Accelerating Antimicrobial Peptide Discovery with Latent Sequence-Structure Model

Danqing Wang[1,2]*, **Zeyu Wen**[1], **Fei Ye**[1], **Lei Li**[2], **Zhou Hao**[3]
[1]ByteDance Research, Shanghai, China [2]University of California, Santa Barbara
[3]Insititute for AI Industry Research, Tsinghua University
danqingwang,leili@ucsb.edu, wenzeyu,yefei.joyce@bytedance.com
zhouhao@air.tsinghua.edu.cn

Abstract

Antimicrobial peptides (AMPs) offer a promising approach for treating a wide range of antibiotic-resistant infections. Recently, there has been a surge of interest in using deep generative models to expedite the discovery of AMPs. However, most current research focuses on sequence characteristics and overlooks structural information, which is crucial for AMP biological function. In this paper, we present a latent sequence-structure model for AMPs (`LSSAMP`) that employs multi-scale VQ-VAE to integrate secondary structures. By sampling from the latent space, `LSSAMP` can concurrently generate peptides with optimal sequence properties and secondary structures. Experimental outcomes indicate that the peptides produced by `LSSAMP` exhibit a high likelihood of being AMPs, and two out of 21 candidates have been confirmed to possess potent antimicrobial activity. We will release our model to facilitate the generation of high-quality AMP candidates for subsequent biological experimentation and expedite the overall AMP discovery process[1].

1 Introduction

In recent times, the application of neural networks to drug discovery has garnered increased interest, as it can expedite the identification of potential treatments while decreasing drug development time and costs (Stokes et al., 2020). Notable progress has been made in using deep generative models to hasten the discovery of drug-like molecules (Jin et al., 2018; Shi et al., 2019; Schwalbe-Koda & Gómez-Bombarelli, 2020; Xie et al., 2020).

Antimicrobial peptides (AMPs) represent one of the most promising new therapeutic agents to supplant antibiotics. These short proteins can eliminate bacteria by disrupting their membranes (Aronica et al., 2021; Cardoso et al., 2020). Unlike the chemical interactions between antibiotics and bacteria that can be circumvented through bacterial evolution, this physical mechanism is more challenging to resist.

A conventional antimicrobial discovery process typically comprises four stages, depicted in Figure 1. Initially, a candidate library is constructed based on the existing AMPs database. Candidates can be created using manual heuristic methods or by training deep generative models. Next, various sequence-based filters are established to screen candidate peptides according to diverse chemical features, including computational metrics and predictive models trained to estimate ideal properties. Subsequently, to ensure that these sequences can adopt appropriate biologically functional structures, peptide structure predictors such as PEPFold 3 (Shen et al., 2014) are used to model the sequences' structures, followed by molecular dynamics simulations. Finally, the filtered sequences are synthesized and examined in wet laboratory experiments. In Figure 1, the grep region represents the bacterial suspension, and the white area signifies a low bacterial concentration in this region.

*Work was done when Danqing Wang was in Bytedance Research.
[1]The code is available at https://github.com/dqwang122/LSSAMP

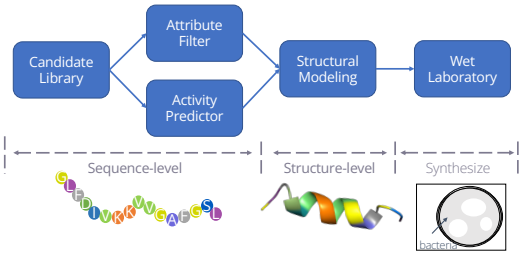


Figure 1: The overview of AMP discovery. The first two steps focus on sequence attributes and the third models the structure. The final step is to verify the antimicrobial activity by inhibiting the growth of bacteria.

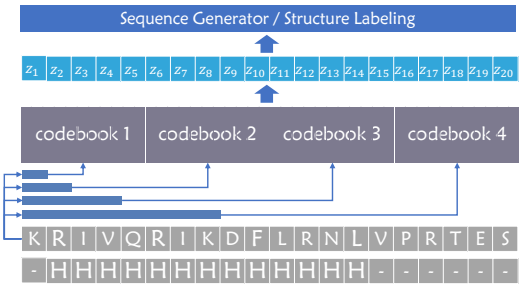


Figure 2: The encoder of LSSAMP. Here, we use $N = 4$ pattern selectors to select various local patterns for each position. The number of selectors is further discussed in Section 3.4.

Deep generative models have recently achieved considerable success in accelerating AMP discovery by using sequence attributes to control generation and directly produce peptides with ideal attributes (Das et al., 2018; 2021; Van Oort et al., 2021). However, these studies solely consider sequence features and neglect the structure-activity relationship. Generated sequences still require processing by structure predictors and manual verification, slowing down the discovery process. Furthermore, the structure significantly influences biological properties, thereby aiding attribute control (Chen et al., 2019; Torres et al., 2018; Tucker et al., 2018).

In this paper, we integrate structure information into the generative model and introduce a Latent Sequence-Structure model for AntiMicrobial Peptide (LSSAMP). It concurrently maps sequence features and secondary structures into a shared latent space and samples peptides with optimal sequence compositions and structures. LSSAMP controls generation in a more refined manner by assigning a latent variable to each position rather than a continuous variable to control the attributes of the entire sequence. We utilize a multi-scale vector quantized-variational autoencoder (VQ-VAE) (van den Oord et al., 2017) to capture sequence and structure patterns of varying lengths. During the generation process, LSSAMP samples from the latent space and generates a peptide sequence along with its secondary structure. Public AMP predictor-based experimental results demonstrate that the peptides generated by LSSAMP exhibit a high AMP probability. Our comprehensive qualitative analysis reveals that our model captures the sequence and structure distribution. We select 21 generated peptides for wet laboratory experimentation and discover that 2 of them exhibit potent antimicrobial activity against Gram-negative bacteria.

To conclude, our contributions are as follows:

- We propose LSSAMP, a sequence-structure generative model that combines secondary structure information into the generation. It can further accelerate AMP discovery by merging the first three steps together.
- We develop a multi-scale VQ-VAE to control the generation in a fine-grained manner and map patterns in sequences and structures into the same latent space.
- Experimental results of AMP predictors show that LSSAMP generates peptides with high probabilities of AMP. Moreover, 2 of 21 generated peptides show strong antimicrobial activities in wet laboratory experiments.

2 Method

In this section, we first discuss how existing generative models expedite this process and their limitations. We then explore previous work employing popular VAE-based models for peptide generation. Building on this, we present the Latent Sequence-Structure model for AMP (LSSAMP), which utilizes the multi-scale VQ-VAE to map sequence and structure distributions into a shared latent space, simultaneously sampling peptides with ideal sequences and structures.

	Uniq	C	H	uH	Combination
VAE	475	18.45% ± 2.92%	2.68% ± 3.28%	-2.78% ± 1.64%	0.29% ± 0.74%
AMP-GAN	1966	2.79% ± 0.50%	2.16% ± 0.34%	-2.29% ± 0.53%	0.17% ± 0.35%
PepCVAE	208	3.87% ± 1.58%	-1.93% ± 1.61%	1.01% ± 2.80%	3.93% ± 1.82%
MLPeptide	2106	-2.48% ± 0.39%	2.01% ± 0.57%	9.24% ± 1.22%	1.12% ± 0.38%

Table 1: The delta ratio of the sequence properties that underwent secondary structure filtering, which reflects the difference in performance before and after the filter was applied. Our experiments were repeated thrice, and we calculated the error bars. **Uniq** is the unique peptide number in 5000 generated sequences. **C**, **H**, **uH** correspond to charge, hydrophobicity, hydrophobic moment. **Combination** is the percentage of satisfying three ranges at the same time.

Notations The peptide[2] is a short protein comprised of amino acids. A peptide of length L can be represented as $\boldsymbol{x} = x_1, x_2, \cdots, x_L$. The amino acid x_i at the i-th position is one of the 20 common types and is also referred to as a *residue*. The *secondary structure* is employed to describe the local form of the peptide's 3D structure. Therefore, the structure of the peptide can be denoted as $\boldsymbol{y} = y_1, y_2, \cdots, y_L$, where y_i is the secondary structure label of the i-th position, belonging to one of eight types [3].

