Epitope Generation for Peptide-based Cancer Vaccine using Goal-directed Wasserstein Generative Adversarial Network with Gradient Penalty

Anonymous authors
Paper under double-blind review

Abstract

We introduce a novel goal-directed Wasserstein Generative Adversarial Network with Gradient Penalty (GD-WGAN-GP) for training a generator capable of producing peptide sequences with high predicted immunogenicity and strong binding affinity to the human leukocyte antigen HLA-A*0201, thereby eliciting cytotoxic T-cell immune responses. The proposed GD-WGAN-GP incorporates a critic network to guide the generator in producing peptides with a strong binding affinity similar to those in the training set and a reward network to steer the generator toward producing sequences with high predicted immunogenicity. To avoid the generator prioritizing the objective of the critic at the expense of immunogenicity, we introduce a scaling factor to balance the influence of the reward in the loss of the generator. To reduce peptide repetition, we integrate the reward into the loss of the generator using two approaches: a switching mechanism that excludes the reward term when duplicated peptides are present in a batch, and otherwise multiplies it by a γ_{max} parameter to control the reward's contribution, and (2) a repetition penalty from ORGAN, where each reward is divided by the number of occurrences of its corresponding peptide within the batch. Experiments on bladder cancer epitope sequences demonstrate that GD-WGAN-GP with the switching mechanism enables a tunable trade-off between the number of unique peptides and the average immunogenicity score via varying γ_{max}. Furthermore, the generator trained by the GD-WGAN-GP with the ORGAN's repetition penalty achieves an optimal balance of uniqueness and immunogenicity. Across multiple datasets, GD-WGAN-GP outperforms existing methods by effectively reducing peptide redundancy while preserving high immunogenicity scores and strong binding affinity. The Python codes are provided at: `https://github.com/AnnonymousForPapers/GP-WGAN-GP_with_switch_and_ORGAN_penalty`.

1 Introduction

Tumor-specific antigens have been utilized in cancer vaccines, a form of cancer immunotherapy, to stimulate the production of tumor-specific T cells (Stevanovic, 2002). These antigens are encoded in the genome and are not present in normal cells, making them representative of aberrant proteins (Gubin et al., 2015). An epitope can be defined as a segment of an antigen that is generated through antigen processing (Sidney et al., 2020). The activation of the tumor-specific T-cell response occurs when epitopes from class I or class II human leukocyte antigen (HLA) molecules are presented to T cell receptors (TCRs), which recognize antigens in the form of peptides (Mohme & Neidert, 2020). The HLA class I molecule serves as the TCR ligand for CD8+ cytotoxic T lymphocytes (CTLs) and binds peptides consisting of 8-11 amino acids. On the other hand, the TCR ligand for CD4+ helper T cells is the HLA class II molecule, and the bound peptides typically consist of 15 amino acids (Sim & Sun, 2022).

Peptide-based cancer vaccines refer to cancer vaccines that employ polypeptides comprised of known or predicted tumor antigen epitopes (Liu et al., 2022). Peptide-based vaccines are widely utilized in cancer vaccination practices and are designed to activate a tumor-specific cytotoxic T lymphocyte (CTL) response (Butterfield, 2015). These vaccines consist of multiple epitopes, typically ranging from 8 to 11 amino acids

in length (Liu et al., 2021). Moreover, these peptides are commonly combined with a carrier protein to facilitate their recognition and processing by antigen-presenting cells (APCs), thereby triggering an immune response that involves CTLs (Abd-Aziz & Poh, 2022).

To develop an effective peptide-based vaccine, it is crucial to ensure that the epitopes are recognizable by T cells, highly prevalent, and exclusive to tumor cells (Nelde et al., 2021). The process of identifying immunogenic epitopes begins by obtaining samples of both cancerous and normal cells through biopsy, followed by comparing their DNA sequences using techniques like Whole Exome Sequencing (WES) or Whole Genome Sequencing (WGS) to identify tumor-specific mutations (Richard et al., 2022). Somatic variant callers can be employed to detect single nucleotide variants from the WES and WGS data, and subsequently, peptides containing the mutated regions are extracted using sliding windows from the varied protein sequence (Richters et al., 2019). Epitope candidates can be identified using computational prediction tools, an approach that offers significant advantages in terms of time and cost when compared to traditional methods such as mass spectrometry techniques (Lemmel & Stevanović, 2003; Chen et al., 2020; Hensen et al., 2022; Parvizpour et al., 2020).

Immunogenicity refers to the capacity of a substance to initiate an immune response, and the binding of a peptide to HLA molecules is essential for epitope immunogenicity (Antunes et al., 2018). (Wu et al., 2019) devised DeepHLApan, a recurrent neural network-based method that integrates binding affinity and immunogenicity details of peptide-HLA complexes to predict CD8+ T-cell epitopes. (Li et al., 2021), on the other hand, utilized a convolutional neural network (CNN)-based method called DeepImmuno, where they utilized linear peptides of 9 to 10 amino acids and 4-digit class I HLA alleles as input to forecast the immunogenicity score of peptide-HLA pairs for T-cell immune responses. (Diao et al., 2022) introduced Seq2Neo, a CNN-based pipeline that leverages the binding affinity and transporters associated with antigen processing (TAP) transport efficiency of a given peptide-HLA pair to improve the prediction of immunogenicity scores for T-cell immune responses.

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are deep learning models with a generative model trying to learn and capture the distribution of training data, thereby being capable of synthesizing samples from the learned distribution (Gonog & Zhou, 2019; Creswell et al., 2018). GAN algorithms have found application in generating novel protein and peptide structures for use in drug screening and the discovery stage (Lin et al., 2022). Dutta (2022) developed a GAN equipped with a graph CNN that predicts solubility and toxicity from molecular descriptors, guiding the generator network to produce new small molecules with desired drug properties. Putin et al. (2018) proposed a GAN architecture known as "RANC," which incorporates reinforcement learning to generate chemically diverse structures with desired features, with a focus on maintaining similar lengths of the SMILES string as their training data. Karimi et al. (2020) developed a semi-supervised, guided, conditional, Wasserstein generative adversarial network capable of generating proteins with desired structure folds, while incorporating greater sequence diversity and novelty compared to conditional variational auto-encoder designs. Li et al. (2021) employed a generator trained with a Wasserstein generative adversarial network with gradient penalty (WGAN-GP) to generate immunogenic epitopes binding to HLA-A*0201 and demonstrated that their generated peptides exhibited features and amino acid sequences similar to the epitopes in their training dataset.

In this study, we introduce a novel training framework, termed goal-directed Wasserstein Generative Adversarial Network with Gradient Penalty (GD-WGAN-GP), for generating immunogenic peptide sequences. Built upon the WGAN-GP architecture, known for its enhanced training stability (Gulrajani et al., 2017), our method integrates both a critic network and a reward network to guide the generator. The critic is trained on bladder cancer-specific peptides that bind to HLA-A*0201, assigning higher values to sequences similar to those in the training data. The reward network, implemented as a convolutional neural network and trained using immunogenicity scores predicted by DeepImmuno (Li et al., 2021), directs the generator toward producing peptides with high immunogenicity. Its output is scaled by a factor S_{scale} to balance the influence of the reward and the critic in the loss of the generator. To reduce repetition among generated sequences, we apply two strategies: a switching mechanism that excludes the reward term when duplicated peptides are present in a batch, and otherwise multiplies it by a γ_{max} parameter to control the reward's contribution to the loss of the generator; and a repetition penalty adapted from ORGAN (Guimaraes et al., 2018), where each reward is divided by the number of occurrences of its corresponding peptide within the

batch. The proposed GD-WGAN-GP with the switching mechanism enables users to control the trade-off between generating peptides with high predicted immunogenicity at the cost of increased repetition, or achieving lower repetition rates with reduced immunogenicity. In both cases, the generated peptides maintain strong binding affinity to HLA-A*0201, a characteristic shared with peptides in the training dataset. The generator trained using the GD-WGAN-GP framework with ORGAN's repetition penalty (Guimaraes et al., 2018) demonstrates the ability to produce peptides with high predicted immunogenicity, low repetition rates, and strong binding affinity to HLA-A*0201. These improvements outperform existing approaches, such as WGAN-GP (Li et al., 2021) and MolGAN (Cao & Kipf, 2022), highlight the potential of our method for the design of immunogenic epitope candidates in cancer immunotherapy. An introduction to our method is shown in Fig. 1.

2 Methods

2.1 Datasets

The tumor-specific neoantigen database (TSNAdb) (Wu et al., 2018) offers 6234 bladder cancer neoepitopes, each featuring a single amino acid mutation, predicted to bind with HLA-A*0201 and having predicted binding affinity IC_{50} <500nM. These epitopes, along with their corresponding HLA molecules, were predicted using NetMHCpan v4.0 (Jurtz et al., 2017) which assesses the binding affinity of potential epitopes within a protein sequence based on mass spectrometry eluted ligands and half-maximal inhibition (IC50) scores, with a threshold of IC50 $<$ 500nM. It is important to note that these epitopes have not been experimentally verified in other literature. We choose HLA-A*0201 as the binding target of the generated peptides since HLA-A*0201 binds with most of the bladder cancer neoepitopes in the TSNAdb (Wu et al., 2018).

