SELF-CONTEMPLATING IN-CONTEXT LEARNING ENHANCES T CELL RECEPTOR GENERATION FOR NOVEL EPITOPES

Pengfei Zhang
School of Computing and Augmented Intelligence
Arizona State University
Tempe, AZ 85281
pzhang84@asu.edu

Sonal Sujit Prabhu
School of Computing and Augmented Intelligence
Arizona State University
Tempe, AZ 85281
sprabh35@asu.edu

Gloria Grama
School of Life Sciences
Arizona State University
Tempe, AZ 85281
ggrama@asu.edu

Seojin Bang
Google DeepMind
Mountain View, CA 94043
seojinb@google.com

Heewook Lee
School of Computing and Augmented Intelligence
Arizona State University
Tempe, AZ 85281
heewook.lee@asu.edu

ABSTRACT

Computational design of T cell receptors (TCRs) that bind to epitopes holds the potential to revolutionize targeted immunotherapy. However, computational design of TCRs for novel epitopes is challenging due to the scarcity of training data, and the absence of known cognate TCRs for novel epitopes. In this study, we aim to generate high-quality cognate TCRs particularly for *novel epitopes* with no known cognate TCRs, a problem that remains under-explored in the field. We propose to incorporate in-context learning, successfully used with large language models to perform new generative tasks, to the task of TCR generation for novel epitopes. By providing cognate TCRs as additional context, we enhance the model's ability to generate high-quality TCRs for novel epitopes. We first unlock the power of in-context learning by training a model to generate new TCRs based on both a target epitope and a small set of its cognate TCRs, so-called in-context training (ICT). We then self-generate its own TCR contexts based on a target epitope, as novel epitopes lack known binding TCRs, and use it as an inference prompt, referred to as self-contemplation prompting (SCP). Our experiments first demonstrate that aligning training and inference distribution by ICT is critical for effectively leveraging context TCRs. Subsequently, we show that providing context TCRs significantly improves TCR generation for novel epitopes. Furthermore, we show TCR generation using SCP-synthesized context TCRs achieves performance comparable to, and sometimes surpassing, ground-truth context TCRs, especially when combined with refined prompt selection based on binding affinity and authenticity metrics. We assess the designed sequences' binding probability and sequence authenticity using seven diverse computational models.

1 Introduction

Genetically engineered T cells equipped with therapeutic T cell receptors (TCRs) have emerged as a transformative approach in personalized immunotherapy for treating diseases such as cancer [1, 2]. Cognate TCRs play crucial roles for T cells in the identification of abnormal cells by recognizing disease-specific epitopes–an epitope is a part of antigen that sits in the binding pocket of cognate receptor–presented by major histocompatibility complex (MHC) on cell surface [3] (Figure 1A). Computational generation and validation of cognate TCRs for target antigens can expedite the process of developing personalized engineered T cells (Figure 1B). It significantly reduces the number of candidate TCRs subject to wet-lab validation, resulting in substantial reductions in time and cost. It particularly presents a unique opportunity for *novel epitopes*, where timely identification of cognate TCR is essential. Novel epitopes, by definition, are peptides with no known cognate TCRs. Examples include those from newly emerging pathogenic viral strains or patient-specific cancer-induced neoantigens.

Cognate TCR generation requires reliable, rapid validation of generated TCRs. Recent success in predicting the binding affinity of TCR epitopes [4, 5, 6, 7, 8, 9], with a recent work achieving high accuracy (AUC > 0.94 even for novel epitopes), opens a door to tackling the generation task. Despite promising advances in the affinity validation, the generation of cognate TCRs remains largely unexplored. Current efforts are limited to generation of TCR repertoires (not considering cognate epitopes), or only in cases where known TCR-epitope pairs exist [10, 11].

TCR generation for novel epitopes requires models with enhanced generalization capacity to produce TCRs for previously unseen and potentially out-of-distribution epitopes. Large language models (LLMs) such as GPT, demonstrated remarkable success in generalizing to new tasks, stemming from their training on extensive and diverse datasets [12]. LLMs adapt to new tasks via in-context learning, dynamically adjusting responses on the context of a new task, typically provided as a few examples of the task during inference [12]. However, applying LLMs to TCR generation for novel epitopes presents unique challenges. Existing datasets of TCR-epitope pairs cover only a limited range of epitopes and each epitope has far fewer corresponding TCRs than it can recognize. The scarcity of data hinders generalization to novel epitopes which are unseen during training. Furthermore, the lack of known TCRs precludes the use of standard in-context learning for novel epitopes, since it relies on providing contextual TCR examples during inference.

