Deep Generative Models for Therapeutic Peptide Discovery: A Comprehensive Review

LESHAN LAI, College of Information Science and Engineering, Hunan University, Changsha, China
YUANSHENG LIU, College of Information Science and Engineering, Hunan University, Changsha, China
BOSHENG SONG, College of Information Science and Engineering, Hunan University, Changsha, China
KEQIN LI, Department of Computer Science, State University of New York, New York, United States
XIANGXIANG ZENG, College of Information Science and Engineering, Hunan University, Changsha, China

Deep learning tools, especially deep generative models (DGMs), provide opportunities to accelerate and simplify the design of drugs. As drug candidates, peptides are superior to other biomolecules because they combine potency, selectivity, and low toxicity. This review examines the fundamental aspects of current DGMs for designing therapeutic peptide sequences. First, relevant databases in this field are introduced. Next, the current situation of data representation and where it can be optimized are discussed. Then, after introducing the basic principles and variants of diverse DGM algorithms, the applications of these methods to design and optimize peptides are stated. Finally, we present several challenges to devising a powerful model that can meet the requirements of learning the different biological properties of peptides, as well as future research directions to address these challenges.

CCS Concepts: • **Applied computing** → **Bioinformatics**; • **Computing methodologies** → **Artificial intelligence;**

Additional Key Words and Phrases: Bioinformatics, deep generative model, deep learning, peptides design

ACM Reference Format:
Leshan Lai, Yuansheng Liu, Bosheng Song, Keqin Li, and Xiangxiang Zeng. 2025. Deep Generative Models for Therapeutic Peptide Discovery: A Comprehensive Review. *ACM Comput. Surv.* 57, 6, Article 155 (February 2025), 29 pages. https://doi.org/10.1145/3714455

1 Introduction

Peptides can be used as therapeutic and diagnostic agents in biotechnology applications [76] owing to numerous characteristics, including high specificity, high selectivity, safety, accessibility, and

This work was supported by the National Natural Science Foundation of China (62372159, 62425204, 62122025, U22A2037, 62450002, 62432011, 62272151), and the Science and Technology Innovation Program of Hunan Province (2022RC1099, 2022RC1100).

Authors' Contact Information: Leshan Lai, College of Information Science and Engineering, Hunan University, Changsha, Hunan, China; e-mail: allysa_lai@hnu.edu.cn; Yuansheng Liu (Corresponding author), College of Information Science and Engineering, Hunan University, Changsha, Hunan, China; e-mail: yuanshengliu@hnu.edu.cn; Bosheng Song (Corresponding author), College of Information Science and Engineering, Hunan University, Changsha, Hunan, China; e-mail: boshengsong@hnu.edu.cn; Keqin Li, Department of Computer Science, State University of New York, New York, Washington, USA; e-mail: lik@newpaltz.edu; Xiangxiang Zeng, College of Information Science and Engineering, Hunan University, Changsha, Hunan, China; e-mail: xzeng@hnu.edu.cn.

ACM 0360-0300/2025/02-ART155
https://doi.org/10.1145/3714455

less immunogenicity. Previous drug design research has focused on small molecules or proteins [52, 90, 98]. The Global Peptide Therapeutics Market & Clinical Trials Insight 2028 Report revealed that more than 200 approved peptide drugs are currently used to treat patients with various diseases. Furthermore, 800 peptide therapeutics are in clinical trials, and the market opportunity for peptide drugs is predicted to exceed USD 75 billion by 2028 [1], indicating that the design of therapeutic peptide is of high interest and promising. However, traditional experimental approaches to new therapeutic peptide design are time-consuming and costly. On average, more than 10 years and two to three billion dollars are needed before a newly designed drug enters the market, with a success rate of less than 1% [32]. To overcome this, peptide therapeutic designs are usually conducted with computation-aided methods [76]. The world's leading pharmaceutical companies have begun to improve their research using artificial intelligence, which has show great potential in many areas of healthcare [100]. In 2020, the sudden outbreak of Covid-19 further accelerated the integration of biology and artificial intelligence [101]. Improvements in computing power and algorithms, as well as the accumulation of large amounts of data, have optimized the research conditions of computational biology [70].

Machine learning is a data-driven approach that is increasingly applied in the field of bioinformatics [78, 163]. It accelerates drug screening and reduces costs. Algorithms such as support vector machines [78], random forests [89], and Bayesian networks [5, 132] have been applied to identify and generate peptides and to predict their properties [6]. The effectiveness of machine learning in peptide research demonstrates the potential of data-driven approaches. However, conventional machine learning techniques are limited at processing data. A machine learning system requires considerable domain expertise to design a feature extractor for suitable representation [75]. Among the many types of machine learning models, deep learning with artificial neural networks can be used to learn data representations automatically, which averts the strenuous task of feature selection. Deep learning models are apt for discovering intricate structures in high-dimensional data, and they outperform machine learning models at many problems given a sufficient set of data [91].

Deep learning-based peptide drug discovery includes prediction, classification, and generation tasks. Much of the focus of past works was on the prediction task [91], with relatively little attention on the efficiency of screening for therapeutic peptides. For example, Lei et al. [79] designed a deep learning model for protein–peptide interaction prediction, successfully capturing the binary interactions between peptides and proteins and identifying the binding residues of the peptides involved. Nevertheless, prediction tasks, which learn a mapping from input to label, can only identify or predict some properties and cannot generate novel peptides. Unlike prediction tasks, generative tasks learn the underlying data distribution and develop de novo peptide design. Current generative models are effective at learning and generating novel data of various sorts, such as the generation of images [110], text [11], music [35], and molecules [119]. A generative model has also shown preliminary effects on peptide design [13, 23] (see the section "Architecture Division and Evaluation Techniques for Deep Generative Models in Peptide Design" for a detailed description of the generative model). Furthermore, improved variants and up-and-coming mechanisms or models offer possible avenues for future research. Therefore, it is important to combine the basic process of peptide generation with future developments.

In this review, peptides applied in generative models are classified into three groups [74]. Group I includes peptides that interfere with a molecule or organism, generally **cationic host defense peptides (CHDPs)** [95] that resolve harmful inflammation, such as **antimicrobial peptides (AMPs)** and **antiviral peptides (AVPs)**. AMPs act as non-specific antibacterial agents by directly destroying bacterial membranes, which reduces the evolutionary probability of bacterial resistance [85, 95, 133]. Group II consists of peptides that form functional polymers with proteins, such as **signal peptides (SPs)** and **cell-penetrating peptides (CPPs)**. SPs are short peptide sequences that

direct newly synthesized proteins to various export pathways and are designed for therapeutic purposes, such as increasing the therapeutic levels of proteins secreted from hosts [166]. CPPs enhance the intracellular delivery of biomolecules such as proteins [138]. Group III includes antigenic peptides that can be candidates for vaccines, such as **human leukocyte antigen (HLA)**-binding peptides. HLA is an antigen-presenting protein that binds to peptides with strong HLA-binding affinity to form **peptide–HLA complexes (pHLA)**, which trigger an immune response similar to vaccine action. Compared with proteins, peptides speed up vaccine development and reduce costs [57].

Different extension tasks based on generative models have been attempted in an effort to design the ideal peptide. These studies often integrate generative models with classifiers [150], impose property constraints on the generated peptides by incorporating conditions to achieve more optimal outcomes [41, 83, 108], or utilize structural evaluation tools [12, 15]. Additionally, advancements include efforts in all-atom generation [81] and multi-modality approaches [155]. In what follows, we first introduce common databases of popular peptides around these tasks and some related requirements. Next, data representation methods for peptide sequences are outlined, and we describe where they can be optimized. Then, we provide a comprehensive description of the basic principles and application notes of generative models, and analyze the structures or algorithmic mechanisms of variant models that enable the optimization of generative tasks. Finally, we discuss challenges and potential directions for future research on generative models for therapeutic peptide design. Figure 1 illustrates a workflow of peptide design using a **deep generative model (DGM)**.

2 Peptide Databases: Critical Resources for Generative Modeling in Therapeutic Peptide Design

Over the last few decades, a large number of databases containing specific functional peptides have been developed [107, 140, 141, 146]. Table 1 summarizes and categorizes the common databases of various therapeutic peptides that are currently of interest. **Universal Protein (UniProt)** [22] is an extensive database that contains information on labeled and unlabeled protein/peptide sequences and their functions. These comprehensive data can be utilized in pretraining to enhance the understanding of the general grammar of peptides. Another general database, THPdb, contains complete information on US-FDA-approved protein and peptide therapeutics, such as their half-life, chemical modifications, immunogenicity, solubility, and toxicity— properties that most candidates should test [141]. The Database of Antimicrobial Activity and Structure of Peptides is a large, comprehensive repository of experimental data from in vitro tests assessing the antimicrobial/cytotoxic activities of peptides, to facilitate the de novo design of AMPs with desired properties [107]. The Immune Epitope Database is one of the largest repositories of immunological epitopes and includes published experimental data on infectious diseases and allergens [145].

Data need to be accurate, meaningful, and representative to maximize the characteristics learned in model construction, so the data collection step is crucial. This process in the design of therapeutic peptides commonly involves filtering for various attributes, including those that satisfy a specific therapeutic function and those that meet the primary states required for therapeutic drugs (Figure 1(a)), such as toxicity, hydrophobicity, secondary structures, and chemical modifications. These are conducive to druggability [133], due to some disadvantages of peptide therapeutics, such as low solubility and instability. Collecting or building different and more comprehensive databases is vital because models with extensive and diverse training data can produce more attributes and novelty than models trained with one data type. For example, the AMP generation frame CLaSS [23], which can controllably generate AMPs with desired properties in ample peptide space, is pre-trained with unlabeled data from UniProt and combined with a classifier that is trained with labeled data of toxic/hemolytic and structural properties from several specific databases.

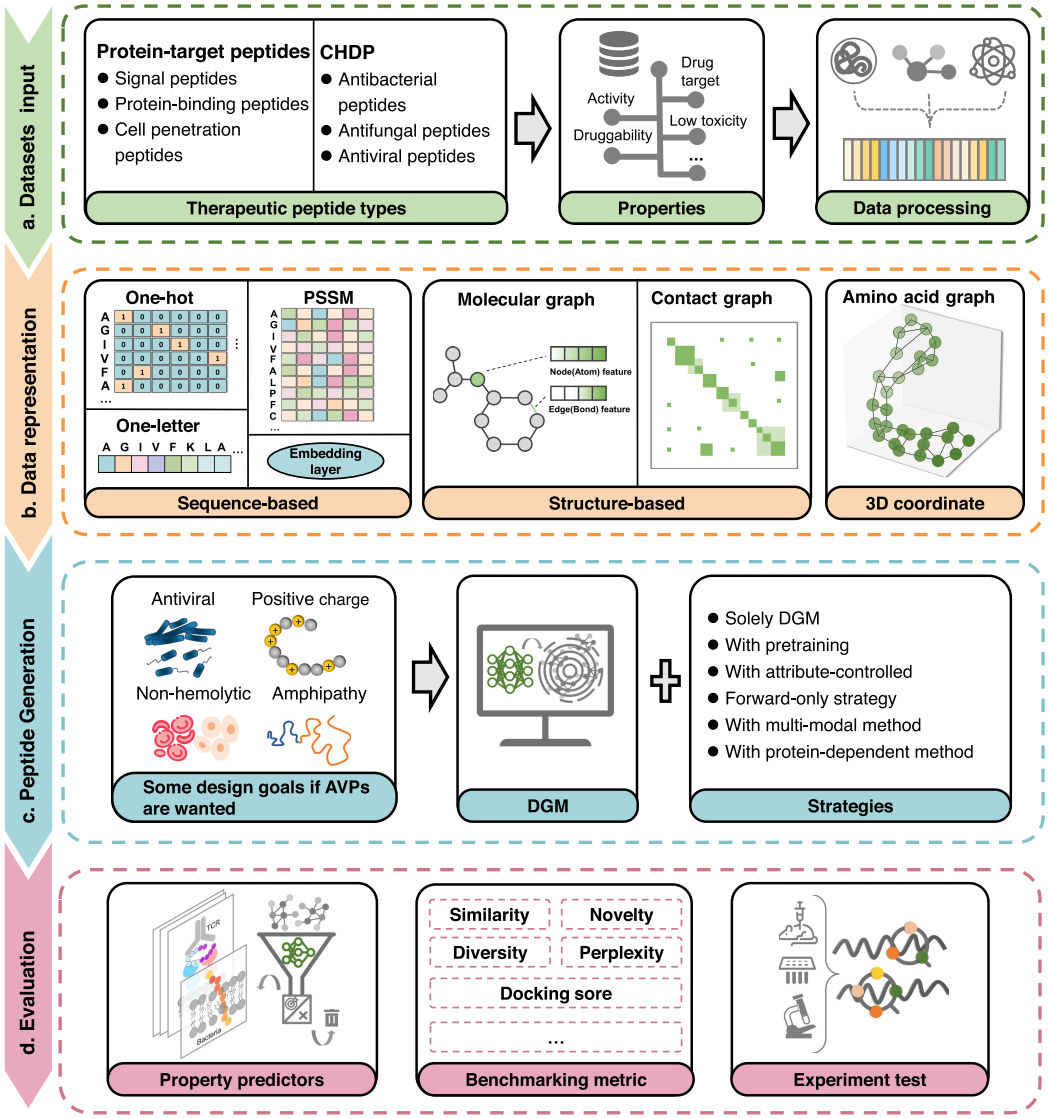

Fig. 1. Workflow for the generation of a specific peptide using DGM. (a) Dataset input: considering appropriate therapeutic peptide properties and processing the data. (b) Data representation: including sequence-based, structure-based, and 3D coordinate representations commonly used in deep learning. (c) Peptide generation: designed according to specific peptide goals and combined with optimization strategies. (d) Evaluation: using property predictors, existing or custom metrics, or wet lab experiments.