2.1 Antimicrobial Peptide Discovery

Deep generative models have demonstrated potential in accelerating AMP discovery by merging sequence-based filters with the generation process and directly producing sequences with user-defined properties as the candidate library. However, previous studies exclusively focus on learning sequence features. They still need to verify and filter structures using external computational tools after generating sequences, rendering the generation process inefficient. For instance, Van Oort et al. (2021) chose 12 cationic and helical peptides among generated peptides, and Capecchi et al. (2021) employed the predicted α-helix structure fraction percentage to filter peptides post-generation.

Additionally, there is a strong relationship between peptide structure and activity. We examine the impact of secondary structure on sequence properties by filtering generated sequences based on the proportion of α-helices, the most prevalent secondary structure in AMPs. In Table 1, we use three sequence attributes (*charge*, *hydrophobicity*, *hydrophobic moment*) vital for the AMP mechanism to assess generation performance (Yeaman & Yount, 2003; Gidalevitz et al., 2003; Wimley, 2010). The ratio in Table 1 represents the performance difference before and after the secondary structure filter (discussed further in §A.3). We observe that most results are improved by limiting α-helical structures. These findings indicate that by controlling the structure, sequence properties can be enhanced. Therefore, incorporating structure information into generative models can not only speed up discovery by combining all steps before the wet laboratory but also improve sequence properties, making the generative process more efficient.

2.2 VAE-based Generative Models

Given a sequence $\boldsymbol{x}$, the variational auto-encoders assume that it depends on a continuous latent variable $\boldsymbol{z}$. Thus the likelihood can be denoted as: $p(\boldsymbol{x}) = \int p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})d\boldsymbol{z}$. The controlled sequence generation incorporates the attribute a and models the conditional probability $p(\boldsymbol{x}|a)$. Based on the dependency between latent variable $\boldsymbol{z}$ and attribute a, these peptide generative models can be divided into semi-VAEs, such as PepCVAE (Das et al., 2018), and GM-VAEs, such as CLaSS (Das et al., 2021).

[2]Here, we use the term peptide to refer to both oligopeptides (< 20 amino acids) and polypeptides (< 50 amino acids).

[3]The three alpha helices are denoted as H, G, and I based on their angles. The two beta sheets are distinguished by E and T according to their shape. The others are random coil structures (Kabsch & Sander, 1983).

The vanilla VAE are usually trained in an auto-encoder framework with regularization. The encoder parameterizes an approximate posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and the decoder reconstructs $\boldsymbol{x}$ based on the latent $\boldsymbol{z}$. The models optimizes a evidence lower bound (ELBO):

$$L_r = E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[log(p_\theta(\boldsymbol{x}|\boldsymbol{z}))] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})), \tag{1}$$

where the $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[log(p_\theta(\boldsymbol{x}|\boldsymbol{z}))]$ is the reconstruction loss and the KL divergence is the regularization. For the conditional generation, the attributes are directly added to the latent variable z, or trained on the latent space to get an attribute-conditioned posterior distribution $p(\boldsymbol{z}|a)$. The VAE-based peptide generative models are first trained on the unsupervised peptide or protein sequences and then trained with specific sequences with biological attribute labels.

A latent variable $\boldsymbol{z}$ is sampled from the latent space and then fed to the decoder to generate a new sequence. The attributes control the generation with the latent $\boldsymbol{z}$.

2.3 Latent Sequence-Structure Model

Traditional VAE models for peptides only learn the sequence distribution. To combine the secondary structure with sequence attributes to further accelerate the discovery and build a more effective candidate library. Different from previous work, we assign a latent variable z_i for each x_i instead of a continuous $\boldsymbol{z}$ for the whole sequence. This gives our model a more fine-grained control on each position. Since it is computationally intractable to sum continuous latent variables over the sequence, we use VQ-VAE (van den Oord et al., 2017) to lookup the discrete embedding vector $\boldsymbol{z}_q = \{z_q(x_1), \cdots, z_q(x_L)\}$ for each position by vector quantization.

Specifically, the encoder output $z_i = z_e(x_i) \in \mathbb{R}^d$ will be replaced by the codebook entry $z_q(x_i) \in \mathbb{R}^d$ via a nearest neighbors lookup from the codebook $\boldsymbol{B} \in \mathbb{R}^{K\times d}$:

$$z_q(x_i) = e_k, \text{and } k = \mathrm{argmin}_{j\in\{1,\cdots,K\}} \left\| z_e(x_i) - e_j \right\|_2 . \tag{2}$$

Here, K is the slot number of the codebook and d is the dimension of the codebook entry e. Then, the generator will take $z_q(x_i)$ as its input and reconstruct x_i. The training objective L_r is defined as:

$$L_r = \sum_{i=1}^{L} \log p\left(x_i|z_q(x_i)\right) + \left\|\mathrm{sg}\left[z_e(x_i)\right] - z_q(x_i)\right\|_2^2 + \beta \left\| z_e(x_i) - \mathrm{sg}[z_q(x_i)]\right\|_2^2 . \tag{3}$$

Here, $sg(\cdot)$ is the stop gradient operator, which becomes 0 at the backward pass. β is the commit coefficient. The $\sum_{i=1}^{L} \log p\left(x_i|z_q(x_i)\right)$ is the discrete reconstruction loss and the rest components perform as the KL divergence regularization, like the second term in Eqn. 1.

For secondary structure modeling, we predict the structure label y_i for i-th residue. We add a separate decoder on top of the latent representation $z_{q'}(x_i)$. The training objective L_s is similar to Eqn. 3 except that the first term is a supervised version $\sum_{i=1}^{L} \log p\left(y_i|z_{q'}(x_i)\right)$. The sequence and structure codebook are not necessarily the same, thus we use $z_{q'}(x_i)$ to indicate the structure latent variable.

Multi-scale VQ-VAE The structure motifs are often longer than sequence patterns. For example, a valid α- helix contains at least 4 residues and may be longer than 12. However, sequence patterns with specific biological functions are much shorter, usually between 1 and 8 residues. To capture these features and map them into the same latent space, we first apply N multi-scale pattern selectors F_n. Then, we establish multiple codebooks and use Eqn. 2 to look up the nearest codebook embedding $z_{q_n}(x_i)$. We share the codebooks between sequence reconstruction and secondary structure prediction to capture common features and relationships between the residue and its structure. The concatenated multi-scale codebook embedding is fed to the sequence generator: $z_q(x_i) = \|_{n\in N} z_{q_n}(x_i)$, Based on Eqn. 3, the reconstruction training objective for multi-scale VQ-VAE can be adapted as the sum of loss on multiple codebooks.

Thus, the total training loss is composed of the reconstruction loss and the labeling loss, which can be denoted as: $L = L_r + \gamma L_s$ (4) where the γ is the weight of the structure prediction task.

Training As VAE-based generative models, we first train LSSAMP in an unsupervised manner with protein sequences via L_r. Then, we incorporate the structure information by jointly training L_r and L_s on a smaller protein dataset with secondary structure annotation. Finally, we finetune our model on the AMP dataset to capture the specific AMP characteristics. The whole training process is described in Algorithm 1 of §.

Prior Model The prior distribution over the codebook is a categorical distribution and can be made auto-regressive by the extra prior model. To model the dependency between $z_{1:L}$, following (van den Oord et al., 2017) we train Transformer-based language models on the embedding entries. We extract the index sequences generated by Eqn. 2 for each codebook n and then train M_{prior_n} on them, as shown in Line 4-9 in Algorithm 1.

Sampling We sample several index sequences from the prior models for each codebook n, and then lookup the codebook to get the embedding vector $\boldsymbol{z}_{q_n}$. Finally, $\boldsymbol{z}_{q_n}$ is fed to the generator and classifier to generate the sequence with its secondary structure. We also try to control the secondary structure by existing AMP structure patterns to further improve the generation quality.

