DNA-Stabilized Silver Nanocluster Design via Regularized Variational Autoencoders

Fariha Moomtaheen*
Matthew Killeen*
University at Albany—SUNY
Computer Science
USA

James Oswald
Rensselaer Polytechnic Institute
Computer Science
USA

Anna Gonzàlez-Rosell
Peter Mastracco
University of California, Irvine
Materials Science and Engineering
USA

Alexander Gorovits
Regeneron Genetics Center
USA

Stacy M. Copp
University of California, Irvine
Materials Science and Engineering
USA

Petko Bogdanov†
University at Albany—SUNY
Computer Science
USA

ABSTRACT

DNA-stabilized silver nanoclusters (Ag_N-DNAs) are a class of nanomaterials comprised of 10-30 silver atoms held together by short synthetic DNA template strands. Ag_N-DNAs are promising biosensors and fluorophores due to their small sizes, natural compatibility with DNA, and bright fluorescence—the property of absorbing light and re-emitting light of a different color. The sequence of the DNA template acts as a "genome" for Ag_N-DNAs, tuning the size of the encapsulated silver nanocluster, and thus its fluorescence color. However, current understanding of the Ag_N-DNA genome is still limited. Only a minority of DNA sequences produce highly fluorescent Ag_N-DNAs, and the bulky DNA strands and complex DNA-silver interactions make it challenging to use first principles chemical calculations to understand and design Ag_N-DNAs. *Thus, a major challenge for researchers studying these nanomaterials is to develop methods to employ observational data about studied Ag_N-DNAs to design new nanoclusters for targeted applications.*

In this work, we present an approach to design Ag_N-DNAs by employing variational autoencoders (VAEs) as generative models. Specifically, we employ an LSTM-based β-VAE architecture and regularize its latent space to correlate with Ag_N-DNA properties such as color and brightness. The regularization is adaptive to skewed sample distributions of available observational data along our design axes of properties. We employ our model for design of Ag_N-DNAs in the near-infrared (NIR) band, where relatively few Ag_N-DNAs have been observed to date. Wet lab experiments validate that when employed for designing new Ag_N-DNAs, our model significantly shifts the distribution of Ag_N-DNA colors towards the NIR while simultaneously achieving bright fluorescence. This work shows that VAE-based generative models are well-suited for the design of Ag_N-DNAs with multiple targeted properties, with significant potential to advance the promising applications of these nanomaterials for bioimaging, biosensing, and other critical technologies.

*Authors contributed equally to this research.
†Corresponding Author. Email: pbogdanov@albany.edu

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539032

CCS CONCEPTS

• **Information systems → Data mining**; • **Computing methodologies → Learning latent representations**.

KEYWORDS

nanomaterials design; DNA; variational autoencoders

ACM Reference Format:
Fariha Moomtaheen, Matthew Killeen, James Oswald, Anna Gonzàlez-Rosell, Peter Mastracco, Alexander Gorovits, Stacy M. Copp, and Petko Bogdanov. 2022. DNA-Stabilized Silver Nanocluster Design via Regularized Variational Autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3534678.3539032

1 INTRODUCTION

DNA is a sequence-encoded building block for nanomaterials. By harnessing the well-understood base pairing rules of natural DNA, researchers have developed ways to engineer DNA sequences to fold DNA "origami" [29], build with DNA "bricks" [19], and wire DNA logic circuits [9]. DNA can also imbue sequence-encoded properties to the tiniest of nanoparticles: nanoclusters composed of just a few metal atoms. Of particular interest are DNA-stabilized silver nanoclusters (Ag_N-DNAs), which contain 10-30 silver atoms that are stabilized by 1 or 2 short DNA strands [10]. Ag_N-DNAs are colloidal nanomaterials that are synthesized in solution by mixing Ag atoms and DNA template strands (Fig. 1, top panel), yielding fluorescent nanoclusters with remarkable sequence-encoded properties. The DNA sequence controls the size and shape of the silver nanocluster, thereby tuning the fluorescence color of Ag_N-DNAs from blue wavelengths (~ 400 nm) to near-infrared (NIR) wavelengths (at least 1,000 nm) [6]. This bright, tunable fluorescence, combined with inherent biological compatibility and sensitivity to the local molecular environment, makes Ag_N-DNAs promising for a range of applications, from bioimaging and sensing to nanophotonics.

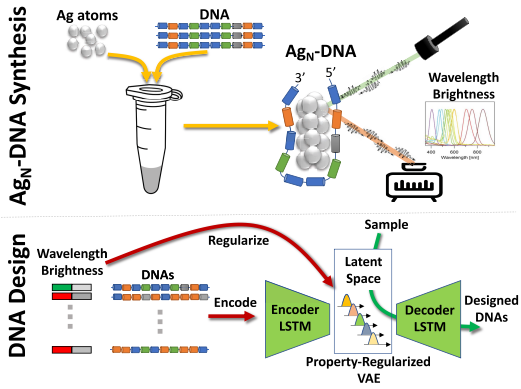


Figure 1: Overview of Ag$_N$-DNA synthesis and our regularized VAE approach for designing new DNA templates with desired wavelength and brightness. Ag$_N$-DNAs are formed by mixing Ag atoms (in the form of Ag salts) and single-strand DNA in aqueous solution, followed by gentle chemical reduction. The resulting nanocluster is a fluorophore, i.e. when excited by light, it re-emits photons with a fixed wavelength that depends on the size and shape of the cluster stabilized by the DNA strand(s) [10]. We propose to design new Ag$_N$-DNAs by training a VAE that learns to encode and decode the DNA sequence while some of its latent dimensions are regularized to correlate with the brightness and wavelength of training Ag$_N$-DNAs. To design DNA sequences for new Ag$_N$-DNAs of interest we perform truncated sampling in latent space and decode them via the trained decoder in our model.

However, a major challenge faces the development of applications of Ag$_N$-DNAs. Unlike the well-known Watson-Crick base pairing rules of natural DNA, the sequence rules that govern how DNA interacts with silver atoms and thereby select for Ag$_N$-DNA fluorescence color are not well-understood. Most researchers have used combinatorial screening or intuition to design the DNA template sequences for Ag$_N$-DNAs reported in the literature, which is a time-consuming and inefficient process. To enable data-driven approaches to map DNA sequence onto Ag$_N$-DNA color, we developed a high-throughput experimental platform for Ag$_N$-DNA synthesis and characterization, producing a library linking DNA sequences to the fluorescence colors of Ag$_N$-DNAs they stabilize. We previously utilized this library to train classifiers based on subsequence motifs to predict Ag$_N$-DNA fluorescence brightness [3] and fluorescence color [5] given an input DNA sequence. (Classification schemes are motivated by the naturally discretized properties of Ag$_N$-DNA colors [6].) We then employed the most discriminative subsequence motifs to create new DNA templates. While this approach led to discovery of new Ag$_N$-DNAs, it has several limitations: it relies on (i) discriminative as opposed to generative models to sample new DNA templates, (ii) ad hoc feature generation by sub-sequence mining, and (iii) discretization of continuous design properties like brightness and color into balanced classes.

The discovery of Ag$_N$-DNAs with NIR fluorescence emission is especially important for bioimaging applications. Biological tissues are much more transparent to NIR light than to visible light, and there is great effort to develop small, nontoxic, and bright fluorescent biolabels in the NIR spectral region. Few NIR Ag$_N$-DNAs were reported before 2018, when the discovery of 161 new NIR Ag$_N$-DNAs [7, 31, 32] suddenly presented the opportunity to extend machine learning-guided design of Ag$_N$-DNAs into the NIR. Because data in the NIR remains scarce, effective approaches to this challenging problem must be sufficiently sensitive to rare data.