One limitation of these peptide databases is the lack of function-specific negative data crucial for classifiers, especially data derived from experimental validation. This is partly due to the ambiguous definition of negative data. Most studies retrieve random peptide segments from UniProt, such as AMPEP [10], which filters negative AMP samples using UniProt's "NOT" keyword search function. Additionally, more advanced models can handle complex data beyond amino acid sequences, such as binding pockets and 3D coordinate information. Therefore, to generate therapeutic peptides with multiple attributes, the database needs to have rich data structures and

Table 1. Comprehensive and Specific Databases Available for Peptide

Type	Specific	Database	Data size	Link	Year
Comprehensive	All kinds of peptide and protein data	PepBank [123]	21,691	http://pepbank.mgh.harvard.edu/	2007
		SatPDB [124]	37,100	https://bit.ly/3W020dJ	2016
		UniProt [7]	>227M	https://www.uniprot.org/	2023
	Peptide-protein complexes	PepBDB[154]	13,299	https://bit.ly/4bywVDz	2019
	Peptide drugs	THPdb [141]	852	http://crdd.osdd.net/raghava/thpdb	2017
		THPdb2 [66]	894	https://bit.ly/3WdbO5s	2024
CHDP that are inti-inflammatory	AMPs/ Non-AMPs	DADP [102]	2,571	http://split4.pmfst.hr/dadp/	2012
		YADAMP [106]	2,133	http://www.yadamp.unisa.it/	2012
		APD3 [149]	2,619	https://aps.unmc.edu/AP/	2016
		AMPEP [10]	170,059	http://crdd.osdd.net/raghava/AmPEP/	2018
		LAMP [159]	23,253	https://bit.ly/3LhZr1G	2020
		DBAASP [107]	35,700	http://dbaasp.org/	2021
		dbAMP [67]	26,447	https://awi.cuhk.edu.cn/dbAMP/	2022
		DRAMP [121]	20,407	http://dramp.cpu-bioinfor.org/	2022
		CAMP [44]	24,243	http://crdd.osdd.net/raghava/camp	2023
	AVPs	AVPpred [136]	1,245	http://crdd.osdd.net/servers/avppred	2012
		AVPdb [109]	2,683	http://crdd.osdd.net/servers/avpdb	2014
	ACPs	CancerPPD [140]	3,491	http://crdd.osdd.net/raghava/cancerppd/	2015
Cell penetrating	CPPs	CPPsite2.0 [2]	1,850	http://crdd.osdd.net/raghava/cppsite/	2016
immunogenic/ antigenic	HLA-binding peptides	SYFPEITHI [113]	2,000	http://www.syfpeithi.de/0-Home.htm	1999
		EPIMHC [114]	4,867	https://bit.ly/3WhJq29	2005
		MHCBN [73]	25,857	http://crdd.osdd.net/raghava/mhcbn/	2009
		IEDB [144]	>1.6M	http://www.iedb.org/	2018

Abbreviations: ACPs, anti-cancer peptides. AMPs, antimicrobial peptides; AVPs, antiviral peptides; CHDP, cationic host defense peptides; CPPs, cell-penetrating peptides; HLA, human leukocyte antigen; M, million.

attributes, efficient search capabilities, and regular updates. Finally, a benchmark dataset should be established for specific peptides to evaluate generative models more universally and conveniently.

3 Peptide Data Representation for Generative Modeling

How the data are represented is essential to extracting useful information from them, whether in conventional machine learning or deep learning [9, 29]. A representation is an extract of original data into an abstract, high-level, and usually low-dimensional feature space. Most conventional machine learning predictive models encode the features of peptides through descriptors [153] such as the amino acid composition, which computes the occurrence frequency of each amino acid type in a peptide sequence or composition–transition–distribution that can describe the global composition of the amino acid property for each sequence [82]. These representations are often unsuitable for generative models because of they lack complete sequence information. There are three main types of data representation for DGMs in peptide design: sequence-based, structure-based, and 3D coordinate representations (Figure 1(b)). Below, we introduce the commonly used representations in each category.

Among sequence-based representations, the simplest, most commonly used representation is one-hot encoding based on binary vectors, where 1 indicates the presence of a character, and 0

indicates an empty slot. Another sequence-based coding method is the embedding layer. The embedding layer is a learnable neural network layer, typically used as the first layer in a model. It essentially performs a multiplication of a matrix with the original vector (one-hot or one-letter encoding). The advantage of learnable embeddings is their adaptability to the model, as the embedding layer is trained simultaneously with the entire model. In addition, the evolutionary-based amino acid coding obtained through sequence alignment tools can capture evolutionarily conserved features of a sequence, and this naturally serves as a matrix representation that provides a richer biological meaning. One evolutionary-based coding method considered effective at improving the performance of deep learning models is the position-specific scoring matrix representation [19, 79, 115, 152], a multiple sequence alignment generated by PSI-BLAST [3] that records the probability that each amino acid at a different position in the macromolecule sequence is transformed into another amino acid.

However, a sequence-based representation limits the model to learning only sequence information. By contrast, some biological functions are contained in the structure of peptides, such as the α-helix, which has a high probability of occurring in AMPs. Structure-based representations provide richer information for models to learn. Regarding structure-based representations, the much shorter sequences in peptides than in proteins make a molecular graph representation feasible, where the atoms are the nodes and the bonds are the edges of a molecular graph. The RDKit tool can convert an amino acid sequence into a molecular graph format. Molecular graph representations can be efficiently learned and used for downstream tasks by graph neural networks with a range of strategies [158, 162, 167]. Another structure-based representation is the contact map, which shows the interaction of residue pairs as a matrix, where the nodes are individual residues. The edges can be determined by a physical distance threshold [63]. It is an efficient representation for describing the spatial structure of macromolecules [68, 94, 105]. In the last category, 3D coordinate representations of peptides supplement the 2D connectivity and topology with spatial geometric information, such as bond lengths and bond angles at the molecular level [64], or relative spatial positions at the amino acid level [64], which more closely reproduces real-world scenarios. Table 2 summarizes different categories of generative models and their corresponding databases and data representation methods used in peptide design, aiding in understanding and selecting appropriate representation techniques.

Typically, CHDPs like AMPs are generated using one-hot encoding or other sequence-based representations. However, linear representations of amino acid sequences are often superficial and limited, though they can be supplemented by structure-based representations. Recently, when generating protein-binding peptides, structure-based representations are increasingly used to capture evolutionary information and binding key atoms or coordinate structures, such as the BLOSUM matrix or 3D coordinates. In the future, more encoding methods will be integrated to better represent data. Due to the significant computational resources required for 3D coordinate representations and the large amount of information to be processed, a comprehensive encoding method similar to SMILES or SELFIES, or a molecular graph, can be constructed to create so-called peptide fingerprints. To accelerate the encoding process, this method can be compiled into packages for easy and rapid invocation.

4 Architecture Division and Evaluation Techniques for Deep Generative Models in Peptide Design

The choice of model is the key to the quality of the generated sequence. In this section, to help readers better understand DGM-assisted drug discovery, the basic principles and optimization strategy of the DGM are briefly outlined. The generative models introduced here are divided into two categories. One category comprises those oriented to language modeling, usually accompanied by

Table 2. Databases and Data Representation Methods for Different Categories of Generative Models in Peptide Design

Category	Model	Database	Representation	Ref.
RNN	LSTM-RNN	ADAM, APD, DADP	One-hot	[97]
	LSTM_Pep	PeptideAtlas/APD3	One-hot	[164]
	PepPPO	IEDB	One-hot, BLOSUM matrix, Embedding layer	[17]
VAE	CLaSS	UniProt/satPDB, DBAASP, AMPEP	Embedding layer	[23]
	PepVAE	APD, DADP, DBAASP, DRAMP, YADAMP	One-hot	[26]
	HydrAMP	dbAMP, AMP Scanner8, DRAMP6	One-hot	[131]
GAN	GANDALF	THPdb	Text, 4D tensor	[116]
	PepGAN	APD, CAMP, LAMP, DBAASP	None	[139]
	DeepImmuno	IEDB	AAindex PCA	[80]
	AMPGAN v2	DBAASP, AVPdb, UniProt	One-hot	[142]
	Pandoragan	AVPDb, AVPpred, CAMP, Dramp, APD3, dbAMP	Embedding layer	[129]
SA-based	Transformer	Swiss-Prot	One-hot	[156]
	TransMut	IEDB, EPIMHC, MHCBN, SYFPEITHI	Embedding layer	[20]
Diffusion model	HelixDiff	PDB	One-hot, 3D coordinate	[157]
	AMP-Diffusion	dbAMP, AMP Scanner, DRAMP	Embedding layer	[16]
	MMCD	APD3, CAMP, DBAMP, DRAMP, SATPdb, YADAMP, LAMP, CancerPPD	One-hot, 3D coordinate	[151]
	HYDRA	PepBDB	One-hot, 3D coordinate	[112]

specific generative tasks. These models are RNN-, attention-, or self-attention-based [143]. The other category consists of models oriented toward fitting the real data distribution, including **generative adversarial networks (GANs)** [46] and the **variational autoencoder (VAE)** [71], which usually use an RNN [117] as the generator. Figure 2 depicts the basic architecture of generative models.

4.1 Data Representation Learning Architecture

4.1.1 Recurrent Neural Network. An RNN has a core recurrent layer that allows information to flow across time steps. This recurrent layer shares parameters at each time step, storing and calculating previous states to process sequential input. During training, the network predicts the next token in a sequence and computes loss by comparing predictions with actual data. RNNs excel at capturing sequential patterns, but face challenges like vanishing or exploding gradients [60] and difficulty learning long-term dependencies [72]. To mitigate these issues, various recurrent unit variants have been proposed. The **long short-term memory (LSTM)** [60] unit efficiently retains information over extended periods through specialized gating mechanisms. Additionally, bidirectional architectures leverage past and future information to enhance context understanding [47]. Another variant, the **gated recurrent unit (GRU)** [18], offers a simpler alternative with fewer parameters than LSTM [21]. Despite advancements, training RNNs typically relies on labeled data, which may be sparse and expensive to obtain at scale, although RNNs can also be applied to unsupervised and self-supervised learning tasks. Additionally, simple RNNs tend to capture only rudimentary patterns within the data, resulting in a lack of diversity in generated sequences. Figure 2(a) illustrates the process of generating peptide sequences with an RNN and a simplified diagram of the variants of the recurrent unit.

4.1.2 Transfomer-based Models. RNNs process inputs sequentially, limiting the use of parallel computing hardware. To address this, Vaswani et al. introduced the transformer model [143], based entirely on the self-attention mechanism. Compared to RNNs, transformers allow for better parallelization and facilitate the modeling of long-term texts. The core of the transformer,

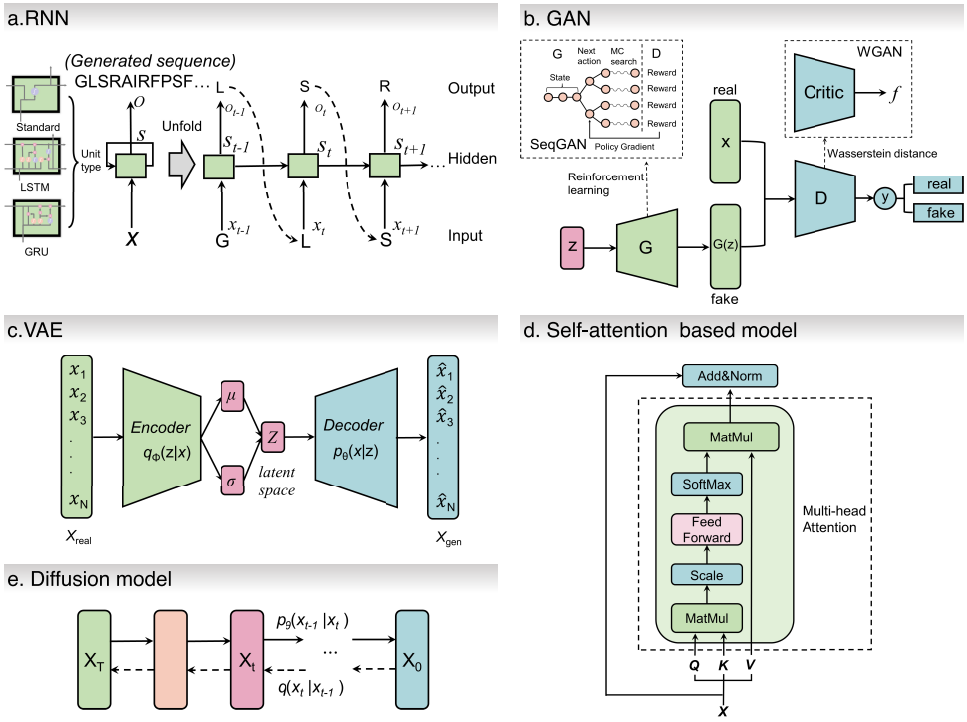

Fig. 2. Basic architecture of generative models. (a) RNN unfolded in time for sequence-based peptide generation and the core of three recurrent unit variants. (b) Schematic diagram of the basic architecture of GAN. GAN based on RL can generate sequences and utilize the Wasserstein distance to improve loss calculation. (c) Diagram of VAE, which trains the encoder and the decoder on the distributions that generate latent space z and reconstruct input. (d) Sketch of self-attention layer. (e) The diffusion model, gradually adds Gaussian noise from X_0 and then reverses.

self-attention, captures interdependencies among sequence elements to compute sequence representations. It computes query (Q), key (K), and value (V) vectors for each word, calculates attention scores $\alpha_{i,j}$, and outputs a weighted sum using SoftMax (Figure 2(d)). The architecture consists of self-attention and position-wise fully connected feed-forward network layers for the encoder and decoder. The decoder's first multi-head self-attention employs a mask operation to prevent data leakage. Additionally, the decoder performs multi-head self-attention over each output word of the encoder, providing a different perspective on the input sequence. Position encoding is added to word embedding to maintain word order.

