Supporting Information: Multi-Objective Design of DNA-Stabilized Nanoclusters Using Variational Autoencoders With Automatic Feature Extraction

Elham Sadeghi,[†,⊥] Peter Mastracco,[‡,⊥] Anna Gonzàlez-Rosell,[‡] Stacy M. Copp,[*,‡,¶,§,∥] and Petko Bogdanov[*,†]

†*Department of Computer Science, University of Albany- SUNY Albany, Albany, New York, 12222, United States*

‡*Department of Materials Science and Engineering, University of California, Irvine, California, 92697, United States*

¶*Department of Chemistry, University of California, Irvine, California, 92697, United States*

§*Department of Chemical and Biomolecular Engineering, University of California, Irvine, California, 92697, United States*

∥*Department of Physics and Astronomy, University of California, Irvine, California, 92697, United States*

⊥*Authors contributed equally to this work*

E-mail: stacy.copp@uci.edu; pbogdanov@suny.edu

1. Detailed VAE Model Description

This section provides details of the VAE model to inform reproducibility of the computational results. Source code and documentation for the model is available at `http://www.cs.albany.edu/~petko/lab/code.html`.

1.1 VAE background and description of the training loss function

In this study, we employ a variational autoencoder (VAE) model tailored for Ag_N-DNA design. Unlike class-based methods, this model directly maps DNA sequences onto multiple Ag_N-DNA properties. The model automatically extracts features during training and serves as a generative model for the multi-objective design of new Ag_N-DNA. Additionally, it provides interpretability for gaining insights into the relationship between DNA sequence and Ag_N-DNA properties.

The model is adapted from the generative model proposed by Moomtaheen et al.[1] with specific modifications. This model employs a bidirectional LSTM-based β-VAE, consisting of separate encoder and decoder neural networks (Figure S1). The encoder is designed to learn the posterior distribution of the latent space representation z of input DNA sequences x $q_\phi(z|x)$. It takes input DNA sequences represented by one-hot encoding, x, and maps input sequences to a lower-dimensional latent space z. One-hot encoding is a method of converting categorical data into a binary vector representation, where each category is represented by a vector with a single high (1) value and the rest as low (0) values. In this context, each nucleobase in a DNA sequence is converted into a binary vector, making it suitable for input into the neural network. The decoder, denoted by $p_\theta(x|z)$, is trained to reconstruct the latent representations z back into the original DNA sequences x, thereby learning the likelihood distribution and mapping from latent space back to sequences.[2–5]

In our case, the model input is a training set (S, A) that consists of sequences, S, and their corresponding properties represented as feature vectors, A. The input sequence rep-

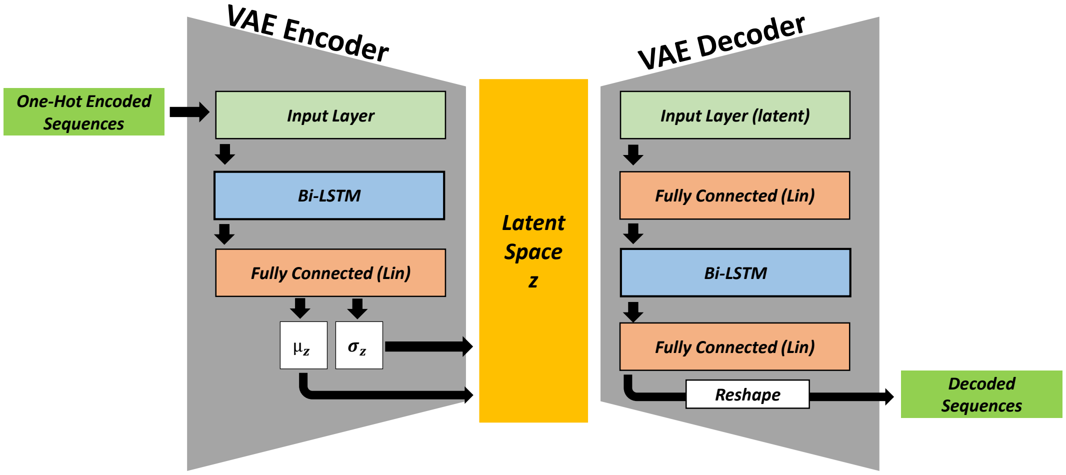


Fig. S1: The VAE model comprises an encoder and decoder. The encoder includes input, Bi-LSTM, fully connected, and output layers for μ_z and σ_z. The decoder employs an inverse architectural approach, wherein the LSTM layer is succeeded by a linear (Lin) fully connected layer, and a reshape operation is applied to generate decoded sequences.

resentation is fed into a bi-directional LSTM layer[1] that comprises the first block of the encoder. Including a bi-LSTM layer with a hidden size of h is essential for capturing long-term dependencies in the preceding and succeeding context of a given DNA nucleobase. The LSTM output is then reshaped and passed into a fully-connected layer that employs the rectilinear (ReLU) activation function. The final layer of the encoder consists of dense layers representing the mean (μ_z) and variance (σ_z) of the latent encoding in z.

In the decoder, a dense layer is first used to reconstruct DNA sequences based on the information in the latent space. Subsequently, the LSTM block converts the output into DNA sequences encoded in a one-hot representation.

To ensure a decoupled and distinct latent space, as well as achieve a precise reconstruction of the input DNA sequence, we utilize a loss function, which consists of three elements:

reconstruction L_{REC}, a Kullback-Leibler (KL)-divergence L_{KL}, and regularization L_{REG}:

$$L_{VAE} = L_{REC}(\phi, \theta) + \beta L_{KL}(\phi, \theta) + \gamma \sum_{a \in A} L_a \tag{1}$$

The first component in the loss function, L_{REC}, encourages the decoder to effectively reconstruct the original samples x using the latent representations z. This component quantifies the accuracy of reconstruction. The second component of the loss function employs the the Kullback-Leibler (KL) to penalize departure of the approximated distribution $q_(z|x_i)$ and a prior distribution $P(z)$, typically a standard multivariate normal distribution. This component encourages independence of the dimensions in the learned latent representation z, and its importance is controlled by hyperparameter β, where higher values enforce a closer alignment with the prior distribution.