3 Experiment

	SVM	RF	DA	Scanner	AMPMIC	IAMPE	amPEP	Average
APD	87.78%	91.24%	86.24%	94.66%	98.42%	97.83%	91.50%	92.52%
Decoy	17.43%	13.71%	16.04%	0.25%	18.07%	23.53%	52.92%	20.28%
Random $p = 0.1$	86.06%	86.12%	84.01%	93.23%	79.14%	95.60%	91.74%	87.99%
Random $p = 0.2$	76.66%	76.64%	74.83%	86.95%	68.57%	91.14%	87.89%	80.38%
VAE	24.90%	15.30%	13.83%	15.12%	15.25%	40.31%	24.30%	21.29%
AMP-GAN	78.62%	87.29%	83.82%	82.17%	89.58%	93.88%	80.52%	85.13%
PepCVAE	82.84%	85.96%	93.33%	85.44%	**98.44**%	**98.14**%	80.77%	89.27%
MLPeptide	90.43%	92.55%	93.08%	**93.72**%	96.34%	97.05%	91.37%	93.51%
LSSAMP	**92.03**%	**92.60**%	**93.45**%	91.52%	95.84%	96.64%	**93.23**%	**93.62**%
LSSAMP **w/o cond**	78.98%	80.24%	80.01%	86.73%	83.81%	93.80%	85.32%	84.13%

Table 2: The percentage of generated sequences being predicted as AMP. The classifiers are described in §3.2. The first part is the prediction results on AMP and non-AMP dataset as the reference. The bold ones are the best model results.

3.1 Experiment Setup

Dataset The Universal Protein Resource (UniProt)[4] is a comprehensive protein dataset. We download reviewed protein sequences (550k) with the limitation of 100 in length as D_r (57k examples). Then we use a community reimplementation of AlphaFold(AlQuraishi, 2019), which is called ProSPr[5] (Billings et al., 2019) to predict the secondary structure for D_r. After filtering low-quality examples with all coil or unknown secondary structures, we obtain D_s with 46k examples, including both sequence and secondary structure information. Here, we use the predicted secondary structures to augment the limited size of the existing secondary structures. For the antimicrobial peptide dataset, we download from Antimicrobial Peptide Database (APD3)[6] (Wang et al., 2016) and filter repeated ones to get 3222 AMPs as D_{amp}. We randomly extract 3,000 examples as validation and 3,000 as test on D_r and D_s. For D_{amp}, the size of validation and test is both 100. Following Veltri et al. (2018), we create a decoy set of negative examples without antimicrobial activities for comparison. It removes peptide sequences with antimicrobial activity from Uniprot, and sequences with length < 10 or > 40, resulting in 2021 non-AMP sequences.

Baseline Traditional methods usually randomly replace several residues on existing AMPs and conduct biological experiments on them. Thus, we use **Random** baseline to represent

[4]https://www.uniprot.org/

[5]https://github.com/dellacortelab/prospr/tree/prospr1

[6]https://aps.unmc.edu/

the method of replacing each residue with probability p. Following Dean & Walper (2020), we use **VAE** to embed the peptides into the latent space and sample latent variable z from the standard Gaussian distribution $p \sim N(0, 1)$. For a fair comparison, we use the same Transformer architecture as our model `LSSAMP` and train on the Uniprot D_r and APD dataset D_{amp}. **AMP-GAN** is proposed by Van Oort et al. (2021), which uses a BiCGAN architecture with convolution layers. It consists of three parts: the generator, discriminator, and encoder. The generator and discriminator share the same encoder. It is trained on 49k false negative sequences from UniProt and 7k positive AMP sequences. **PepCVAE** is a semi-VAE generative model that concatenates the attribute features to the latent variable for conditional generation (Das et al., 2018). Since the authors did not release their code, we use the model architecture from Hu et al. (2017) and modify the reproduced code[7] for AMPs, as described in their paper. The original paper uses 93k sequences from UniProt and 7960/6948 positive/negative AMPs for training. For comparison, we use UniProt dataset D_r and ADP dataset D_{amp} to train it. **MLPeptide** (Capecchi et al., 2021) is RNN-based generator. It is first trained on 3580 AMPs and then transferred to specific bacteria. `LSSAMP` is implemented as described in §2.3. The detailed implementation is discussed in §C.1.

3.2 Evaluation Metric

Following previous work (Das et al., 2020; Van Oort et al., 2021), we use open-source AMP prediction tools to estimate the AMP probability of the generated sequence. Since these open-source AMP predictors are trained and report results in different AMP datasets, we use APD and decoy datasets as a reference of their performance. We also evaluate the generative diversity of these models and the sequence attributes in §B.2.

AMP Classifiers Thomas et al. (2010) trained on the AMP database of 3782 sequences with random forest (**RF**), discriminant analysis (**DA**), support vector machines (**SVM**)[8], and artificial neural network (ANN)[9] respectively. AMP Scanner v2[10] (Veltri et al., 2018), short as **Scanner**, is a CNN-&LSTM-based deep neural network trained on 1778 AMPs picked from APD. **AMPMIC**[11] (Witten & Witten, 2019) trained a CNN-based regression model on 6760 unique sequences and 51345 MIC measurement to predict MIC values. **IAMPE**[12] (Kavousi et al., 2020) is a model based on Xtreme Gradient Boosting. It achieves the highest correct prediction rate on a a set of ten more recent AMPs (Aronica et al., 2021). **ampPEP**[13] (Lawrence et al., 2021) is a random forest based model which is trained on 3268 AMP sequences. It has the best performance across multiple datasets (Aronica et al., 2021).

Wet Laboratory Experiments Following the previous AMP design (Capecchi et al., 2021; Das et al., 2021), we use minimum inhibitory concentration (MIC) to indicate peptide activity, which is defined as the lowest concentration of an antibiotic that prevents the visible growth of bacteria. A lower MIC means a higher antimicrobial activity. To determine MIC, the broth microdilution method was used. The detail setting is put in §C.2

3.3 Experimental Results

We generate 5000 sequences for each baseline. During the generation process, we add some structural restrictions on positions based on the antimicrobial mechanism. Specifically, we reject peptides with more than 30% coil structure ('-'), which can hardly fold in the solution environment and insert into the bacterial membrane in silico screening. Besides, we limit the minimum length of a continuous helix ('H') to 4 according to physical rules. We name our model with structural control as `LSSAMP` and the model without extra conditions as `LSSAMP` **w/o cond**.

[7] https://github.com/wiseodd/controlled-text-generation
[8] http://www.camp3.bicnirrh.res.in/prediction.php
[9] We drop the ANN model because its accuracy on APD is low (82.83%).
[10] https://www.dveltri.com/ascan/v2/ascan.html
[11] https://github.com/zswitten/Antimicrobial-Peptides
[12] http://cbb1.ut.ac.ir/AMPClassifier/Index
[13] https://github.com/tlawrence3/amPEPpy

AMP Prediction The results of prediction tools are shown in Table 2. `LSSAMP` performs best in four of seven and has the highest average score across all classifiers, indicating its advantage over baselines. PepCVAE performs best on the AMPMIC and IAMPE predictors, however, it performs poorly on the other predictors and gets a low average score. MLPeptide performs relatively evenly across predictors, outperforming other models on only Scanner and slightly underperforming our model on the average score. The comparison of `LSSAMP` and `LSSAMP` w/o cond indicates that adding fine-grained control on the secondary structure can further improve the generation performance.

No	Activity (ug/mL) ↓			Sequence identity ↓	Hemolysis/Toxicity ↓
	A. Baumannii	P. aeruginose	E. coli		
P1	16-32	/	32-64	83.30%	Low
P2	8	32	/	75.00%	Low

Table 3: Wet laboratory experiment results. P1 is GAFGNFLKNVAKKAGIYLLSI-AQCKLFGTP and P2 is FIGFLFKLAKKIIPSLFQTKTE. 'Sequence identity' measures the similarity with existing AMPs and 'Hemolysis/Toxicity' measures the damage to other cells.