During the data cleaning process, we specifically selected bladder cancer epitope sequences predicted to bind with HLA-A*0201 and having a length of either 9 or 10 amino acids. Our resulting training dataset comprised a total of 6234 epitopes. For standardization, we employed the same coding strategy as in (Li et al., 2021). This involved padding 9-mer peptides to become 10-mers by joining the first five amino acids and the last four amino acids with a placeholder "-". Thus, the peptide sequences in our study are represented as a sequence consisting of a placeholder "-" and the 20 amino acid types: Alanine (A), Arginine (R), Asparagine (N), Aspartate (D), Cysteine (C), Glutamine (Q), Glutamate (E), Glycine (G), Histidine (H), Isoleucine (I), Leucine (L), Lysine (K), Methionine (M), Phenylalanine (F), Proline (P), Serine (S), Threonine (T), Tryptophan (W), Tyrosine (Y), and Valine (V). The amino acids and placeholder were converted into one-hot encoded matrices (Jiang et al., 2022).

As of March 2023, the Immune Epitope Database (IEDB) (Vita et al., 2019) reports only 24 experimentally tested linear bladder cancer epitopes that bind with HLA class I molecules, as identified by Wang et al. (2020). In addition, the neoepitopes from the TSNAdb (Wu et al., 2018) are not predicted to be immunogenic. Thus, we aim to design a goal-directed generator to provide potential immunogenic peptide sequences and increase the pool of peptides worthy of experimental testing.

2.2 Goal-directed WGAN-GP

The designed GD-WGAN-GP architecture comprises a peptide sequence generator, a critic, an immunogenicity predictor, and a reward network. The generator and critic are CNN-based models from (Li et al., 2021). The critic is trained using the predicted bladder cancer peptides obtained from TSNAdb (Wu et al., 2018) with IC50 $<$ 500nM and the generated epitopes, thereby guiding the training of the generator. Before being input to the critic, the predicted bladder cancer epitopes are encoded as a one-hot matrix, as depicted in Figure 2 (**a**). Conversely, the output of the generator is a matrix of probabilities, which can be decoded as a peptide sequence, as shown in Figure 2 (**b**). The peptide sequence generator tries to generate sequences that are closely similar to the predicted bladder cancer epitopes in the training dataset.

The immunogenicity predictor utilized is the Deepimmuno-CNN from (Li et al., 2021). This predictor takes a 9-mer or 10-mer generated peptide sequence, along with a 46-mer HLA-A*0201 sequence, as input, predicting their respective immunogenicities. The reward network is a CNN-based model that outputs the

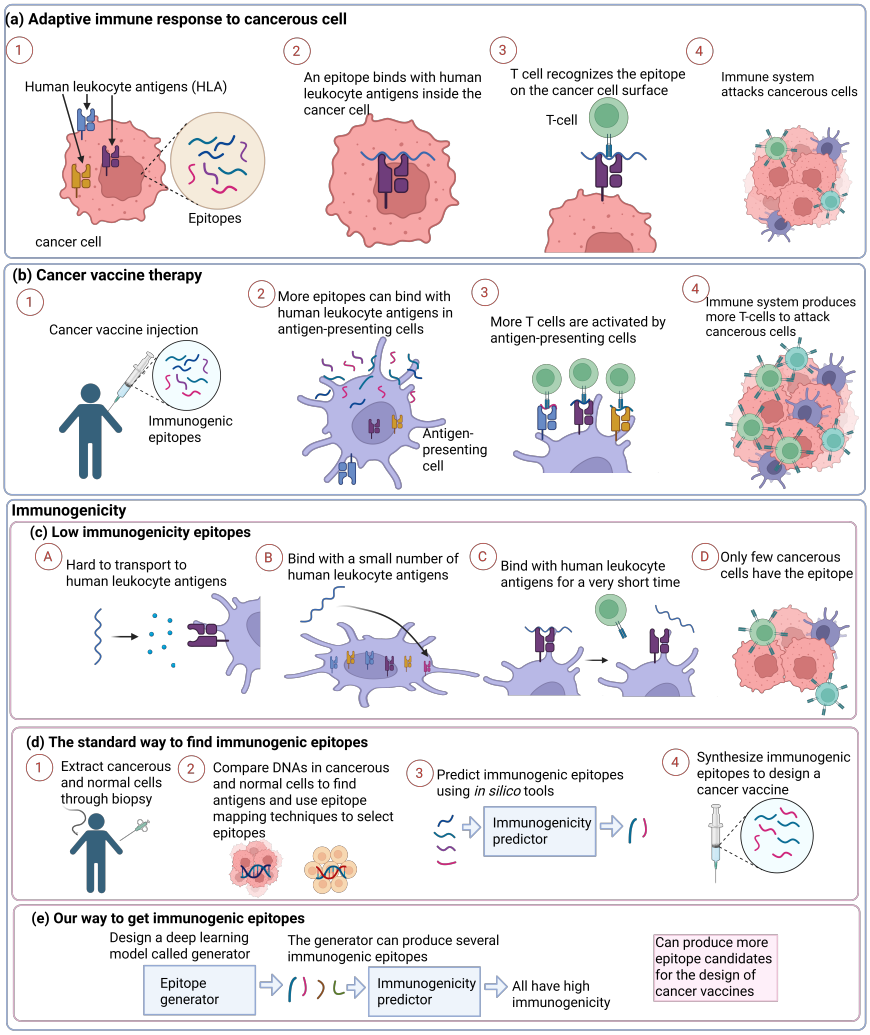

Figure 1: Introduction to peptide-based cancer vaccines and our proposed method. (a) Epitopes in cancerous cells may bind to HLA and the resulting HLA-peptide complex may be transferred to the cell surface to be recognized by T cells Zhao et al. (2021). (b) The polypeptides in peptide-based cancer vaccines are processed in APCs, the epitopes are bound to HLA, and the HLA-complexes present the epitopes on the cell surface to trigger T cell immune responses Bartnik et al. (2013). (c) The immunogenicity of a peptide can be influenced by its delivery system Li et al. (2014), the binding diversity between peptide and HLA Stephens et al. (2021), the stability of the HLA-peptide complex Van Der Burg et al. (1996), and the abundance and density of the antigen that is presented on a cell surface Purcell et al. (2007). (d) The standard way of finding the immunogenic peptides for the design of cancer vaccines Guo et al. (2018). (e) The designed generator can produce more epitope candidates with high immunogenicity to facilitate peptide-based cancer vaccines. Created with BioRender.com.

immunogenicity score. In contrast to the immunogenicity predictor, the reward network is trained during the training of the GD-WGAN-GP. The reward network takes the generated sequences or the sequences in the training dataset as input and the output of the immunogenicity predictor given the reward network's input as the target value. Throughout the training, the placeholder '-' remains in the peptide sequence and is only removed when utilizing the trained generator to produce peptide sequences, as depicted in the procedure outlined in Figure 2 (**b**), step **(3)**.

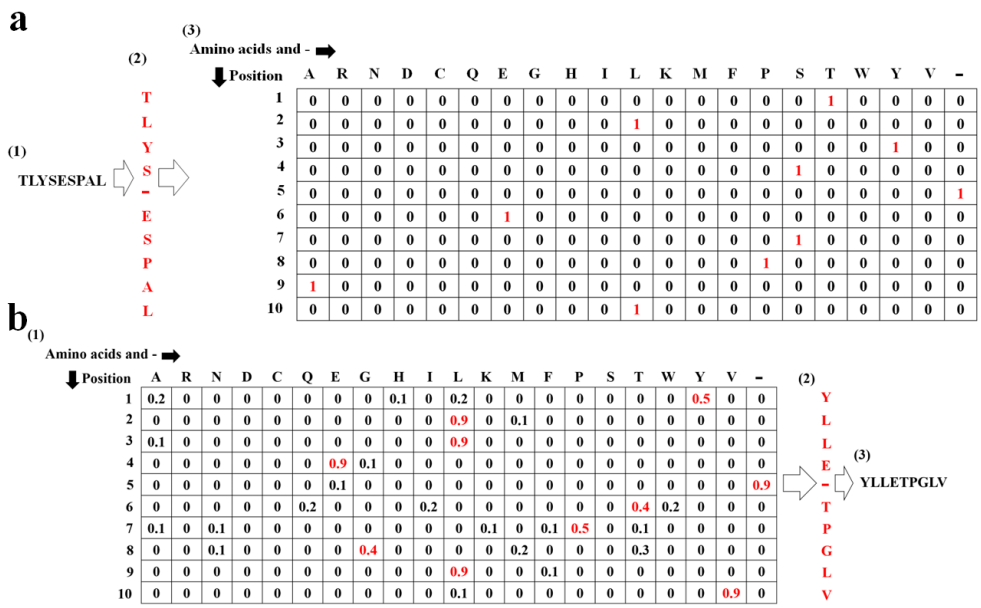

Figure 2: The encoding and the decoding methods in our goal-directed Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) adopted from Li et al. (2021). In (**a**), the peptide is encoded from a peptide sequence to a one-hot matrix x. **(1)** A 9-mer peptide sequence, such as TLYSESPAL, is prepared before input to the critic. **(2)** If the peptide sequence has only 9 amino acids, a placeholder '-' is inserted at the fifth position to extend the sequence to a length of 10. **(3)** The encoded peptide is represented as a one-hot matrix, where each row corresponds to a position in the peptide sequence, and columns represent amino acids along with a placeholder '-'. The element in the one-hot matrix is 1 if the corresponding position contains the amino acid or the placeholder at the corresponding column. For example, 'T' is at position 1, so the first row of the one-hot matrix will have 1 at the 17th column, corresponding to the character 'T', and 0 in the other columns. In (**b**), the peptide is decoded through the generated one-probability matrix $G_\theta(z)$. **(1)** The output of the generator $G_\theta(z)$ is a one-probability matrix, where each row represents the position of a peptide sequence, and columns represent amino acids along with a placeholder '-'. Each element in $G_\theta(z)$ is the probability that the position corresponding to the row contains the amino acid or the placeholder. **(2)** A peptide sequence is decoded by selecting the amino acid or the placeholder with the highest probability for each row. For instance, if 0.5 is the highest value in the first row, the first character in the decoded sequence is 'Y', corresponding to the column with the value 0.5 in the first row. **(3)** If the peptide sequence contains a placeholder, it is removed to form a peptide sequence shorter than 10.