Two essential variants of the transformer are **generative pretraining (GPT)** [111] and **bidirectional encoder representations from transformers (BERT)** [30]. GPT is an autoregressive language model that uses the transformer's decoder structure to predict the next token based on the previous one. BERT is an autoencoder language model that predicts masked tokens based on unmasked tokens (bidirectional context).

4.2 Real Data Distribution Learning Architecture

4.2.1 Variational Autoencoder. The variational autoencoder (VAE) is a probabilistic graphical model commonly used for unsupervised learning [71]. They learn a latent representation z of input

data x to generate new data points by sampling from this latent space (Figure 2(c)). VAEs approximate the true posterior distribution of latent variables via variational inference, optimizing the **evidence lower bound (ELBO)**. The ELBO comprises a reconstruction loss, measuring disparity between input and reconstructed data, and a KL divergence term, regularizing the latent space towards a prior distribution. The reparameterization trick enables efficient training by decoupling stochasticity from model parameters. By learning meaningful latent features, VAEs can generate data that resembles but are not identical to the input distribution. Consequently, VAEs exploit unlabeled data more efficiently and generate more diverse outputs than RNNs. The encoder in a VAE can be an RNN, CNN, or any model learning latent representations, while the decoder is typically an RNN for data generation. However, VAEs suffer from entangled latent attributes, limiting independent attribute generation. To address this, Hu et al. [62] proposed a semi-supervised approach that combines a VAE with the wake–sleep procedure [58], disentangling latent codes, enabling controlled generation, and achieving semi-supervised learning using labeled data. The properties of latent vectors facilitate the addition of conditional variables. Sohn et al. [126] proposed the **conditional variational autoencoder (CVAE)**, which concatenates the encoded latent vector with data labels before inputting them into the decoder. This approach enables the generation of desired data by specifying the corresponding labels during the generation process.

4.2.2 Generative Adversarial Networks. A generative adversarial network (GAN) [46] consists of two networks: a generator G and a discriminator D (Figure 2(b)). The training process involves a minimax game where the generator aims to create realistic data to fool the discriminator, and the discriminator aims to distinguish real from generated data accurately. Through iterative training, the generator improves its ability to generate data that the discriminator cannot distinguish from real data. The generator learns to generate data indistinguishable from real samples through iterative training. A GAN learns the data distribution without prior knowledge, which makes it a powerful tool for various generative tasks, including sequence generation, despite the challenges posed by the non-differentiability of discrete data. Variants of GANs have been proposed as researchers tackle the challenges of sequence generation. One of the biggest challenges is the non-differentiability of discrete data. SeqGAN [161] addresses this by leveraging **reinforcement learning (RL)** techniques to frame sequence generation as a sequential decision-making problem. The generator learns a stochastic policy through policy gradient [130], using the discriminator's evaluations of complete sequences as rewards. SeqGAN uses a Monte Carlo search with a rollout strategy to assess intermediate steps. In addition to SeqGAN, notable variants include LeakGAN [51], CGAN [93], WGAN [4], and BiGAN [33]. LeakGAN utilizes hierarchical RL to enhance feedback in the discriminator for sequential data. CGAN adds conditional information to control sequence generation, while WGAN addresses issues like mode collapse using the Wasserstein distance. BiGAN incorporates an encoder for inverse mapping from data to latent representation.

4.2.3 Denoising Diffusion Probabilistic Models. **Denoising diffusion probabilistic models (DDPMs)** [59] are based on a process that iteratively refines samples by adding and then removing noise (Figure 2(e)), ultimately generating data that resembles the target distribution. A diffusion model is typically composed of two key components: the forward diffusion process and the reverse denoising process. The forward process is defined as a Markov chain that adds small amounts of noise at each step. Mathematically, given data X_0, the forward process produces X_t by adding noise according to a predefined schedule. The reverse process is parameterized by a neural network that predicts the original data from the noisy observations and is trained to minimize the difference between the predicted and actual noise. Recent advancements in diffusion models have demonstrated their potential to generate high-quality images. One of the notable benefits of these models is their ability to produce diverse and realistic samples, which has been evidenced by their

performance at image synthesis tasks. Unlike other generative models, such as GANs, diffusion models do not suffer from mode collapse, where the model generates a limited variety of outputs. This makes diffusion models particularly advantageous in applications that require high diversity in generated samples.

However, DDPMs require a large number of diffusion steps to perform well, and this slows down the sample generation process. To address this issue, **denoising diffusion implicit models (DDIMs)** [127] were introduced. Although DDIMs share the same training objectives as DDPMs, they do not require the diffusion process to be a Markov chain. By setting the variance to zero at all time steps, the randomness of Gaussian noise is eliminated, resulting in deterministic sampling. This reduction in sampling steps accelerates the generation process.

The noise addition process in DDPMs is executed through discrete time steps, with both the forward and reverse diffusion processes divided into T steps. By contrast, **score-based generative modeling (SBGM)** [128] interprets these steps as a continuous transformation that can be described by a **stochastic differential equation (SDE)**, thereby simplifying the solution process. SBGM provides a robust theoretical foundation for training diffusion models, ensuring that they can effectively learn the underlying data distribution.

4.3 Comprehensive Evaluation Methods for Peptide Generative Models

4.3.1 Data Partitioning. Data partitioning is crucial to evaluate generative models accurately. Proper partitioning helps prevent overfitting and underfitting. A common approach is the holdout method, which divides the dataset into training, validation, and test sets. The training set is for learning parameters, the validation set for tuning hyperparameters, and the test set for the final evaluation of the model's generalization ability [45]. For low-data scenarios, k-fold cross-validation is effective at preventing overfitting. This involves splitting the data into k subsets, using each subset as a test set, and then averaging the results. The approach ensures that every data point is used for training and testing, providing a more reliable performance estimate. This method is prevalent in peptide studies.

4.3.2 Evaluating Peptide Generation Models.

Functional Validation. Functional validation directly reflects the effectiveness of peptide generation models. Although wet lab experiments are the gold standard for testing peptide performance due to their accuracy, they are resource-intensive and time-consuming. Fortunately, computational models have advanced such that they can evaluate generated peptides effectively. Prediction tools estimate peptide function likelihood or provide experimentally confirmed data values. For example, CAMPR3 [146], CAMP [42], SignalP 6.0 [135], and TransPHLA [20] demonstrate excellent performance in predicting peptide attributes. Molecular dynamics simulations reveal peptide–biomolecule interactions [20, 23, 37], aiding in understanding stability, flexibility, and binding affinity.

After preliminary computational assessments, promising peptides can be selected for further experimental validation. Various functional assays are employed depending on the specific application of the peptide. For example, evaluating the efficacy of AMPs involves determining the **minimum inhibitory concentration (MIC)** against particular bacteria (e.g., *Escherichia coli*), where the MIC is defined as the minimum concentration of an antimicrobial agent that inhibits microbial growth. The lower the MIC value, the higher the activity of the peptide, indicating greater potency against the target microorganism.

Quantitative Evaluation.

— **Statistical analysis for sequences**: Methods such as the Pearson correlation coefficient, cosine similarity, and Euclidean distance are used to compare the representation of the

generated sequences with the ground truth. BLEU and perplexity (PPL) scores reflect the reconstruction quality of the generative model. Novelty is assessed by calculating the proportion of generated peptides not present in the training set, indicating whether overfitting has occurred. Diversity among the generated peptides is also crucial for evaluating the potential of drug candidates and determining whether the model suffers from mode collapse.

- **Physicochemical properties**: Tools like the "Peptides" R package [103] and the Python library "modlAMP" [96] estimate properties such as length, molar weight, amino acid frequency, net charge, aromaticity, hydrophobicity, and hydrophobic moment. These properties are statistically compared with the training data to assess model learning.
- **Secondary structure analysis**: Specific functions often relate to specific structures. Secondary structure analysis (e.g., α-helix, β-sheet, and random coil) can be performed using methods like GOR IV to evaluate the helicity for designed AMPs [27].

4.3.3 Classification Metrics for Evaluating Generated Peptides. Effective evaluation of generated peptide sequences is paramount in peptide design. Classification models are indispensable for screening effective peptides or reducing the proportion of unexpected ones [97, 148]. Several methods have been developed to improve classification models, including the confusion matrix and **receiver operating characteristic (ROC)** curve. The confusion matrix, which delineates true positives, true negatives, false positives, and false negatives, provides insights into classification accuracy. Derived metrics like sensitivity (recall), specificity, precision, and accuracy standardize model efficacy measures. Metrics such as the F1 score balance precision and recall, while the ROC curve plots the true positive rate against the false positive rate, with the area under the curve (AUC) as a benchmark, quantifying the discriminative power of the classification model. Additionally, the Matthews correlation coefficient and perplexity offer nuanced insights, particularly useful in diverse contexts. In summary, these metrics underscore the importance of classification models in peptide design and their efficacy in evaluating generative model performance, contributing to the advancement of peptide design strategies.

4.4 Summary of DGM

Table 3 shows the comparison of various generative models and their variants that have adopted different optimization strategies. Recently, diffusion models have become prevalent in generating peptide sequences and structures, due to their superior capability of fitting distributions compared to earlier models. Therefore, the integration of diffusion with diverse techniques for peptide design is a field with significant potential. Additionally, establishing benchmarking metrics is necessary to enable unbiased and objective comparisons of different models across various domains.

5 Applications and Optimization Strategies of Generative Models in Therapeutic Peptide Design

Novel algorithm-assisted de novo peptide design and evaluation are increasingly considered an effective means of searching the desired part of the vast chemical space, as starting points for hit-to-lead optimization, to expand compound libraries, and as a tool for encoding before performing other deep learning tasks. There have been several applications for peptide generation based on generative models. This section underlines the achievements and optimization of deep generative models on peptide design. We categorize the applications into six types. According to these six categories, Table 4 summarizes the architecture, data sets, data size, and target peptides used in the applications mentioned below.

Table 3. Comparison of Different Generative Models and Variants that Adopted Different Optimization Strategies

Techniques	Optimization	Model	Year	Summary
Data Representation Learning Architecture	Small-scale architecture	RNN [34]	1990	RNNs are suitable for processing and predicting time series data but struggle with long-timescale dependencies. LSTMs address gradient vanishing and explosion issues, with four times more parameters than RNNs. GRUs have a simpler structure than LSTMs, improving training effectiveness. All three can handle inputs of arbitrary length and consider historical information, but their Markov chain nature makes them work serially and ignore future information.
		LSTM [60]	1997	
		GRU [18]	2014	
	Self-attention-based large-scale architecture	Transformer [143]	2017	The Transformer excels in processing long sequences using self-attention, which allows for better interpretability, parallelization, and modeling of long-term texts compared to RNNs. GPT, the Transformer's decoder, is a unidirectional autoregressive model. BERT, the Transformer's encoder, learns bidirectionally to capture context information but trains more slowly. These models have many parameters, making them suitable for pretraining to obtain a universal data representation, requiring only a small amount of data for fine-tuning. However, their training is slow and compute-intensive.
		GPT [111]	2018	
		BERT [30]	2018	
Real Data Distribution Learning Architecture	Directed probability graphical	VAE [71]	2013	VAEs can learn a smooth hidden state but may result in a biased representation of the input data.
	Adversarial training	GAN [46]	2014	GANs can learn the unbiased distribution of real data, generating samples closer to the source data. However, they are not well-suited for generating discrete data. Their alternate optimization makes training challenging, and they also suffer from poor interpretability.
	Conditional generation	CGAN [93]	2014	Constrains the model with additional information to guide data generation.
		CVAE [126]	2015	
	Latent space encoder	BiGAN [33]	2016	Achieves inverse mapping from data to latent representation.
	Improvement of loss function	WGAN [4]	2017	Solves the pattern collapse problem by adopting the Wasserstein distance to measure the sample distribution distance.
		WAE [137]	2017	
	Improvement of regularization	WGAN-GP [50]	2017	Replaces WGAN's weight clipping with a gradient penalty strategy allows for more stable training.
	Improvement in sequence generation	SeqGAN [161]	2017	Solves the problem of backpropagation of discrete data and the evaluation of intermediate sequences with the aid of reinforcement learning.
	Long sequence generation	LeakGAN [51]	2018	Introduces the ideas of hierarchical RL to solve the problems of insufficient feedback information and sparse feedback in discriminator.
	The forward diffusion process and the reverse denoising process	DDPM [59]	2020	Generates diverse and realistic samples, but the generation speed is slow.
		DDIM [127]	2020	Deterministic sampling, more efficient and faster.
		SBGM [128]	2020	The generated results are diverse, and the solving process is simplified.