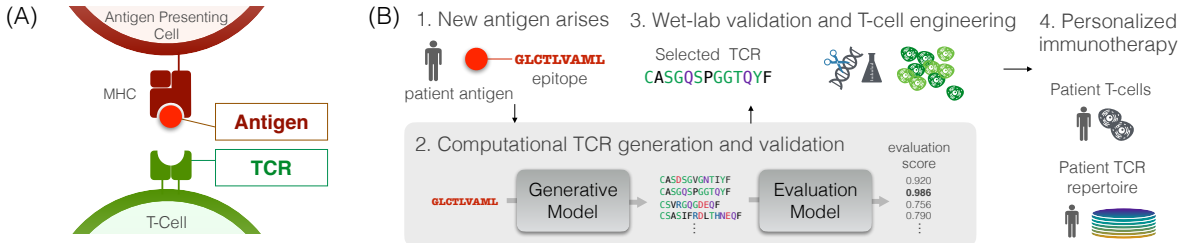

Figure 1: (A) TCR, a protein complex located on the surface of T cells, recognizes and binds to a specific part of antigen (epitope). It allows T-cell to recognize and kill abnormal cells. (B) This study investigates computational generation and validation of TCRs (gray background) within personalized immunotherapy development. Personalized immunotherapy development is powered by the computational approaches: The computational approaches prior to wet-lab validation can offer substantial cost savings and improved workflow efficiency.

Our goal is to develop a computational model that can generate cognate TCR sequences for novel epitopes. The core idea is to provide the model with both the target epitope sequence and a few of its associated TCRs as input. This enables the model to leverage additional information from the provided TCR context when generating cognate TCRs for the target epitope.

We address two challenges encountered in generating TCRs for novel epitopes. First, to overcome the data scarcity and enhance generalizability to unseen epitopes, we align the model's training with our inference goal. It involves training a model to generate a new TCR based not only on a target epitope, but also a small set of its cognate TCRs as additional input context, and so-called in-context training (ICT)[1]. Next, to address the issue of unavailability of known context TCRs for novel epitopes, we eliminate the need for having known binding TCRs at the inference time by leveraging self-contemplation prompting (SCP). It begins self-generating an initial set of TCRs based only on a given epitope, from which high-quality TCRs are selected and used as input context (prompt). We assess the generated TCRs' authenticity and binding potential using state-of-the-art computational methods as a proxy for wet-lab validation.

This study pioneers the generation of potentially cognate TCRs for novel epitopes, a challenging task because these epitopes fall outside the distribution of known epitopes. We believe our work establishes a foundation for developing more sophisticated TCR generative models for novel epitopes with in-context learning. More research is needed to refine and validate the generated TCRs. Crucially, wet-lab validation remains essential to confirm the functionality of generated TCRs. The code and models are available in a public repository[2].

2 Data

We collect experimentally validated TCR-epitope pairs from three publicly available databases: McPAS [13], IEDB [14], VDJdb [15]. We preprocess the data as described in [6]. Only pairs of human MHC class I epitopes (linear) and TCRβ

[1]Various terms such as few-shot learning and in-context tuning have been employed to describe this technique as mentioned in Section A. We refer to this approach as in-context training (ICT) for consistency with the established terminology of in-context learning (ICL).