Consider a generative network $G_\theta : \mathbb{R}^m \to \mathbb{R}^{p\times q}$ with a set of parameters $\theta \in \mathbb{R}$ and a regression model $P : \mathbb{R}^{p\times q} \to [0, 1]$. We aim to update the set of parameters θ such that the expected value $E_{G_\theta(z)\sim\mathbb{P}_G}[\hat{p}]$ is maximized, where $\hat{p} = P(G_\theta(z))$, $z \in \mathbb{R}^m$ is a random noise vector, and $\mathbb{P}_G$ is the generated data distribution.

The loss function of the critic D_ω is the same as in (Gulrajani et al., 2017), which is

$$\begin{aligned} L_\omega =& E_{\tilde{x}\sim\mathbb{P}_G}[D_\omega(\tilde{x})] - E_{x\sim\mathbb{P}_r}[D_\omega(x)] \\ &+ \lambda E_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D_\omega(\hat{x})\|_2 - 1)^2], \end{aligned} \tag{1}$$

where

$$\tilde{x} = G_\theta(z), \tag{2}$$

$$\hat{x} = \epsilon x + (1 - \epsilon)\tilde{x}, \tag{3}$$

$E_{x\sim\mathbb{P}}[f(x)]$ is the expected value of a function $f(x)$ with x sampled from the distribution $\mathbb{P}$, $\mathbb{P}_r$ and $\mathbb{P}_G$ are the training data distribution and the generated data distribution, respectively, $\mathbb{P}_{\hat{x}}$ is the distribution of the

data sampled from the training and generated distribution as defined in (3), $G_\theta(\cdot)$ and $D_\omega(\cdot)$ is the function of the generator and the critic in WGAN-GP, λ is the penalty coefficient, $\|\cdot\|_2$ is the L-2 norm, $\epsilon \in [0,1]$ is a random real number between zero and one, z is a random noise vector and each element is sampled from a normal distribution with zero mean and unit variance, θ is the weights in the generative network G_θ, and ω is the weights in the critic network D_ω.

2.2.1 Generator loss with repetition penalty

The first approach: switching between vanilla WGAN-GP and a combined loss with the output of the reward network The designed loss function of the generator G_θ with a switching mechanism is

$$L_\theta^{D1} = -E_{\tilde{x}\sim\mathbb{P}_G}[D_\omega(\tilde{x})] - \gamma S_{scale} E_{\tilde{x}\sim\mathbb{P}_G}[S_\phi(\tilde{x})], \tag{4}$$

where

$$\gamma = \begin{cases} \gamma_{max} & \text{if } n(A) = N_{batch} \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

$\gamma_{max} \in [0,1]$, A is a set composed of the generated one-probability matrix $G_\theta(z)$ in a batch, $n(A)$ is the cardinality of A (the number of elements in the set A), N_{batch} is the number of generated peptides in a batch,

$$S_{scale} = 1 + |E_{\tilde{x}\sim\mathbb{P}_G}[D_\omega(\tilde{x})] - E_{\tilde{x}\sim\mathbb{P}_G}[S_\phi(\tilde{x})]|, \tag{6}$$

$|\cdot|$ denotes as modulus, S_ϕ is the reward network, and ϕ is the weights in the reward network S_ϕ.

The loss function of the reward network S_ϕ is defined as

$$L_\phi = E_{\tilde{x}\sim\mathbb{P}_G}[(S_\phi(\tilde{x}) - P(\tilde{x}))^2] + E_{x\sim\mathbb{P}_r}[(S_\phi(x) - P(x))^2], \tag{7}$$

where $P(\cdot) \in [0,1]$ is the output of the immunogenicity predictor.

The variable S_{scale} in the loss function of the generator (4) ensures that the expected value of the critic output will not dominate the generator's loss function. The variable γ in the loss function of the generator (4) is used to force the generator to produce diverse peptide sequences in the training stage. If $\gamma = 1$, (4) becomes a weighted sum of the expected value of the critic output and the expected value of the reward network output given the generated one-probability matrices $G_\theta(z)$ as input. If $\gamma = 0$, the training of the GD-WGAN-GP is the vanilla WGAN-GP in Gulrajani et al. (2017). The architecture of GD-WGAN-GP using the first generator loss design is illustrated in Figure 3.

The second approach: dividing each reward in the batch by its number of occurrences (Guimaraes et al., 2018) The loss function of the generator G_θ with the repetition penalty from ORGAN (Guimaraes et al., 2018) is

$$L_\theta^{D2} = -E_{\tilde{x}\sim\mathbb{P}_G}[D_\omega(\tilde{x})] - S_{scale} E_{\tilde{x}\sim\mathbb{P}_G}[\hat{S}_\phi(\tilde{x})], \tag{8}$$

where $\hat{S}_\phi(\tilde{x})$ is the output of the reward network,$S_\phi(\tilde{x})$, divided by the number of the occurrences of the generated one-hot matrix, $\tilde{x}$, in a batch.

2.2.2 Training of the GD-WGAN-GP

In this paper, the batch size is 64 ($N_{batch} = 64$). $z \in \mathbb{R}^{128}$ is defined to be a random noise vector with 128 elements, and each element is sampled from a normal distribution with zero mean and unit variance. The generative network, the critic network, and the reward network network are defined as $G_\theta : \mathbb{R}^{128} \to \mathbb{R}^{10\times21}$, $D_\omega : \mathbb{R}^{10\times21} \to \mathbb{R}$, and $S_\phi : \mathbb{R}^{10\times21} \to \mathbb{R}$, respectively. For each epoch, the critic is trained 97 times, the generator is trained 10 times, and the reward network is trained 97 times. The architecture of the generator, the critic, the reward network, and the residual block in these three networks can be found in Table 3, Table 4, Table 5, and Table 6 in Appendix A, respectively. The algorithm of the GD-WGAN-GP with switching mechanism in (4) is shown in Algorithm 1. The same hyperparameters are used for the GD-WGAN-GP with the ORGAN's repetition penalty (Guimaraes et al., 2018) in (8).

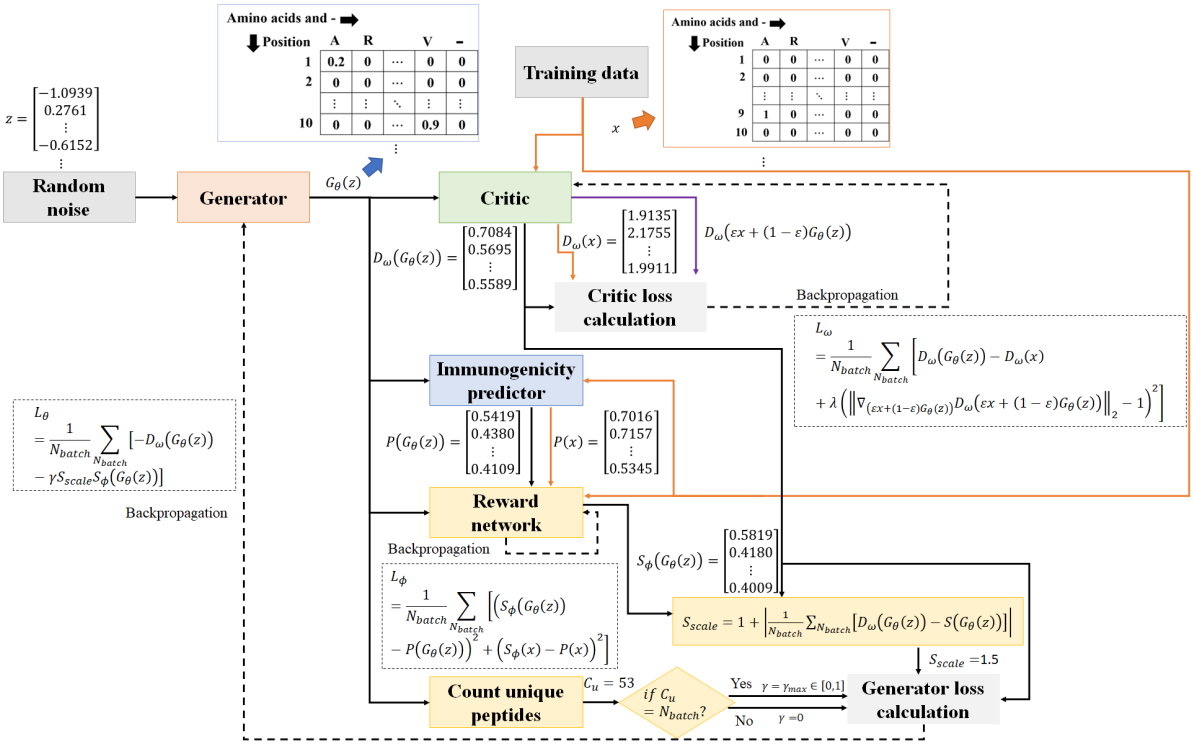

Figure 3: Training scheme of the designed GD-WGAN-GP with the switching mechanism in (4). The training of the GD-WGAN-GP involves incorporating an immunogenicity predictor from Li et al. (2021) to train a generator and a reward network. The objective of the reward network is to assist the generator in producing highly immunogenic peptides, while the critic aims to guide the generator in generating peptides with high similarity (though not identical) to those in the training dataset. The generator takes a set of N_{batch} random noise vectors z, each composed of 128 elements sampled from a zero-mean, unit-variance Gaussian distribution, as input. Implemented as a deep convolutional neural network, the generator produces N_{batch} one-probability matrices $G_\theta(z)$, which serve as input for the critic, immunogenicity predictor, and reward network. The critic's output is a vector of real values $D_\omega(G_\theta(z))$, the output of the immunogenicity predictor $P(G_\theta(z))$ is a vector of real values between 0 and 1, and the reward network's output is a vector with real values $S_\phi(G_\theta(z))$. The reward network is trained using the mean squared error between the output of the immunogenicity predictor and its own output, considering the generated one-probability matrices $G_\theta(z)$ and the one-hot encoded matrix of peptides in the training data x as input. The outputs of the reward network $S_\phi(G_\theta(z))$ and the output of the critic $D_\omega(G_\theta(z))$ are used to compute a scaling variable S_{scale}. The variable γ indicates whether the generated peptides are repeated in a batch ($\gamma = 0$ for repetitions, $\gamma = \gamma_{max}$ otherwise). The critic is trained using the standard critic loss in a WGAN-GP, while the generator is trained with a loss aiming to minimize the critic's output given the generated one-probability matrices $D_\omega(G_\theta(z))$and to minimize the reward network's output given the generated one-probability matrices $S_\phi(G_\theta(z))$ multiplied by the scaling variable S_{scale} and the variable γ.