In this work, we set out to address the limitations of our earlier Ag$_N$-DNA design approaches and employ our new model to enrich the space of known NIR Ag$_N$-DNAs. We propose and deploy a regularized variational autoencoder (VAE) model for the design of Ag$_N$-DNAs with desired properties summarized in Fig. 1, lower panel. Inputs to our model are sequences for synthesized Ag$_N$-DNAs and their measured wavelengths and brightness levels. We train the VAE to encode and decode DNA sequence by employing a bi-directional LSTM architecture. Instead of learning a fully latent space, we regularize a subset of its dimensions to correlate with design properties of interest. Our regularization scheme also accounts for bias in the observations along the design parameters, by compensating for rarer sample Ag$_N$-DNAs in the NIR band. We employ the trained model to design Ag$_N$-DNA template sequences by truncated sampling from latent space, thus biasing samples towards high wavelength and brightness while obeying the distribution of the remaining latent dimensions. We experimentally test the proposed VAE model on 20 new DNA sequences, finding that all of them produce Ag$_N$-DNAs with bright fluorescence and high wavelengths, including a bright NIR Ag$_N$-DNA with 845 nm peak fluorescence that has never been observed before.

Our contributions in this work are as follows:

- **Novelty.** We propose, test and deploy the first approach for rational design of Ag$_N$-DNAs with multiple continuous properties of interest via a VAE architecture.
- **Generality.** Our framework is general, in that it can extend to more design properties of interest, variable length of DNAs, and for designing other biological sequences with desired properties.
- **Applicability.** We experimentally demonstrate the utility of our approach, employing it to sample and synthesize 20 new Ag$_N$-DNAs in the lab, and discover a previously unreported NIR Ag$_N$-DNA.

2 RELATED WORK

Ag$_N$-DNA design. The vast majority of studies on Ag$_N$-DNAs employ nanoclusters designed by a combination of combinatorial screening and intuition, which is highly inefficient. To overcome these challenges, we developed high-throughput experimental synthesis and characterization of Ag$_N$-DNAs [6], producing a large training dataset that enabled early machine learning approaches based on support vector machine classifiers [4, 5, 7]. These approaches rely on bioinformatics techniques for feature engineering and discretization of a single design property into classes (e.g., high/low fluorescence yield in [4] and color in [5, 7]); as Ag$_N$-DNA colors are naturally discretized due to their structural properties, this approach is motivated by physics/chemistry [6]. Perhaps most importantly, these prior approaches rely on discriminative, as opposed to generative, models and *ad hoc* heuristics to sample from the complex space of all possible DNA sequences. The proposed VAE approach in this work addresses the above limitations: it maps both DNA sequences *and* multiple design properties into a continuous space from which one can perform truncated sampling to tune properties of interest and decode the samples into DNA sequences.

Generative models based on VAEs. Our proposed model is a generative VAE that builds on prior autoencoder (AE) research. AEs have been in use since the mid 1980s, but were initially used for dimensionality reduction and denoising, with little generative ability [11]. In 2014, Kingma and Welling proposed the Variational

Autoencoder (VAE) [22], modifying the latent space of the VAE architecture to hold latent distributions, which are then sampled during the training process. This change allows for VAEs to be used for generative tasks. A drawback of classical VAEs is the inability to control various properties of the features in latent space such as disentanglement, regularization, and monotonicity. Higgins and colleagues proposed the β-VAE framework [16] to control the level of entanglement in latent space by incorporating a Kullback–Leibler divergence term of the latent distributions from a normal prior. Regularization of VAEs aimed to impose monotonicity of the learned latent space with respect to features of the input was originally introduced in the context of Fader Networks [24] as part of the GLSR-VAE [14] model and later employed in the ARVAE [30] model for image and music datasets. Our proposed model follows a similar property monotonicity approach; however, we apply it to DNA sequences and further consider non-uniform coverage of properties in the training which is inherent to the problem of discovering new Ag_N-DNAs where new samples are both laborious and expensive to obtain.

Machine learning for biological sequences. Many sequence embedding approaches build upon on word2vec [28], which was designed to represent words as vectors by enforcing low cosine similarity between the vector representations of semantically similar words. FastText [1, 17] is an alternative employing *n*-grams within words as opposed to whole words. Biological sequence (e.g., RNA, DNA, and proteins) embedding techniques also utilize and extend the above frameworks to obtain representations employed in promoter region [25] and protein [36] classification, taxonomy [34] and neural distance learning [8] and others. Generators for protein or DNA sequences have also been of high interest [35]. Specifically, both Generative Adversarial Network (GAN) [2, 18, 20] and VAE-based [12, 15] generators have been employed for creating nucleotide or amino acid sequences. Gupta and Zou's FBGAN [13] incorporates an additional feedback component that guides the generator towards desired features, such as peptides with antimicrobial activities. The VAE methods in this group are employed to edit sequences for downstream targets as opposed to direct targeted synthesis [12, 15]. Distinct from our work, the majority of these approaches focus on large biological datasets, both in terms of the available input data as well as the lengths of the encoded biological sequences. Methods tuned for long, information-rich sequences are unlikely to perform as well on shorter strands, like the short 10-base DNA strands that we employ to stabilize Ag_N-DNAs. Additionally, the incorporation of additional information is limited to either direct annotation or a semi-supervised editing between runs as in the FBGAN approach.

3 PROBLEM FORMULATION

Our goal is to design Ag_N-DNAs of specific properties tuned by their stabilizing DNA template sequence. The input to our problem is a training set (S, A) of sequences S and their corresponding properties represented as numeric feature vectors A. Specifically, our training data consist of a set of 10-base DNA sequences annotated by (i) fluorescence emission color quantified as the peak *wavelength (WAV)* of the emission spectrum of the corresponding Ag_N-DNA and (ii) its fluorescence brightness quantified as the *local integrated intensity (LII)* of a Gaussian fitted to the fluorescence spectral peak. In other words, the input property matrix is 2-dimensional $A \in \mathcal{R}^{|S|\times 2}$. Our past work describes the data set acquisition, processing, and curation in detail [5].

Given that input, we aim to learn a generative model for the joint distribution of DNA sequence and properties $M : p(S, A)$ based on the training observations (note that we overload the notation and use S and A as the corresponding random variables as well). We can then employ M to sample unobserved sequences S' with desired properties A', i.e. $S' \sim p(S|A = A')$. Specifically, we aim to design DNA templates that stabilize bright Ag_N-DNAs with NIR emission, i.e. $WAV > 800nm$ and as high fluorescence yield (LII) as possible. A few such Ag_N-DNAs were only recently synthesized for the first time [31, 32]. In this regime ($WAV > 800nm$) biological tissues become increasingly transparent to light and the Ag_N-DNAs can be employed as effective and non-toxic biosensors.

4 METHODOLOGY

Estimating the joint distribution of DNA sequences and properties $p(S, A)$ is challenging with limited training data, since the discrete space of all possible sequences is exponential and testing the properties of all sequences by Ag_N-DNA synthesis is impossible. Hence, we seek to learn a joint low-dimensional numeric embedding for sequences and properties that allows for two-way transformation to and from the input space. To this end, we employ the Variational Autoencoders (VAEs) framework [23] which allows for the desired two-way transformation and can flexibly incorporate appropriate encoder/decoder architectures for sequential data such as DNA sequences (Sec. 4.1). To enable sampling from the learned latent space while controlling for WAV and LII of interest, we regularize a subset of the latent dimensions in the VAE to correlate with the observed Ag_N-DNA properties from training (Sec. 4.2) and handle imbalanced coverage of property samples (Sec. 4.3). Finally, since we employ a β-VAE architecture that enforces decoupling and normality of the latent space, we can efficiently sample from the conditional latent distribution employing the truncated normal distribution (Sec. 4.4).