[2]`https://github.com/Lee-CBG/TCRGen`

sequences are included. Pairs containing wildcards such as * or X and VDJdb pairs with zero confidence scores are excluded. To ensure both training stability and reliable evaluation, we require that each training epitope have at least 100 distinct binding TCRs. This filtering preserves 146 epitopes—retaining over 93% of all interactions—and yields approximately 140K unique TCR–epitope pairs (Figure A1). To assess how well our models can generalize to *novel epitopes*, we divide the 146 unique epitopes into three distinct sets: training (64%), validation (16%), and test (20%). Each set contains a completely distinct list of epitopes and their associated TCRs. This ensures that the models are trained on a representative subset of data and tested on entirely *unseen epitopes*–epitopes that do not appear in training. This results in about 96.7K training, 23.5K validation, and 19.7K test TCR-epitope pairs. We also source four million TCR sequences (unlabeled) from healthy human TCR repertoire (ImmunoSeq, [16]) for developing our evaluation framework. Throughout this paper, we use the term 'TCRs' to refer to TCR β CDR3 (complementarity-determining region 3) sequences as it is the most important CDR in antigen recognition.

3 Method

Our approach is twofold: (1) in-context training to maximize generalizability of in-context inference for TCR generation in Section 3.1 and (2) self-contemplation prompting to enable in-context inference even for novel epitopes without known cognate TCRs in Section 3.2. We then outline our evaluation metrics and baselines to assess the quality of the generated TCRs in Section 3.3.

3.1 In-context training

Due to the limited diversity of the training epitopes, the model has difficulty leveraging cognate TCRs provided as in-context information when generating TCRs for unseen epitopes. We address this challenge by training our model in the way that mirrors the inference process itself. This approach, referred to as in-context training (ICT), tailors training samples to resemble few-shot prompting tasks. This prepares the model to effectively leverage a few-shot style contextual information during inference, despite the scarcity of training epitopes.

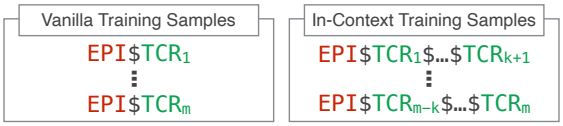

Figure 2: Vanilla training (Vanilla) and in-context training (ICT). The key difference lies in the context window. ICT involves multiple in-context TCRs within each training sample, allowing the model to leverage contextual TCRs when generating new sequences. However, vanilla training lacks the contextual information, relying solely on TCR-epitope pairs.

Vanilla training. The vanilla instances (Figure 2, left) are formed as `EPI$TCR`. `EPI` denotes an epitope sequence, and `TCR` denotes a corresponding TCR sequence, sepearated by a token delimiter `$`. The model parameterized by θ is trained to maximize (log) likelihood of a TCR sequence T given a target epitope E, expressed as:

$$\text{argmax}_\theta \; p_\theta \left(T \mid E \right) \tag{1}$$

In-context training. The ICT instances (Figure 2, right) are formed as $\texttt{EPI\$TCR}_1\texttt{\$}\cdots\texttt{\$TCR}_{k+1}$, leveraging multiple binding TCRs as context for each subsequent TCR. The model parameterized by θ is trained to maximize (log) likelihood of a TCR sequence T_{k+1} given the other TCRs $T_1, \cdots, T_k$ as context and a target epitope E, expressed as:

$$\text{argmax}_\theta \; p_\theta \left(T_{k+1} \mid E, T_1, \cdots, T_k \right) \tag{2}$$

The training objective is in line with the inference objective of in-context learning via few-shot prompting (Equation 3). This refined design is intended to ultimately enhance the model's capacity to generate high-quality TCRs for *novel epitopes* (proposed in Sections 3.2).

Implementation detail. We fine-tune a pre-trained protein language model, `RITA_m` [17], using both the vanilla and ICT training. `RITA_m` consists of 300 million parameters, pre-trained on over 280 million protein sequences. For the vanilla training, we fine-tune on 96.7K unique TCR-epitope pairs, formatted as `EPI$TCR` (Section 2). For ICT training, we curate the samples by randomly grouping $k+1$ TCRs that bind the same epitope into a single instance ($k = 4$ for our model), formatted as $\texttt{EPI\$TCR}_1\texttt{\$}\cdots\texttt{\$TCR}_{k+1}$. While the same TCR can be included in multiple instances as long as there are distinct epitopes where the same TCR is known to bind, it cannot appear multiple times with the same epitope. Comprehensive details about model training and hyperparameter tuning are provided in Section B.

Figure 3: Comparison of the self-contemplation prompting approach (SCP) and standard few-shot prompting (FSP). Both leverage TCRs interacting with a target epitope. While FSP requires known TCRs as a basis for generating new ones, SCP does not need pre-existing TCRs which makes it feasible to target novel epitopes.