The final component of the loss function introduces property regularization, which is governed by a hyperparameter γ. The goal is that specific latent dimensions in z "align" well with Ag_N-DNA properties encoded in A for each input sequence. Specifically, we align the first two dimensions of z to the measured wavelength (WAV) and local integrated intensity (LII) of the highest spectral peak of nanocluster products. The VAE is trained to align the ordering of input clusters for WAV with a latent dimension serving as a WAV proxy, and similarly for LII in z, aligning it with a corresponding LII proxy dimension in the latent space. The explicit form of the L_a term is as follows:

$$L_a = \text{MAE}(\tanh(\delta D_r) - \text{sign}(D_a)), \tag{2}$$

where MAE denotes Mean Absolute Error, $tanh()$ denotes the hyperbolic tangent function applied element-wise, δ is a scaling parameter, $sign()$ represents the sign function applied element-wise, and D_r and D_a are matrices of batch-specific square difference in $R^{b \times b}$.

We apply a penalty term to pairs of instances whose embeddings in regularized dimensions deviate from the expected order of their properties. This definition assumes that attribute values are uniformly distributed within batches and throughout the entire dataset. However, when the training dataset contains attribute values that are not uniformly distributed, the regularization process tends to assign more weight to intervals with higher occurrence(probability) while disregarding rare values. We alleviate this shortcoming by stratification of the training batches discussed below.

1.2 Grid search and hyperparameter tuning

Hyperparameters can be classified into two main categories: architectural and loss hyperparameters. The parameters $L_w, L_d, |z|, h/2$ and w are architectural, as they define the configuration of the model's layers. Parameters $\alpha, \beta, \gamma, \delta$ are related to loss, controlling the evaluation of the loss function and how the model is trained. Each hyperparameter has a distinct impact on specific metrics. We conduct a wide grid search for all hyperparameters to optimize our model. The model selection was based on achieving high accuracy and correlation for the desired properties. Furthermore, we carefully considered ordering of the mean wavelength for different clusters in the latent space (Figure S2). We defined 4 different color ranges based on the magic number properties of Ag_N-DNAs,[6] as follows:

- Green: $\lambda_p < 590$ nm, $LII > 0.5$
- Red: 590 nm $< \lambda_p < 660$ nm, $LII > 0.5$
- Far Red 660 nm $< \lambda_p < 680$ nm, $LII > 0.5$
- NIR: $\lambda_p > 800$ nm, $LII > 0.5$

We similarly tracked the ordering of brightness (local integrated intensity, LII) proxy to ensure the mean of $VeryDark < Dark < Bright < VeryBright$.

- Very Dark: $0.5 < LII < 1$
- Dark: $1 < LII < 3$

- Bright $3 < LII < 10$
- Very Bright: $LII > 10$

The values we tested for each parameter are listed in Table SS1. The best-performing values are in bold. Batch size was set to 32 based on the prior detailed analysis by Moomtaheen et al.[1], who experimented with different batch sizes on the same problem and dataset. Batch size is just one of the many hyperparameters of the VAE, and the values of other hyperparameters were screened as shown in Table S1 to find the best model.

Table S1: Hyper-parameters tuned during the model training phase and their ranges of possible settings. Highlighted values indicate the selected optimal hyperparameters.

Hyperparameter	Values used for grid search
α	0.001, **0.003**, 0.005, 0.007, 0.01, 0.02
β	0.002, 0.004, **0.007**, 0.009
γ	1, 1.2, 1.5, 1.8, **2**, 2.5, 2.7, 3
δ	**1**, 3, 5
Latent Dimensions ($\lvert z \rvert$)	15, 16, **17**, 18, 19, 20
LSTM Layers (L_w)	**1**, 3
LSTM Dropout (L_d)	**0**, 0.02, 0.3
LSTM Info ($h/2$)	10, 12, 13, 14, **15**, 16, 17, 18, 19, 20, 21, 24, 30
Encoder Width (w)	12, **16**

1.3 Improving VAE Performance by Batch Stratification

To improve the performance of our VAE model, we implemented a stratified sampling method to ensure that each training batch reflects the overall distribution of nanocluster properties. The goal is to address the challenge of imbalanced λ_p values, particularly the under-representation of NIR sequences in the training data.[7] Intuitively, we "discretize" the two-dimensional property space of LII and wavelength into intervals, depicted in Figure S3, and ensure that the properties of each training batch is representative of the overall dataset.

First, we convert the continuous values of λ_p and LII into discrete bins by employing the quantile-based variable discretization implemented in the Pandas python library (specifically