4.1 VAE Encoder/Decoder architecture

Our VAE model is composed of two distinct networks, an encoder mapping DNA sequences S to distributions in latent space $p(z)$ and a decoder mapping samples from latent space back to sequences Fig. 2. Observed sequences $S_i, |S_i| = l$ of length l are encoded using one-hot encoding into matrices $X_i \in \mathcal{R}^{l\times 4}$ since our DNA alphabet can take one of 4 possible DNA base values $\{A, C, T, G\}$.

Encoder: The one-hot encoding input matrices X_i are grouped into training batches of size b and fed into the first block of the encoder, followed by a many-to-many bi-directional LSTM (*Bi-LSTM*) with hidden state size h. We select this sequential architecture due to its wide-adoption for sequence learning [27], yet it is among the simplest sequential models with relatively few parameters to tune. The bi-directionality is essential to capture the context both before and after a given DNA base, which we expect to control the 3D local structure of the DNA strand and its interactions with silver atoms in the Ag_N-DNAs. Each Bi-LSTM layer in the block has one Bi-LSTM cell per base position resulting in a total of l cells. Each cell contains forward and backwards regular LSTM cells. The output of

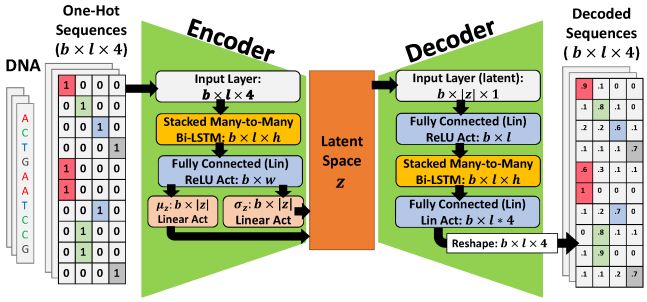


Figure 2: The architecture of the LSTM encoder and decoder in our VAE model. The Encoder is a sequence of (i) input, (ii) LSTM, (iii) fully connected and (iv) latent mean μ_z and variance σ_z output layers. The Decoder follows a "reversed" architecture with the difference that the LSTM layer is followed by a fully-connected linear layer and a reshape transformation to obtain decoded sequences. *Notation:* l is the DNA sequence length, b is the training batch size, h is the hidden state size of Bi-LSTM cells, w is the linear width of the fully connected layer, and $|z|$ is the dimensionality of the latent space.

each Bi-LSTM cell is the concatenation of the hidden states of its forward and backward LSTM cell, which is a vector of size h. The LSTM block output is a tensor of shape $\mathcal{R}^{b \times l \times h}$. We experimented with multiple LSTM layers and by adding dropout layers, but the simplest architecture of one LSTM layer and no dropout resulted in optimal performance for our dataset (details on our hyperparameter search are available in Tbl. 1).

The LSTM output is flattened and fed into a fully-connected layer with ReLU activation. The output size of the fully connected layer is $b \times w$, where w is the layer width, i.e., the number of neurons in the layer. The last layer of the encoder includes the latent mean μ_z and variance σ_z dense layers which are traditionally employed in VAEs. Their outputs represent the corresponding distributional parameters of an input's latent encoding in z.

Decoder. The decoder takes as an input a batch of samples from latent space and is trained to reconstruct the DNA sequences in the batch. The first decoder layer is dense and features ReLU activations. Its output size is the same as that of the DNA sequence length. The output from the latter is passed to a bidirectional many-to-many LSTM block with the same architecture as its encoder counterpart. Finally we transform the LSTM block output into the shape of a one-hot encoded DNA sequence using a dense layer and the output of the latter is transformed in to a batch reconstruction tensor Y of size $b \times l \times 4$.

4.2 Property-regularized loss function

The loss function of the basic VAE architecture features a reconstruction L_{REC} and a Kullback-Leibler (KL)-divergence L_{KL} term:

$$L_{VAE} = L_{REC}(\phi, \theta) + L_{KL}(\phi, \theta), \tag{1}$$

where ϕ and θ are the parameters of the encoder and decoder respectively. The first term is the reconstruction loss between the input tensor X and the decoded output tensor Y, both of shape $b \times l \times 4$:

$$L_{REC}(\phi, \theta) = \frac{1}{bl} \sum_{u=0}^{b} \sum_{i=0}^{l} \left(- \sum_{j=0}^{4} X_{uij} \log \left(\frac{\exp(Y_{uij})}{\sum_{k=0}^{4} \exp(Y_{uik})} \right) \right) \tag{2}$$

VAEs model the latent representation Z as a random variable and hence the end-to-end encoding-decoding process is viewed as a sequence of sampling operations: (i) the input is sampled from the conditional distribution represented by the decoder $X \sim p_\theta(X|Z)$ and (ii) Z is sampled from an approximation of the true posterior distribution $Z \sim q_\phi(Z|X)$ realized by the encoder and parameterized by the variational parameters ϕ [23]. Similar to other variational methods, variational inference in VAEs is performed by maximizing the *evidence lower bound (ELBO)*:

$$\log_p(X) \geq E_{Z \sim q_\phi(Z|X)}[\log(p_\theta(X|Z))] - D_{KL}(q_\phi(Z|X)||p(Z)),$$

where $p(Z)$ is a prior distribution for the latent representation and $D_{KL}(q_\phi(Z|X)||p(Z))$ is the Kullback-Leibler (KL) divergence between the approximation to the posterior distribution and a prior distribution for Z. Minimizing the KL divergence improves the tightness of the ELBO bound and how well the approximate posterior aligns with the prior [23] and hence it is the second loss term in Eq. 1, i.e. $L_{KL} = D_{KL}(q_\phi(Z|X)||p(Z))$. A typical prior distribution employed for the latent variable Z is multivariate normal with zero co-variances which promotes independence between the dimensions in the latent space. A follow-up model called β-VAE introduces a weight β which multiplies L_{KL} and allows for more control for decoupling of the dimensions of the latent representation [16].

The loss function from Eq. 1 ensures a decoupled latent space and a good reconstruction of the input DNA sequence. Our goal for the latent representation is to also jointly reflect the properties A of Ag_N-DNAs (WAV and LII) that are of interest for design. Thus, we extend a property regularization variant of the VAE model [30] that ensures that a subset of the dimensions of Z encode properties, that corresponding latent dimensions monotonically increase in unison with the observed properties A. To this end, we add a property regularization term to the basic β-VAE loss as follows:

$$L_{REC}(\phi, \theta) + \beta L_{KL}(\phi, \theta) + \gamma \sum_{a \in A} L_a, \tag{3}$$

where the last term, which we will also refer to as L_{REG}, adds property regularization controlled by a hyper-parameter γ. The individual summands L_a in L_{REG} enforce alignment between a single property (e.g., WAV, LII) and a corresponding latent dimension:

$$L_a = MAE(tanh(\delta D_r) - sign(D_a)), \tag{4}$$

where MAE stands for the mean absolute error, δ is a scaling parameter, $tanh()$ denotes the hyperbolic tangent function applied element wise to its argument, $sign()$ is the sign function also applied element-wise, and $D_r, D_a \in \mathcal{R}^{b \times b}$ are batch-specific square difference matrices whose elements are defined as follows:

$$D_r(i, j) = Z_r(i) - Z_r(j) \text{ and } D_a(i, j) = A_a(i) - A_a(j),$$

where i and j are training instance indices within a given batch, $Z_r(i)$ is the r-th dimension of the embedding of instance i that is mapped to attribute index a and $A_a(i)$ is the a-th attribute value of instance i provided as input to out model. Intuitively, L_{REG} introduces cost for instance pairs which are ordered differently based on their training attribute with index a and their latent embedding in a corresponding dimension r. As a result the VAE will be trained to match the WAV ordering to a WAV proxy latent dimension in Z and similarly the LII to a corresponding LII proxy dimension in latent space. Optimization of the overall objective is performed using standard neural network batch-gradient methods, since all components of the loss function are differentiable with respect to the parameters of the encoder ϕ and decoder θ.