Table 4. Six Types of Applications in Generative Models for Peptide Design with Corresponding Dataset Sizes and Target Peptides

Application type	Architecture	Dataset size	Target	Year	Ref.
Solely deep generative models	LSTM	1, 554	AMP	2018	[97]
	LSTM	1, 011	AMP	2018	[99]
	GRU	4, 774/797[a]	AMP	2021	[13]
	VAE	5, 942	AMP	2020	[27]
	VAE	1, 554 AMP, 299 ACP	AMP, ACP	2019	[14]
	LSTM	600	CPP	2021	[118]
	LeakGAN	553	AVP	2023	[129]
With pretraining	CVAE	1.7M/7, 960[a]	AMP	2018	[24]
	LSTM	3, 274, 675/-[a]	Bioactive peptides	2023	[164]
	DDIM, ESM	8M/195, 121[a]	AMP	2023	[16]
With attribute-controlled	LeakGAN	16, 648	AMP	2020	[139]
	BiCGAN	496, 891	AMP	2021	[142]
	CVAE	247, 506	AMP	2023	[131]
	DDPM	3, 118	Helix-peptide	2024	[157]
Forward-only strategy	WAE	> 1.7 M/8, 683[a]	AMP	2021	[23]
With multi-modal method	DM, Transformer, EGNN	20, 129 AMP, 4, 381 ACP	AMP, ACP	2024	[151]
With protein-dependent method	Transformer	25, 000 SP-protein pairs	SP	2020	[156]
	GAN	380 peptides, 25, 239 proteins	Protein-binding peptides	2020	[116]
	SA	7, 320 peptide–HLA complexes	HLA-binding peptides	2022	[20]
	LSTM	31, 3652 peptides, 10, 551 proteins	MHC-binding peptides	2023	[17]
	DM, SE(3)-EGNN	9, 225 protein-peptide complexes	Protein-binding peptides	2024	[112]

[a]The models in these applications are pre-trained, and the dataset size is the number of pretraining/fine-tuning.

5.1 Designing Functional or Optimized Peptides Solely using Deep Generative Models

One of the fundamental approaches to generating therapeutic peptides with desired properties is to design therapeutic peptides solely using DGMs. Initially, functional peptide data are represented as simple one-dimensional sequences. Subsequently, a vanilla DGM is trained on these data. Finally, an independent predictor or other evaluative method is employed for screening or evaluation. For example, amphipathicity is a crucial property of AMPs. Müller et al. [97] trained an LSTM generative model using one-hot encoded sequences. The generated peptides were evaluated using the AMP classifier CAMP [147] and compared for helix structure similarity with AMP templates [38] to demonstrate the presence of amphipathicity. Nagarajan et al. [99] employed a model and training method similar to Müller et al. However, they conducted a more comprehensive evaluation by constructing a Bi-LSTM-based MIC prediction model. Peptides were scored using this model, and those with low MIC scores were subsequently tested in experiments. The results demonstrated that the generated peptides exhibited broad-spectrum activity and well-folded structures [148].

Considering that the design of amphipathic peptides can sometimes lead to cytotoxicity, that is, hemolytic activity, which greatly limits their use as drugs—researchers have sought ways to balance amphipathicity and toxicity. To address this challenge, Capecchi et al. [13] developed an RNN-based AMP generative model. This model was trained with both AMP and non-hemolytic peptides and screened using predictors for both properties. AMP design tends to favor short sequences [13, 99, 148], as shorter peptides are less likely to form complex three-dimensional structures that sequence models cannot accurately interpret. This simplicity in structure not only makes short peptides easier and cheaper to manufacture but also ensures that their predicted sequences are more likely to match their actual structures.

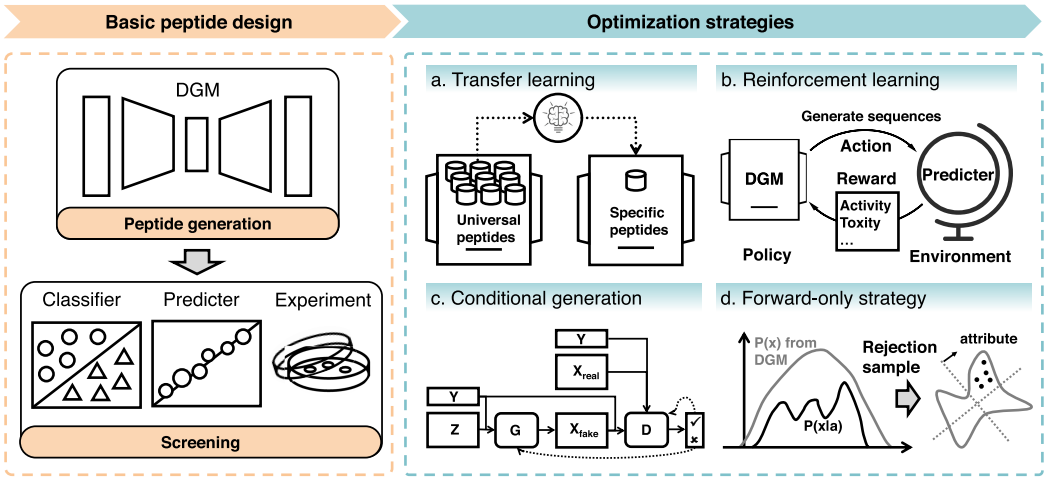

Fig. 3. A diagram illustrating the optimization strategy for designing peptides based on deep generative model (DGM). The sequence generated by individual DGM is far from satisfying the required properties as a drug; thus, a tedious series of screenings has to be performed. Optimization strategy can improve the efficiency of peptide generation in terms of (a) generic model learning, (b) controlled generation, (c) semi-supervised learning, and (d) forward-controlled sampling strategies.

In applying CPPs, to enhance the delivery of specific biomolecules within cells, Schissel et al. [118] designed an LSTM-based CPP generative model. By combining this model with a prediction model and directed evolution, they discovered "Mach" nuclear-targeting miniproteins, which are highly effective delivery structures for a specific type of therapeutic drug, the PMO.

One drawback to de novo generation of AVPs is the limited data available. To address this, Surana et al. [129] were the first to use a GAN to generate AVPs. They trained the model on a very limited dataset of 553 high-quality AVPs. Utilizing LeakGAN to leak learned features from the discriminator to the generator, they successfully generated an optimal set of antiviral peptides.

VAEs offer an alternative approach by sampling distributions, facilitating diversity while maintaining semantic continuity. For example, Dean and Walper [27] performed interpolation between the vectors of AMPs and scrambled peptides in the latent space of a VAE. They validated the VAE's ability to capture functional differences in sequences within the latent space using the activity predictor CAMPR3 [146] and the secondary structure evaluation method GOR [43]. Chen and Kim [14] used a temperature factor ranging from 0 to 1 during the decoding stage of the VAE, making the generated peptide sequences more deterministic.

5.2 Generation of Peptides with Pretraining

More specific data feeding is required to train deep model parameters adequately and prevent overfitting, which is usually a limited resource. Pre-training leads to remarkable performance improvements for many NLP tasks [86]. Pre-training on massive unlabeled data can obtain more general language representations and benefit downstream tasks. Training of the pretraining model consists of two stages (Figure 3(a)). First, massive unlabeled data are incorporated into general representations via a self-supervised learning strategy. Second, the model is fine-tuned via supervised learning, which can be adapted to different kinds of tasks with only slight modifications. In the case of peptides, this global approach allows for meaningful density modeling across multiple families and better learning of the "grammar" of peptides. For example, Das et al. [24] designed a VAE-based AMP controllable generation model PepCVAE, which controls the properties of generated

AMPs by combining VAE with a CNN to learn a space of disentangled antimicrobial properties. Pre-trained models with all known short peptide sequences in UniProt explored domains beyond the known antimicrobial templates, and the results demonstrated that these pre-trained models produced sequences with high reconstruction accuracy and diversity. Zhang et al. [164] used the general database PeptideAtlas [28] to pre-train an LSTM model, enabling it to learn the general syntax of peptides. They then fine-tuned the model with specific peptides and employed structure-based screening to discover bioactive peptides capable of binding to a particular target.

Instead of pretraining a model by arduously collecting vast amounts of data, Chen et al. [16] leveraged the state-of-the-art pre-trained protein language model ESM-2 [84] to enhance the diffusion model's ability to capture the essence of AMPs in the latent space. They used the ESM-2 encoder to map peptide sequences into a continuous latent space and designed a latent-space diffusion protein language model based on DDIMs. The denoising process employed pre-trained ESM-2-8M attention blocks. Utilizing the prior knowledge acquired by the ESM model enables the generation of biologically plausible peptides.

5.3 Generation of Peptides with Attribute Control

A generative model may not always go in the right direction individually, and a separate predictor screening tool does not improve the generative model at the source. To improve the generation efficiency and allow the generative model to autonomously guide the generation of the desired data, an advanced strategy is to optimize the generative model by using constraints on important properties-related information. One of the strategies is RL (Figure 3(a)). For example, the LeakGAN-based PepGAN [139] sets a mixing constant λ as the weight of the discriminator output and $(1-\lambda)$ as the weight of the activity predictor output, with the weighted sum as the reward score. PepGAN is hence more perceptive of activity than LeakGAN and avoids generating sequences in the distribution of negative examples. The statistical results of the match between the generated sequences and the positive samples in terms of physicochemical properties prove the favorable impact of the activity predictor in statistical fidelity. One of the generated peptides was shown to have vigorous antimicrobial activity, with a minimum inhibitory concentration of 3.1 μgmL, which is twice as strong as the widely used antibiotic ampicillin.

The design of peptides can be seen as a problem of approximating the conversion of energy functions into generative distributions. Bengio et al. [8] suggested that the goal of scoring generated drug candidates high in a virtual screen is modality-singular, and so diversity should be a key consideration in sequence design. Thus, they proposed GFlowNet, a flow network-based generative model that treats the generative process as a flow network and takes the flow consistency equation as a learning objective. So that the exploration of RL is not limited to a single mode, GFlowNet proportionalizes the probability of a sampled object to the given reward of that object $P(T) \propto R(T)$. When combined with active learning, diverse peptides are generated that are more promising for meeting multiple drug properties [65].

In addition to RL, another constraint strategy is to perform controlled generation by adding conditional variables (Figure 3(c)). For example, Van Oort et al. [142] designed AMPGAN v2, which is a BiCGAN that associates conditional variables with features learned by the discriminator to encourage the generator to consider the same associations and then controls the output of the generator. In addition, latent vectors of data are generated by applying the architecture of BiGAN. AMPGAN v2 generates diverse, novel sequences while maintaining key AMP features. Szymczak et al. [131] developed a framework based on CVAE, treating antimicrobial and anti-*E. coli* activities as two conditions. They employed various regularization techniques and specific classifiers to enhance the stability of controllable effects. They used Jacobian decoupling regularization to disentangle the latent representations of peptides and conditions. Additionally, they introduced analogue generation,

which enhanced the diversity of generated peptides by initiating generation from negative samples and employing Gumbel softmax approximation to regularize the reconstruction of latent vector.

In bioactive peptide design, structural stability is crucial, often achieved through well-designed α-helices. Therefore, one method of controlling generation involves identifying α-helical structures as hotspots and encoding them as conditional constraints. Xie et al. [157] developed a hotspot-specific generation algorithm using data selected from the PDB database. The data are represented by a combination of one-hot encoding of amino acid sequences and 3D coordinate structural encoding, forming an image-like representation. The framework employs a score-based generative diffusion model [128] and a conditional mask constraint for the full-atom design of α-helix peptides. The energy minimization of the generated data is optimized using the advanced protein prediction software, Rosetta.

5.4 Accelerating Controlled Generation with a Forward-only Strategy

The methods mentioned above suffer from additional (computational) complexity, where RL requires policy learning and conditional vectors require collecting different conditional data and making conditional labels. A simple and efficient alternative to achieve controlled generation is to combine forward-only mathematical strategies that can subtly incorporate information from the classifier into the generative model (Figure 3(d)). One such strategy is rejection sampling, a Monte Carlo random sampling method for complex problems with hard-to-find cumulative distribution functions. The basic idea is to sample the data with a proposed distribution that can be directly sampled and then use the target distribution ratio of the proposed distribution as the acceptance probability of samples. The sample is accepted when a random number from the uniform distribution is larger than the acceptance probability. Specifically, the CLaSS [23] method takes an unknown and complex conditional data distribution as the objective function and the VAE explicit density model (i.e., posterior distribution) as the proposed distribution. The functional attribute prediction of the classifier on samples from the proposed distribution will naturally appear as the rejection rate. This method is simple and highly parallelized, implementing a joint generative model and classifier to approximate the controllable generation.

5.5 Generation of Peptides with Multi-modal Method

The proposal of multi-modal generative methods arises from two key points. First, using sequences alone or simply concatenating sequences with structures impedes generation performance. Second, most models rely on positive samples of therapeutic peptides, where the learning of negative samples proves valuable to meet the demands for more data and diversity. For example, Wang et al. [151] introduced a **multi-modal contrastive diffusion model (MMCD)**. Their model integrates both sequence and structural modalities within a diffusion framework and employs multi-modal **contrastive learning (CL)** strategies at each diffusion time step. MMCD co-generates the sequence and structure of peptides. Sequence information is encoded using one-hot encoding, while structural information involves adding noise directly to atomic coordinates, due to their continuous nature in three-dimensional space. The denoising process utilizes a transformer and EDGG. Multi-modal CL in MMCD comprises inter-CL and intra-CL. Inter-CL maximizes the **mutual information (MI)** between sequence and structure pairs of the same peptide while minimizing the MI between the sequence of one peptide and the structure of another, aiming to align sequence and structure embeddings. Intra-CL maximizes/minimizes MI for positive/negative therapeutic peptides, aiming to distinguish embeddings of therapeutic from non-therapeutic peptides. This method captures consistency between the two modalities, enhancing model performance and reinforcing the diffusion model's ability to generate high-quality therapeutic peptides.