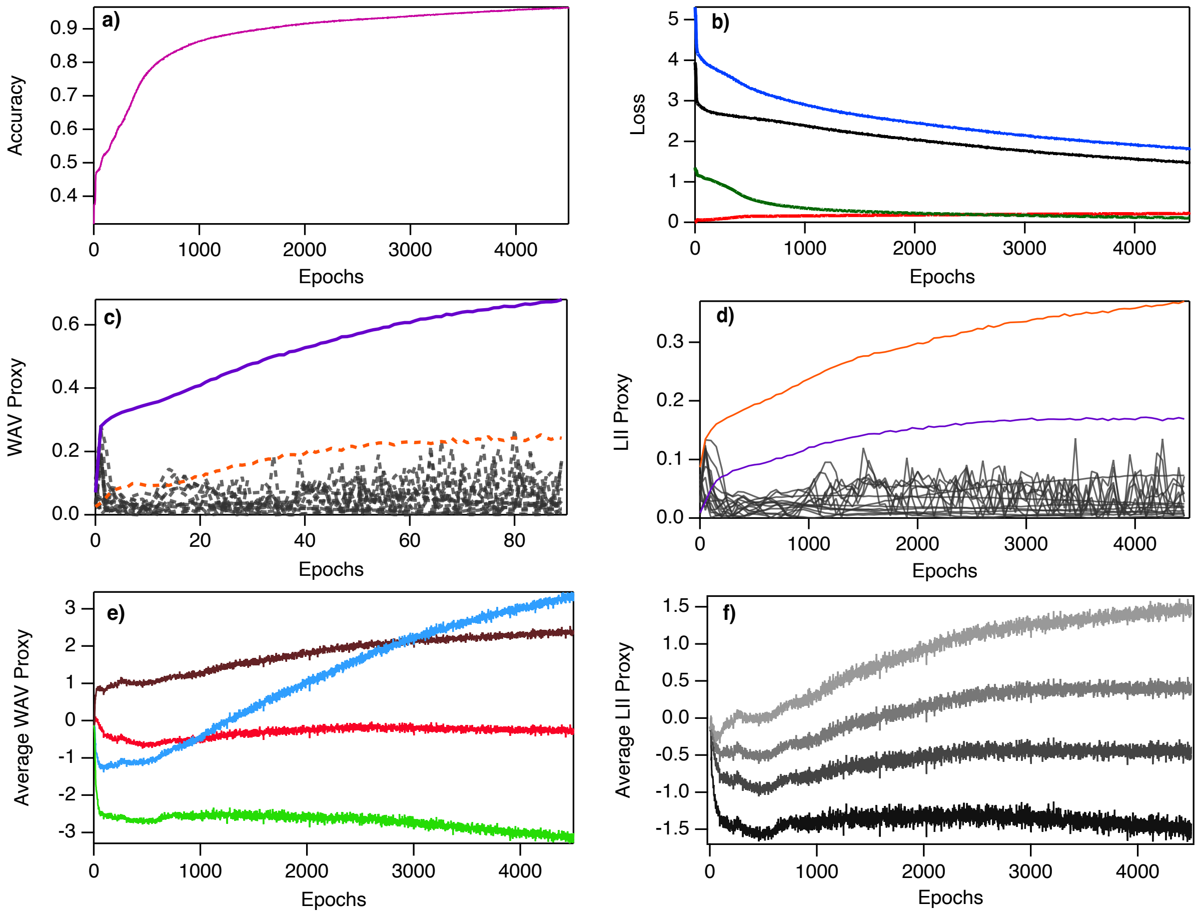
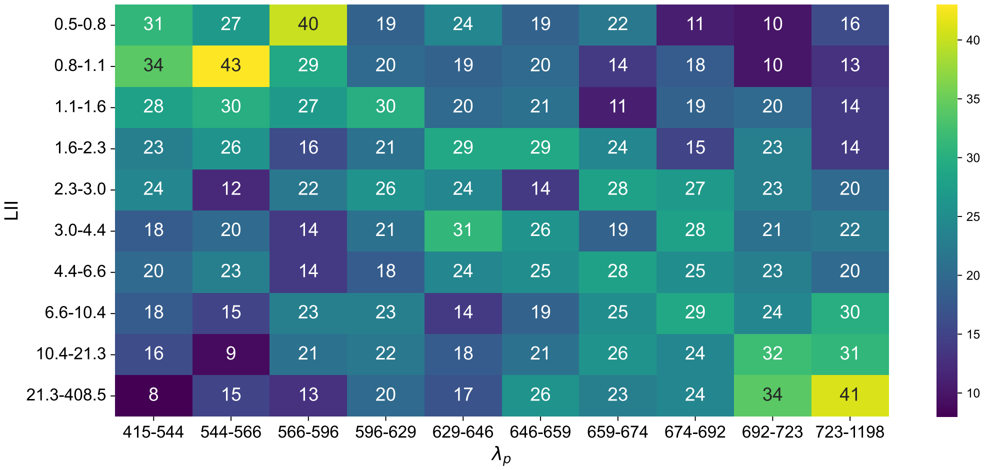


Fig. S2: Training dynamics of the unstratified VAE model utilized for *de novo* Ag$_N$-DNA synthesis across e=4500 epochs. a) Progression of training accuracy. b) Breakdown of loss components ($L_{REC}, L_{KLD}, L_{REG}$). c,d) WAV and LII correlation, with grey dashed curves capturing correlations of other latent dimensions. e) Average proxy wavelength for Green, Red, Far Red, and NIR ranges. f) LII proxy averages for groups: Very Dark, Dark, Bright, and Very Bright (black to light grey).

'*pandas.qcut*' [1]). This method partitions the variable space into a pre-specified number of bins, ensuring that each bin contains approximately an equal number of samples. We specify 10 bins for the two properties λ_p and LII. λ_p values range from 415 to 1200 nm are are divided into 10 bins. Similarly LII values range from 0.5 to 408 and are divided into 10 bins. Values are divided such that the overall 2-dimensional grid contains similar number of samples in each box. Note that the quantile-based binning method ensures equal-sized (similar number of items) bins for each attribute individually but not for their combinations. The resulting

[1] `https://pandas.pydata.org/docs/reference/api/pandas.qcut.html`



LII	415-544	544-566	566-596	596-629	629-646	646-659	659-674	674-692	692-723	723-1198
0.5-0.8	31	27	40	19	24	19	22	11	10	16
0.8-1.1	34	43	29	20	19	20	14	18	10	13
1.1-1.6	28	30	27	30	20	21	11	19	20	14
1.6-2.3	23	26	16	21	29	29	24	15	23	14
2.3-3.0	24	12	22	26	24	14	28	27	23	20
3.0-4.4	18	20	14	21	31	26	19	28	21	22
4.4-6.6	20	23	14	18	24	25	28	25	23	20
6.6-10.4	18	15	23	23	14	19	25	29	24	30
10.4-21.3	16	9	21	22	18	21	26	24	32	31
21.3-408.5	8	15	13	20	17	26	23	24	34	41