4.3 Handling imbalance in the observations

The regularized VAE model imposes a penalty in L_{REG} for pairs of instances whose embeddings in regularized dimensions are ordered differently than the reference values of their properties. Specifically, the regularization loss for a given attribute a within a batch of size b instances is computed as an average of all instance pairs:

$$L_a = \frac{2}{b(b+1)} \sum_{i=1}^{b-1} \sum_{j=i+1}^{b} |\tanh(\delta D_r(i,j)) - \text{sign}(D_a(i,j))|. \quad (5)$$

The above definition assumes that attribute values in batches and in the overall dataset are uniformly randomly distributed. In settings where the training dataset contains non-uniformly distributed attribute values, the regularization, and consequently the trained VAE model, will over-represent intervals of attribute values with high support and neglect rare values. Note that this imbalance is especially relevant to employing VAEs for Ag_N-DNA design. In particular, we have many fewer NIR Ag_N-DNAs in the training data, while at the same time our goal is to design DNAs for NIR Ag_N-DNAs. *Given the relative scarcity of training instances for desired attribute values, how can we "focus" the training on representing that attribute value region well?*

To this end, we propose a weighted MAE alternative in which differences for rare pairs induce higher penalties. Specifically, if $v(i,j)$ is an instance pair score function, we define a weighted MAE attribute loss as follows:

$$\begin{aligned} L^a_{\text{reg}} &= \text{WMAE}(\tanh(\delta D(Z^a)) - \text{sign}(D(A^a))) \\ &= \frac{1}{v} \sum_{i=1}^{b-1} \sum_{j=i+1}^{b} v(i,j) |\tanh(\delta D(Z^a)_{ij}) - \text{sign}(D(A^a)_{ij})|, \end{aligned} \quad (6)$$

where $v = \sum_{i=1}^{b-1} \sum_{j=i+1}^{b} v(i,j)$. To over-represent rare pairs we consider scores that are inversely proportional to the probability of such pairs. In particular we employ the exponential function:

$$v(i,j) = e^{-\alpha p_i^a p_j^a},$$

where p_i^a is the probability of observing the attribute value of the i-th sample and $\alpha \geq 0$ is a parameter controlling the rate of score decrease with increasing pair probability. Note that as $\alpha \to 0$ the score of all pairs approaches 1, regardless of their probability and the weighted MAE loss reduces to the original unweighted version. In our experiments, small values of $\alpha = 0.01$ improve the representation of rare attribute instances without significant impact on the overall reconstruction accuracy.

To estimate the probability of attribute values empirically, we compute a fixed-bin-width frequency histogram from the attributes of training samples and normalize each bin by the total number of training instances. The probability of a specific instance p_i^a is the probability of the bin corresponding to its attribute value.

4.4 Truncated VAE sampling for DNA design

Recall that we proposed the regularized VAE as an approach to represent the joint distribution $p(S,A)$ of DNA sequences and the corresponding Ag_N-DNA properties and our goal is to design DNAs with specific properties, i.e., sample $S' \sim p(S|A \in [A_{lb}, A_{ub}])$, where $[A_{lb}, A_{ub}]$ specifies some property ranges of interest for

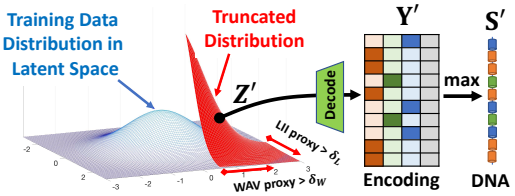


Figure 3: Truncated sampling in regularized latent space to design DNA templates for Ag_N-DNAs with desired properties. We first fit normal distribution of the expected locations of training samples in latent space and sample from the truncated version of the latter, where truncation is performed for proxy dimensions for desired bands of Wavelength (WAV Proxy) and Local Integrated Intensity (LII Proxy).

design. In our specific case, we would like to design bright NIR Ag_N-DNAs, so the range of interest is high WAV and high LII.

The process of sampling from our VAE is demonstrated in Fig. 3. Since we cannot control directly WAV and LII, we sample Z' from the latent space conditional on the property-regularized proxy dimensions being in specified bands. Since the latent distribution is regularized (via the KL divergence loss term L_{KL}) to approximate a normal prior distribution, we can use the embedding of training instances to estimate the parameters of this distribution $\hat{p}(Z_{train})$ and then sample from it. A naive approach to obtain samples from the bands of interest is rejection sampling: sample from the overall distribution $p(Z)$ and retain samples that fall in the bands of interest. However, this will be highly inefficient, especially when trying to sample from the tail (high WAV proxy and LII proxy). Due to the Gaussian assumption for the latent embeddings, we can employ truncated normal sampling—an efficient sampling approach that does not require rejection [26].

Given a sample in latent space Z' we employ the trained decoder to obtain an output approximation for a one-hot encoding $Y' \in \mathcal{R}^{l \times 4}$ (Fig. 3). Finally, to obtain a DNA sequence S' we select the position of maximal weight for each row in Y' and decode it to the corresponding DNA base. Note that this last step introduces non-linear distortion since effectively some non-zero elements in Y' are disregarded and only the maximum is taken to select a DNA base. To quantify this distortion, we re-encode new sampled DNAs S' via the VAE encoder to obtain a re-encoded latent representation Z''. Samples whose re-encoded representation Z'' satisfy our design bands are selected for Ag_N-DNA synthesis.

5 EXPERIMENTAL EVALUATION

This section reports on new Ag_N-DNAs we experimentally synthesize based on sampled sequences from a trained VAE model and discusses implications of tuning, training and deploying the model, including effect of hyper-parameters and lessons learned from this first deployment for design of new Ag_N-DNAs.

5.1 Experimental setup

Data. Our training dataset consists of $|S| = 2661$ DNA sequences of length $l = 10$ together with the properties *WAV* and *LII* of their corresponding stabilized Ag_N-DNAs. The distribution of property values in the training set can be seen in Fig. 4 (grey bars). To visualize results and to select truncation points for sampling and synthesis of new Ag_N-DNAs from higher wavelengths, we adopt the same color class definitions from past work, motivated by the

physical properties of Ag_N-DNAs [6]. We define DNA sequences with *WAV* < 580 nm to be Green, 600 nm < *WAV* < 660 nm to be Red, and 660 nm < *WAV* < 800 nm to be Very Red (details in [5]). We also introduce a new NIR class with *WAV* > 800 nm, which is the particularly rare class (Fig. 4a) that we aim to target with sequence generation. These color definitions play no role in VAE training but are useful for comparing to past work on Ag_N-DNAs.

Metrics. Intuitively, a well-trained VAE for targeted Ag_N-DNA design should (i) reconstruct DNA sequences well and (ii) impose ordering in regularized (proxy) latent dimensions similar to that of their corresponding property observations from training. To quantify sequence reconstruction *Accuracy*, we measure the fraction of correctly recovered DNA bases after decoding, namely: $Accuracy = 1 - d_H(S, S')/l$, where S is the input DNA sequence, S' is its reconstructed DNA sequence after taking the maximum loadings from the VAE output encoding Y' (See Fig. 3), l is the length of sequences and $d_H(\cdot,\cdot)$ is the Hamming distance between the two argument sequences. To quantify the alignment of proxy dimensions with their observed attribute values, we compute the *Correlation* between the regularized latent dimension embedding of training/validation instances and their corresponding properties. Note that the above measures have counterparts in the loss function, but we use these more interpretable measures to select hyper-parameter configurations for synthesis. In tuning the model, we also reserve a random subset of instances for validation to gauge if the VAE's internal representation overfits the training data. Unless stated otherwise, we employ 85% of data for training when tuning the model, but then re-train the best model with all data before employing it for sampling and Ag_N-DNA synthesis.

