Bioinformatics **35**(14), i269–i277 (7 2019). https://doi.org/10.1093/BIOINFORMATICS/BTZ339
17. Tschannen, M., Zurich, E., Google, O.B., Team, B., Lucic, M., Ai, G.: Recent Advances in Autoencoder-Based Representation Learning (12 2018)
18. Wang, Y., Zhao, Y., Bollas, A., Wang, Y., Au, K.F.: Nanopore sequencing technology, bioinformatics and applications. Nature Biotechnology 2021 39:11 **39**(11), 1348–1365 (11 2021). https://doi.org/10.1038/s41587-021-01108-x

A VADA Model Architecture

An embedding size of 64 is used for the embedding layer f_θ, the latent samples have size 32 and the prediction window size is set to 16. Models are trained for 140000 training steps, using Adam [8] with batch size 512. Loss hyperparameters are set to $\beta_{aux} = 0.03$ and $\beta_{KL} = 0.005$.

We use a ResBlock(in_channels, out_channels), consisting of: Conv1D(out_channels, 3), BatchNorm1D, Conv1D(out_channels, 3), BatchNorm1D, and finally a residual (skip) connection [5] of the input to the result.

We let the parameters of Conv1D represent (output channels, kernel size). Furthermore, ConvTranspose1D and Conv1D are used to, respectively, upsample and downsample the input.

Table 2: Architectures for the neural networks used in our experiments.

(a) Architecture for the prior $p_\theta(z|y')$. The model has two heads, one for the mean and one for the scale.

Block	Details
1	Linear(64, 64), LeakyReLU
2	Linear(64, 64), LeakyReLU
3-μ	Linear(64, 32)
3-σ	Linear(64, 32), Softplus

(b) Architecture for the auxiliary regressor $r_\theta(z)$.

Block	Details
1	Linear(32, 64), LeakyReLU
2	Linear(64, 64), LeakyReLU
3	Linear(64, 32)

(c) Architecture for decoder $p_\theta(x^{t:t+\delta}|z, x^{t-\delta:t})$. The network outputs values for the mean and scale is fixed at $1/\sqrt{2}$.

Block	Details
1	Linear(48, 64), LeakyReLU
2	ResBlock(64, 64)
3	ResBlock(64, 32)
4	UpSample
5	ResBlock(32, 32)
6	ResBlock(32, 16)
7	UpSample
8	ResBlock(16, 16)
9	ResBlock(16, 8)
10	UpSample
11	ResBlock(8, 8)
12	ResBlock(8, 4)
13	UpSample
14	Conv1D(1, 1)

(d) Architecture for encoder $q_\theta(z|x^{t:t+\delta})$. The distribution is Gaussian, with the following parameterisation.

Block	Details
1	Conv1d(4, 1)
2	ResBlock(4, 8)
3	ResBlock(8, 8)
4	DownSample
5	ResBlock(8, 16)
6	ResBlock(16, 16)
7	DownSample
8	ResBlock(16, 32)
9	ResBlock(32, 32)
10	DownSample
11	ResBlock(32, 64)
12	ResBlock(64, 64)
13	DownSample
14-μ	Linear(64, 64), LeakyReLU
14-σ	Linear(64, 64), LeakyReLU
15-μ	Linear(64, 32)
15-σ	Linear(64, 32), Softplus