Fig. S3: Heatmap showing the distribution of data points across the bins for batch creation. The x-axis represents wavelength ranges and the y-axis represents LII ranges. Each cell value indicates the count of data points falling within the respective bin, ensuring a diverse and representative sampling across all ranges.

grid is shown as a heatmap in Figure S3, with bin boundaries denoted on the axes and the number of training instances in each bin color-coded. For example, 8 sequences in the dataset have $415 < \lambda_p < 544$ and $21 < LII < 408$. This bin has the lowest number of samples, whereas 43 sequences have values $544 < \lambda_p < 566$ and $0.8 < LII < 1.1$, which is the bin with the most samples.

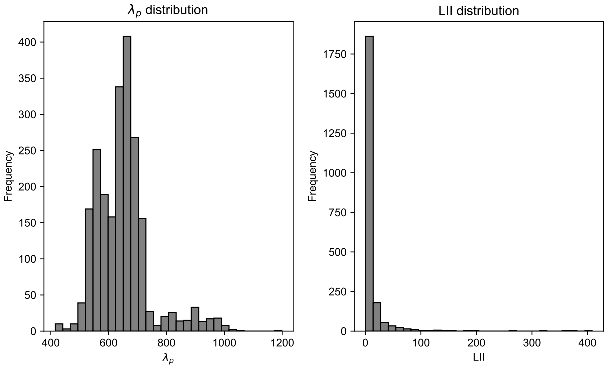


Fig. S4: Left: distribution of λ_p. Right: distribution of LII.

To create batches that reflect the overall distribution of the dataset, we employed a weighted random sampling technique using the '$WeightedRandomSampler$' function from the '*torch.utils.data*' module in PyTorch[2].[8,9] This technique calculates weights for each bin based on the frequency of samples and utilizes an inverse frequency weighting approach to sample balanced batches without replacement. We then train the same model as the unstratified variant using stratification.

We tracked WAV proxy and LII proxy during training to assess if stratification improved the VAE's ability to distinguish Far Red and NIR sequences. The training progress is visualized in Figure S5. Compared to the unstratified model (Figure S2), the stratified VAE orders NIR sequences correctly in fewer epochs, with the average NIR proxy exceeding the Far Red proxy in under 1,000 epochs. This stratified sampling method not only improved the diversity of training batches but also enhanced the VAE's ability to learn mappings across the entire λ_p spectrum, leading to better model performance, particularly for the design of NIR Ag_N-DNAs.

1.4 Sampling new sequences for desired Ag_N-DNA properties

The regularized VAE was introduced as a method to effectively model the joint distribution $p(S, A)$ of DNA sequences and their Ag_N-DNA properties. The primary goal is to generate DNA sequences that possess specific properties. To achieve this, we sample from the latent space of the VAE, considering the property-regularized proxy dimensions falling within predefined ranges. Instead of using a naive (rejection sampling) approach to obtain samples within these ranges, we employed a more efficient method called truncated normal sampling.[10] We estimated a normal distribution in latent space to represent the expected positions of training samples. Our main emphasis was on truncating samples from the tail of the latent space, as this approach aligns with our objective of designing NIR Ag_N-DNAs that exhibit both high λ_p and LII. Specifically, we determined the truncation bounds for the regularized dimensions during the sampling process in the latent space based on the corresponding distribution of NIRs present in the training data. For sequences with high λ_p