Table 5. Peptide Generation Models Evaluated using the Same AMP Database

Model	Type of model	Database	Ant. Score ↑	Similarity ↓	Perplexity ↓
LSTM-RNN [97]	LSTM	ADAM, APD, DADP	0.855	39.6164	20.26
CLaSS [23]	WAE	DBAASP, APD3, LAMP, CAMP	0.8757	\	12.87
AMPGAN v2 [142]	BiCGAN	DBAASP, AVPdb	0.8617	38.308	17.7
HydrAMP [131]	VAE	dbAMP, APD3, DRAMP	0.8145	31.0662	17.27
AMP-Diffusion [16]	DDIM, ESM	dbAMP, APD3, DRAMP	0.81	\	12.84
MMCD [151]	Diffusion, Transformer, EGNN	APD3	0.881	24.4107	\

All metrics are self-reported and evaluated with the APD database.

As shown in Table 5, we compared the performance of peptide DGMs that use similar databases and did not involve wet lab experiments (except CLaSS). All models achieved antimicrobial scores above 0.8. Among these, the diffusion model MMCD stands out: by incorporating structural information and leveraging multimodal contrastive learning, it achieved **state-of-the-art (SOTA)** performance and distinguished itself by its capability to generate more novel peptides.

5.6 Discovery of Peptides that Depended on Proteins

Promoting certain functions of proteins is another goal of designing therapeutic peptides. Peptides influence many cellular processes and metabolic systems, such as signal transduction and regulatory networks [31], by directing the secretion of proteins. Peptide–protein interactions are closely associated with the pathogenesis of human diseases such as cancer and neurodegenerative diseases [92]. In immunology, the binding of peptides to HLAs is essential for antigen presentation and recognition by T cells that trigger an immune response [160]. In protein engineering, SPs can improve the efficiency of the biotechnological production of target proteins [40]. Such peptides are good starting points for the design of novel therapeutics. However, an exhaustive understanding of the experimental details of peptide–protein interactions remains a substantial task [160], thus providing an opportunity for the development of deep learning computational methods.

The transformer has advantages in designing peptides that depend on proteins because the self-attention mechanism is not limited to the length of the input sequence. For example, since the computational prediction of interactions between HLA and peptides can accelerate antigenic epitope screening and vaccine design, Chu et al. [20] constructed TransMut. TransMut is a transformer-based framework for predicting peptide and HLA binding affinities and automatically optimizing mutant peptides. It is trained by uniting the extracted peptides and HLA attention scores, which are then used to discover some of the most important amino acid sites for the most critical peptides and HLAs. Essential sites on these weak affinity peptides are replaced with amino acids that might contribute more to the binding affinity, resulting in mutant peptides with more vital binding ability.

In addition to probing the interaction between peptides and proteins by taking them as inputs, another strategy is to consider peptide generation as machine translation. This involves taking a protein as the input and a peptide with a strong dependence on the protein as the output. For example, Wu et al. [156] approximated the transformer model to generate SPs specific to the desired secretion protein as machine translation, which uses self-attention to explore the underlying semantics in the language of proteins and SPs. Their results showed that the generated SPs are functional and lead to secreted protein activity that is competitive with that of industrially used SPs. The application of transformer-based mini-proteins that bind with protein design research remains to be explored. With effectiveness across a range of domains, transformer-based models have garnered considerable interest lately. Many variants have been proposed that improve computational and memory efficiency, and others are still in progress [61, 134]. Thus, peptide research will further benefit from these prospective approaches.

Table 6. An Overview of Protein-binding Peptide Generation Models

Model	Type of model	Database	Target	Representation	Imm. Score ↑	Affi. Score ↓
PepPPO [17]	LSTM	IEDB	MHC	One-hot, BLOSUM matrix, Embedding layer	91.48	\
DeepImmuno [80]	WGAN	IEDB	MHC	AAindex PCA	67	\
GANDALF [116]	GAN	THPdb	PD-1	Text, 4D tensor	\	-9.191
HYDRA [112]	DM, SE(3)-EGNN	PepBDB	PfEMP1	One-hot, 3D coordinate	\	-4.112

All metrics are self-reported.

Susceptibility to enzymatic degradation restricts the effectiveness of targeted therapeutic peptides. This limitation can be mitigated by enhancing the peptides' binding affinity to proteins. Utilizing 3D information representation enables the generation of biomolecules with more realistic 3D structures and improved binding energies at protein binding sites. For example, Ramasubramanian et al. [112] developed a stable, target-aware peptide design model by integrating 3D structural information into a hybrid diffusion model. They used an SE(3)-equivariant graph neural network to ensure spatial consistency between generated peptides and protein binding pockets throughout the generation process. The binding affinity of these peptides was subsequently optimized using energy minimization techniques and heuristic algorithms for binding affinity maximization.

The generation of binding peptides can be approached as an RL problem. For example, Chen et al. [17] proposed PepPPO, a framework designed to generate qualified peptides for binding motif characterization. PepPPO employs the proximal policy optimization algorithm and uses rewards from a peptide-MHC binding predictor to learn a mutation strategy. This strategy optimizes random initial peptides by gradually mutating their amino acids until the peptides are predicted to be positive binders. Additionally, the model uses a hybrid encoding method for peptides and proteins, combining one-hot encoding, the BLOSUM matrix, and an embedding layer.

Table 6 summarizes the performance of some protein-binding peptide generation models. Since these models utilize protein or structural information, they integrate multiple representations to more accurately represent the data. The evaluation assesses the efficacy of peptides against their targets. Table 6 shows MHC-binding peptide generation models evaluated based on the immunogenicity of the generated peptides. For example, PepPPO [17] employs MHCflurry2.0 [104] to assess the immunogenic score, which is a composite score of antigen processing and the binding affinity. Models considering structural factors typically use docking software to assess the binding affinity to specific proteins. GANDALF [116] uses 25,000 human protein structures as augmented data to train a structural generation model and evaluates binding affinity using PyDockWEB. Similarly, HYDRA [112], which incorporates 3D coordinates of binding pockets, evaluates affinity using Autodock Vina.

6 Challenges and Prospects

Deep learning in peptide design is still in its infancy, and there are not as many research reports as those of proteins. Owing to the homogeneity of peptides and proteins, the increase of peptide data, and the enthusiasm for exploring deep learning models, deep learning will certainly develop rapidly in peptide design. Still, problems remain despite the tremendous effort put into peptide design. It is unclear which methods or models have general advantages in peptide design because many of them are rooted in abiotic fields. Consequently, selecting an appropriate model for designing target peptides from among the various models is challenging. The model should address the interdisciplinary differences and its inherent shortcomings, such as the difficulty in generating discrete data with GANs and the possibility that the generated examples may be of low diversity (mode collapse). Numerous variants of DGMs have been developed as architectures [48, 165]. With different architectures, specific designs must be made on the underlying layer to achieve good task

performance. Three key perspectives— structural, algorithmic, and hyperparametric—should be fully considered to leverage generative models for performing peptide generation tasks better.

6.1 Fuller Utilization of Low Data

The nature of requirements for specific functions in de novo peptide design increases the need for abundant target data. Data scarcity directly leads to data imbalances and model overfitting. However, sequence data with associated experimental data is more limited for peptides with specific functions, especially for some niche peptides targeting particular receptors, and such data are much more costly to come by than small molecule data. For this purpose, unsupervised representation learning on extensive data through pretraining offers a potential solution through which peptide "grammar" can be better learned globally across multiple families. Thus, sequences with high reconstruction accuracy and diversity are more likely to be generated. The WAE-based AMP generation model from ClaSS [23] is pre-trained on UniProt with many peptides. The similarity of the potential space generated by the model was investigated by linear interpolation, and sequence similarity in the potential space was found to be negatively correlated with the Euclidean distance, indicating that WAE essentially captures the sequence relationships within the peptide space. A pre-trained generic generative model can perform transfer learning on peptides with different functions [13, 49]. Studies regarding pretraining on peptide generation models are still rare, albeit promising.

Similar to generic generative models that can better exploit low data on different therapeutic peptides, classification models that make controlled generation more generic can also speed up peptide discovery. One approach to better explore the relationships between data is metric-based meta-learning [125], which makes use of prior knowledge of artificial measures, such as metrics that measure the distance between categories (the Euclidean distance or cosine distance). Although the goal of both meta-learning and pretraining is to obtain a better set of model initialization parameters, pretraining focuses on the model's performance of the current task, whereas meta-learning focuses more on the model's potential to go beyond the local optimum and find the optimal solution for multiple tasks. A recent example is a generic peptide bioactivity predictor [55] that calculates the mean value of each category based on the support sets as prototypes and the loss function of the data in the query sets of each category, together with mutual information that can exploit the unsupervised information of peptides. This first generic predictor predicted 16 different peptide functions. Incorporating more general prediction models into generative models to accelerate the discovery of various kinds of peptides for exploration is one direction for future work.

Another way to address data scarcity is data augmentation, which performs well in image studies [122] because its semantics remain the same with slight changes. By contrast, the augmentation of discrete sequence data is much more difficult [36]. Lee et al. [77] augmented data via random substitutions and insertions of amino acids of experimentally confirmed neurotoxic peptides and helped the model progress in target recognition. When designing protein-binding peptides (HLA–peptides complexes) or protein transportation guides such as SPs, the protein data can be increased by truncating to a different sequence length [156], which also avoids the effect of using one same-length truncation. Generative models themselves can also serve as tools for data augmentation [88]. The challenge of biological sequence data augmentation lies in the randomness of augmentation and the conservation of biological data. Much room remains for further exploration.

6.2 Utilization of Peptide Structure Information

Several well-evidenced findings indicate that exploring information at the biomoleculear atomic and structural levels is as essential as sequence information. One study illustrates that amino acid

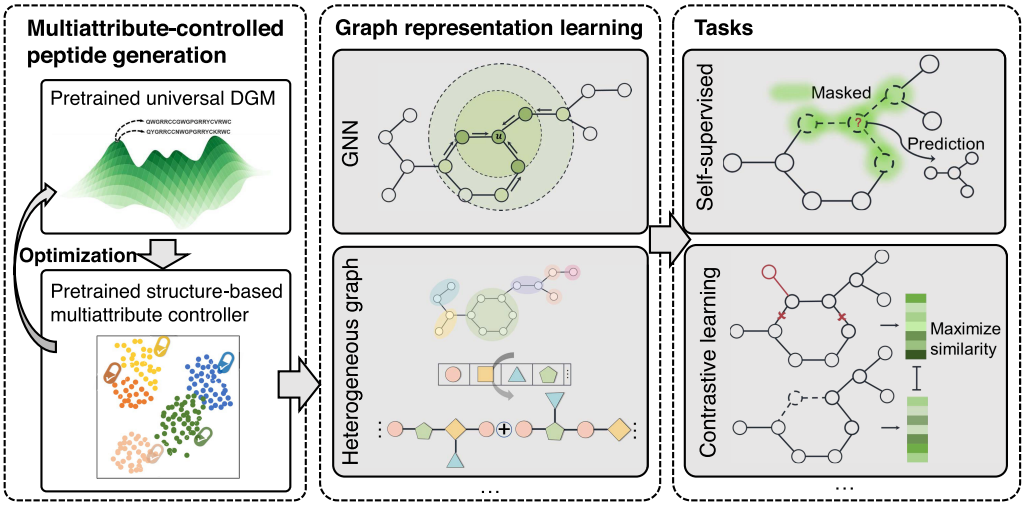

Fig. 4. A hypothetical framework for the structure-assisted multiattribute-controlled generation of therapeutic peptide, including (i) Pretraining phase: pre-train universal DGM and pre-train structure-based multiattribute controller through different graph representation learning methods and tasks; (ii) Optimization phase: Optimize the DGM using feedback from the sampled data through the output of the multiattribute predictor.

sequences that are structurally homologous and have similar biological functions may have low sequential similarity [56]. Some suggest that long-range dependence in a sequence is generally short-range in 3D space [64]. In deep learning, the amino acid linear sequence representation is one-sided and limited. For example, the sequence representation of short peptides lacks rich contextual information, making it difficult to capture discriminative features [152]. The molecular graph representation can solve this deficiency. A graph is a data structure consisting of a set of vertices and a set of edges, where nodes can have multiple adjacent nodes, and edges are connections between two nodes (Figure 4(a)). Molecules are essentially compositions of atoms and bonds between interconnected atoms, which can naturally be described by a graph with rich structural and spatial information. **Graph neural networks (GNNs)** are good at capturing connections between nodes and representation learning of graph data, and they can integrate features obtained from graph embedding. GNNs can process graphs, and have been widely used on small molecules for molecular property prediction [54, 87] and molecular generation [69, 120]. With proteins, for example, Fout et al. [39] predicted protein interfaces with effective graph latent representations that represent the 3D structure of a protein by graph convolutional neural networks. Such graph representations use the sequence information of residues and the degree of residue exposure as node features, and the relative and distance angular description between residues as edge features.