Wet Laboratory Experiment We synthesized and conducted experimental tests on peptides generated using `LSSAMP`. First, we filtered the produced sequences based on their physical attributes (as outlined in §A.1) and employed AMP classifiers to choose those predicted to have antimicrobial properties (as detailed in §3.2). Next, we sorted the sequences according to their novelty (as described in §B.2) and selected those with a distance greater than 5. Ultimately, we obtained 21 peptides and examined their antimicrobial activities against three types of Gram-negative bacteria (*A. Baumannii*, *P. aeruginosa*, *E. coli*), which took approximately 30 days. We utilized minimal inhibitory concentration (MIC) to assess the activity. The wet laboratory experiment specifics are provided in §C.2. As indicated in Table 3, two peptides were found to be effective against A. Baumannii. **P2** against P. aeruginosa and **P1** against E. coli also exhibited activity. Additionally, these two newly identified AMPs differ from existing ones (similarity $< 85\%$) and exhibit low toxicity, making them promising new therapeutic agents. The wet-lab experiment outcomes demonstrate that `LSSAMP` can efficiently identify AMP candidates and decrease the required time.

3.4 Analysis

Ablation Study We conduct the ablation study for our `LSSAMP` and show the results in Table 4. **PPL** is the perplexity of generated sequences that can measure fluency. **Loss** is the model loss on the validation set. **AA Acc.** is the reconstruction accuracy of residue and **SS Acc.** is the prediction accuracy of the secondary structure. We can find that without the first training phase on D_r, the model can hardly generate valid sequences. The second phase to train the model on the large-scale secondary structure dataset D_s will affect the prediction performance on the target AMP dataset. If we remove multiple sub-codebooks (SB) and use a single large codebook with the same size, the performance will decline.

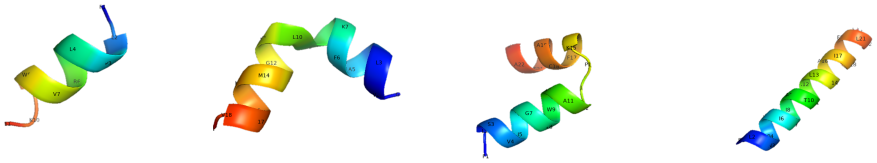
Codebook Number We explore the effect of different numbers of codebooks on generation performance. From Table 5, we find that a single small codebook can hardly learn enough information to reconstruct the sequence. The PPL, Loss, and SS Acc. become better with

	PPL ↓	Loss ↓	AA Acc.↑	SS Acc.↑
`LSSAMP`	**3.12**	**1.14**	**99.93**	**86.76**
w/o D_r	11.56	2.45	66.06	82.78
w/o D_s	3.83	1.34	99.58	85.87
w/o SB	3.49	1.25	99.86	86.61

Table 4: Ablation Study on validation set of D_{amp} . See §3.4 for details. The full table with std is Table 10.

Codebook	PPL ↓	Loss ↓	AA Acc.↑	SS Acc.↑
[1]	19.04	2.94	65.49	83.41
[1, 2]	3.84	1.35	99.40	85.39
[1, 2, 4]	3.32	1.20	**100.00**	85.95
[1, 2, 4, 8]	**3.24**	**1.17**	99.79	**87.20**

Table 5: The influence of the number of codebooks. '[1,2]' indicates that we use 2 codebooks to capture local features with window sizes of 1 and 2. The full table is Table 11.



(a) Y1 (b) Y4 (c) Y9 (d) Y10

Figure 3: 3D structures for sequence Y1, Y4, Y9, and Y10 of Table 8.

the increase of codebook entries. However, the reconstruction accuracy achieves the best performance when the codebook is 3. This may be due to the relatively short local pattern of sequences, making the window of 8 too long for it.

Case Study We show 10 peptides generated by `LSSAMP` in §(Table 8). We further build 3D models of 4 generated sequences by PEPFold 3 (Shen et al., 2014) and draw the picture by PyMOL (Schrödinger, LLC, 2015) in Figure 3. We can find that all these peptides have several helical structures, which make them more likely to have the antimicrobial ability. At the same time, although the model predicts a long continuous helical structure for Y4 and Y9, in fact, they have a small coil structure between the two helical structures. It indicates that our model tends to predict a long continuous secondary structure instead of several discontinuous small fragments.

4 Related work

Antimicrobial Peptide Generation Deep generative models have experienced rapid growth in recent years. Dean & Walper (2020) encode peptides into a latent space and interpolate across a predictive vector between a known AMP and its scrambled version to generate novel peptides. PepCVAE (Das et al., 2018) and CLaSS (Das et al., 2021) utilize the variational auto-encoder model for sequence generation. AMPGAN (Van Oort et al., 2021) employs the generative adversarial network to create new peptide sequences, with a discriminator differentiating real AMPs from artificial ones. To the best of our knowledge, this is the first study incorporating secondary structure information into the generative phase, promoting the efficient generation of well-structured sequences with desired properties.

Sequence Generation via VQ-VAE

Sequence Generation via VQ-VAE Variational auto-encoders (VAEs) were first proposed by Kingma & Welling (2014) for image generation. Instead of mapping input to a continuous latent space as in VAE, vector quantized-variational autoencoder (VQ-VAE) (van den Oord et al., 2017) learns a codebook to obtain a discrete latent representation. This method can circumvent issues of posterior collapse while maintaining performance comparable to VAEs. Building on this, Razavi et al. (2019) employs a multi-scale hierarchical organization to capture global and local features for image generation. Bao et al. (2021) learns implicit categorical information of target words with VQ-VAE and models the categorical sequence using conditional random fields in non-autoregressive machine translation. In this paper, we utilize the multi-scale vector quantized technique to obtain the discrete representation for each position of the peptide.

5 Conclusion

In this paper, we present `LSSAMP`, which employs multi-scale VQ-VAE for fine-grained control of each position. It maps sequence and structure features into a shared latent space, and by sampling the overlapping distribution, it can generate peptides with optimal sequence attributes and secondary structures. `LSSAMP` demonstrates strong performance on AMP predictors and designs two peptides with high activity against Gram-negative bacteria. This suggests that our generative model can effectively produce an AMP library with high-quality candidates for subsequent biological experiments, thereby accelerating AMP discovery.

References

Mohammed AlQuraishi. Alphafold at casp13. *Bioinformatics*, 35(22):4862–4865, 2019.

Pietro GA Aronica, Lauren M Reid, Nirali Desai, Jianguo Li, Stephen J Fox, Shilpa Yadahalli, Jonathan W Essex, and Chandra S Verma. Computational methods and tools in antimicrobial peptide research. *Journal of Chemical Information and Modeling*, 61(7): 3172–3196, 2021.

Yu Bao, Shujian Huang, Tong Xiao, Dongqi Wang, Xinyu Dai, and Jiajun Chen. Non-autoregressive translation by learning target categorical codes. In *Proc. of NAACL-HLT*, pp. 5749–5759, 2021.

Wendy M Billings, Bryce Hedelius, Todd Millecam, David Wingate, and Dennis Della Corte. Prospr: democratized implementation of alphafold protein distance prediction network. *bioRxiv*, pp. 830273, 2019.

HG Boman. Antibacterial peptides: basic facts and emerging concepts. *Journal of internal medicine*, 254(3):197–215, 2003.

Alice Capecchi, Xingguang Cai, Hippolyte Personne, Thilo Kohler, Christian van Delden, and Jean-Louis Reymond. Machine learning designs non-hemolytic antimicrobial peptides. *Chemical Science*, 2021.

Marlon H Cardoso, Raquel Q Orozco, Samilla B Rezende, Gisele Rodrigues, Karen GN Oshiro, Elizabete S Cândido, and Octávio L Franco. Computer-aided design of antimicrobial peptides: are we generating effective drug candidates? *Frontiers in microbiology*, 10:3097, 2020.