[2] `https://pytorch.org/docs/stable/data.html`

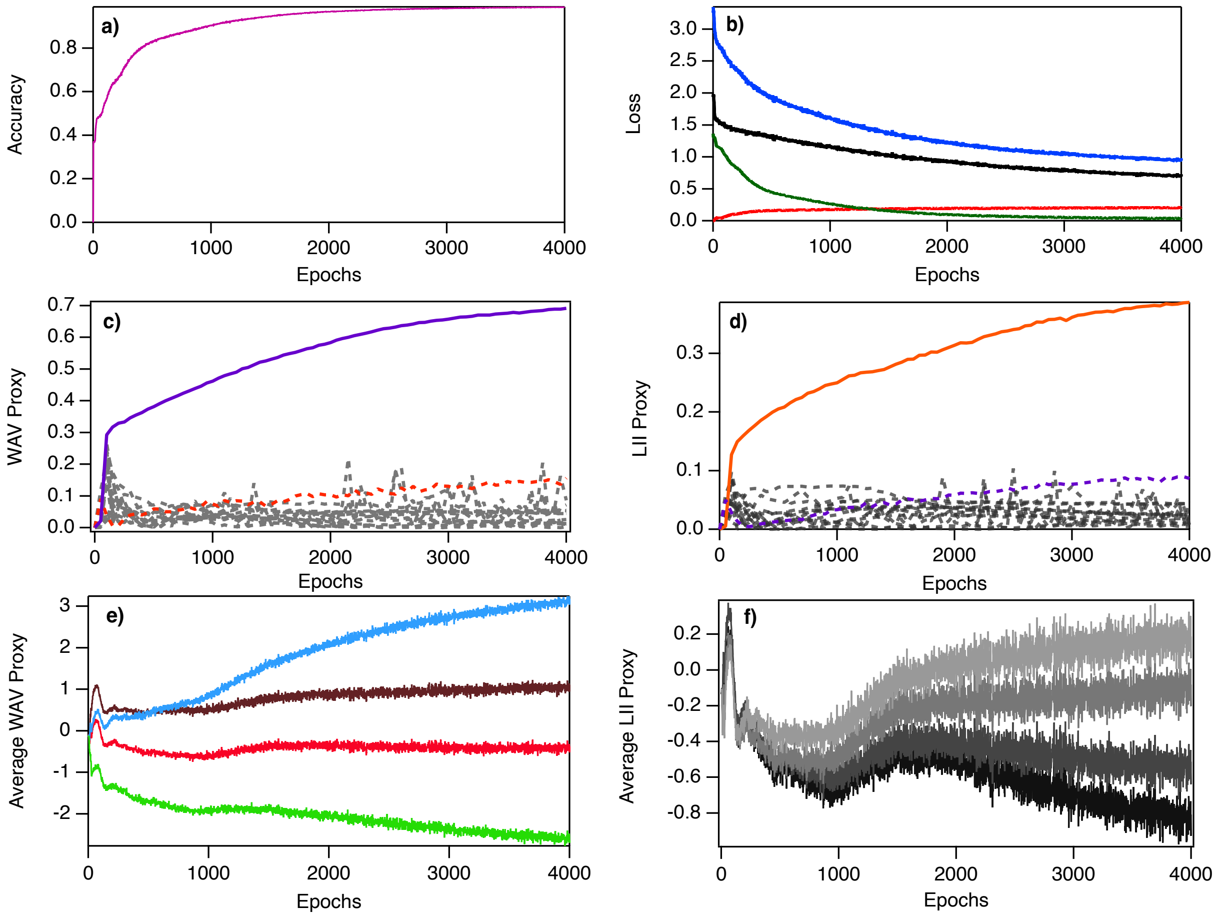


Fig. S5: Stratified VAE model training over 4000 epochs, showing a) training accuracy, b) loss components, c,d) WAV and LII correlations. e) Average proxy wavelengths for Green, Red, Far Red, and NIR ranges, and d) LII proxy averages for Very Dark, Dark, Bright, and Very Bright groups are also displayed.

and LII, we used the training means of the relevant proxies from the NIR training data as truncation cut-offs and sampled from a region above the mean WAV and LII proxies of NIR in latent space.

Similarly, for Green sequences with low wavelengths ($\lambda_p < 590$), we used the training means of the proxies from the Green training data; hence the mean latent value acted as an upper bound, and we subsequently truncated the samples from the region below the mean WAV proxy. Finally, to generate new DNA samples, we employed the trained VAE decoder that provides an approximation for one-hot encoding. The newly generated DNA sequences were then re-encoded using the VAE encoder, resulting in the re-encoded latent representation which was ultimately used to select candidates for synthesis (details of the re-encoding process are available in[1]). Figure S6 illustrates the impact of our sampling technique and how newly sampled proxy values (blue histograms) compare with the proxies of the overall dataset (grey histograms) and those of range specific training instances (red histograms).

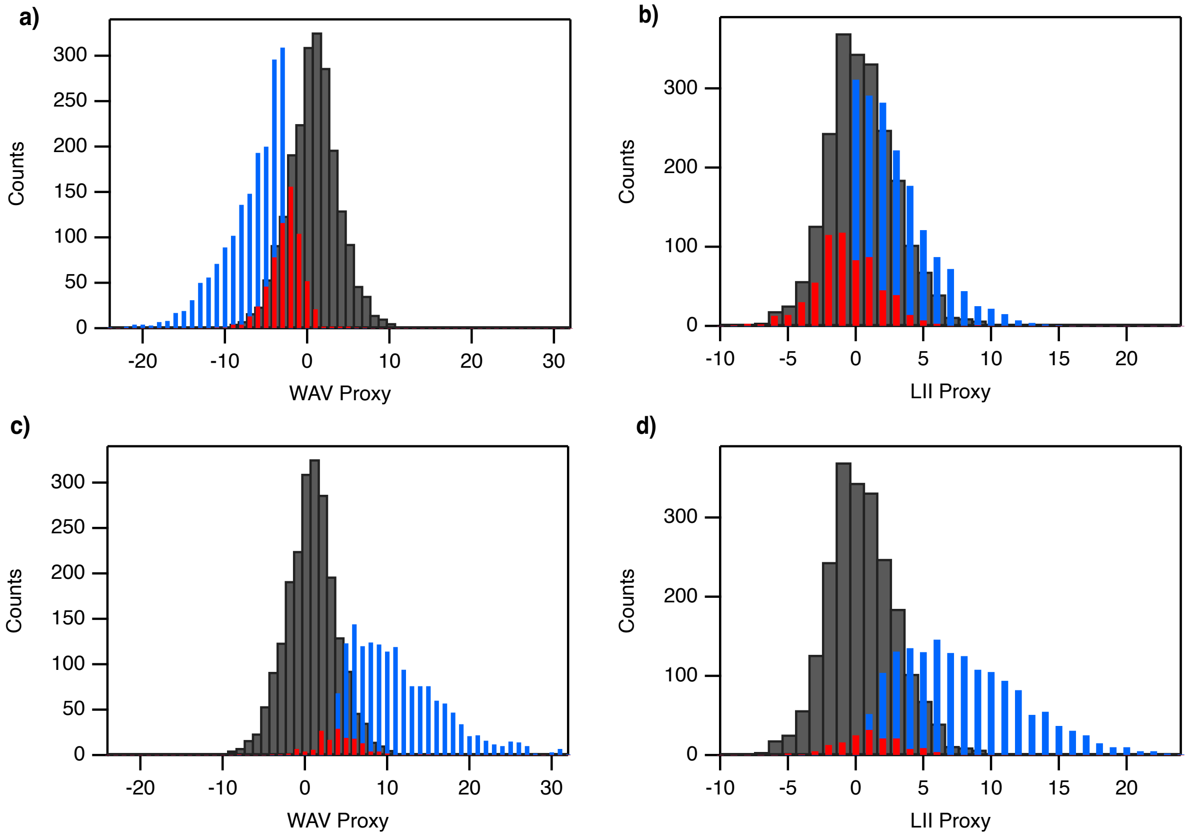


Fig. S6: Distributions of proxy values for generated sequences and training data. (a, b) WAV and LII proxies for the Green spectral region and (c, d) WAV and LII proxies for the NIR spectral region, respectively. The blue histograms represent the distribution of the WAV/LII proxy values of VAE-generated sequences, gray histograms show the WAV/LII distribution of the entire training dataset, and red histograms represent the distribution of the proxy values for range-specific training data instances only, *i.e.* the fraction of the training data that falls within the target property range.