3 Experiments

To validate the efficacy of our proposed generator training scheme, we adopted the same architecture for both the generative and critic networks as outlined in (Li et al., 2021). We then compared the peptides generated by the generator trained with and without our devised training scheme. For assessing immunogenicity, we utilized the CNN-based immunogenicity predictor, DeepImmunoCNN, from (Li et al., 2021). The CNN was retrained using the "remove0123_sample100.csv" data file provided in their source code. The immunogenicity

Algorithm 1 The training procedure of the GD-WGAN-GP with the switching mechanism. The value for the parameters are $k = 1$, $\lambda = 10$, $n_{critic} = 10$, $N_{data} = 6232$, $N_{batch} = 64$, $\alpha = 10^{-4}$, $\beta_1 = 0.5$, and $\beta_1 = 0.9$.

Require: A maximum iteration number N, γ_{max}, and the initial weights ω_0, θ_0, and ϕ_0 for the critic D_ω, the generator G_θ, and the reward network S_ϕ, respectively.
$n_{split} \leftarrow \lfloor \frac{N_{data}}{N_{batch}} \rfloor$
while $k \leq N$ **do**
  Randomly split the training dataset with data distribution $\mathbb{P}_r$ to n_{split} data $\{\mathbb{P}_r^{(1)}, \mathbb{P}_r^{(2)}, \cdots, \mathbb{P}_r^{(n_{split})}\}$
  for $i = 1, \cdots, n_{split}$ **do**
    Sample a batch of noise vectors $\{z^{(j)}\}_{j=1}^{N_{batch}} \sim N(0, 1)$.
    for $j = 1, \cdots, N_{batch}$ **do**
      Sample real data $x \sim \mathbb{P}_r^{(i)}$ without replacement, and a random number $\epsilon \in U[0, 1]$.
      $\tilde{x} \leftarrow G_\theta(z^{(j)})$
      $L_\phi^{(j)} \leftarrow \frac{1}{N_{batch}}((S_\phi(\tilde{x}) - P(\tilde{x}))^2 + (S_\phi(x) - P(x))^2)$
      $\hat{x} \leftarrow \epsilon x + (1 - \epsilon)\tilde{x}$
      $L_\omega^{(j)} \leftarrow \frac{1}{N_{batch}}(-D_\omega(x) + D_\omega(\tilde{x}) + \lambda(\|\nabla_{\hat{x}} D_\omega(\hat{x})\|_2 - 1)^2)$
    end for
    $\phi \leftarrow \text{Adam}(\nabla_\phi \sum_{j=1}^{N_{batch}} L_\phi^{(j)}, \phi, \alpha, \beta_1, \beta_2)$
    $\omega \leftarrow \text{Adam}(\nabla_\omega \sum_{j=1}^{N_{batch}} L_\omega^{(j)}, \omega, \alpha, \beta_1, \beta_2)$
    if $(i \bmod n_{critic}) = 0$ **then**
      Sample a batch of noise vectors $\{z^{(j)}\}_{j=1}^{N_{batch}} \sim N(0, 1)$.
      $A \leftarrow \emptyset$
      for $j = 1, \cdots, N_{batch}$ **do**
        Add $G_\theta(z^{(j)})$ to a set A
      end for
      $S_{scale} \leftarrow 1 + |\frac{1}{N_{batch}} \sum_{j=1}^{N_{batch}} D_\omega(G_\theta(z^{(j)})) - \frac{1}{N_{batch}} \sum_{j=1}^{N_{batch}} S_\phi(G_\theta(z^{(j)}))|$
      if $n(A) = N_{batch}$ **then**
        $\gamma \leftarrow \gamma_{max}$
      else
        $\gamma \leftarrow 0$
      end if
      $L_\theta \leftarrow \frac{1}{N_{batch}} \sum_{j=1}^{N_{batch}} (-D_\omega(G_\theta(z^{(j)}) - \gamma S_{scale} S_\phi(G_\theta(z^{(j)}))$
      $\theta \leftarrow \text{Adam}(\nabla_\theta L_\theta, \theta, \alpha, \beta_1, \beta_2)$
    end if
  end for
  $k := k + 1$
end while

predictor takes a 9- or 10-mer peptide sequence and a 4-digit encoded HLA sequence as inputs, predicting their real-value immunogenicity score within the range of $[0, 1]$.

3.1 Compared method

The designed GD-WGAN-GP is compared with the MolGAN designed by Cao & Kipf (2022). Cao & Kipf (2022) used WGAN without gradient penalty, and the loss functions of their critic and generator can be written as:

$$L_\omega^{MolGAN} = E_{\tilde{x} \sim \mathbb{P}_G}[D_\omega(\tilde{x})] - E_{x \sim \mathbb{P}_r}[D_\omega(x)] \tag{9}$$

and

$$L_\theta^{MolGAN} = -\lambda_M E_{\tilde{x} \sim \mathbb{P}_G}[D_\omega(\tilde{x})] - (1 - \lambda_M) S_\phi(\tilde{x}), \tag{10}$$

respectively, where $\lambda_M \in [0, 1]$. The same hyperparameters as the designed GD-WGAN-GP are used for the training of the MolGAN (Cao & Kipf, 2022) and the weight clipping of $c = 0.01$ is applied to the critic network.

3.2 Immunogenicity

A comparison between the generators trained with the designed GD-WGAN-GP training scheme and the generator trained using the architecture from Li et al. (2021) and MolGAN (Cao & Kipf, 2022) is shown in Table 1. The GANs are trained on bladder cancer epitope data from TSNAdb (Wu et al., 2018). Five variants of the proposed GD-WGAN-GP are evaluated. The first four variants apply the switching mechanism in (4) with $\gamma_{\max}$ set to 0.25, 0.5, 0.75, and 1, respectively. The fifth variant incorporates the repetition penalty from ORGAN (Guimaraes et al., 2018), as defined in (8). All of them are trained after 1000 epochs.

Three variants of the MolGAN (Cao & Kipf, 2022) designs are compared with the proposed GD-WGAN-GP. The first two variants use MolGAN with λ_M set to 0.5 and 0 in (10), respectively. The third variant uses MolGAN with $\lambda_M = 0$ and incorporates the repetition penalty from ORGAN (Guimaraes et al., 2018). For each variant, two generator checkpoints are selected: one after 1000 training epochs, and another selected based on the epoch that yields the highest sum of the immunogenicity score and the ratio of non-repeated peptides in a batch of 64 generated samples across all 1000 epochs, following a model selection strategy similar to that used in Cao & Kipf (2022).

In Table 1, 10,000 peptide sequences are produced by each of the generator given the same input with a size of $\mathbb{R}^{10000\times128}$, where each element is sampled from a normal distribution. The percentage of non-repeated peptides for each generator is shown in the fourth column of Table 1. The five variants of the proposed GD-WGAN-GP achieve percentages above 75%, with one reaching 100%, outperforming the MolGAN (Cao & Kipf, 2022) designs, which remain below 25%.

To ensure that the generated peptides can be used as the input for the immunogenicity predictor (Li et al., 2021), the peptide sequences containing more than 2 placeholders ('-') are removed. In addition, the repeated peptide sequences are excluded prior to evaluating the immunogenicity score. The average immunogenicity score across the remaining peptides for each generator is presented in the second column in Table 1.

The GD-WGAN-GP with the switching machenism allows different generator configurations to optimize the predicted immunogenicity score or the number of non-repeated peptides. The GD-WGAN-GP with ORGAN's (Guimaraes et al., 2018) repetition penalty can achieve a higher predicted immunogenicity score with a similar percentage of non-repeated peptides compared to GD-WGAN-GP with ($\gamma_{max} = 0.5$).

Using a model selection strategy similar to that used in MolGAN (Cao & Kipf, 2022), which selects the generator that maximizes the sum of the predicted immunogenicity score and the ratio of non-repeated peptides, the GD-WGAN-GP with ORGAN's (Guimaraes et al., 2018) repetition penalty achieves the highest sum of 1.7642. The second-highest value, 1.7268, is obtained by the GD-WGAN-GP with $\gamma_{max} = 0.5$.

3.2.1 Evaluation of the immunogenicity on the generated brain cancer epitopes

The same evaluation is conducted on the same set of GANs trained using brain cancer epitopes from TSNAdb (Wu et al., 2018), with results presented in Table 7 in Appendix B. Unlike the result in Table 1, the MolGAN^{best} variant with $\lambda_M = 0.5$ achieves a higher percentage of non-repeated peptides compared to GD-WGAN-GP ($\gamma_{max} = 1$), but yields a lower predicted immunogenicity score of 0.68. Moreover, GD-WGAN-GP variants with $\gamma_{max} = 0.5$, $\gamma_{max} = 0.75$, and the ORGAN's repetition penalty outperform MolGAN^{best} in both predicted immunogenicity score and the percentage of non-repeated peptides.