VAE training, hardware and implementation. We train our model employing a batch size of $b = 32$ and over 2000 epochs using the Adam optimizer [21] with a learning rate of 0.0001. We tune other parameters in order to strike a balance between good DNA reconstruction *Accuracy* and *Correlation* with training properties (Details of hyperparameter tuning are available in the Supplement). Our method is implemented in PyTorch, and we train our models on a Dell server equipped with NVIDIA Tesla V100 (16GB) GPUs. Our code is available at http://www.cs.albany.edu/~petko/lab/code.html.

Wet lab synthesis and spectroscopy. Ag_N-DNAs are synthesized and characterized by the same methods used for the training data library for NIR Ag_N-DNAs [31]. DNA and silver nitrate ($AgNO_3$) are mixed in an aqueous solution of ammonium acetate (NH_4OAc) at neutral pH. After 18 min incubation at room temperature for 18 minutes, the mixture is reduced with sodium borohydride ($NaBH_4$). Final concentrations are 20 μM DNA, 140 μM $AgNO_3$, 70 μM $NaBH_4$, and 10 mM NH_4OAc. Solutions are stored in the dark at 4°C for 7 days, and fluorescence emission spectra are collected on two well plate fluorimeter, from 400 - 850 nm on a commercial Tecan Spark, and from 700 - 1,300 nm on a customized plate reader with enhanced NIR sensitivity [32], using UV excitation at 260 nm to universally excite all fluorescent Ag_N-DNAs. To determine the WAV and LII properties of each designed Ag_N-DNA, we used the same spectral fitting procedures outlined in our past works [5, 31].

5.2 Results from wet lab Ag_N-DNA synthesis

Because high-throughput experiments on hundreds of Ag_N-DNAs is a costly process, we select a smaller set of DNA sequences to test experimentally following our sampling approach (Sec. 4.4). We employ a VAE trained on all instances (no validation set) and with hyper-parameters $\alpha = 0.01, \beta = 0.007, \gamma = 1, \delta = 1, |z| = 19, h = 13, w = 16$, single LSTM layer, and no dropout. We tune hyper-parameters by performing a grid search and select the model with both high Accuracy and Correlation for properties (details in the Supplement). We generate 1000 samples of DNA templates and rank them by their re-encoded WAV proxy Z''_{WAV}. Specifically, each sample Z' is first decoded and translated to a DNA sequence S', which is then re-encoded by the encoder to obtain the re-encoded WAV proxy. The top-20 sequences of highest Z''_{WAV} are selected for synthesis.

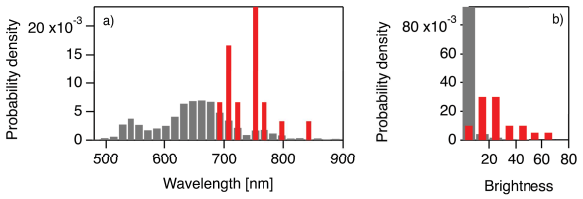


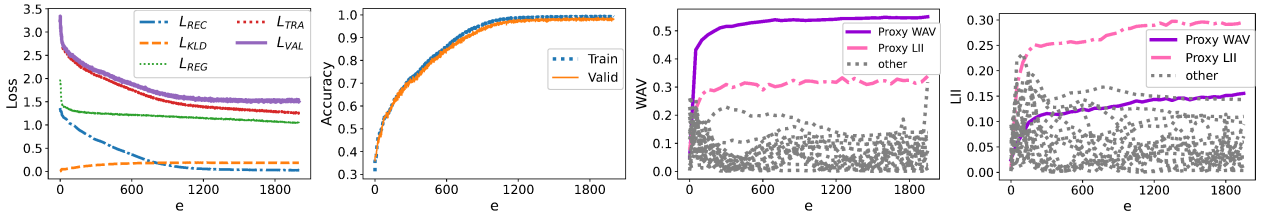
Figure 4: Wet lab synthesis results: Probability density distributions of (a) *WAV* (units of wavelength in nanometers, representing Ag_N-DNA color) and (b) *LII* (brightness) for training (grey) and newly synthesized Ag_N-DNAs.

We experimentally synthesized Ag_N-DNAs using the selected 20 strands and measured their fluorescence properties, finding that all 20 sequences yield brightly fluorescent Ag_N-DNAs with *WAV* between 695nm and 845nm. One Ag_N-DNA falls into our targeted region of *WAV* > 800nm, a 240% increase in NIR frequency compared to the training data. Notably, the other generated sequences form fluorescent Ag_N-DNAs very close to the NIR *WAV* threshold, without a single nonfluorescent sample or Ag_N-DNA at Green or Red *WAV* values (Fig. 4a). Furthermore, the distribution of fluorescence brightness values (*LII*) also increases substantially compared to training data (Fig. 4b).

5.3 Training, latent space and sampling

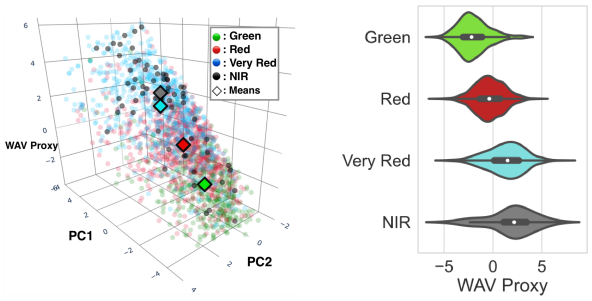
We next provide more insight into the best model employed for synthesis (Sec. 5.2). Fig. 5 summarizes various metrics of the model during training over $e = 2000$ epochs, using 85% of the data for training to also allow characterization of validation statistics. Fig. 5(a) shows the break-down of the loss components. The reconstruction L_{REC} and regularization L_{REG} loss components monotonically decrease as the VAE is learning to both encode-decode sequences and also training properties in their corresponding proxy dimensions in Z. This effect is also evident from Accuracy (Fig. 5(b)) and Correlation (Fig. 5(c),5(d)) profiles. It is important to note that both proxy dimensions tend to retain significant correlation with both WAV and LII. This is because the training WAV and LII are inherently correlated and so are their proxies, regardless of the KLD loss that "works" to de-correlate the latent space.

Fig. 6 presents a visualization of the learned latent representations in terms of the Ag_N-DNA color classes defined in Sec. 5.1. Note that our VAE models the properties (WAV and LII) in continuous space, and we introduce this natural (and physically-motivated) binning into classes only to aid the visualization. The latent embeddings of training samples (both centroids and individual samples) follow the natural WAV order of classes (Fig. 6(a)). Note that the



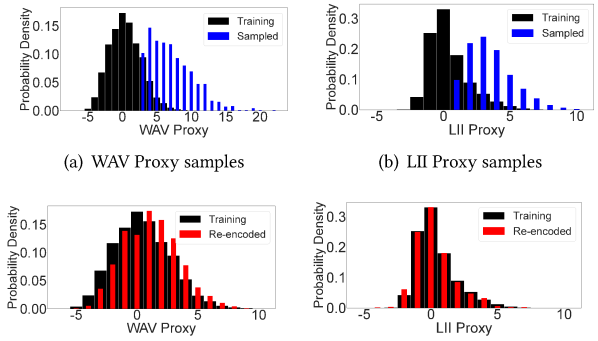
(a) Break-down of loss (b) Accuracy (c) WAV Correlation (d) LII Correlation

Figure 5: Training profiles of the VAE model we employ for de novo Ag_N-DNA synthesis over $e = 2000$ epochs. (a): Break-down of the loss components for training ($L_{REC}, L_{KLD}, L_{REG}$) and the overall training L_{TRA} and validation L_{VAL} loss ; (b): Training and validation accuracy; 5(c): WAV 5(d): LII correlation. The grey dashed curves (other) in the last two figures show the correlation of the remaining latent dimensions with the training properties.