2. Model Interpretation via Shapley analysis

2.1 Background on Shapley analysis

While a model that accurately predicts or generates sequences with specific properties is valuable, understanding the inner workings of such models is crucial for broader scientific acceptance. Deep learning models, like the VAE presented here, are often considered "black boxes", making it challenging to interpret how DNA sequences is mapped onto Ag_N-DNA properties.[11] To elucidate what the VAE model captures about this mapping, we adapted SHapley Additive exPlanations (SHAP) analysis to interpret its model predictions.[12]

Shapley values, based on an approach from game theory that scores team success based on individual sub-team contributions, measure the marginal contribution of each feature to the predictions across all possible subsets of the data. This method can be used to understand the influence of different nucleobase positions within a DNA sequence on the model's predictions. In this context, the "team" corresponds to the complete DNA sequence, the "team members" correspond to nucleobases at specific positions, and "sub-teams" correspond to subsequences (motifs) of nucleobases. Success is quantified by the VAE's ability to order templates by WAV proxy and LII proxy in latent space.

Formally, the Shapley value for a team member i given a value function v is defined as follows:

$$\phi_i(v) = \frac{1}{|N|} \sum_{S \subseteq N \setminus \{i\}} \binom{|N|-1}{|S|} \left(v(S \cup \{i\}) - v(S) \right), \tag{3}$$

where N represents the set of team members, corresponding to the length of a sequence ($N = 10$ in our case), S denotes a subset of N ($S \subseteq N$), $v(S)$ represents the value or payout associated with subset S, and $S \subseteq N \setminus \{i\}$ denotes all subsets that do not include member i.

2.2 Specific value function employed in analysis

To instantiate the SHAP analysis for model, we employ a Gaussian value function to assess motifs based on their proximity to the mean proxy of a specific color (or LII) group as follows:

$$\phi_i(v) = \frac{1}{|N|} \sum_{S \subseteq N \setminus \{i\}} \binom{|N|-1}{|S|} \left(v(S \cup \{i\} \mid \mu, \sigma^2) - v(S \mid \mu, \sigma^2) \right) \tag{4}$$

In this equation, μ and σ^2 are the mean and variance of each group from the training data respectively, where we considered both color and brightness groups based on the partitioning of λ_p and LII defined in the previous section.We scored motifs based on their difference from the cluster mean and selected the top 20 subsequences closest to the mean to identify key nucleobase patterns learned by the VAE for mapping sequences to properties. Figure $S7$ displays the probability of each nucleobase at each position across different λ_p ranges. The top 20 subsequences for a sample of Green($\lambda_p < 590$ nm) and NIR ($\lambda_p > 800$ nm) DNA templates are shown in Table S2.

3. Experimental details

Experimental Ag$_N$-DNA synthesis was performed using previously developed methods; extensive details of these methods can be found in prior work.[13] Details are briefly summarized as follows. Ag$_N$-DNA synthesis was performed on 384 well clear bottom microplates (Corning) using a Tecan Freedom Evo 150 robotic liquid handler equipped with a 96 MultiChannel Arm. DNA oligomers (Integrated DNA Technologies, 40 μM in H_2O, standard desalting) were mixed via pipetting with an aqueous solution of $AgNO_3$ and NH_4OAc (Sigma Aldrich), pH 7. After 18 minutes, Ag$_N$-DNA solutions are partially reduced by a freshly prepared solution of $NaBH_4$ (Sigma Aldrich). Finally, the microplate is centrifuged at low speed for < 60 seconds to remove any small bubbles in wells. Final stoichiometries were selected to match