Charles H Chen, Charles G Starr, Evan Troendle, Gregory Wiedman, William C Wimley, Jakob P Ulmschneider, and Martin B Ulmschneider. Simulation-guided rational de novo design of a small pore-forming antimicrobial peptide. *Journal of the American Chemical Society*, 141(12):4839–4848, 2019.

Payel Das, Kahini Wadhawan, Oscar Chang, Tom Sercu, Cicero Dos Santos, Matthew Riemer, Vijil Chenthamarakshan, Inkit Padhi, and Aleksandra Mojsilovic. Pepcvae: Semi-supervised targeted design of antimicrobial peptide sequences. *arXiv preprint arXiv:1810.07743*, 2018.

Payel Das, Tom Sercu, Kahini Wadhawan, Inkit Padhi, Sebastian Gehrmann, Flaviu Cipcigan, Vijil Chenthamarakshan, Hendrik Strobelt, Cicero dos Santos, Pin-Yu Chen, et al. Accelerating antimicrobial discovery with controllable deep generative models and molecular dynamics. *arXiv preprint arXiv:2005.11248*, 2020.

Payel Das, Tom Sercu, Kahini Wadhawan, Inkit Padhi, Sebastian Gehrmann, Flaviu Cipcigan, Vijil Chenthamarakshan, Hendrik Strobelt, Cicero Dos Santos, Pin-Yu Chen, et al. Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations. *Nature Biomedical Engineering*, 5(6):613–623, 2021.

Scott N Dean and Scott A Walper. Variational autoencoder for generation of antimicrobial peptides. *ACS omega*, 5(33):20746–20754, 2020.

David Eisenberg, Robert M Weiss, and Thomas C Terwilliger. The hydrophobic moment detects periodicity in protein hydrophobicity. *Proceedings of the National Academy of Sciences*, 81(1):140–144, 1984.

David Gidalevitz, Yuji Ishitsuka, Adrian S Muresan, Oleg Konovalov, Alan J Waring, Robert I Lehrer, and Ka Yee C Lee. Interaction of antimicrobial peptide protegrin with biomembranes. *Proceedings of the National Academy of Sciences*, 100(11):6302–6307, 2003.

Robert EW Hancock and Annett Rozek. Role of membranes in the activities of antimicrobial cationic peptides. *FEMS microbiology letters*, 206(2):143–149, 2002.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *International Conference on Machine Learning*, pp. 1587–1596. PMLR, 2017.

Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecule optimization. In *International Conference on Learning Representations*, 2018.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. In *International Conference on Machine Learning*, pp. 2390–2399. PMLR, 2018.

Kaveh Kavousi, Mojtaba Bagheri, Saman Behrouzi, Safar Vafadar, Fereshteh Fallah Atanaki, Bahareh Teimouri Lotfabadi, Shohreh Ariaeenejad, Abbas Shockravi, and Ali Akbar Moosavi-Movahedi. Iampe: Nmr-assisted computational prediction of antimicrobial peptides. *Journal of Chemical Information and Modeling*, 60(10):4691–4701, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *Proc. of ICLR*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. 2014.

Travis J Lawrence, Dana L Carper, Margaret K Spangler, Alyssa A Carrell, Tomás A Rush, Stephen J Minter, David J Weston, and Jessy L Labbé. ampeppy 1.0: a portable and accurate antimicrobial peptide prediction tool. *Bioinformatics*, 37(14):2058–2060, 2021.

Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pp. 707–710. Soviet Union, 1966.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems*, pp. 14866–14876, 2019.

Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. November 2015.

Daniel Schwalbe-Koda and Rafael Gómez-Bombarelli. Generative models for automatic chemical design. In *Machine Learning Meets Quantum Physics*, pp. 445–467. Springer, 2020.

Yimin Shen, Julien Maupetit, Philippe Derreumaux, and Pierre Tuffery. Improved pep-fold approach for peptide and miniprotein structure prediction. *Journal of chemical theory and computation*, 10(10):4745–4758, 2014.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*, 2019.

Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

Shaini Thomas, Shreyas Karnik, Ram Shankar Barai, Vaidyanathan K Jayaraman, and Susan Idicula-Thomas. Camp: a useful resource for research on antimicrobial peptides. *Nucleic acids research*, 38(suppl_1):D774–D780, 2010.

Marcelo DT Torres, Cibele N Pedron, Yasutomi Higashikuni, Robin M Kramer, Marlon H Cardoso, Karen GN Oshiro, Octavio L Franco, Pedro I Silva Junior, Fernanda D Silva, Vani X Oliveira Junior, et al. Structure-function-guided exploration of the antimicrobial peptide polybia-cp identifies activity determinants and generates synthetic therapeutic candidates. *Communications biology*, 1(1):1–16, 2018.

Ashley T Tucker, Sean P Leonard, Cory D DuBois, Gregory A Knauf, Ashley L Cunningham, Claus O Wilke, M Stephen Trent, and Bryan W Davies. Discovery of next-generation antimicrobials through bacterial self-screening of surface-displayed peptide libraries. *Cell*, 172(3):618–628, 2018.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6309–6318, 2017.

Colin M Van Oort, Jonathon B Ferrell, Jacob M Remington, Safwan Wshah, and Jianing Li. Ampgan v2: Machine learning-guided design of antimicrobial peptides. *Journal of Chemical Information and Modeling*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Proc. of NeurIPS*, pp. 5998–6008, 2017.

Daniel Veltri, Uday Kamath, and Amarda Shehu. Deep learning improves antimicrobial peptide recognition. *Bioinformatics*, 34(16):2740–2747, 2018.

Guangshun Wang, Xia Li, and Zhe Wang. Apd3: the antimicrobial peptide database as a tool for research and education. *Nucleic acids research*, 44(D1):D1087–D1093, 2016.

William C Wimley. Describing the mechanism of antimicrobial peptide action with the interfacial activity model. *ACS chemical biology*, 5(10):905–917, 2010.

Jacob Witten and Zack Witten. Deep learning regression model for antimicrobial peptide design. *BioRxiv*, pp. 692681, 2019.

Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. Mars: Markov molecular sampling for multi-objective drug discovery. In *International Conference on Learning Representations*, 2020.

Michael R Yeaman and Nannette Y Yount. Mechanisms of antimicrobial peptide action and resistance. *Pharmacological reviews*, 55(1):27–55, 2003.

A Appendix

A.1 Sequence Attributes

According to biological studies (Yeaman & Yount, 2003; Gidalevitz et al., 2003; Wimley, 2010), there are several physical properties crucial for the antimicrobial activity of peptides based on the mechanism. For example, amino acids with positive charges are more likely to bind with bacterial membranes as most bacterial surfaces are anionic, while those with high hydrophobicities tend to move from the solution environment to the bacterial membrane. Here, we introduce three important sequence attributes for AMPs.

Charge The bacterial membrane usually takes a negative charge. Peptides with a positive charge are more likely to bind with the membrane. The whole charge of the peptide sequence S is defined as the sum of the charge of all its residues $C(x_i)$ at pH 7.4, which is

$$C(S) = \sum_{x_i \in S} C(x_i). \tag{5}$$

We only take integer charges into consideration.

Hydrophobicity The hydrophobicity reflects the tendency to bind lipids on the bacterial membrane. A peptide with a high hydrophobicity is easy to move from the solution environment to the bacterial membrane. We use the hydrophobicity scale $H(x_i)$ in Eisenberg et al. (1984) to calculate the hydrophobicity of a sequence, which is

$$H(S) = \sum_{x_i \in S} H(x_i). \tag{6}$$

Amphipathicity / Hydrophobic Momentum The amphipathicity measures the ability of the peptide to bind water and lipid at the same time, which is a definitive feature of antimicrobial peptides(Hancock & Rozek, 2002). It can be quantified by the hydrophobic momentum $uH(S, \theta)$, defined by Eisenberg et al. (1984). The hydrophobic momentum is determined by the hydrophobicity $H(x_i)$ of each residue x_i, along with the angle θ between residues. The angle can be estimated by the secondary structure. For the α-helix structure, θ is 100° and for β-sheet, θ is 180°.

$$uH(S, \theta) = \sqrt{R^2_{cos}(S, \theta) + R^2_{sin}(S, \theta)}, \tag{7}$$

$$R_{cos}(S, \theta) = \sum_{x_i \in S} H(x_i) * cos(i * \theta), \tag{8}$$

$$R_{sin}(S, \theta) = \sum_{x_i \in S} H(x_i) * sin(i * \theta). \tag{9}$$

For each peptide, we calculate the above attributes to measure its antimicrobial activity. For comparison, we draw the distribution on the APD and decoy dataset and select a range for each attribute based on the biological mechanism (Section A.2). We use the percentage of peptides in each attribute range to exploit the generation performance and use **Combination** to measure the percentage of peptides that satisfy three conditions at the same time.