(a) Training data in z space (b) Class-based distributions

Figure 6: Visualization of the learned latent representation Z for training data instances partitioned into WAV classes. (a) shows the means μ_z of latent representation of samples in 3D space, where the horizontal axes are the first two principal components of the 19-dimensional embeddings and the vertical axis is the WAV proxy dimension. Individual samples are depicted as class-colored circles, while class-based centroids are shown as diamonds. (b): Color class-based distributions of training samples in latent space along the WAV proxy dimension.



(a) WAV Proxy samples (b) LII Proxy samples

(c) WAV Proxy re-encoded (d) WAV Proxy re-encoded

Figure 7: WAV (a) and LII (b) proxy distributions of 1000 latent samples and their corresponding distributions after re-encoding the sequences (c),(d). Samples and re-encoded samples are superimposed over the training distributions depicted as wider black-bar histograms.

variance in these 19 dimensional points is not sufficiently captured by only the top-2 PCA components but we use those to qualitatively visualize the spread in the data. The natural ordering of classes is also visible in the overall distributions of WAV proxy (Fig. 6(b)).

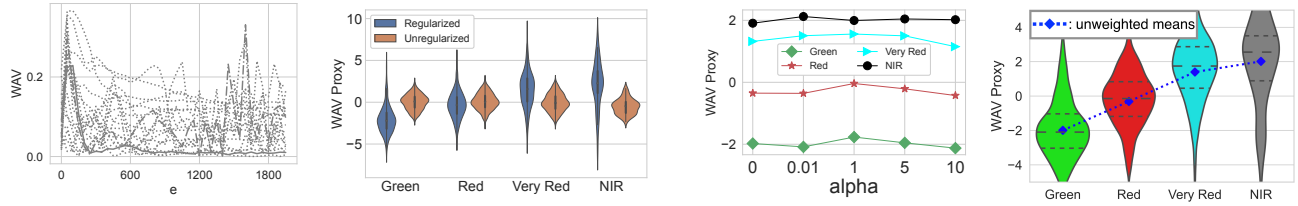
Next we demonstrate the effect of our sampling approach and the distributional shift of samples after re-encoding (Fig. 7). Since our goal is to design bright NIR Ag_N-DNAs, our truncation bounds for δ_{WAV} and δ_{LII} for sampling in latent space are informed by the corresponding distribution of NIRs in the training data. Specifically, we use the mean proxy WAV and LII of all training NIR samples as truncation cut-offs. The sampled WAV and LII distributions (1000 samples) are significantly shifted to the right compared to training data (black bars) and, as expected, "follow" the truncated normal distribution past the cut-off point (Figs. 7(a),7(b)). Note that the "dip" in the left-most blue bars is due to truncation values "falling" in the window corresponding to that bar. The re-encoded sample distributions (Figs. 7(c),7(d)) are shifted back "closer" to the training counterparts. This is due to two factors: (i) loss of information in the translation from continuous encoding to discrete DNA sequences (last step in Fig. 3) and (ii) the "position" of the sample in latent space may have been "outside the feasible bounds" for DNA sequences. This shift motivates ranking the candidate templates for synthesis based on re-encoded proxy properties.

5.4 Ablation analysis

In this section, we remove essential components of the VAE model to test their impact on model quality (Fig. 8). Particularly, we are interested in answering the following questions: *Is property regularization necessary for the VAE architecture to "isolate" the LII and WAV proxies into interpretable latent dimensions? Does the weighting of rare property samples improve the representation of corresponding samples in latent space?*

We first characterize the latent space learned by an *Unregularized* β-VAE, i.e., $\gamma = 0$ and all other hyper-parameters set in the same way as in Sec. 5.2,5.3. Fig. 8(a) shows the correlation of all its latent dimensions with the WAV property. Unlike its regularized counterpart (training profile in Fig. 5(c)), none of this model's latent dimensions correlate as strongly with WAV. A side-by-side comparison of the color class distributions of training samples of our Regularized model and its Unregularized counterpart is in Fig. 8(b) (first latent dimensions to Unregularized). We observe a similar behavior for the other regularized property, LII (figures not included). Without regularization, the β-VAE architecture cannot learn to isolate the Ag_N-DNAs properties in its latent representation.

We also investigate the effect of weighted regularization (Sec. 4.3) on the representation of rare property values. Fig. 8(c) shows centroids of latent samples in the WAV proxy dimension for the four WAV classes as a function of α—the hyper-parameter controlling importance of rare pairs of properties in the regularization. We aim to "well-separate" rare property values. As α (and hence relative



(a) WAV Correlation, Unregularized (b) WAV distribution comparison (c) Separation of means (d) Color class-based shift

Figure 8: Ablation analysis. (a): Lack of correlation of latent dimensions with training WAV when the attribute regularization is omitted; (b): class-based comparison of the distributions of regularized and unregulated latent representations of training instances. (c): Separation of color class means in latent space as a function of α. (d): Mean shift due to weighted regularization.

importance of rare observations) increases, the centroid of the rare NIR class first diverges from the Very Red centroid, then approaches, and then diverges again for large α. Note that at very large α, the overall correlation of all instances with WAV deteriorates. In particular, we determine an optimal setting for α of 0.01 (discussed in the following section). If we "zoom in" on the effect of α values from 0 (no weighting) to 0.01 (our optimal setting) in Fig. 8(d), the distributions of both Very Red and NIR diverge (their means increase) and that of Red decreases slightly. Overall, weighting allows for improved decoupling at the class level and is especially important for equitable representation of rare samples like NIR.

5.5 Varying sequence length

Ideally, our model should be generalizable to other sequence lengths. Thus, we ask: *Can the model trained for sequences of length* $l = 10$ *be employed for other values of* l*?* To study this, we employ a small sample of Ag_N-DNAs stabilized by sequences of lengths $l = 8, 12, 16$ with measured WAV and LII from our recent work [7]. We investigate the quality of embedding for different length sequences in our model trained for $l = 10$. For this, we must first choose how to represent variable length sequences within a $l = 10$ VAE model.

For $l = 8$, we pad the sequence to increase length to $l = 10$. The padding characters can be placed on either side of the sequence (8-FB), the front only (8-F) and in the back only (8-B). Padding positions feature uniform distributions in the one-hot encoding (i.e. all four positions have a value of 0.25). For sequences of lengths longer than $l = 10$, we apply a sliding window approach and represent a single long sequence as a set of its sliding size-10 windows, all sharing the same WAV and LII properties.

We encode the sequences with our trained $l = 10$ model to characterize how the model embeds them. Fig. 9 presents the accuracy and correlation values for all l and padding options. While accuracies are comparable to validation results for sequence $l = 10$ (note that the expected Accuracy of $l = 8$ sequences is 0.8 as successful matching of padding characters is random), the WAV and LII correlations, however, are significantly lower than validation results for $l = 10$. This outcome suggests that simple padding and sliding window approaches are insufficient to generalize to varying l. It may be necessary to consider alternatives in which l and "do-not-matter" positions are explicitly modeled.

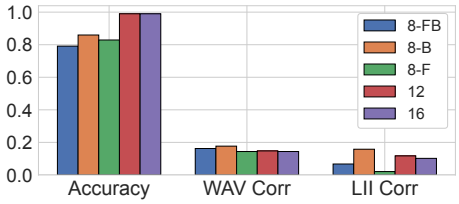


Figure 9: Accuracy, WAV and LII correlation for Ag_N-DNAs of sequence length 8 (8-FB, 8-B, 8-F), 12 and 16, encoded by our model trained on length $l = 10$.