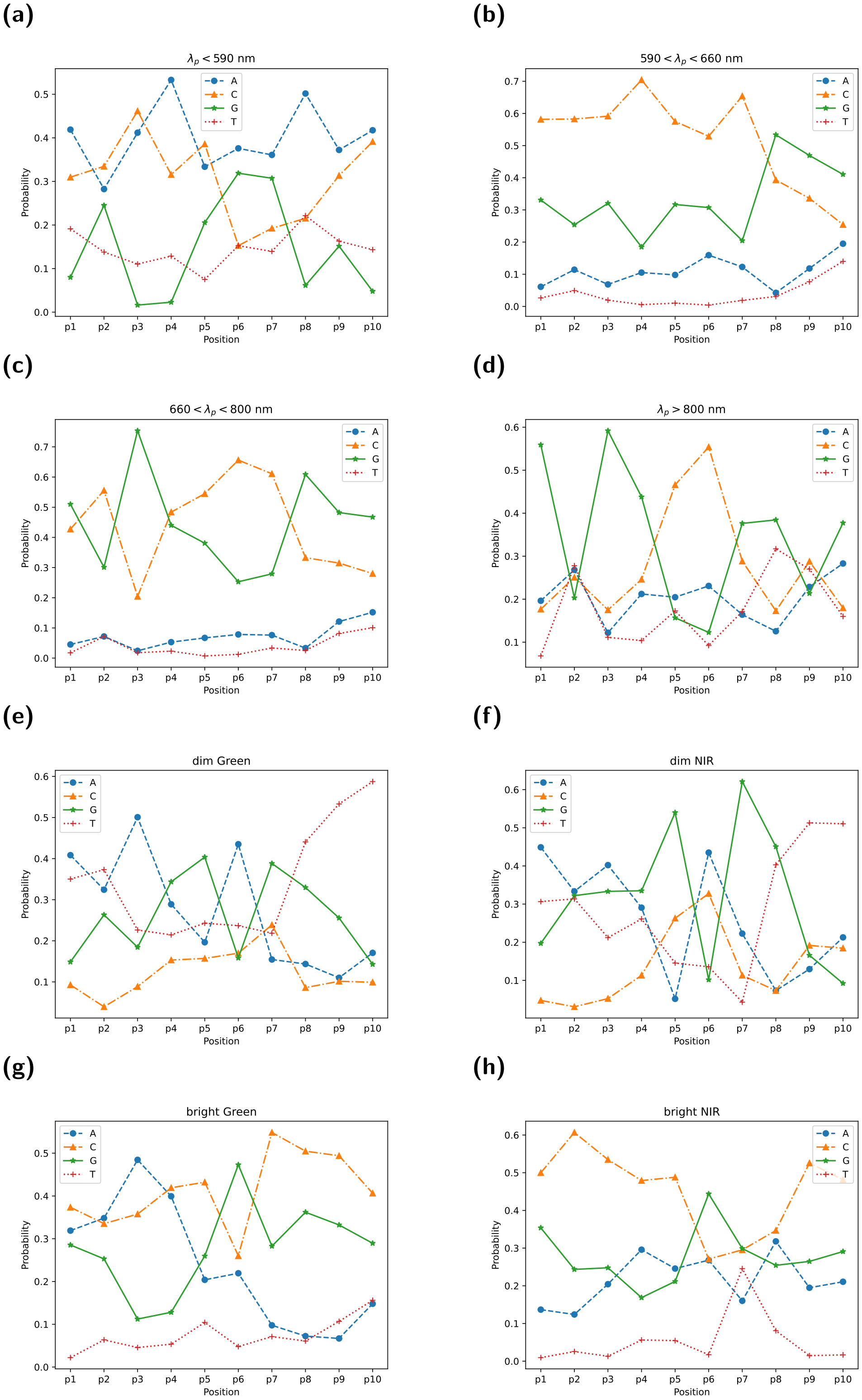


Fig. S7: Top 20 sub-sequences identified from primary sequences, ranked by Shapley scores, representing proximity to the mean λ_p range for (a) Green, (b) Red, (c) Far Red, and (d) NIR sequences. The linegraphs show the top 20 sub-sequences from the Shapley value analysis for (e) dim Green, (f) dim NIR, (g) bright Green, and (h) bright NIR sequences. ”Dim” denotes sequences within the lowest 30% of LII values, while ”bright” indicates sequences within the highest 30% of LII values

Table S2: Top 20 sub-sequences derived from primary sequences, ranked by Gaussian Shapley scores that reflect their proximity to the mean of their respective group. Presented in ascending order, the sub-sequences at the top possess the highest scores, highlighting their pivotal role in the model's predictions.

	$\lambda_p < 590nm$		$\lambda_p > 800nm$	
Index	**AAAATCCCTA**	**AGAGTCCAAC**	**GGGGACCTAA**	**CGAGAACTCA**
1	$A - - - -C - - - -$	$A - - - - - - - - -$	$-G - - - -C - --$	$-G - - - - - - - -$
2	$- - - - - - - - T-$	$- - A - - - - - - -$	$-G - - - -C - -A$	$-G - - - - - - - A$
3	$-A - - - -C - --$	$A - A - - - - - - -$	$-G - - - - - - - A$	$-G - - - -C - --$
4	$A - - - -C - - - A$	$A - - - -C - - - -$	$GG - - - -C - --$	$-G - - - - - -C-$
5	$A - - - -CC - -A$	$A - A - - - - - -C$	$G - - - - - C - -A$	$- - - - - - - - -A$
6	$- - A - T - - - --$	$A - - - - - C - --$	$-G - - - C - - - A$	$-G - - - -C - C-$
7	$-A - -TC - - - -$	$- - A - -C - - - -$	$G - - - - - - - A$	$-G - - - -C - -A$
8	$A - - - -CC - --$	$- - A - - - C - --$	$-G - - - CC - -A$	$-G - - - - - -CA$
9	$- - A - T - C - --$	$A - A - T - - - --$	$G - - - -C - - - A$	$- - - - - - C - --$
10	$-A - - - -CCT-$	$- - A - T - - - --$	$GG - - - CC - --$	$-G - G - - - - - A$
11	$-A - - - C - - - -$	$A - A - - - C - --$	$GG - - - - - --$	$-G - - - -C - CA$
12	$A - - - - - C - --$	$A - - - - - -A - -$	$G - - - - - --$	$-G - -A - - - --$
13	$- - - - -C - - - -$	$A - - - T - - - --$	$G - - - - - C - --$	$CG - - - -C - --$
14	$- - -ATC - - - -$	$- - AG - - - - - -$	$GG - - - C - - - -$	$-G - G - - - -C-$
15	$A - - - T - - - --$	$A - AG - - - - - -$	$G - - - -CC - -A$	$-G - G - -C - C-$
16	$-A - - - - - - - A$	$A - - - -CC - --$	$-G - - - C - - - -$	$-G - G - -C - --$
17	$- - A - - - C - --$	$A - A - -C - - - -$	$- - - - - - C - --$	$-G - G - - - - - -$
18	$A - A - -CC - -A$	$A - A - - - -A - C$	$-G - - - - - - -AA$	$-G - G - - - -CA$
19	$- - A - - - CCT-$	$- - A - - - -A - -$	$- - G - - - C - -A$	$-GA - - - - - -A$
20	$AA - - - CC - -A$	$AGA - - - - - --$	$-G - - - CC - --$	$-G - -A - C - C-$

conditions used for training data collection in previous work.[13–15] 5 Ag^+/DNA stoichiometry was used for measurements in the visible spectrum, whereas 7 Ag^+/DNA stoichiometry was used for measurements in the NIR. The final concentration of DNA was 20 μM in both cases, and $NaBH_4/AgNO_3$ was always maintained at 0.5. Well plates were stored in the dark at 4 °C and measured 7 days after synthesis. Full experimental details are provided in freely available supporting information of past publications.[13]