A.2 Attribute Distribution

To determine the effective threshold of charge, hydrophobicity, and hydrophobic moment of AMP, we analyze the sequence distribution in APD and decoy in Figure 4. For charge, we follow the rule summarized by experts and choose sequences whose net charge is $+2$ to $+10$. For the remaining two characters, we draw a histogram and compare the proportion in each box. If the proportion of APD is larger than that in the decoy, we add a bin to the acceptance range of the evaluation metric. The final ranges are $C \in [2, 10]$, $H \in [0.25, \infty]$, and $uH \in [0.5, 0.75] \cup [1.75, \infty]$.

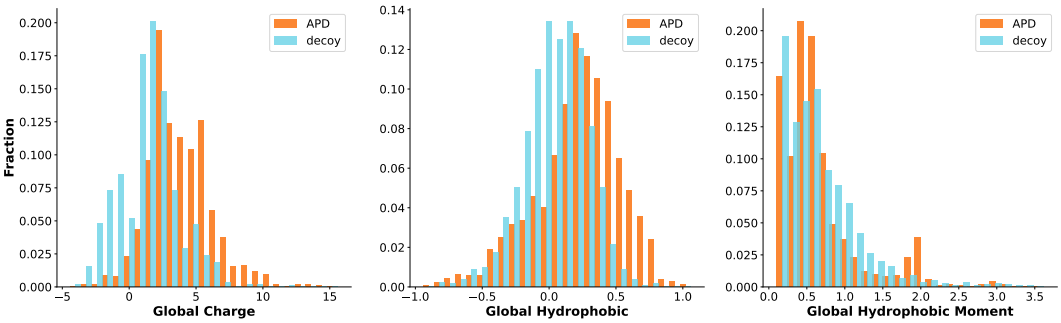


Figure 4: The histogram of charge, hydrophobicity, and hydrophobic moment on APD and decoy dataset.

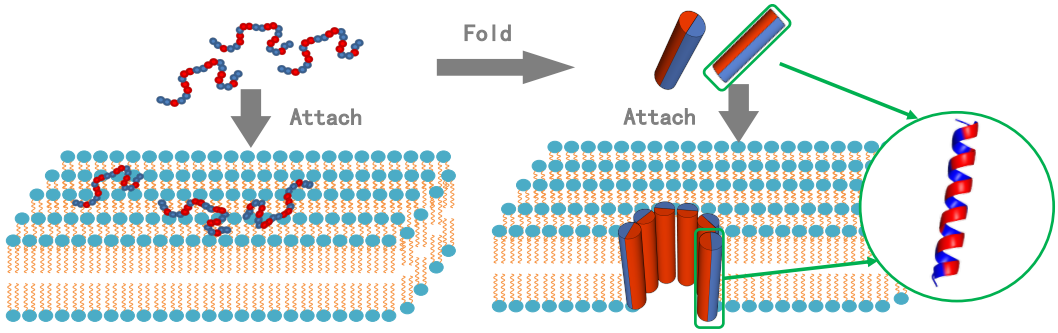


Figure 5: An example of the antimicrobial mechanism. The blue indicates the hydrophobic amino acids, and the red ones are hydrophilic. On the left, although the peptides with reasonable amino acids have attached to the bacterial membrane, they still can not insert into it. However, by folding into the helix structure, as shown on the right, the peptides maintain a stable hole that breaks the membrane of the bacterium.

A.3 Secondary Structure Filter

Similar to proteins, the biological functions of AMPs are determined by their amino acid sequences and folded structures (Boman, 2003). If the peptide can not fold into an appropriate structure, it is still difficult to take effect. For example, by forming a helical structure, the peptide can gather hydrophobic amino acids on one side and hydrophilic amino acids on the other. This amphiphilic structure helps the peptide insert into the membrane and maintain

	Uniq	C	H	uH	Combination
Random $p = 0.1$	2055	7.38% ± 11.01%	37.93% ± 0.44%	4.61% ± 0.31%	4.34% ± 0.41%
Random $p = 0.2$	1831	6.87% ± 0.31%	9.52% ± 0.31%	1.91% ± 1.06%	2.19% ± 0.66%
VAE	475	18.45% ± 2.92%	2.68% ± 3.28%	-2.78% ± 1.64%	0.29% ± 0.74%
AMP-GAN	1966	2.79% ± 0.50%	2.16% ± 0.34%	-2.29% ± 0.53%	0.17% ± 0.35%
PepCVAE	208	3.87% ± 1.58%	-1.93% ± 1.61%	1.01% ± 2.80%	3.93% ± 1.82%
MLPeptide	2106	-2.48% ± 0.39%	2.01% ± 0.57%	9.24% ± 1.22%	1.12% ± 0.38%
`LSSAMP`	4876	0.30% ± 0.37%	3.96% ± 0.64%	7.53% ± 0.41%	1.87% ± 0.07%

Table 6: The delta ratio of sequence properties filtered by secondary structures. **Uniq** is the uniq peptide number among 5000 generated sequences. **C**, **H**, **uH** correspond to charge, hydrophobicity, hydrophobic moment. **Combination** is the percentage satisfying three ranges at the same time.

a stable hole with other molecules in the membrane, as shown in Figure 5. Without it, the peptide can hardly penetrate the membrane and attach to the surface.

But does controlling secondary structure also affect sequence attributes? To answer this question, we control the secondary structure of the generated peptides to α-helix for our baseline. The performance gaps are shown in Table 6. From Table 6, we can find that most of the results are improved by limiting sequences to the α-helix structures. It shows that by controlling the structure, the sequence attributes can be improved, which verifies the importance of introducing secondary structures to the controlled generation process. However, the sequence size has decreased significantly, indicating that this generate-then-filter pipeline is inefficient.

	Uniq	**C**	**H**	**uH**	**Combination**
APD	3222	68.75%	27.96%	4.72%	6.15%
Decoy	2020	21.83%	8.81%	1.98%	0.10%
Random $p = 0.1$	4978	65.86% ± 0.19%	**26.80**% ± 0.23%	23.10% ± 0.58%	4.38% ± 0.16%
Random $p = 0.2$	5000	62.13% ± 0.39%	24.87% ± 0.29%	20.79% ± 0.76%	2.47% ± 0.17%
VAE	4988	38.00% ± 0.36%	21.07% ± 0.58%	12.43% ± 0.66%	0.34% ± 0.11%
AMP-GAN	4976	**87.66**% ± 0.45%	17.31% ± 0.74%	23.45% ± 0.73%	1.92% ± 0.05%
PepCVAE	1346	15.61% ± 0.06%	14.54% ± 0.55%	11.65% ± 0.23%	2.75% ± 0.25%
MLPeptide	4486	77.95% ± 0.72%	8.11% ± 0.27%	32.91% ± 0.60%	2.90% ± 0.16%
`LSSAMP`	4876	81.88% ± 0.31%	25.06% ± 0.45%	**37.10**% ± 0.33%	**6.26**% ± 0.07%
`LSSAMP` **w/o cond**	4903	82.04% ± 0.42%	21.32% ± 0.34%	30.51% ± 0.51%	4.46% ± 0.20%

Table 7: Physical attributes of generated sequences. We use the percentage of peptides meeting the range to measure the performance. **Uniq** is the number of unique generated sequences. **C**, **H**, **uH** correspond to charge, hydrophobicity, hydrophobic moment described in Section A.1. **Combination** is the percentage satisfying three ranges at the same time. The best results are bold.