A feasible peptide graph representation is a graph with atoms as nodes when the peptide chain length is not too long. For example, Wei et al. [152] proposed a peptide toxicity predictor ATSE based on a GNN and attention mechanism. The structural features of peptide molecular graphs are extracted using the GNN. In terms of generation, molecules generated node by node are hardly compatible with chemical validity. A more desirable approach takes a valid chemical substructure (subgraph) as the generating unit [69]. However, unlike small molecules with relatively simple structures, peptides are more complex with 3D folds and thus face more significant challenges in graph generation. Recently, the transformer-based model ProteinMPNN [25], which solves the long-standing problem of inverse protein folding, can determine the amino acids sequence

of a protein based on its 3D structure. ProteinMPNN incorporates the distance, direction, and orientation in the side features in the protein. Thus, the 3D structure of the protein is modeled by capturing the dependencies in its amino acid sequence to produce protein embedding. Thus, the use of graph information can potentially help speed up the design of therapeutic peptides.

6.3 Automatic Multi-attribute Optimization

Although the de novo generation of peptides allows the development of novel and valid sequences with a desired function or with few optimized properties, DGMs consider only the target potency in most cases. Experiments are still needed to screen for sequences that better match the necessary properties required for a drug, such as physicochemical properties, selectivity, specificity, novelty, druggability, synthetic feasibility, and so on. If multi-attribute optimization (or multi-objective optimization) is set up in a computational framework, the drug discovery process will be greatly accelerated. A major challenge is that some specific properties for output sequences are difficult to formulate or optimize. It is difficult to automate the generation for multi-attribute optimization. Classifiers for models trained on domain-specific datasets increase the probability that candidates have the desired feature [23]. Therefore, one way to incorporate multiple attributes into generative models is to combine predictive or classification models whose output can be used as a reward for RL. There is already precedent for this in molecular generation, such as the new small molecule design method GENTRL [168], which is used to generate effective inhibitors of the discoid protein structural discoidin domain receptor 1 (a promising target for tumor therapy). GENTRL constrains the generated data of VAE by RL reward from assessing the self-organizing maps of the attributes, including their novelty compared to already existing drugs, general kinase inhibitors, and discoidin domain receptor one inhibitor. It also utilizes tensor decompositions to tie latent codes and properties (parameterization). GENTRL identified and tested potent inhibitors of discoidin domain receptor 1 for only 46 days by controlling the synthetic feasibility, novelty, and biological activity. Similarly, in the case of therapeutic peptides, blocking specific receptors or enzymes of the host is a therapeutic category of the recent worldwide outbreak of the SARS-CoV-2 coronavirus. The S-protein and 3C-like protease of the SARS-CoV-2 virion are potential drug targets [53], so exploring sequences with similar characteristics similar to existing S-protein and 3C-like protease inhibitors could provide essential clues for optimizing therapeutic peptides [53]. In short, more rapid and effective multi-attribute automatic machine generation is a significant trend in peptide design.

Combining the perspectives mentioned above, the formation of a structure-assisted multiattribute-controlled peptide generation framework is promising. Specifically, as shown in Figure 4, training the framework includes the following. (i) Pretraining phase: To maximize the utilization of graph structure-based multiattribute controllers (predictors), a large amount of non-specific peptide sequence data are applied to pre-train a universal DGM that guarantees the legitimacy of the generated sequences and, at the same time, pre-train the multiattribute controllers with a variety of graph structure representations of specific peptides and extract structural information through different graph representation learning methods and tasks. (ii) Optimization phase: The sequences sampled by a DGM are converted into molecular graphs using the RDKit tool and sent to the multiattribute controllers. The prediction results are used as feedback to optimize the DGM using suitable algorithms.

7 Conclusion

Carefully designing biological peptides is essential for discovering and developing efficient drugs, but traditional experimental methods are time-consuming and labor-intensive. Compared with machine learning, which requires an expert to extract features manually and has weak learnability at shallow layers, deep learning has powerful nonlinear modeling capabilities and excellent

performance for complex real-world tasks. In this review, we first provided an overview of data processing and representation learning. DGMs and (pre-trained) language models based on self-attention mechanisms were introduced, and various evaluation methods were reviewed to help researchers evaluate, select, and use deep learning models for peptide generation. Then, some peptide design scenarios and related research results were illustrated. Finally, we discussed some challenges and future directions related to data scarcity and model optimization. Deep learning models have a variety of strengths and weaknesses that can be exploited based on design and experimental constraints to improve the stability, affinity, and specificity of the generated peptides. In addition, the number of deep learning model variants is thriving, so cross-domain applicability should be considered before selecting a model.

Finally, while current practice is still plagued by the issues above, the field is rapidly evolving, and more emerging technologies can address these issues. This review can serve as a practical reference for better understanding the research progress of peptide design and to develop more effective models.

References

[1] 2022. Global peptide therapeutics market & clinical trials insight 2028. (2022). https://www.researchandmarkets.com/r/3og5pw

[2] Piyush Agrawal, Sherry Bhalla, Salman Sadullah Usmani, Sandeep Singh, Kumardeep Chaudhary, Gajendra P. S. Raghava, and Ankur Gautam. 2016. CPPsite 2.0: A repository of experimentally validated cell-penetrating peptides. *Nucleic Acids Research* 44, D1 (2016), D1098–D1103.

[3] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 25, 17 (1997), 3389–3402.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 214–223. https://proceedings.mlr.press/v70/arjovsky17a.html

[5] Rainier Barrett, Shaoyi Jiang, and Andrew D. White. 2018. Classifying antimicrobial and multifunctional peptides with Bayesian network models. *Peptide Science* 110, 4 (2018), e24079.

[6] Shaherin Basith, Balachandran Manavalan, Tae Hwan Shin, and Gwang Lee. 2020. Machine intelligence in peptide therapeutics: A next-generation tool for rapid disease screening. *Medicinal Research Reviews* 40, 4 (2020), 1276–1314.

[7] Alex Bateman, Maria-Jesus Martin, Sandra Orchard, Michele Magrane, Shadab Ahmad, Emanuele Alpi, Emily H. Bowler-Barnett, Ramona Britto, Austra Cukura, Paul Denny, et al. 2023. UniProt: The universal protein knowledgebase in 2023. *Nucleic Acids Research* 51, D 1 (2023), D523–D531.

[8] Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. 2021. Flow network based generative models for non-iterative diverse candidate generation. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 27381–27394. https://proceedings.neurips.cc/paper_files/paper/2021/file/e614f646836aaed9f89ce58e837e2310-Paper.pdf

[9] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828.

[10] Pratiti Bhadra, Jielu Yan, Jinyan Li, Simon Fong, and Shirley W. I. Siu. 2018. AmPEP: Sequence-based prediction of antimicrobial peptides using distribution patterns of amino acid properties and random forest. *Scientific Reports* 8, 1 (2018), 1–10.

[11] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, Stefan Riezler and Yoav Goldberg (Eds.). Association for Computational Linguistics, Berlin, Germany, 10–21.

[12] Qiushi Cao, Cheng Ge, Xuejie Wang, Peta J. Harvey, Zixuan Zhang, Yuan Ma, Xianghong Wang, Xinying Jia, Mehdi Mobli, David J. Craik, et al. 2023. Designing antimicrobial peptides using deep learning and molecular dynamic simulations. *Briefings in Bioinformatics* 24, 2 (2023), bbad058.

[13] Alice Capecchi, Xingguang Cai, Hippolyte Personne, Thilo Köhler, Christian van Delden, and Jean-Louis Reymond. 2021. Machine learning designs non-hemolytic antimicrobial peptides. *Chemical Science* 12, 26 (2021), 9221–9232.

[14] Shuan Chen and Hyun Uk Kim. 2019. Designing novel functional peptides by manipulating a temperature in the softmax function coupled with variational autoencoder. In *2019 IEEE International Conference on Big Data (Big Data'19)*. IEEE, 6010–6012.

[15] Sijie Chen, Tong Lin, Ruchira Basu, Jeremy Ritchey, Shen Wang, Yichuan Luo, Xingcan Li, Dehua Pei, Levent Burak Kara, and Xiaolin Cheng. 2024. Design of target specific peptide inhibitors using generative deep learning and molecular dynamics simulations. *Nature Communications* 15, 1 (2024), 1611.

[16] Tianlai Chen, Pranay Vure, Rishab Pulugurta, and Pranam Chatterjee. 2023. AMP-Diffusion: Integrating latent diffusion with protein language models for antimicrobial peptide generation. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*. https://openreview.net/forum?id=145TM9VQhx

[17] Ziqi Chen, Baoyi Zhang, Hongyu Guo, Prashant Emani, Trevor Clancy, Chongming Jiang, Mark Gerstein, Xia Ning, Chao Cheng, and Martin Renqiang Min. 2023. Binding peptide generation for MHC Class I proteins with deep reinforcement learning. *Bioinformatics* 39, 2 (2023), btad055.

[18] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). Association for Computational Linguistics, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179

[19] Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Christina Floristean, Anant Kharkar, Koushik Roy, Charlotte Rochereau, Gustaf Ahdritz, Joanna Zhang, George M. Church, et al. 2022. Single-sequence protein structure prediction using a language model and deep learning. *Nature Biotechnology* 40, 11 (2022), 1617–1623.

[20] Yanyi Chu, Yan Zhang, Qiankun Wang, Lingfeng Zhang, Xuhong Wang, Yanjing Wang, Dennis Russell Salahub, Qin Xu, Jianmin Wang, Xue Jiang, et al. 2022. A transformer-based model to predict peptide–HLA class I binding and optimize mutated peptides for vaccine design. *Nature Machine Intelligence* 4, 3 (2022), 300–311.

[21] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

[22] UniProt Consortium. 2019. UniProt: A worldwide hub of protein knowledge. *Nucleic Acids Research* 47, D1 (2019), D506–D515.

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

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

[25] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Alexis Courbet, Rob J. de Haas, Neville Bethel, et al. 2022. Robust deep learning–based protein sequence design using ProteinMPNN. *Science* 378, 6615 (2022), 49–56.

[26] Scott N. Dean, Jerome Anthony E. Alvarez, Daniel Zabetakis, Scott Allen Walper, and Anthony P. Malanoski. 2021. PepVAE: Variational autoencoder framework for antimicrobial peptide generation and activity prediction. *Frontiers in Microbiology* (2021), 2764.

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

[28] Frank Desiere, Eric W. Deutsch, Nichole L. King, Alexey I. Nesvizhskii, Parag Mallick, Jimmy Eng, Sharon Chen, James Eddes, Sandra N. Loevenich, and Ruedi Aebersold. 2006. The PeptideAtlas project. *Nucleic Acids Research* 34, suppl_1 (2006), D655–D658.

[29] Nicki Skafte Detlefsen, Søren Hauberg, and Wouter Boomsma. 2022. Learning meaningful representations of protein sequences. *Nature Communications* 13, 1 (2022), 1–12.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, MN, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[31] Francesca Diella, Niall Haslam, Claudia Chica, Aidan Budd, Sushama Michael, Nigel P. Brown, Gilles Travé, and Toby J. Gibson. 2008. Understanding eukaryotic linear motifs and their role in cell signaling and regulation. *Frontiers in Bioscience-Landmark* 13, 17 (2008), 6580–6603.

[32] Joseph A. DiMasi, Henry G. Grabowski, and Ronald W. Hansen. 2016. Innovation in the pharmaceutical industry: New estimates of R&D costs. *Journal of Health Economics* 47 (2016), 20–33.

[33] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. 2016. Adversarial feature learning. *arXiv preprint arXiv:1605.09782* (2016).

[34] Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science* 14, 2 (1990), 179–211.
[35] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. 2017. Neural audio synthesis of musical notes with wavenet autoencoders. In *International Conference on Machine Learning*. PMLR, 1068–1077.
[36] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 968–988.
[37] Jonathon B. Ferrell, Jacob M. Remington, Colin M. Van Oort, Mona Sharafi, Reem Aboushousha, Yvonne Janssen-Heininger, Severin T. Schneebeli, Matthew J. Wargo, Safwan Wshah, and Jianing Li. 2021. A generative approach toward precision antimicrobial peptide design. *BioRxiv* (2021), 2020–10.
[38] Christopher D. Fjell, Jan A. Hiss, Robert E. W. Hancock, and Gisbert Schneider. 2012. Designing antimicrobial peptides: Form follows function. *Nature Reviews Drug Discovery* 11, 1 (2012), 37–51.
[39] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. 2017. Protein interface prediction using graph convolutional networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, CA, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6533–6542.
[40] Roland Freudl. 2018. Signal peptides for recombinant protein secretion in bacterial expression systems. *Microbial Cell Factories* 17, 1 (2018), 1–10.
[41] Itsuki Fukunaga, Yuki Matsukiyo, Kazuma Kaitoh, and Yoshihiro Yamanishi. 2024. Automatic generation of functional peptides with desired bioactivity and membrane permeability using Bayesian optimization. *Molecular Informatics* 43, 4 (2024), e202300148.
[42] Musa Nur Gabere and William Stafford Noble. 2017. Empirical comparison of web-based antimicrobial peptide prediction tools. *Bioinformatics* 33, 13 (2017), 1921–1929.
[43] Jean Garnier, Jean-François Gibrat, and Barry Robson. 1996. GOR method for predicting protein secondary structure from amino acid sequence. In *Methods in Enzymology*. Vol. 266. Elsevier, 540–553.
[44] Ulka Gawde, Shuvechha Chakraborty, Faiza Hanif Waghu, Ram Shankar Barai, Ashlesha Khanderkar, Rishikesh Indraguru, Tanmay Shirsat, and Susan Idicula-Thomas. 2023. CAMPR4: A database of natural and synthetic antimicrobial peptides. *Nucleic Acids Research* 51, D1 (2023), D377–D383.
[45] Benyamin Ghojogh and Mark Crowley. 2019. The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial. *arXiv preprint arXiv:1905.12787* (2019).
[46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, Vol. 27.
[47] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5-6 (2005), 602–610.
[48] Karol Gregor, George Papamakarios, Frederic Besse, Lars Buesing, and Theophane Weber. 2018. Temporal difference variational auto-encoder. In *International Conference on Learning Representations (ICLR'18)*.
[49] Francesca Grisoni, Claudia S. Neuhaus, Gisela Gabernet, Alex T. Müller, Jan A. Hiss, and Gisbert Schneider. 2018. Designing anticancer peptides by constructive machine learning. *ChemMedChem* 13, 13 (2018), 1300–1302.
[50] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, Vol. 30.
[51] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
[52] Sudheer Gupta, Pallavi Kapoor, Kumardeep Chaudhary, Ankur Gautam, Rahul Kumar, Open Source Drug Discovery Consortium, and Gajendra P. S. Raghava. 2013. In silico approach for predicting toxicity of peptides and proteins. *PloS One* 8, 9 (2013), e73957.
[53] Yanxiao Han and Petr Král. 2020. Computational design of ACE2-based peptide inhibitors of SARS-CoV-2. *ACS Nano* 14, 4 (2020), 5143–5147.
[54] Zhongkai Hao, Chengqiang Lu, Zhenya Huang, Hao Wang, Zheyuan Hu, Qi Liu, Enhong Chen, and Cheekong Lee. 2020. ASGN: An active semi-supervised graph neural network for molecular property prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 731–752. https://doi.org/10.1145/3394486.3403117
[55] Wenjia He, Yi Jiang, Junru Jin, Zhongshen Li, Jiaojiao Zhao, Balachandran Manavalan, Ran Su, Xin Gao, and Leyi Wei. 2022. Accelerating bioactive peptide discovery via mutual information-based meta-learning. *Briefings in Bioinformatics* 23, 1 (2022), bbab499.
[56] Yi He, Gia G. Maisuradze, Yanping Yin, Khatuna Kachlishvili, S. Rackovsky, and Harold A. Scheraga. 2017. Sequence-, structure-, and dynamics-based comparisons of structurally homologous CheY-like proteins. *Proceedings of the National Academy of Sciences* 114, 7 (2017), 1578–1583.