The remaining five MolGAN (Cao & Kipf, 2022) variants achieve a maximum of only 22.30% non-repeated peptides, whereas all GD-WGAN-GP variants maintain a minimum of 60.15%. In addition, the GD-WGAN-GP with ORGAN's (Guimaraes et al., 2018) repetition penalty achieves the highest sum of the predicted immunogenicity score and the ratio of non-repeated peptides with a value of 1.7917 compared to other methods. These results further demonstrate that the proposed GD-WGAN-GP outperforms MolGAN across different datasets.

3.2.2 Immunogenicity score and unique peptide rate during training

The average immunogenicity scores of the generated peptide sequences from the same set of generators in Section 3.2 during the 1000 epochs of training are presented in Figure 4 (a), where the solid lines represent

Table 1: Comparison of different designs of GANs trained to generate peptide vaccine candidates for bladder cancer for 1000 epochs except for MolGAN^{best} (Cao & Kipf, 2022), where the generator checkpoint is selected from the epoch that yields the highest sum of the immunogenicity score (imm. score) and the ratio of non-repeated peptides among a batch of 64 generated samples across all 1000 epochs. The imm. score is predicted by the predictor from Li et al. (2021) and its average is across the generated non-repeated peptides after removing peptides more than 10-mer and less than 9-mer. The percentage (percent.) of non-repeated peptides is the percentage of the value of the number of non-repeated peptides divided by the number of generated peptides (10,000). "min." indicates minute. In the Algorithm column, "with ORGAN" indicates the reward for each generated data is penalized by divided the number of its repetition in a batch during training.

Algorithm	**Average imm. score**	**Percent. of peptides with 9-10-mer (%)**	**Percent. of non-repeated peptides (%)**	**Total training time (min.)**
WGAN-GP	0.57	99.07	100.00	32.57
MolGAN ($\lambda_M = 0.5$)	0.73	53.16	20.07	185.34
MolGAN^{best} ($\lambda_M = 0.5$)	0.73	62.59	17.34	166.75
MolGAN ($\lambda_M = 0$)	0.89	100.00	0.01	184.58
MolGAN^{best} ($\lambda_M = 0$)	0.73	100.00	17.55	200.27
MolGAN ($\lambda_M = 0$) with ORGAN	0.89	100.00	0.01	228.33
MolGAN^{best} ($\lambda_M = 0$) with ORGAN	0.73	100.00	16.95	160.56
GD-WGAN-GP ($\gamma_{max} = 0.25$)	0.63	99.86	**100.00**	229.36
GD-WGAN-GP ($\gamma_{max} = 0.5$)	0.73	99.52	99.68	173.25
GD-WGAN-GP ($\gamma_{max} = 0.75$)	0.88	99.97	81.42	198.71
GD-WGAN-GP ($\gamma_{max} = 1$)	**0.93**	99.98	77.33	235.92
GD-WGAN-GP with ORGAN	0.77	99.93	99.42	213.57

the averaged scores calculated using

$$P_{epoch}^{(s)} = \frac{1}{10}\sum_{k=1}^{10} P_{mean}^{(k)}, \tag{11}$$

where $k = 1, 2, \cdots, 10$ is the number of times the generator is trained in an epoch, $s = 1, 2, \cdots, N$ is the number of epochs, $N = 1000$ is the maximum iteration number, and

$$P_{mean}^{(k)} = \frac{1}{N_{batch}} \sum_{j=1}^{N_{batch}} P(G_\theta(z^{(j)})), \tag{12}$$

Notably, the immunogenicity scores of sequences generated from the proposed GD-WGAN-GP continue to increase as the training epoch increases, reaching approximately 0.93 with the rate of non-repeated peptides (unique rate) of around 0.99 at epoch 1000, as shown in the gray line in Figure 4. Conversely, sequences generated without the immunogenicity predictor from WGAN-GP only reach around 0.6 and show no further increase. The MolGAN ($\lambda_M = 0$) and MolGAN ($\lambda_M = 0$) with ORGAN's repetition penalty achieve an immunogenicity scores of roughly 0.88, but the rate of non-repeated peptides decreases to around 0.02, as shown in the red and green lines in Figure 4 (**b**).

3.3 Binding affinity

The binding affinity between the generated peptides and HLA-A*0201 is quantified by the half-maximal inhibition (IC_{50}) values, representing the concentration of the test peptide resulting in 50% inhibition of the binding of a probe peptide (Jurewicz et al., 2019). In the training dataset, the bladder cancer epitopes are predicted to have a binding affinity smaller than 500nM using NetMHCpan v4.0 (Jurtz et al., 2017)

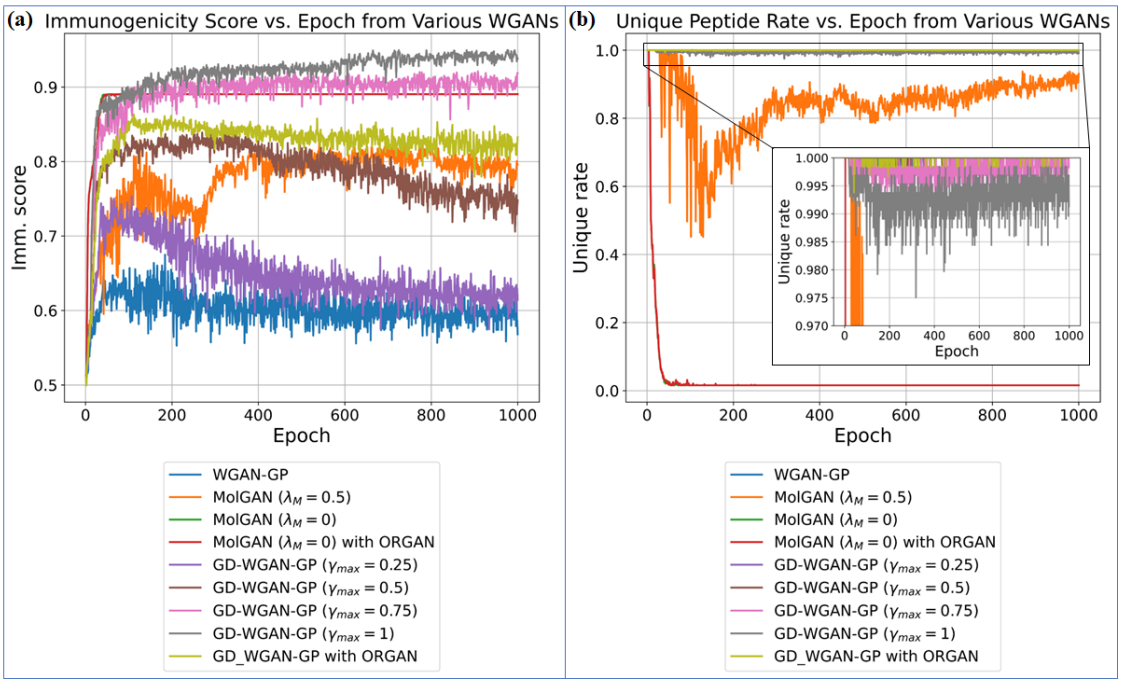

Figure 4: (**a**) The averaged immunogenicity score and (**b**) the unique rate of the peptide sequences generated from the proposed GD-WGAN-GPs (purple, brown, pink, gray, and apple green lines) compared to the WGAN-GP without the immunogenicity predictor (blue line) and MolGANs (Cao & Kipf, 2022) (orange, green, and red lines) through 1000 epochs of training. The proposed GD-WGAN-GP with $\gamma_{max} = 1$ achieves a higher immunogenicity and unique rate compared to MolGANs (Cao & Kipf, 2022) after 400 epochs.

by (Wu et al., 2018). To investigate whether the generated peptides also exhibit a strong binding affinity (< 500nM), we employed the same prediction tool NetMHCpan 4.0 (Jurtz et al., 2017) to predict the IC_{50} values. Peptides with a predicted binding affinity IC_{50} below 500nM are considered binders to HLA-A*0201 (Lundegaard et al., 2008).

The percentage of the number of binders of the generated peptides divided by the total number of generated peptides from the designed GD-WGAN-GP is presented in the second column of Table 2. The GD-WGAN-GP with ORGAN's repetition penalty shows that the generated peptides have a similar property to the training set (binders to HLA-A*0201) compared to WGAN-GP and . The variants from the designed GD-WGAN-GP with the switching mechanism achieve a higher binder rate with a minimum of 62.45% compared to MolGAN's variants with no more than 5.82%.

The prediction of the binding affinity between HLA-A*0201 and the generated brain cancer epitopes candidates from the same set of generators is presented in Table 8 of Appendix B. The results show that WGAN-GP achieves the highest percentage of unique binders at 93.79%, followed by GD-WGAN-GP with $\gamma_{max} = 0.25$ at 89.45%. The lowest percentage of unique binders produced by the proposed method is 51.89% (from GD-WGAN-GP with $\gamma_{max} = 1$), which is higher than the MolGAN variants (Cao & Kipf, 2022), where the best-performing model, $\text{MolGAN}^{\text{best}}$ ($\lambda_M = 0.5$), achieves only 14.72%.

The results from Table 2 and Table 8 suggest that increasing γ_{max} improves the average predicted immunogenicity score but reduces both the number of unique binders and the percentage of non-repeated peptides.