A.4 Results of Sequence Attributes

Following the previous AMP design (Das et al., 2018; Van Oort et al., 2021; Capecchi et al., 2021; Das et al., 2021), we use the above three sequence attributes to evaluate the generation performance. As listed in Table 7, `LSSAMP` outperforms 1.88% on the combination percentage, which indicates that our model can generate sequences satisfying multiple properties at the same time. Besides, the combination percentage is similar to APD, which means that our model learns the sequence distribution of APD. `LSSAMP` tends to generate peptides with higher hydrophobicity, while AMP-GAN and MLPeptide sample more cationic sequences. Besides, `LSSAMP` can better capture the amphiphilic secondary structure indicated by the highest uH. Compared with other models, PepCVAE inefficiently generates redundant sequences, which results in a significant decrease in the number of unique sequences. Furthermore, we can find that by further controlling the secondary structure, H, uH and Combination can be improved. This verifies that secondary structure information has a great influence on sequence attributes.

Structure Condition As described above, controlling the secondary structure can affect the attributes of generated peptides. Thus we limit the percentage of the coil structure with different ratios and calculate the sequence attributes of generated peptides. The results are shown in Figure 6. We can find that with the decrease in the number of coil structures, the percentage of positive peptides keeps growing. However, for hydrophobicity and hydrophobic moment, the percentage drop after 0.3. Therefore, we limit the length of the coil structure to 30% in our main experiments.

Visualization of Residue Distribution To illustrate the distribution of residues in the generated peptides, we plot tSNE, shown in Figure 7. We transform the vector with each dimension representing the probability of a certain residue to represent the peptide. Then we use tSNE to convert the high-dimensional vector to 2D and visualize them. We find that

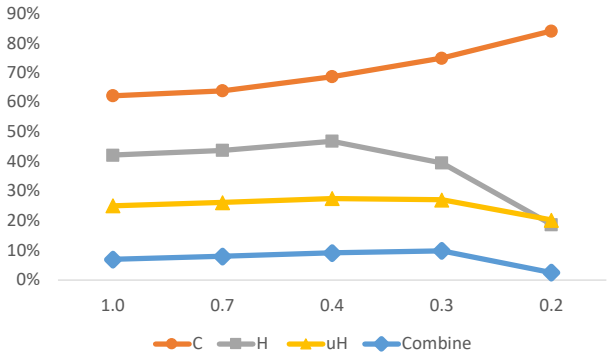


Figure 6: The physical properties of peptides with different percentages of the coil structure. The x-axis is the maximum percentage and the y-axis is the percentage of peptides that meet the property range.

there is a large overlap between LSSAMP w/o condition and APD, which indicates that our model has captured the global distribution of APD instead of collapsing to a local mode. Furthermore, LSSAMP covers APD and has some outliers. The results show that with the secondary structure condition, our model can not only learn the existing AMP distribution but also explore more possible spaces.

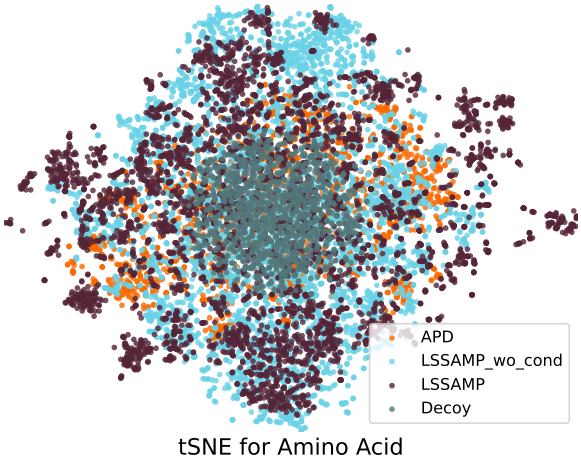


Figure 7: The tSNE plot for the distribution of residue in each sequence on four datasets.

Visualization of LSSAMP Distribution We plot the distribution of residues, charge, sequence length, hydrophobicity, and hydrophobic momentum for APD, Decoy, and our models in Figure 8. Without condition, the distribution of LSSAMP is similar to APD, which indicates that LSSAMP successfully learns the sequence distribution of AMP. However, if we control the secondary structure, it is more likely to generate sequences with longer lengths and more positive charges. For hydrophobicity and hydrophobic momentum, the distribution of the generated sequences is more concentrated.

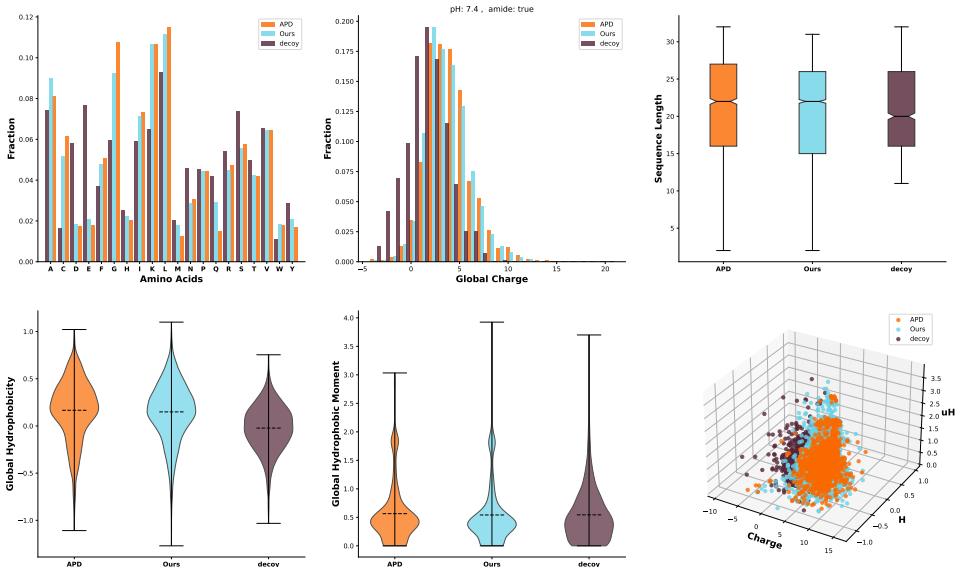


Figure 8: The distribution of residues, charge, sequence length, hydrophobicity, hydrophobic momentum, and a 3D visualization for three sequence attributes.

B More Experimental Results

B.1 Case Study

We show 10 peptides generated by `LSSAMP` with the sequence, the secondary structure and attributes discussed above. The peptides all have a long alpha-helix, which makes them more possible to be AMP.

ID	Sequence	Secondary Structure	C	H	uH
Y1	FLPLVRVWAKLI	–HHHHHHHHHH	2	0.471	0.723
Y2	FLSTVPYVAFKVVPTLFCPIAKTC	–HHHHHHHHHHHHHHHHHHHHT–	2	0.446	1.812
Y3	FFGVLARGIKSVVKHVMGLLMG	–HHHHHHHHHHHHHHHHHH–	3	0.420	0.549
Y4	GVLPAFKQYLPGIMKIIVKF	–HHHHHHHHHHHHHHH—	3	0.419	0.523
Y5	VFTLLGAIIHHLGNFVKRFSHVF	-HHHHHHHHHHHHHHHHHHHH–	2	0.416	0.514
Y6	FVPGLIKAAVGIGYTIFCKISKACYQ	–HHHHHHHHHHHHHHHHHHHHT—-	3	0.394	1.815
Y7	ALWCQMLTGIGKLAGKA	–HHHHHHHHHHHHHH	2	0.344	0.506
Y8	LLTRIIVGAISAVTSLIKKS	–HHHHHHHHHHHHHHH–	3	0.334	0.531
Y9	FLSVIKGVWAASLPKQFCAVTAKC	–HHHHHHHHHHHHHHHHHHHT–	3	0.334	0.660
Y10	FLNPIIKIATQILVTAIKCFLKKC	–HHHHHHHHHHHHHHHHHHHT–	4	0.334	1.940

Table 8: Ten generated peptides and their physical properties and predicted structures. ‘H’ is the α-helix, ‘T’ is the Turn and ‘-’ is the coil.