Fluorescence emission spectra were collected using two microplate readers. A Tecan Spark was used to acquire emission in the visible range (400-850 nm). NIR emission (675-1,425 nm) was measured in a Tecan Infinity 200 Pro with a custom-built InGaAs photodetector,[16] using 50 nm bandpass filters and posteriorly correcting for detector spectral responsivity. In both instruments, 280 nm light was used to universally excite all Ag_N-DNAs.[17] To extract peak wavelength from visible emission spectra, the 400-800 nm range is fit to the sum of three Gaussians in terms of energy (in eV). The LII is normalized to a well studied Ag_N-DNA with green and red emission across all training data.[18] Any spectra with more than 3 peaks or with a normalized LII less than 0.5 were discarded. For NIR emission, peak wavelengths were extracted using a weighted average of the of the maximum, and two neighboring points to the right and left. The resulting training dataset comprises 2,204 DNA sequences, each with 10 nucleobases, and includes the wavelength and brightness properties of the stabilized Ag_N-DNAs.

References

(1) Moomtaheen, F.; Killeen, M.; Oswald, J.; Gonzàlez-Rosell, A.; Mastracco, P.; Gorovits, A.; Copp, S. M.; Bogdanov, P. DNA-Stabilized Silver Nanocluster Design via Regularized Variational Autoencoders. Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022; pp 3593–3602.

(2) Ochiai, T.; Inukai, T.; Akiyama, M.; Furui, K.; Ohue, M.; Matsumori, N.; Inuki, S.; Uesugi, M.; Sunazuka, T.; Kikuchi, K.; others Variational autoencoder-based chemical latent space for large molecular structures with 3D complexity. *Communications Chemistry* **2023**, *6*, 249.

(3) Mansoor, S.; Baek, M.; Park, H.; Lee, G. R.; Baker, D. Protein Ensemble Generation through Variational Autoencoder Latent Space Sampling. *Journal of Chemical Theory and Computation* **2024**, *20*, 2689–2695.

(4) Lew, A. J.; Buehler, M. J. Encoding and exploring latent design space of optimal material structures via a VAE-LSTM model. *Forces in Mechanics* **2021**, *5*, 100054.

(5) Iovanac, N. C.; Savoie, B. M. Improved chemical prediction from scarce data sets via latent space enrichment. *The Journal of Physical Chemistry A* **2019**, *123*, 4295–4302.

(6) Copp, S. M.; Schultz, D.; Swasey, S.; Pavlovich, J.; Debord, M.; Chiu, A.; Olsson, K.; Gwinn, E. Magic Numbers in DNA-Stabilized Fluorescent Silver Clusters Lead to Magic Colors. *The Journal of Physical Chemistry Letters* **2014**, *5*, 959–963.

(7) Peng, D.; Gu, T.; Hu, X.; Liu, C. Addressing the multi-label imbalance for neural networks: An approach based on stratified mini-batches. *Neurocomputing* **2021**, *435*, 91–102.

(8) Hughes, C. Demystifying pytorch's Weightedrandomsampler by example. 2022; `https://towardsdatascience.com/demystifying-pytorchs-weightedrandomsampler-by-example-a68aceccb452`.

(9) Ansel, J. et al. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24). 2024.

(10) Li, Y.; Ghosh, S. K. Efficient sampling methods for truncated multivariate normal and student-t distributions subject to linear inequality constraints. *Journal of Statistical Theory and Practice* **2015**, *9*, 712–732.

(11) Loecher, A.; Bruyns-Haylett, M.; Ballester, P. J.; Borros, S.; Oliva, N. A machine learning approach to predict cellular uptake of pBAE polyplexes. *Biomaterials Science* **2023**, *11*, 5797–5808.

(12) Remmer, E. Explainability Methods for Transformer-based Artificial Neural Networks:: a Comparative Analysis. 2022.

(13) Mastracco, P.; Gonzàlez-Rosell, A.; Evans, J.; Bogdanov, P.; Copp, S. M. Chemistry-informed machine learning enables discovery of DNA-stabilized silver nanoclusters with near-infrared fluorescence. *ACS nano* **2022**, *16*, 16322–16331.

(14) Copp, S. M.; Gorovits, A.; Swasey, S. M.; Gudibandi, S.; Bogdanov, P.; Gwinn, E. G. Fluorescence color by data-driven design of genomic silver clusters. *ACS nano* **2018**, *12*, 8240–8247.

(15) Swasey, S. M.; Copp, S. M.; Nicholson, H. C.; Gorovits, A.; Bogdanov, P.; Gwinn, E. G. High throughput near infrared screening discovers DNA-templated silver clusters with peak fluorescence beyond 950 nm. *Nanoscale* **2018**, *10*, 19701–19705.

(16) Swasey, S. M.; Nicholson, H. C.; Copp, S. M.; Bogdanov, P.; Gorovits, A.; Gwinn, E. G. Adaptation of a visible wavelength fluorescence microplate reader for discovery of near-infrared fluorescent probes. *Review of Scientific Instruments* **2018**, *89*, 095111.

(17) O'Neill, P. R.; Gwinn, E. G.; Fygenson, D. K. UV excitation of DNA stabilized Ag cluster fluorescence via the DNA bases. *The Journal of Physical Chemistry C* **2011**, *115*, 24061–24066.

(18) Cerretani, C.; Vosch, T. Switchable dual-emissive DNA-stabilized silver nanoclusters. *ACS omega* **2019**, *4*, 7895–7902.