These findings indicate that the proposed GD-WGAN-GP achieves a higher percentage of unique binders than the MolGAN variants (Cao & Kipf, 2022), implying that the generated peptides are more similar to the training data, all of which are binders to HLA-A*0201 with $IC_{50} < 500$ nM predicted by NetMHCpan v4.0 (Jurtz et al., 2017).

Table 2: Comparison of the predicted binding affinity between different designs of GANs trained to generate peptide vaccine candidates for bladder cancer after training for 1000 epochs except for MolGAN^{best} (Cao & Kipf, 2022), where the generator checkpoint is selected from the epoch that yields the highest sum of the immunogenicity score and the ratio of non-repeated peptides among a batch of 64 generated samples across all 1000 epochs. The percentage (percent.) of strong binders is the percentage of the ratio between the number of generated unique sequences having $IC_{50} < 500nM$ binding with HLA-A*0201 predicted by NetMHCpan v4.0 (Jurtz et al., 2017) and the number of generated peptides (10,000). In the Algorithm column, "with ORGAN" indicates the reward for each generated data is penalized by divided the number of its repetition in a batch during training.

Algorithm	**Percent. of unique binders (%)**	**Number of generated unique peptides with 9-10-mer**	**Number of generated peptides in the training set**
WGAN-GP	69.23	9907	0
MolGAN ($\lambda_M = 0.5$)	4.68	1163	0
MolGAN^{best} ($\lambda_M = 0.5$)	5.82	1222	0
MolGAN ($\lambda_M = 0$)	0.00	1	0
MolGAN^{best} ($\lambda_M = 0$)	0.00	1749	0
MolGAN ($\lambda_M = 0$) with ORGAN	0.00	1	0
MolGAN^{best} ($\lambda_M = 0$) with ORGAN	0.00	1688	0
GD-WGAN-GP ($\gamma_{max} = 0.25$)	77.36	9986	1
GD-WGAN-GP ($\gamma_{max} = 0.5$)	77.98	9919	0
GD-WGAN-GP ($\gamma_{max} = 0.75$)	62.45	8139	0
GD-WGAN-GP ($\gamma_{max} = 1$)	77.30	7730	0
GD-WGAN-GP with ORGAN	**89.69**	9934	0

4 Conclusion

This paper presents various designs of goal-directed Wasserstein Generative Adversarial Networks with Gradient Penalty (GD-WGAN-GP) for training generators capable of producing 9- to 10-mer peptide sequences with high predicted immunogenicity, low repetition rate, and strong binding affinity. The output of a reward network, which is concurrently trained to predict immunogenicity scores during the training of the generator network, is incorporated into the GD-WGAN-GP architecture by multiplying it with a scaling factor, S_{scale}, to prevent the generator from solely maximizing the critic output regardless of the reward signal. To reduce repetition, two approaches are proposed: (1) a switching mechanism in which the reward term is excluded in the generator loss when duplicated peptides are present in a batch, and otherwise multiplied by a γ_{max} parameter to control the reward's contribution to the generator loss, and (2) a repetition penalty from ORGAN, which divides each reward by the number of occurrences of its corresponding peptide within a batch. A strong binding affinity is achieved by using a training dataset comprising 6,234 bladder cancer epitope sequences with predicted binding affinity $IC_{50} < 500$ nM to HLA-A*0201.

The GD-WGAN-GP variant with the switching mechanism achieves the highest average immunogenicity score among all compared methods when $\gamma_{max} = 1$. The number of repeated peptides can be reduced by decreasing γ_{max}, although this comes at the cost of lower average immunogenicity. The GD-WGAN-GP variant using ORGAN's repetition penalty achieves the highest combined score of immunogenicity and uniqueness, suggesting it as the most balanced and effective design. All GD-WGAN-GP variants outperform existing goal-directed GANs in terms of the percentage of unique binders with predicted $IC_{50} < 500$ nM, indicating that the proposed models can generate peptides with properties similar to those in the training dataset while maximizing immunogenicity and minimizing redundancy.

Broader Impact Statement

We present novel algorithms that enable goal-directed generation within the Wasserstein generative adversarial network with gradient penalty framework, specifically used to design peptide sequences with enhanced predicted immunogenicity score for peptide vaccine development against bladder cancer. Our approach not only demonstrates improved performance in generating peptides with high predicted immunogenicity, strong binding affinity, and low repetition for bladder cancer, but also generalizes to brain cancer, indicating its broader applicability across multiple cancer types.

References

Noraini Abd-Aziz and Chit Laa Poh. Development of peptide-based vaccines for cancer. *Journal of Oncology*, 2022, 2022.

Dinler A Antunes, Jayvee R Abella, Didier Devaurs, Maurício M Rigo, and Lydia E Kavraki. Structure-based methods for binding mode and binding affinity prediction for peptide-mhc complexes. *Current topics in medicinal chemistry*, 18(26):2239–2255, 2018.

Aleksandra Bartnik, Ajit Johnson Nirmal, and Shi-Yu Yang. Peptide vaccine therapy in colorectal cancer. *Vaccines*, 1(1):1, 2013.

Lisa H Butterfield. Cancer vaccines. *Bmj*, 350, 2015.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs, 2022. URL `https://arxiv.org/abs/1805.11973`.

Xiaotong Chen, Ju Yang, Lifeng Wang, and Baorui Liu. Personalized neoantigen vaccination with synthetic long peptides: recent advances and future perspectives. *Theranostics*, 10(13):6011, 2020.

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

Kaixuan Diao, Jing Chen, Tao Wu, Xuan Wang, Guangshuai Wang, Xiaoqin Sun, Xiangyu Zhao, Chenxu Wu, Jinyu Wang, Huizi Yao, et al. Seq2neo: A comprehensive pipeline for cancer neoantigen immunogenicity prediction. *International Journal of Molecular Sciences*, 23(19):11624, 2022.

Abhishek Dutta. Deep graph generation of small molecules guided by drug-likeness. In *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*, pp. 1–2. IEEE, 2022. URL `https://cmte.ieee.org/cis-bbtc/wp-content/uploads/sites/172/IEEE_CIBCB_2022_paper_6136.pdf`.

Liang Gonog and Yimin Zhou. A review: generative adversarial networks. In *2019 14th IEEE conference on industrial electronics and applications (ICIEA)*, pp. 505–510. IEEE, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Matthew M Gubin, Maxim N Artyomov, Elaine R Mardis, Robert D Schreiber, et al. Tumor neoantigens: building a framework for personalized cancer immunotherapy. *The Journal of clinical investigation*, 125 (9):3413–3421, 2015.

Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models, 2018. URL `https://arxiv.org/abs/1705.10843`.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Yugang Guo, Kewen Lei, and Li Tang. Neoantigen vaccine delivery for personalized anticancer immunotherapy. *Frontiers in immunology*, 9:1499, 2018.

Luca Hensen, Patricia T Illing, Louise C Rowntree, Jane Davies, Adrian Miller, Steven YC Tong, Jennifer R Habel, Carolien E van de Sandt, Katie L Flanagan, Anthony W Purcell, et al. T cell epitope discovery in the context of distinct and unique indigenous hla profiles. *Frontiers in Immunology*, 13, 2022.

Limin Jiang, Jijun Tang, Fei Guo, and Yan Guo. Prediction of major histocompatibility complex binding with bilateral and variable long short term memory networks. *Biology*, 11(6):848, 2022.

Mollie M Jurewicz, Richard A Willis, Vasanthi Ramachandiran, John D Altman, and Lawrence J Stern. Mhc-i peptide binding activity assessed by exchange after cleavage of peptide covalently linked to β2-microglobulin. *Analytical biochemistry*, 584:113328, 2019.

Vanessa Jurtz, Sinu Paul, Massimo Andreatta, Paolo Marcatili, Bjoern Peters, and Morten Nielsen. Netmhcpan-4.0: improved peptide–mhc class i interaction predictions integrating eluted ligand and peptide binding affinity data. *The Journal of Immunology*, 199(9):3360–3368, 2017.

Mostafa Karimi, Shaowen Zhu, Yue Cao, and Yang Shen. De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks. *Journal of chemical information and modeling*, 60(12):5667–5681, 2020.

Claudia Lemmel and Stefan Stevanović. The use of hplc-ms in t-cell epitope identification. *Methods*, 29(3): 248–259, 2003.

Guangyuan Li, Balaji Iyer, VB Surya Prasath, Yizhao Ni, and Nathan Salomonis. Deepimmuno: deep learning-empowered prediction and generation of immunogenic peptides for t-cell immunity. *Briefings in bioinformatics*, 22(6):bbab160, 2021.

Weidang Li, Medha D Joshi, Smita Singhania, Kyle H Ramsey, and Ashlesh K Murthy. Peptide vaccine: progress and challenges. *Vaccines*, 2(3):515–536, 2014.

Eugene Lin, Chieh-Hsin Lin, and Hsien-Yuan Lane. De novo peptide and protein design using generative adversarial networks: an update. *Journal of Chemical Information and Modeling*, 62(4):761–774, 2022.

Jian Liu, Minyang Fu, Manni Wang, Dandan Wan, Yuquan Wei, and Xiawei Wei. Cancer vaccines as promising immuno-therapeutics: Platforms and current progress. *Journal of Hematology & Oncology*, 15 (1):28, 2022.

Wensi Liu, Haichao Tang, Luanfeng Li, Xiangyi Wang, Zhaojin Yu, and Jianping Li. Peptide-based therapeutic cancer vaccine: current trends in clinical application. *Cell Proliferation*, 54(5):e13025, 2021.

Claus Lundegaard, Kasper Lamberth, Mikkel Harndahl, Søren Buus, Ole Lund, and Morten Nielsen. Netmhc-3.0: accurate web accessible predictions of human, mouse and monkey mhc class i affinities for peptides of length 8–11. *Nucleic acids research*, 36(suppl_2):W509–W512, 2008.