6 DISCUSSION

Compared to past machine learning models for Ag_N-DNA design [4, 5, 7], the VAE generative model presented here has several new advantages. First, past models learned only a single Ag_N-DNA property (*WAV* or *LII*), while the generative model here can distinctly target multiple Ag_N-DNA properties, namely both *WAV* and *LII*. It is advantageous to design DNA sequences correlated with multiple Ag_N-DNA properties (e.g. fluorescence color, brightness, chemical stability, and sensitivity to an analyte of interest). Thus, multi-objective design methods like the models introduced here are critically needed to advance Ag_N-DNA applications. Second, this method does not require strictly defined Ag_N-DNA "color classes" to learn Ag_N-DNA color; this is particularly ideal for rare Ag_N-DNAs in the newly explored NIR spectral range, where little chemical information exists to motivate learning Ag_N-DNA color as a classification problem. Also of note is that our VAE model is more successful in targeting the high WAV space for generated sequences (despite no explicit "class" targeting), and in particular the high LII space. Finally, future examination of the latent space may provide new insights into how DNA sequence selects for Ag_N-DNA properties, advancing fundamental science of these nanomaterials.

While the current implementation of our model yields strong experimental performance, expansions for future work can lead to further improvements. Among these are expanding the attributes used to yield further classifying information, and more aggressive truncation of the sampling distribution to more strongly target the higher end of the WAV scale.

7 CONCLUSIONS

In this paper we proposed, evaluated and deployed a β-VAE generative model for the design of Ag_N-DNA nanomaterials. Our model was able to learn a joint representation for stabilizing DNA templates and Ag_N-DNA properties, including fluorescence color and fluorescence brightness, from a highly imbalanced training data set by regularizing the latent space to correlate with Ag_N-DNA properties. To counteract imbalanced training samples, our model

employed weighting scheme to over-represent such instances. To test the model's efficacy, we targeted the design of DNA template sequences for especially rare NIR-fluorescent Ag_N-DNAs, which represent only 2% of training instances. Our experiments showed that out of 20 DNA template sequences generated by the VAE-based model, all succeeded in producing Ag_N-DNAs with both bright *and* high wavelength fluorescence, including a new NIR-emissive Ag_N-DNA with 840 nm peak fluorescence. The successful selection of a NIR Ag_N-DNA in this test set represents a 240% increase in the target Ag_N-DNA color class, an improvement upon past machine learning models for Ag_N-DNA design despite a significantly imbalanced training data. In addition to enhanced predictive power, our model is the first to learning multiple Ag_N-DNA properties, with significant implications for the advancement of Ag_N-DNA applications in bioimaging and biosensing. Our results show that VAE-based generative models are highly promising for the design of nanomaterials whose properties are encoded by biomolecular sequence and for which only sparse experimental observations may be available. As the fields of DNA and protein nanotechnology [29, 33] continue to expand, such computational models may be crucial in the advancement of biomolecule-based nanotechnologies.

ACKNOWLEDGEMENTS

Research supported by NSF CBET awards #2025793 and #2025790.

REFERENCES

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

[2] Prabal Chhibbar and Arpit Joshi. 2019. Generating protein sequences from antibiotic resistance genes data using generative adversarial networks. *arXiv preprint arXiv:1904.13240* (2019).

[3] Stacy Copp, Petko Bogdanov, Mark Debord, Ambuj Singh, and Elisabeth Gwinn. 2014. Base Motif Recognition and Design of DNA Templates for Fluorescent Silver Clusters by Machine Learning. In *Journal of Advanced Materials*.

[4] Stacy Copp, Petko Bogdanov, Mark Debord, Ambuj K. Singh, and Elisabeth Gwinn. 2014. Motif-based design of DNA templates for fluorescent silver clusters. In *FNANO*.

[5] Stacy M Copp, Alexander Gorovits, Steven M Swasey, Sruthi Gudibandi, Petko Bogdanov, and Elisabeth G Gwinn. 2018. Fluorescence color by data-driven design of genomic silver clusters. *ACS nano* 12, 8 (2018), 8240–8247. https://pubs.acs.org/doi/abs/10.1021/acsnano.8b03404

[6] Stacy M Copp, Danielle Schultz, Steven Swasey, James Pavlovich, Mark Debord, Alexander Chiu, Kevin Olsson, and Elisabeth Gwinn. 2014. Magic numbers in DNA-stabilized fluorescent silver clusters lead to magic colors. *The journal of physical chemistry letters* 5, 6 (2014), 959–963.

[7] Stacy M Copp, Steven M Swasey, Alexander Gorovits, Petko Bogdanov, and Elisabeth G Gwinn. 2019. General approach for machine learning-aided design of DNA-stabilized silver clusters. *Chemistry of Materials* 32, 1 (2019), 430–437.

[8] Gabriele Corso, Zhitao Ying, Michal Pándy, Petar Veličković, Jure Leskovec, and Pietro Liò. 2021. Neural Distance Embeddings for Biological Sequences. *Advances in Neural Information Processing Systems* 34 (2021).

[9] Brian M Frezza, Scott L Cockroft, and M Reza Ghadiri. 2007. Modular multilevel circuits from immobilized DNA-based logic gates. *Journal of the American Chemical Society* 129, 48 (2007), 14875–14879.

[10] Anna Gonzalez-Rosell, Cecilia Cerretani, Peter Mastracco, Tom Vosch, and Stacy M Copp. 2021. Structure and luminescence of DNA-templated silver clusters. *Nanoscale Advances* 3, 5 (2021), 1230–1260.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[12] Anvita Gupta and Anshul Kundaje. 2019. Targeted optimization of regulatory DNA sequences with neural editing architectures. *bioRxiv* (2019), 714402.

[13] Anvita Gupta and James Zou. 2018. Feedback GAN (FBGAN) for DNA: A novel feedback-loop architecture for optimizing protein functions. *arXiv preprint arXiv:1804.01694* (2018).

[14] Gaëtan Hadjeres, Frank Nielsen, and François Pachet. 2017. GLSR-VAE: Geodesic latent space regularization for variational autoencoder architectures. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)* (2017), 1–7.

[15] Alex Hawkins-Hooker, Florence Depardieu, Sebastien Baur, Guillaume Couairon, Arthur Chen, and David Bikard. 2021. Generating functional protein variants with variational autoencoders. *PLoS computational biology* 17, 2 (2021), e1008736.

[16] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. β-VAE: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations (ICLR)* (Toulon, France).

[17] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[18] Mostafa Karimi, Shaowen Zhu, Yue Cao, and Yang Shen. 2020. De novo protein design for novel folds using guided conditional Wasserstein generative adversarial networks. *Journal of Chemical Information and Modeling* 60, 12 (2020), 5667–5681.

[19] Yonggang Ke, Luvena L Ong, William M Shih, and Peng Yin. 2012. Three-dimensional structures self-assembled from DNA bricks. *science* 338, 6111 (2012), 1177–1183.

[20] Nathan Killoran, Leo J Lee, Andrew Delong, David Duvenaud, and Brendan J Frey. 2017. Generating and designing DNA with deep generative models. *arXiv preprint arXiv:1712.06148* (2017).

[21] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2015).

[22] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. *CoRR* abs/1312.6114 (2014).

[23] Diederik P Kingma, Max Welling, et al. 2019. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning* 12, 4 (2019), 307–392.

[24] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Fader Networks: Manipulating Images by Sliding Attributes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 5969–5978.

[25] Nguyen Quoc Khanh Le, Edward Kien Yee Yapp, N Nagasundaram, and Hui-Yuan Yeh. 2019. Classifying promoters by interpreting the hidden information of DNA sequences via deep learning and combination of continuous fasttext N-grams. *Frontiers in bioengineering and biotechnology* 7 (2019), 305.

[26] Yifang Li and Sujit K Ghosh. 2015. Efficient sampling methods for truncated multivariate normal and student-t distributions subject to linear inequality constraints. *Journal of Statistical Theory and Practice* 9, 4 (2015), 712–732.