B.2 Novelty

To measure the novelty of the generated peptides, we define three evaluation metrics: Uniqueness, Diversity, and Similarity. **Uniqueness** is the percentage of unique peptides in the generation phase. **Diversity** measures the similarity among the generated peptides. We calculate the Levenshtein distance (Levenshtein et al., 1966) between every two sequences and normalize it by the sequence length. Then we average the normalized distance to get the mean as its diversity. The higher the diversity, the more dissimilar the generated peptides are. **Novelty** is the difference between the generated peptides and the training AMP set. For each generated sequence, we search the training set for a peptide which has the smallest

	Uniqueness ↑	Diversity ↑	Novelty ↑
Random $p = 0.1$	0.995 ± 0.000	0.871 ± 0.021	0.078 ± 0.001
Random $p = 0.2$	**0.999** ± 0.000	0.971 ± 0.022	0.160 ± 0.001
VAE	0.986 ± 0.001	**1.011** ± 0.038	**0.584** ± 0.002
AMP-GAN	0.995 ± 0.001	0.907 ± 0.023	0.565 ± 0.007
PepCVAE	0.265 ± 0.006	0.367 ± 0.007	0.423 ± 0.005
MLPeptide	0.900 ± 0.003	0.850 ± 0.016	0.416 ± 0.010
`LSSAMP`	0.981 ± 0.001	0.878 ± 0.018	0.503 ± 0.005
`LSSAMP` **w/o cond**	0.976 ± 0.002	0.901 ± 0.013	0.515 ± 0.008

Table 9: The novelty of the sampling. ↑ means higher is better. The detailed descriptions are in Section B.2.

Algorithm 1 Training and Sampling phase of `LSSAMP`

Require: A protein dataset D_r, a peptide dataset with secondary structure D_s, and the AMP dataset D_{amp}. The model M_θ with N codebooks. A set of N prior models M_{prior_n}.

```
Train on D_r and update M_θ via Eqn. 3.
Train on D_s and update the M_θ via Eqn. 4.
Finetune M_θ on D_amp via Eqn. 4.
for each codebook n = 1, 2, ··· , N do
    Create an empty dataset C_n.
    for x_i ∈ D_amp do
        Save the n-th codebook index of x_i via Eqn. 2 to C_n
    end for
    Train an auto-regressive language model M_prior_n on C_n.
end for
```

Levenshtein distance from it and normalize the distance according to its length. We calculate the average length as the Novelty.

From Table 9, we can see that VAE has the highest diversity and novelty. However, from Table 2, we can find that the peptides generated by VAE do not have a high probability of AMP. It means that the vanilla VAE trained on AMP datasets without attribute control can hardly capture the antimicrobial features. It randomly samples in the latent space. At the same time, `LSSAMP` has a significant advantage over the above strong baseline PepCVAE and MLPeptide. It means that our model can generate promising AMPs with relatively high novelty. Besides, the limitation of secondary structure will lead to a decline in diversity. However, it does not result in more redundant peptides because the uniqueness does not decrease. It indicates that the restrictions make the model capture similar local patterns, but not generate the exact same sequence.

C Reproduction

We run the model several times and calculate the mean and variance of the main experimental results and analysis. The algorithm can be descibed as Alg. 1. Following Kaiser et al. (2018), we use Exponential Moving Average (EMA) to update the embedding vectors in the codebooks. Specifically, we keep a count c_k measuring the number of times that the embedding vector e_k is chosen as the nearest neighbor of $z_e(x_i)$ via Eqn. 2. Thus, the counts are updated with a sort of momentum: $c_k \leftarrow \lambda c_k + (1-\lambda)\sum_i \mathbb{I}[z_q(x_i) = e_k]$, with the embedding e_k being updated as: $e_k \leftarrow \lambda e_k + (1-\lambda)\sum_i \frac{\mathbb{I}[z_q(x_i)=e_k] z_e(x_i)}{c_k}$. Here, λ is the decay parameter.

C.1 Model Implementation

There are three main modules for `LSSAMP`. The encoder and decoder are based on a 2-layer Transformer (Vaswani et al., 2017) with $d_{model} = 128$, $head = 8$. The size of the FFN projection is $d_{ffn} = 512$ and all dropout rates are 0.1. For the classifier, we use the same CNN block as Billings et al. (2019) with 32 input channels and a dilation scale of $[1, 2, 4, 8, 10]$.

	PPL ↓	Loss ↓	AA Acc.↑	SS Acc.↑
LSSAMP	**3.24** ± 0.16	**1.17** ± 0.05	99.79 ± 0.20	**87.20** ± 0.62
w/o D_r	11.56 ± 3.81	2.45 ± 0.94	66.06± 0.67	82.78 ± 0.57
w/o D_s	3.83 ± 0.32	1.34 ± 0.04	99.58± 0.26	85.87± 0.35
w/o subbook	3.49 ± 0.20	1.25 ± 0.05	99.86 ± 0.36	86.61 ± 0.95

Table 10: Ablation Study on validation set of D_{amp}. 'w/o' indicates that we remove the module from LSSAMP. ↑ means higher is better, and ↓ is the opposite.

Codebook	PPL ↓	Loss ↓	AA Acc.↑	SS Acc.↑
[1]	19.04 ± 2.84	2.94 ± 0.14	65.49 ± 3.49	83.41 ± 2.34
[1, 2]	3.84 ± 0.09	1.35 ± 0.02	99.40 ± 0.45	85.39 ± 0.26
[1, 2, 4]	3.32 ± 0.03	1.20 ± 0.01	**100.00** ± 0.00	85.95 ± 0.42
[1, 2, 4, 8]	**3.24** ± 0.16	**1.17** ± 0.05	99.79 ± 0.20	**87.20** ± 0.62

Table 11: The influence of the number of codebooks. '[1,2,4,8]' indicates that we use 4 codebooks with window sizes of 1,2,4,8. The meanings of symbols are the same as Table 4.

For multi-scale codebooks, we first apply CNN as $F^{(n)}$ to extract features. We set $n = 4$ and kernel width ranging in $[1, 2, 4, 8]$. The features will be padded to the same length as the input sequence. Then, we use 4 codebooks with $K = 8$ and $d = 128$. The reconstruction and prediction share the same codebooks, which means $N_r = N_s = 4$. The commit coefficient is set to $\beta = 0.05$.

We use PyTorch to implement our model and train it on a single Tesla-V100-32GB. We optimize the parameter with Adam Optimizer (Kingma & Ba, 2015). During pre-training for sequence construction on D_r, we set the maximum token in a batch bz as 30,000, learning rate lr as 0.01 with 8,000 warm-up steps, and decoy weight for EMA as $\lambda = 0.8$. For secondary structure prediction on D_s, the max length is limited to 32, $bz = 10,000$, $lr = 0.003$, $\lambda = 0.95$, and the prediction loss coefficient $\gamma = 1$. Finally, we transfer to D_{amp} with the same hyperparameters except the $lr = 0.001$.

C.2 Wet Experiment Implementation

A colony of bacteria was grown in LB (Lysogeny broth) medium overnight at 37 degrees. A peptide concentration range of 0.25 to 128 mg/liter was used for MIC assay. The concentration of bacteria was quantified by measuring the absorbance at 600 nm and diluted to OD600 = 0.022 in MH medium. The sample solutions(150uL) were mixed with a 4uL diluted bacterial suspension and finally inoculated with about 5 * 10E5 CFU. The Plates were incubated at 37 degrees until satisfactory growth 18h. For each test, two columns of plates were reserved for sterile control (broth only) and growth control (broth with bacterial inoculum, no antibiotics). The MIC was defined as the lowest concentration of the peptide dendrimer that inhibited the growth of bacteria visible after treatment with MTT.

C.3 Full Ablation results

Table 10 and 11 the full results with error bars for Table 4 and 5 in Section 3.4.