Malte Mohme and Marian Christoph Neidert. Tumor-specific t cell activation in malignant brain tumors. *Frontiers in immunology*, 11:205, 2020.

Annika Nelde, Hans-Georg Rammensee, and Juliane S Walz. The peptide vaccine of the future. *Molecular & Cellular Proteomics*, 20, 2021.

Sepideh Parvizpour, Mohammad M Pourseif, Jafar Razmara, Mohammad A Rafi, and Yadollah Omidi. Epitope-based vaccine design: a comprehensive overview of bioinformatics approaches. *Drug Discovery Today*, 25(6):1034–1042, 2020.

Anthony W Purcell, James McCluskey, and Jamie Rossjohn. More than one reason to rethink the use of peptides in vaccine design. *Nature reviews Drug discovery*, 6(5):404–414, 2007.

Evgeny Putin, Arip Asadulaev, Yan Ivanenkov, Vladimir Aladinskiy, Benjamin Sanchez-Lengeling, Alán Aspuru-Guzik, and Alex Zhavoronkov. Reinforced adversarial neural computer for de novo molecular design. *Journal of chemical information and modeling*, 58(6):1194–1204, 2018.

Guilhem Richard, Michael F Princiotta, Dominique Bridon, William D Martin, Gary D Steinberg, and Anne S De Groot. Neoantigen-based personalized cancer vaccines: the emergence of precision cancer immunotherapy. *Expert Review of Vaccines*, 21(2):173–184, 2022.

Megan M Richters, Huiming Xia, Katie M Campbell, William E Gillanders, Obi L Griffith, and Malachi Griffith. Best practices for bioinformatic characterization of neoantigens for clinical utility. *Genome medicine*, 11(1):1–21, 2019.

John Sidney, Bjoern Peters, and Alessandro Sette. Epitope prediction and identification-adaptive t cell responses in humans. In *Seminars in immunology*, volume 50, pp. 101418. Elsevier, 2020.

Malcolm JW Sim and Peter D Sun. T cell recognition of tumor neoantigens and insights into t cell immunotherapy. *Frontiers in Immunology*, 13, 2022.

Alexander J Stephens, Nicola A Burgess-Brown, and Shisong Jiang. Beyond just peptide antigens: the complex world of peptide-based cancer vaccines. *Frontiers in Immunology*, 12:696791, 2021.

Stefan Stevanovic. Identification of tumour-associated t-cell epitopes for vaccine development. *Nature Reviews Cancer*, 2(7):514–514, 2002.

Sjoerd H Van Der Burg, MJ Visseren, RM Brandt, W Martin Kast, and CJ Melief. Immunogenicity of peptides bound to mhc class i molecules depends on the mhc-peptide complex stability. *Journal of immunology (Baltimore, Md.: 1950)*, 156(9):3308–3314, 1996.

Randi Vita, Swapnil Mahajan, James A Overton, Sandeep Kumar Dhanda, Sheridan Martini, Jason R Cantrell, Daniel K Wheeler, Alessandro Sette, and Bjoern Peters. The immune epitope database (iedb): 2018 update. *Nucleic acids research*, 47(D1):D339–D343, 2019.

Chen Wang, Yu Ding, Yuanyong Liu, Qingchen Zhang, Shiqiang Xu, Liliang Xia, Huangqi Duan, Shujun Wang, Ping Ji, Weiren Huang, et al. Identification of mutated peptides in bladder cancer from exomic sequencing data reveals negative correlation between mutation-specific immunoreactivity and inflammation. *Frontiers in Immunology*, 11:576603, 2020.

Jingcheng Wu, Wenyi Zhao, Binbin Zhou, Zhixi Su, Xun Gu, Zhan Zhou, and Shuqing Chen. Tsnadb: a database for tumor-specific neoantigens from immunogenomics data analysis. *Genomics, proteomics & bioinformatics*, 16(4):276–282, 2018.

Jingcheng Wu, Wenzhe Wang, Jiucheng Zhang, Binbin Zhou, Wenyi Zhao, Zhixi Su, Xun Gu, Jian Wu, Zhan Zhou, and Shuqing Chen. Deephlapan: a deep learning approach for neoantigen prediction considering both hla-peptide binding and immunogenicity. *Frontiers in immunology*, 10:2559, 2019.

Yue Zhao, Alexey V Baldin, Orkhan Isayev, Jens Werner, Andrey A Zamyatnin Jr, and Alexandr V Bazhin. Cancer vaccines: antigen selection strategy. *Vaccines*, 9(2):85, 2021.

A Detailed description of the network architecture

We present the details of the generator, the critic, the reward network, and the residual block in Table 3, Table 4, Table 5, and Table 6, respectively.

In the residual block shown in Table 6, the residual connection is defined to be

$$z_{res} = x_{res} + 0.3y_{res}, \tag{13}$$

where x_{res} is the input data in a residual block, y_{res} is the output from the fifth layer (convolution 1D) in the residual block, and z_{res} is the output of the residual connection.

Table 3: Details of the generator network adopted from Li et al. (2021)

Layer	Type	Kernel size	Filter	Stride	Padding	Output shape	Number of parameters
1	Input	-	-	-	-	128	-
2	Fully Connected	-	-	-	-	1280	165120
3	Reshape	-	-	-	-	128×10	-
4	Residual block	-	-	-	-	128×10	98560
5	Residual block	-	-	-	-	128×10	98560
6	Residual block	-	-	-	-	128×10	98560
7	Residual block	-	-	-	-	128×10	98560
8	Residual block	-	-	-	-	128×10	98560
9	Convolution 1D	1	21	1	no	21×10	2709
10	Transpose	-	-	-	-	10×21	-
11	Gumbel-Softmax	-	-	-	-	10×21	-

We denote the output matrix of the Convolution 1D in the residual block as a two-dimensional matrix Y^{res}. The element at the i-th row and the j-th column of Y^{res} is denoted Y^{res}_{ij}. The input matrix of the Convolution 1D in the residual block is denoted as X^{res}. In the residual block, the dimension of the input to the Convolution 1D is defined to be $X^{res} \in \mathbb{R}^{r_X^{res} \times c_X^{res}}$, where $r_X^{res} = 128$ and $c_X^{res} = 10$ are the number of rows and columns of the input matrix, respectively. The input matrix with one zero padding is defined as $X^{pad} = [\mathbf{0}^{res} \quad X^{res} \quad \mathbf{0}^{res}] \in \mathbb{R}^{r_X^{res} \times (c_X^{res}+2)}$, where $\mathbf{0}^{res} = [0 \quad 0 \quad \cdots \quad 0]^T \in \mathbb{R}^{r_X^{res} \times 1}$ is an all zero column vector with dimension r_X^{res}. The dimension of the output matrix after the Convolution 1D will be $Y^{res} \in \mathbb{R}^{f_{res} \times c_X^{res}}$, where $f_{res} = 128$ is the number of kernels. Each element in the output matrix of the Convolution 1D layer in the residual block is computed by

$$Y^{res}_{i,j} = \sum_{m=1}^{r_X^{res}} \sum_{n=1}^{c_w^{res}} (k^{res,i}_{m,n} \cdot X^{pad}_{m,n+j}) + b^{res,i}, \tag{14}$$

where $1 \leq i \leq f_{res}$, $1 \leq j \leq c_X^{res}$, $w^{res,i} \in \mathbb{R}^{r_X \times c_w^{res}}$ is the i-th convolutional kernel, $c_w^{res} = 3$ is the number of columns in the kernel, and $b^{res,i} \in \mathbb{R}$ is the bias for the i-th kernel.

For The Convolution 1D in the 9-th layer of the generator, $X^{gen} \in \mathbb{R}^{r_X^{gen} \times c_X^{gen}}$ is denoted as the input matrix, where $r_X^{gen} = 128$ and $c_X^{gen} = 10$ is the number of rows and columns of the input matrix, respectively. The output matrix will be $Y^{gen} \in \mathbb{R}^{f_{gen} \times c_X^{gen}}$, where $f_{gen} = 21$ is defined to be the number of kernels for this Convolution 1D. The Convolution 1D in the 9-th layer of the generator is defined to be

$$Y^{gen}_{i,j} = \sum_{m=1}^{r_X} \sum_{n=1}^{c_w^{gen}} (k^{gen,i}_{m,n} \cdot X^{gen}_{m,n+j}) + b^{gen,i}, \tag{15}$$

where $1 \leq i \leq f_{gen}$, $1 \leq j \leq c_X^{gen}$, $w^{gen,i} \in \mathbb{R}^{r_X \times c_w^{gen}}$ is the i-th convolutional kernel, $c_w^{gen} = 1$ is the number of columns in the kernel, and $b^{gen,i} \in \mathbb{R}$ is the bias for the i-th kernel.