[27] Zachary Chase Lipton. 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning. *ArXiv* abs/1506.00019 (2015).

[28] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[29] Jeanette Nangreave, Dongran Han, Yan Liu, and Hao Yan. 2010. DNA origami: a history and current perspective. *Current opinion in chemical biology* 14, 5 (2010), 608–615.

[30] Ashis Pati and Alexander Lerch. 2020. Attribute-based regularization of latent spaces for variational auto-encoders. *Neural Computing and Applications* (2020), 1–16.

[31] Steven M Swasey, Stacy M Copp, Hunter C Nicholson, Alexander Gorovits, Petko Bogdanov, and Elisabeth G Gwinn. 2018. High throughput near infrared screening discovers DNA-templated silver clusters with peak fluorescence beyond 950 nm. *Nanoscale* 10, 42 (2018), 19701–19705.

[32] Steven M Swasey, Hunter C Nicholson, Stacy M Copp, Petko Bogdanov, Alexander Gorovits, and Elisabeth G Gwinn. 2018. Adaptation of a visible wavelength fluorescence microplate reader for discovery of near-infrared fluorescent probes. *Review of Scientific Instruments* 89, 9 (2018), 095111. https://aip.scitation.org/doi/10.1063/1.5023258

[33] Rein V Ulijn and Roman Jerala. 2018. Peptide and protein nanotechnology into the 2020s: beyond biology. *Chemical Society Reviews* 47, 10 (2018), 3391–3394.

[34] Stephen Woloszynek, Zhengqiao Zhao, Jian Chen, and Gail L Rosen. 2019. 16S rRNA sequence embeddings: Meaningful numeric feature representations of nucleotide sequences that are convenient for downstream analyses. *PLoS computational biology* 15, 2 (2019), e1006721.

[35] Zachary Wu, Kadina E Johnston, Frances H Arnold, and Kevin K Yang. 2021. Protein sequence design with deep generative models. *Current Opinion in Chemical Biology* 65 (2021), 18–27.

[36] Kevin K Yang, Zachary Wu, Claire N Bedbrook, and Frances H Arnold. 2018. Learned protein embeddings for machine learning. *Bioinformatics* 34, 15 (2018), 2642–2648.

SUPPLEMENTAL MATERIAL

Our model includes multiple hyperparameters (listed in Table 1) that can be grouped into two categories:

(1) Architectural hyperparameters: $(|z|, L_w, L_d, h/2, w)$ controlling the shape and function of layers in the architecture; and
(2) Loss hyperparameters: $(\alpha, \beta, \gamma, \delta)$ controlling the behavior of the loss function.

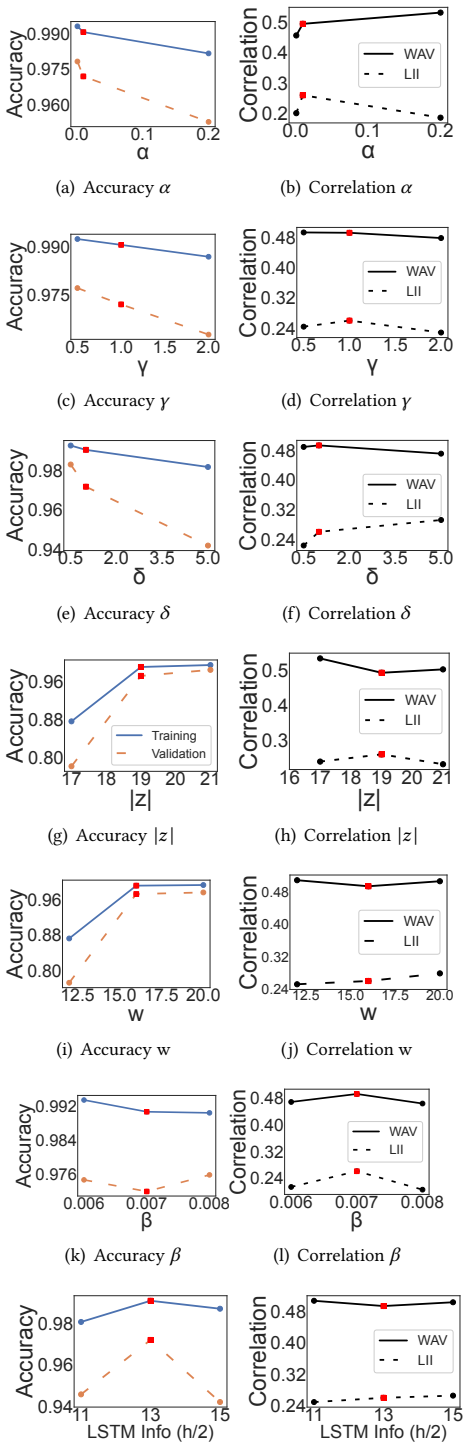
Hyperparameters have varying impacts on different metrics. We perform a grid search across all our hyperparameters to optimize our model for both accuracy and latent space correlation. Tested value of each parameter are listed in Table 1 and optimal parameters denote with bold font.

Hyper-parameter	Values used for grid search
α	0.005, **0.01**, 0.02
β	0.001, 0.003, 0.005, 0.006, **0.007**, 0.008, 0.06, 0.07, 0.08
γ	**1.0**, 3.0, 5.0, 10.0
δ	**1.0**, 5.0, 10.0
Latent Dimensions ($\|z\|$)	10.0, 15.0, 16.0, 17.0, 18.0, **19.0**, 20.0, 30.0
LSTM Layers (L_w)	**1.0**, 3.0, 5.0
LSTM Dropout (L_d)	**0.0**, 0.3, 0.5
LSTM Info ($h/2$)	5.0, 10.0, 12.0, **13.0**, 14.0, 15.0, 16.0, 20.0, 22.0, 26.0, 30.0
Encoder Width (w)	12.0, **16.0**, 20.0

Table 1: Table of hyperparameters that were used during the model testing phase, highlighted values represent the chosen hyperparameters. Note: h is the size of the concatenated output of both the forward and backward LSTM cells, hence the size of each LSTM cell hidden state (LSTM info) is $h/2$.

To illustrate the effect of individual hyperparameters on the model's reconstruction accuracy and correlations, we plot changes of these metrics in an interval around the optimal hyperparameters value while keeping the rest of the parameters set to their optimal values (bold in Tbl. 1).

We present results from this experiment in Figure 10, and indicate the optimal hyperparameter values by red squares. Consider first the figures on reconstruction accuracy as a function of hyperparameters. For α, γ and δ it is evident that, as these hyperparameters increase in value, both training and validation accuracy decrease (w has a similar trend, as the change from 12 to 16 starkly increases training and validation accuracy, though stays stagnant upon further increase). $|z|$ has the opposite effect: as we increase the values of this hyperparameter, training and validation accuracy both increase. Finally, our chosen β results in marginally smaller training and validation accuracy when compared to the other values of β, and our chosen value for LSTM Info (h/2) results in larger training and validation accuracy than its counterpart values. The corresponding correlation figures, however, (WAV and LII Proxies represented by solid and dashed lines here, respectively) demonstrate that increases in reconstruction accuracy often results in deteriorated correlation. To achieve a good balance between reconstruction accuracy and latent correlation, we therefore choose our optimal hyperparameters indicated in Table 1.



(a) Accuracy α (b) Correlation α
(c) Accuracy γ (d) Correlation γ
(e) Accuracy δ (f) Correlation δ
(g) Accuracy $|z|$ (h) Correlation $|z|$
(i) Accuracy w (j) Correlation w
(k) Accuracy β (l) Correlation β
(m) Accuracy LSTM Info (h/2) (n) Accuracy LSTM Info (h/2)

Figure 10: Effect of hyper parameters on reconstruction accuracy and correlations. All parameters, apart from the one varied in each figure, are set to the optimal regimes denoted by bold values in Tbl. 1.