[57] Jonas S. Heitmann, Tatjana Bilich, Claudia Tandler, Annika Nelde, Yacine Maringer, Maddalena Marconato, Julia Reusch, Simon Jäger, Monika Denk, Marion Richter, et al. 2022. A COVID-19 peptide vaccine for the induction of SARS-CoV-2 T cell immunity. *Nature* 601, 7894 (2022), 617–622.

[58] Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, and Radford M. Neal. 1995. The "wake-sleep" algorithm for unsupervised neural networks. *Science* 268, 5214 (1995), 1158–1161.

[59] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 6840–6851.

[60] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[61] Dichao Hu. 2019. An introductory survey on attention mechanisms in NLP problems. In *Proceedings of SAI Intelligent Systems Conference*. Springer, 432–448.

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

[63] Jun Huan, Deepak Bandyopadhyay, Wei Wang, Jack Snoeyink, Jan Prins, and Alexander Tropsha. 2005. Comparing graph representations of protein structure for mining family-specific residue-based packing motifs. *Journal of Computational Biology* 12, 6 (2005), 657–671.

[64] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. 2019. Generative models for graph-based protein design. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/f3a4ff4839c56a5f460c88cce3666a2b-Paper.pdf

[65] Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, et al.. 2022. Biological sequence design with GFlowNets. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 9786–9801. https://proceedings.mlr.press/v162/jain22a.html

[66] Shipra Jain, Srijanee Gupta, Sumeet Patiyal, and Gajendra P. S. Raghava. 2024. THPdb2: Compilation of FDA approved therapeutic peptides and proteins. *Drug Discovery Today* (2024), 104047.

[67] Jhih-Hua Jhong, Lantian Yao, Yuxuan Pang, Zhongyan Li, Chia-Ru Chung, Rulan Wang, Shangfu Li, Wenshuo Li, Mengqi Luo, Renfei Ma, et al. 2022. dbAMP 2.0: Updated resource for antimicrobial peptides with an enhanced scanning method for genomic and proteomic data. *Nucleic Acids Research* 50, D1 (2022), D460–D470.

[68] Mingjian Jiang, Zhen Li, Shugang Zhang, Shuang Wang, Xiaofeng Wang, Qing Yuan, and Zhiqiang Wei. 2020. Drug–target affinity prediction using graph neural network and contact maps. *RSC Advances* 10, 35 (2020), 20701–20712.

[69] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*. PMLR, 2323–2332.

[70] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 7873 (2021), 583–589.

[71] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR'13)*.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, Vol. 25.

[73] Sneh Lata, Manoj Bhasin, and Gajendra P. S. Raghava. 2009. MHCBN 4.0: A database of MHC/TAP binding peptides and T-cell epitopes. *BMC Research Notes* 2 (2009), 1–6.

[74] Benjamin Leader, Quentin J. Baca, and David E. Golan. 2008. Protein therapeutics: A summary and pharmacological classification. *Nature Reviews Drug Discovery* 7, 1 (2008), 21–39.

[75] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[76] Andy Chi-Lung Lee, Janelle Louise Harris, Kum Kum Khanna, and Ji-Hong Hong. 2019. A comprehensive review on current advances in peptide drug development and design. *International Journal of Molecular Sciences* 20, 10 (2019), 2383.

[77] Byungjo Lee, Min Kyoung Shin, In-Wook Hwang, Junghyun Jung, Yu Jeong Shim, Go Woon Kim, Seung Tae Kim, Wonhee Jang, and Jung-Suk Sung. 2021. A deep learning approach with data augmentation to predict novel spider neurotoxic peptides. *International Journal of Molecular Sciences* 22, 22 (2021), 12291.

[78] Ernest Y. Lee, Benjamin M. Fulan, Gerard C. L. Wong, and Andrew L. Ferguson. 2016. Mapping membrane activity in undiscovered peptide sequence space using machine learning. *Proceedings of the National Academy of Sciences* 113, 48 (2016), 13588–13593.

[79] Yipin Lei, Shuya Li, Ziyi Liu, Fangping Wan, Tingzhong Tian, Shao Li, Dan Zhao, and Jianyang Zeng. 2021. A deep-learning framework for multi-level peptide–protein interaction prediction. *Nature Communications* 12, 1 (2021), 1–10.
[80] Guangyuan Li, Balaji Iyer, V. B. Surya Prasath, Yizhao Ni, and Nathan Salomonis. 2021. DeepImmuno: Deep learning-empowered prediction and generation of immunogenic peptides for T-cell immunity. *Briefings in Bioinformatics* 22, 6 (2021), bbab160.
[81] Jiahan Li, Chaoran Cheng, Zuofan Wu, Ruihan Guo, Shitong Luo, Zhizhou Ren, Jian Peng, and Jianzhu Ma. 2024. Full-atom peptide design based on multi-modal flow matching. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*, Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). 27615–27640.
[82] Ze-Rong Li, Hong Huang Lin, L. Y. Han, L Jiang, X Chen, and Yu Zong Chen. 2006. PROFEAT: A web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Research* 34, suppl_2 (2006), W32–W37.
[83] Po-Yu Liang and Jun Bai. 2024. E (3)-invaraint diffusion model for pocket-aware peptide generation. *arXiv preprint arXiv:2410.21335* (2024).
[84] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. 2023. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 379, 6637 (2023), 1123–1130.
[85] Losee L. Ling, Tanja Schneider, Aaron J. Peoples, Amy L. Spoering, Ina Engels, Brian P. Conlon, Anna Mueller, Till F. Schäberle, Dallas E. Hughes, Slava Epstein, et al. 2015. A new antibiotic kills pathogens without detectable resistance. *Nature* 517, 7535 (2015), 455–459.
[86] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York, New York, USA) *(IJCAI'16)*. AAAI Press, 2873–2879.
[87] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. 2022. Pre-training molecular graph representation with 3D geometry. In *International Conference on Learning Representations (ICLR)*.
[88] Yuansheng Liu, Zhenran Zhou, Xiaofeng Cao, Dongsheng Cao, and Xiangxiang Zeng. 2024. Effective drug-target affinity prediction via generative active learning. *Information Sciences* (2024), 121135.
[89] Balachandran Manavalan, Tae H. Shin, Myeong O. Kim, and Gwang Lee. 2018. AIPpred: Sequence-based prediction of anti-inflammatory peptides using random forest. *Frontiers in Pharmacology* 9 (2018), 276.
[90] Susan Marqus, Elena Pirogova, and Terrence J. Piva. 2017. Evaluation of the use of therapeutic peptides for cancer treatment. *Journal of Biomedical Science* 24, 1 (2017), 1–15.
[91] Jesse G. Meyer. 2021. Deep learning neural network tools for proteomics. *Cell Reports Methods* 1, 2 (2021), 100003.
[92] Uros Midic, Christopher J. Oldfield, A Keith Dunker, Zoran Obradovic, and Vladimir N. Uversky. 2009. Protein disorder in the human diseasome: Unfoldomics of human genetic diseases. *BMC Genomics* 10, 1 (2009), 1–24.
[93] M. Mirza and S. Osindero. 2014. Conditional generative adversarial nets. *Computer Science* (2014), 2672–2680.
[94] Somesh Mohapatra, Joyce An, and Rafael Gómez-Bombarelli. 2022. Chemistry-informed macromolecule graph representation for similarity computation, unsupervised and supervised learning. *Machine Learning: Science and Technology* 3, 1 (2022), 015028.
[95] Neeloffer Mookherjee, Marilyn A. Anderson, Henk P. Haagsman, and Donald J. Davidson. 2020. Antimicrobial host defence peptides: Functions and clinical potential. *Nature Reviews Drug Discovery* 19, 5 (2020), 311–332.
[96] Alex T. Müller, Gisela Gabernet, Jan A. Hiss, and Gisbert Schneider. 2017. modlAMP: Python for antimicrobial peptides. *Bioinformatics* 33, 17 (2017), 2753–2755.
[97] Alex T. Müller, Jan A. Hiss, and Gisbert Schneider. 2018. Recurrent neural network model for constructive peptide design. *Journal of Chemical Information and Modeling* 58, 2 (2018), 472–479.
[98] Markus Muttenthaler, Glenn F. King, David J. Adams, and Paul F. Alewood. 2021. Trends in peptide drug discovery. *Nature Reviews Drug Discovery* 20, 4 (2021), 309–325.
[99] Deepesh Nagarajan, Tushar Nagarajan, Natasha Roy, Omkar Kulkarni, Sathyabaarathi Ravichandran, Madhulika Mishra, Dipshikha Chakravortty, and Nagasuma Chandra. 2018. Computational antimicrobial peptide design and evaluation against multidrug-resistant clinical isolates of bacteria. *Journal of Biological Chemistry* 293, 10 (2018), 3492–3509.
[100] Nagasundaram Nagarajan, Edward K. Y. Yapp, Nguyen Quoc Khanh Le, Balu Kamaraj, Abeer Mohammed Al-Subaie, and Hui-Yuan Yeh. 2019. Application of computational biology and artificial intelligence technologies in cancer precision drug discovery. *BioMed Research International* 2019 (2019).
[101] Nina Notman. 2022. Teaching old drugs new tricks to treat COVID-19. *Nature Synthesis* 1 (2022), 2–5.
[102] Mario Novković, Juraj Simunić, Viktor Bojović, Alessandro Tossi, and Davor Juretić. 2012. DADP: The database of anuran defense peptides. *Bioinformatics* 28, 10 (2012), 1406–1407.