For the Convolution 1D in the 3-rd layer of the critic and the reward network, $X^{critic} \in \mathbb{R}^{r_X^{critic} \times c_X^{critic}}$ is denoted as its input matrix, where $r_X^{critic} = 21$ and $c_X^{critic} = 10$ is the number of rows and columns of its input matrix, respectively. The output matrix will be $Y^{critic} \in \mathbb{R}^{f_{critic} \times c_X^{critic}}$, where $f_{critic} = 128$ is defined to be the number of kernels for this Convolution 1D. The operation of the Convolution 1D in the 3-rd layer of the critic and the reward network is similar to (15) but with the change of the input size and the number of kernels and it is defined as

$$Y^{critic}_{i,j} = \sum_{m=1}^{r_X} \sum_{n=1}^{c_w^{critic}} (k^{dis,i}_{m,n} \cdot X^{critic}_{m,n+j}) + b^{dis,i}, \tag{16}$$

Table 4: Details of the critic network adopted from Li et al. (2021)

Layer	Type	Kernel size	Filter	Stride	Padding	Output shape	Number of parameters
1	Input	-	-	-	-	10×21	-
2	Transpose	-	-	-	-	21×10	-
3	Convolution 1D	1	128	1	no	128×10	2816
4	Residual block	-	-	-	-	128×10	98560
5	Residual block	-	-	-	-	128×10	98560
6	Residual block	-	-	-	-	128×10	98560
7	Residual block	-	-	-	-	128×10	98560
8	Residual block	-	-	-	-	128×10	98560
9	Reshape	-	-	-	-	1280	-
10	Fully Connected	-	-	-	-	1	1281

Table 5: Details of the reward network network

Layer	Type	Kernel size	Filter	Stride	Padding	Output shape	Number of parameters
1	Input	-	-	-	-	10×21	-
2	Transpose	-	-	-	-	21×10	-
3	Convolution 1D	1	128	1	no	128×10	2816
4	Residual block	-	-	-	-	128×10	98560
5	Residual block	-	-	-	-	128×10	98560
6	Residual block	-	-	-	-	128×10	98560
7	Residual block	-	-	-	-	128×10	98560
8	Residual block	-	-	-	-	128×10	98560
9	Reshape	-	-	-	-	1280	-
10	Fully Connected	-	-	-	-	1	1281

where $1 \leq i \leq f_{critic}$, $1 \leq j \leq c_X^{critic}$, $w^{dis,i} \in \mathbb{R}^{r_X \times c_w^{critic}}$ is the i-th convolutional kernel, $c_w^{critic} = 1$ is the number of columns in the kernel, and $b^{dis,i} \in \mathbb{R}$ is the bias for the i-th kernel.

In the critic and the reward network, the reshape function aligns each row into a row vector. The reshape function acted on the matrix $A \in \mathbb{R}^{p \times q}$ can be represented by

$$vec(A) \triangleq \begin{bmatrix} A_{r1} & A_{r2} & \cdots & A_{rp} \end{bmatrix}, \tag{17}$$

where A_{ri} is denoted as the i-th row vector in the matrix A.

In the generator, the reshape function at the third layer converts a vector $B \in \mathbb{R}^{p \cdot q}$ into a matrix $\hat{B} \in \mathbb{R}^{p \times q}$ by putting the p-th q elements of B to the p-th row of the matrix $\hat{B}$. It can be represented as

$$\hat{B} = \begin{bmatrix} B_1 & B_2 & \cdots & B_q \\ B_{q+1} & B_{q+2} & \cdots & B_{2q} \\ \vdots & & & \\ B_{(p-1)q+1} & B_{(p-1)q+2} & \cdots & B_{pq} \end{bmatrix}, \tag{18}$$

where B_i is denoted as the i-th elements in the matrix B.

Let $X^{GS} \in \mathbb{R}^{r_X^{GS} \times c_X^{GS}}$ be the input and $Y^{GS} \in \mathbb{R}^{r_X^{GS} \times c_X^{GS}}$ be the output of the Gumbel-Softmax in the 11-th layer of the generator network, where $r_X^{GS} = 10$ is the number of row for the input matrix X^{GS} and $c_X^{GS} = 21$ is the number of column in X^{GS}, respectively. The Gumbel-Softmax in the generator network is computed

Table 6: Details of the residual block adopted from Li et al. (2021)

Layer	Type	Kernel size	Filter	Stride	Padding	Output shape	Number of parameters
1	Input	-	-	-	-	128×10	-
2	ReLU	-	-	-	-	128×10	-
3	Convolution 1D	3	128	1	1	128×10	49280
4	ReLU	-	-	-	-	128×10	-
5	Convolution 1D	3	128	1	1	128×10	49280
6	Residual connection	-	-	-	-	128×10	-

as

$$Y_{i,j}^{GS} = \frac{e^{X_{i,j}^{Gumbel}}}{\sum_{m=1}^{c_X^{GS}} e^{X_{i,m}^{Gumbel}}}, \tag{19}$$

where $1 \leq i \leq r_X^{GS}$, $1 \leq j \leq c_X^{GS}$, $Y_{i,j}^{GS}$ is denoted as the element of Y^{GS} at the i-th row and the j-th column, X^{Gumbel} is evaluated as

$$X^{Gumbel} = \frac{X^{GS} - ln(g^{GS})}{\tau}, \tag{20}$$

where $g^{GS} \in \mathbb{R}^{r_X^{GS} \times c_X^{GS}}$ is a matrix with its value generated by the exponential distribution $f(x) = \lambda e^{-\lambda x}$, $\lambda = 1$, and $\tau = 0.75$.

B Experiment using brain cancer epitopes

In this section, the comparison of the proposed GD-WGAN-GP with the generator trained using the architecture from Li et al. (2021) and MolGAN (Cao & Kipf, 2022) is presented in Table 7 using the 2,454 brain cancer epitopes from TSNAdb (Wu et al., 2018) with IC50 < 500nM. The network architecture and hyperparameters are the same as in Section 2.

The results in Table 7 demonstrate that the proposed GD-WGAN-GP with a switching mechanism enables users to balance the average immunogenicity score and the uniqueness of the generated peptides. These peptides also exhibit high binding affinity to HLA-A*0201, as shown in Table 8, outperforming those generated by MolGAN (Cao & Kipf, 2022). Among all models, the generator trained with the GD-WGAN-GP incorporating ORGAN's repetition penalty (Guimaraes et al., 2018) achieves the highest sum of average immunogenicity score and ratio of non-repeated peptides (1.7917), indicating it as the most effective design.

Table 7: Comparison of different designs of GANs trained to generate peptide vaccine candidates for brain cancer for 1000 epochs except for MolGAN^{best} (Cao & Kipf, 2022), where the generator checkpoint is selected from the epoch that yields the highest sum of the immunogenicity score (imm. score) and the ratio of non-repeated peptides among a batch of 64 generated samples across all 1000 epochs. The imm. score is predicted by the predictor from Li et al. (2021) and its average is across the generated non-repeated peptides after removing peptides more than 10-mer and less than 9-mer. The percentage (percent.) of non-repeated peptides is the percentage of the value of the number of non-repeated peptides divided by the number of generated peptides (10,000). "min." indicates minute. In the Algorithm column, "with ORGAN" indicates the reward for each generated data is penalized by divided the number of its repetition in a batch during training (Guimaraes et al., 2018).

Algorithm	**Average imm. score**	**Percent. of peptides with 9-10-mer (%)**	**Percent. of non-repeated peptides (%)**	**Total training time (min.)**
WGAN-GP	0.57	99.85	**99.41**	12.80
MolGAN ($\lambda_M = 0.5$)	0.76	95.56	10.28	70.90
MolGAN^{best} ($\lambda_M = 0.5$)	0.68	99.98	73.61	76.14
MolGAN ($\lambda_M = 0$)	0.89	100.00	0.01	70.82
MolGAN^{best} ($\lambda_M = 0$)	0.73	100.00	22.30	73.91
MolGAN ($\lambda_M = 0$) with ORGAN	0.89	100.00	0.01	70.71
MolGAN^{best} ($\lambda_M = 0$) with ORGAN	0.73	100.00	22.85	71.15
GD-WGAN-GP ($\gamma_{max} = 0.25$)	0.64	97.99	98.88	64.83
GD-WGAN-GP ($\gamma_{max} = 0.5$)	0.82	99.86	97.04	87.53
GD-WGAN-GP ($\gamma_{max} = 0.75$)	0.86	99.94	76.77	75.70
GD-WGAN-GP ($\gamma_{max} = 1$)	**0.90**	99.86	60.15	76.88
GD-WGAN-GP with ORGAN	0.83	99.96	96.17	74.66

Table 8: Comparison of the predicted binding affinity between different designs of GANs trained to generate peptide vaccine candidates for brain cancer after training for 1000 epochs except for MolGAN^{best} (Cao & Kipf, 2022), where the generator checkpoint is selected from the epoch that yields the highest sum of the immunogenicity score and the ratio of non-repeated peptides among a batch of 64 generated samples across all 1000 epochs. The percentage (percent.) of strong binders is the percentage of the ratio between the number of generated unique sequences having $IC_{50} < 500nM$ binding with HLA-A*0201 predicted by NetMHCpan v4.0 (Jurtz et al., 2017) and the number of generated peptides (10,000). In the Algorithm column, "with ORGAN" indicates the reward for each generated data is penalized by divided the number of its repetition in a batch during training (Guimaraes et al., 2018).

Algorithm	**Percent. of unique binders (%)**	**Number of generated unique peptides with 9-10-mer**	**Number of generated peptides in the training set**
WGAN-GP	**93.79**	**9796**	0
MolGAN ($\lambda_M = 0.5$)	6.34	998	0
MolGAN^{best} ($\lambda_M = 0.5$)	14.72	7352	0
MolGAN ($\lambda_M = 0$)	0.00	1	0
MolGAN^{best} ($\lambda_M = 0$)	0.00	2225	0
MolGAN ($\lambda_M = 0$) with ORGAN	0.00	1	0
MolGAN^{best} ($\lambda_M = 0$) with ORGAN	0.00	2281	0
GD-WGAN-GP ($\gamma_{max} = 0.25$)	89.45	9686	0
GD-WGAN-GP ($\gamma_{max} = 0.5$)	88.37	9690	0
GD-WGAN-GP ($\gamma_{max} = 0.75$)	72.10	7671	0
GD-WGAN-GP ($\gamma_{max} = 1$)	51.89	6001	0
GD-WGAN-GP with ORGAN	89.29	9613	0