[103] Daniel Osorio, Paola Rondón-Villarreal, and Rodrigo Torres. 2015. Peptides: A package for data mining of antimicrobial peptides. *Small* 12 (2015), 44–444.
[104] Timothy J. O'Donnell, Alex Rubinsteyn, and Uri Laserson. 2020. MHCflurry 2.0: Improved pan-allele prediction of MHC class I-presented peptides by incorporating antigen processing. *Cell Systems* 11, 1 (2020), 42–48.
[105] Robin Pearce, Yang Li, Gilbert S. Omenn, and Yang Zhang. 2022. Fast and accurate Ab Initio Protein structure prediction using deep learning potentials. *PLOS Computational Biology* 18, 9 (2022), e1010539.
[106] Stefano P. Piotto, Lucia Sessa, Simona Concilio, and Pio Iannelli. 2012. YADAMP: Yet another database of antimicrobial peptides. *International Journal of Antimicrobial Agents* 39, 4 (2012), 346–351.
[107] Malak Pirtskhalava, Anthony A. Amstrong, Maia Grigolava, Mindia Chubinidze, Evgenia Alimbarashvili, Boris Vishnepolsky, Andrei Gabrielian, Alex Rosenthal, Darrell E. Hurt, and Michael Tartakovsky. 2021. DBAASP v3: Database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics. *Nucleic Acids Research* 49, D1 (2021), D288–D297.
[108] Ying Qi, Xuanpei Jiang, Yongquan Jiang, Yan Yang, Qiangwei Zhang, and Yuan Tian. 2024. Antimicrobial peptide sequence generation based on conditional diffusion model. In *Proceedings of the 2024 16th International Conference on Bioinformatics and Biomedical Technology*. 102–107.
[109] Abid Qureshi, Nishant Thakur, Himani Tandon, and Manoj Kumar. 2014. AVPdb: A database of experimentally validated antiviral peptides targeting medically important viruses. *Nucleic Acids Research* 42, D1 (2014), D1147–D1153.
[110] A. Radford, L. Metz, and S. Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *Computer Science* (2015).
[111] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).
[112] Vishva Saravanan Ramasubramanian, Soham Choudhuri, and Bhaswar Ghosh. 2024. A hybrid diffusion model for stable, affinity-driven, receptor-aware peptide generation. *bioRxiv* (2024), 2024–03.
[113] H.-G. Rammensee, Jutta Bachmann, Niels Philipp Nikolaus Emmerich, Oskar Alexander Bachor, and Sanja Stevanović. 1999. SYFPEITHI: Database for MHC ligands and peptide motifs. *Immunogenetics* 50 (1999), 213–219.
[114] Pedro A. Reche, Hong Zhang, John-Paul Glutting, and Ellis L. Reinherz. 2005. EPIMHC: A curated database of MHC-binding peptides for customized computational vaccinology. *Bioinformatics* 21, 9 (2005), 2140–2141.
[115] Nicolas Renaud, Cunliang Geng, Sonja Georgievska, Francesco Ambrosetti, Lars Ridder, Dario F. Marzella, Manon F. Réau, Alexandre M. J. J. Bonvin, and Li C. Xue. 2021. DeepRank: A deep learning framework for data mining 3D protein-protein interfaces. *Nature Communications* 12, 1 (2021), 1–8.
[116] Allison Rossetto and Wenjin Zhou. 2020. GANDALF: Peptide generation for drug design using sequential and structural generative adversarial networks. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 1–10.
[117] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1985. *Learning Internal Representations by Error Propagation*. Technical Report. California Univ. San Diego La Jolla Inst. for Cognitive Science.
[118] Carly K. Schissel, Somesh Mohapatra, Justin M. Wolfe, Colin M. Fadzen, Kamela Bellovoda, Chia-Ling Wu, Jenna A. Wood, Annika B. Malmberg, Andrei Loas, Rafael Gómez-Bombarelli, et al. 2021. Deep learning to design nuclear-targeting abiotic miniproteins. *Nature Chemistry* 13, 10 (2021), 992–1000.
[119] Daniel Schwalbe-Koda and Rafael Gómez-Bombarelli. 2020. Generative models for automatic chemical design. In *Machine Learning Meets Quantum Physics*. Springer, 445–467.
[120] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. 2020. GraphAF: A flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations (ICLR)*.
[121] Guobang Shi, Xinyue Kang, Fanyi Dong, Yanchao Liu, Ning Zhu, Yuxuan Hu, Hanmei Xu, Xingzhen Lao, and Heng Zheng. 2022. DRAMP 3.0: An enhanced comprehensive data repository of antimicrobial peptides. *Nucleic Acids Research* 50, D1 (2022), D488–D496.
[122] Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6, 1 (2019), 1–48.
[123] Timur Shtatland, Daniel Guettler, Misha Kossodo, Misha Pivovarov, and Ralph Weissleder. 2007. PepBank-a database of peptides based on sequence text mining and public peptide data sources. *BMC Bioinformatics* 8 (2007), 1–10.
[124] Sandeep Singh, Kumardeep Chaudhary, Sandeep Kumar Dhanda, Sherry Bhalla, Salman Sadullah Usmani, Ankur Gautam, Abhishek Tuknait, Piyush Agrawal, Deepika Mathur, and Gajendra P. S. Raghava. 2016. SATPdb: A database of structurally annotated therapeutic peptides. *Nucleic Acids Research* 44, D1 (2016), D1119–D1126.
[125] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, Vol. 30.
[126] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, Vol. 28.

[127] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).
[128] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=PxTIG12RRHS
[129] Shraddha Surana, Pooja Arora, Divye Singh, Deepti Sahasrabuddhe, and Jayaraman Valadi. 2023. PandoraGAN: Generating antiviral peptides using generative adversarial network. *SN Computer Science* 4, 5 (2023), 607.
[130] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, Vol. 12.
[131] Paulina Szymczak, Marcin Możejko, Tomasz Grzegorzek, Radosław Jurczak, Marta Bauer, Damian Neubauer, Karol Sikora, Michał Michalski, Jacek Sroka, Piotr Setny, et al. 2023. Discovering highly potent antimicrobial peptides with deep generative model HydrAMP. *Nature Communications* 14, 1 (2023), 1453.
[132] Lorillee Tallorin, JiaLei Wang, Woojoo E. Kim, Swagat Sahu, Nicolas M. Kosa, Pu Yang, Matthew Thompson, Michael K. Gilson, Peter I. Frazier, Michael D. Burkart, et al. 2018. Discovering de novo peptide substrates for enzymes using machine learning. *Nature Communications* 9, 1 (2018), 1–10.
[133] Peng Tan, Huiyang Fu, and Xi Ma. 2021. Design, optimization, and nanotechnology of antimicrobial peptides: From exploration to applications. *Nano Today* 39 (2021), 101229.
[134] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *ACM Computing Surveys (CSUR)* (2020).
[135] Felix Teufel, José Juan Almagro Armenteros, Alexander Rosenberg Johansen, Magnús Halldór Gíslason, Silas Irby Pihl, Konstantinos D. Tsirigos, Ole Winther, Søren Brunak, Gunnar von Heijne, and Henrik Nielsen. 2022. SignalP 6.0 predicts all five types of signal peptides using protein language models. *Nature Biotechnology* (2022), 1–3.
[136] Nishant Thakur, Abid Qureshi, and Manoj Kumar. 2012. AVPpred: Collection and prediction of highly effective antiviral peptides. *Nucleic Acids Research* 40, W1 (2012), W199–W204.
[137] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2017. Wasserstein auto-encoders. In *International Conference on Learning Representations (ICLR)*.
[138] Iva Trenevska, Demin Li, and Alison H. Banham. 2017. Therapeutic antibodies against intracellular tumor antigens. *Frontiers in Immunology* 8 (2017), 1001.
[139] Andrejs Tucs, Duy Phuoc Tran, Akiko Yumoto, Yoshihiro Ito, Takanori Uzawa, and Koji Tsuda. 2020. Generating ampicillin-level antimicrobial peptides with activity-aware generative adversarial networks. *ACS Omega* 5, 36 (2020), 22847–22851.
[140] Atul Tyagi, Abhishek Tuknait, Priya Anand, Sudheer Gupta, Minakshi Sharma, Deepika Mathur, Anshika Joshi, Sandeep Singh, Ankur Gautam, and Gajendra P. S. Raghava. 2015. CancerPPD: A database of anticancer peptides and proteins. *Nucleic Acids Research* 43, D1 (2015), D837–D843.
[141] Salman Sadullah Usmani, Gursimran Bedi, Jesse S. Samuel, Sandeep Singh, Sourav Kalra, Pawan Kumar, Anjuman Arora Ahuja, Meenu Sharma, Ankur Gautam, and Gajendra P. S. Raghava. 2017. THPdb: Database of FDA-approved peptide and protein therapeutics. *PloS One* 12, 7 (2017), e0181748.
[142] Colin M. Van Oort, Jonathon B. Ferrell, Jacob M. Remington, Safwan Wshah, and Jianing Li. 2021. AMPGAN v2: Machine learning-guided design of antimicrobial peptides. *Journal of Chemical Information and Modeling* 61, 5 (2021), 2198–2207.
[143] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30.
[144] Randi Vita, Swapnil Mahajan, James A. Overton, Sandeep Kumar Dhanda, Sheridan Martini, Jason R. Cantrell, Daniel K. Wheeler, Alessandro Sette, and Bjoern Peters. 2019. The Immune Epitope Database (IEDB): 2018 update. *Nucleic Acids Research* 47, D1 (2019), D339–D343.
[145] Randi Vita, Laura Zarebski, Jason A. Greenbaum, Hussein Emami, Ilka Hoof, Nima Salimi, Rohini Damle, Alessandro Sette, and Bjoern Peters. 2010. The Immune Epitope Database 2.0. *Nucleic Acids Research* 38, suppl_1 (2010), D854–D862.
[146] Faiza Hanif Waghu, Ram Shankar Barai, Pratima Gurung, and Susan Idicula-Thomas. 2016. CAMPR3: A database on sequences, structures and signatures of antimicrobial peptides. *Nucleic Acids Research* 44, D1 (2016), D1094–D1097.
[147] Faiza Hanif Waghu, Lijin Gopi, Ram Shankar Barai, Pranay Ramteke, Bilal Nizami, and Susan Idicula-Thomas. 2014. CAMP: Collection of sequences and structures of antimicrobial peptides. *Nucleic Acids Research* 42, D1 (2014), D1154–D1158.
[148] Christina Wang, Sam Garlick, and Mire Zloh. 2021. Deep learning for novel antimicrobial peptide design. *Biomolecules* 11, 3 (2021), 471.
[149] Guangshun Wang, Xia Li, and Zhe Wang. 2016. APD3: The antimicrobial peptide database as a tool for research and education. *Nucleic Acids Research* 44, D1 (2016), D1087–D1093.

[150] Xue-Fei Wang, Jing-Ya Tang, Jing Sun, Sonam Dorje, Tian-Qi Sun, Bo Peng, Xu-Wo Ji, Zhe Li, Xian-En Zhang, and Dian-Bing Wang. 2024. ProT-Diff: A modularized and efficient strategy for de novo generation of antimicrobial peptide sequences by integrating protein language and diffusion models. *Advanced Science* (2024), 2406305.

[151] Yongkang Wang, Xuan Liu, Feng Huang, Zhankun Xiong, and Wen Zhang. 2024. A multi-modal contrastive diffusion model for therapeutic peptide generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 3–11.

[152] Lesong Wei, Xiucai Ye, Yuyang Xue, Tetsuya Sakurai, and Leyi Wei. 2021. ATSE: A peptide toxicity predictor by exploiting structural and evolutionary information based on graph neural network and attention mechanism. *Briefings in Bioinformatics* 22, 5 (2021), bbab041.

[153] Leyi Wei, Chen Zhou, Huangrong Chen, Jiangning Song, and Ran Su. 2018. ACPred-FL: A sequence-based predictor using effective feature representation to improve the prediction of anti-cancer peptides. *Bioinformatics* 34, 23 (2018), 4007–4016.

[154] Zeyu Wen, Jiahua He, Huanyu Tao, and Sheng-You Huang. 2019. PepBDB: A comprehensive structural database of biological peptide–protein interactions. *Bioinformatics* 35, 1 (2019), 175–177.

[155] Fang Wu, Tinson Xu, Shuting Jin, Xiangru Tang, Zerui Xu, James Zou, and Brian Hie. 2024. D-Flow: Multi-modality flow matching for D-peptide design. *arXiv preprint arXiv:2411.10618* (2024).

[156] Zachary Wu, Kevin K. Yang, Michael J. Liszka, Alycia Lee, Alina Batzilla, David Wernick, David P. Weiner, and Frances H. Arnold. 2020. Signal peptides generated by attention-based neural networks. *ACS Synthetic Biology* 9, 8 (2020), 2154–2161.

[157] Xuezhi Xie, Pedro A. Valiente, Jisun Kim, and Philip M. Kim. 2024. HelixDiff, a score-based diffusion model for generating all-atom α-helical structures. *ACS Central Science* 10, 5 (2024), 1001–1011.

[158] Yaochen Xie, Zhao Xu, and Shuiwang Ji. 2022. Self-supervised representation learning via latent graph prediction. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 24460–24477. https://proceedings.mlr.press/v162/xie22e.html

[159] Guizi Ye, Hongyu Wu, Jinjiang Huang, Wei Wang, Kuikui Ge, Guodong Li, Jiang Zhong, and Qingshan Huang. 2020. LAMP2: A major update of the database linking antimicrobial peptides. *Database* 2020 (2020), baaa061.

[160] Jonathan W. Yewdell and Jack R. Bennink. 1999. Immunodominance in major histocompatibility complex class I-restricted T lymphocyte responses. *Annual Review of Immunology* 17 (1999), 51.

[161] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[162] Zhaoning Yu and Hongyang Gao. 2022. Molecular representation learning via heterogeneous motif graph neural networks. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 25581–25594. https://proceedings.mlr.press/v162/yu22a.html

[163] Xiangxiang Zeng, Xinqi Tu, Yuansheng Liu, Xiangzheng Fu, and Yansen Su. 2022. Toward better drug discovery with knowledge graph. *Current Opinion in Structural Biology* 72 (2022), 114–126.

[164] Haiping Zhang, Konda Mani Saravanan, Yanjie Wei, Yang Jiao, Yang Yang, Yi Pan, Xuli Wu, and John Z. H. Zhang. 2023. Deep learning-based bioactive therapeutic peptide generation and screening. *Journal of Chemical Information and Modeling* 63, 3 (2023), 835–845.

[165] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. 2017. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 5907–5915.

[166] Lei Zhang, Qixin Leng, and A. James Mixson. 2005. Alteration in the IL-2 signal peptide affects secretion of proteins in vitro and in vivo. *The Journal of Gene Medicine: A Cross-disciplinary Journal for Research on the Science of Gene Transfer and Its Clinical Applications* 7, 3 (2005), 354–365.

[167] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Chee-Kong Lee. 2021. Motif-based graph self-supervised learning for molecular property prediction. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 15870–15882.

[168] Alex Zhavoronkov, Yan A. Ivanenkov, Alex Aliper, Mark S. Veselov, Vladimir A. Aladinskiy, Anastasiya V. Aladinskaya, Victor A. Terentiev, Daniil A. Polykovskiy, Maksim D. Kuznetsov, Arip Asadulaev, et al. 2019. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature Biotechnology* 37, 9 (2019), 1038–1040.

Received 21 December 2022; revised 26 November 2024; accepted 16 January 2025