3.2 Self-contemplation prompting

Although we believe that the use of known TCRs interacting with a target epitope can lead to more reliable TCR generation, the absence of known TCRs makes in-context learning with the standard few-shot prompting (Figure 3, left) infeasible for *novel epitopes*. To address this, we leverage a technique called self-contemplation prompting (SCP) [18]. The core idea is to have the model create its own prompts. The step-by-step procedure (Figure 3, right) is outlined below.

Prompt generation. First, we generate a set of n candidate TCRs ($n = 300$) for each epitope via 0-shot prompting. The prompt simply consists of the epitope sequence followed by a delimiter (formatted as `EPI$`).

Prompt selection. We evaluate the binding affinity and authenticity of each candidate TCR using computational models. Detailed evaluation procedure is demonstrated in Section 3.3. From this evaluation, we select best k high-quality TCRs to serve as prompts.

In-context inference. The selected TCRs are combined into a k-shot prompt and used as input for the standard few-shot inference. The inference objective is to generate the most likley TCR sequence t given the other TCRs $T_1, \cdots, T_k$ as context and a target epitope E:

$$\text{argmax}_t \; p_\theta \left(t \mid E, T_1, \cdots, T_k \right) \tag{3}$$

The inference objective is analogous with the standard few-shot prompting (FSP) but utilizes machine-generated TCRs as a prompt, excluding the need for known TCRs.

3.3 TCR quality evaluation

3.3.1 Metrics

We consider two key evaluation factors of the generated TCRs: binding affinity to target epitopes and similarity to naturally occurring TCRs. These metrics act as pre-screening filters, expediting the development process by reducing the number of candidates requiring subsequent wet-lab validation.

Binding affinity: Binding affinity measures a TCR's ability to recognize and bind a specific antigen (epitope). We evaluate binding affinity using three computational models. Each model takes an epitope–TCR sequence pair as input and outputs a predicted affinity score. The first model, *BAP MLP*, is a state-of-the-art predictor built on catELMo embeddings [8]. In addition, we implement two alternative architectures—*BAP LSTM* and *BAP CNN*—both of which employ BLOSUM62 embeddings as input features and use an LSTM or CNN backbone, respectively.

Authenticity: Authenticity quantifies how closely a generated TCR resembles naturally occurring TCRs, which is essential for downstream functionality and safety in immunotherapy. We use four metrics in total: one novel likelihood-based score (GPT-LL) and three existing methods. First, following a common anomaly-detection framework [19], we fine-tune a GPT-style protein language model [17] on four million real TCR β CDR3 sequences from ImmunoSeq [16]. We call this fine-tuned model *GPT-LL*. For each generated TCR, we compute the average log-likelihood across all amino acids; higher average log-likelihood indicates greater authenticity. Second, we adopt *TCRMatch* [20], which compares two TCR sequences via k-mer–level similarity. TCRMatch assigns a score between 0 and 1; a score of 1 implies identical binding profiles. For a generated TCR, its authenticity relative to a target epitope is defined as the maximum TCRMatch score against any ground-truth TCR for that epitope. Because TCRMatch relies on known binding TCRs, we use it only for final evaluation (not for selecting SCP prompts). Although originally designed for binding prediction, we found it more effective for authenticity assessment. Finally, we include two additional sequence-similarity metrics—bit score and BLOSUM62 score—computed against ground-truth TCRs. These scores provide complementary measures of how "natural" a generated sequence is.

A generated TCR is deemed high-quality ("good") if it exceeds predefined thresholds for both binding affinity and authenticity. For binding affinity, we consider all three BAP models (BAP MLP, BAP LSTM, BAP CNN); for authenticity, we use GPT-LL, TCRMatch score, bit score, and BLOSUM62 score. To determine the optimal threshold for each metric, we maximize Youden's Index J [21], thereby correctly distinguishing known strong versus weak binders (for the binding models) and authentic versus non-authentic TCRs (for GPT-LL, TCRMatch score, bit score, and BLOSUM62). In addition, the BAP MLP threshold is adjusted so that at least 80% of non-epitope-specific TCRs (as estimated via GPT-LL on background sequences) are classified as negative. These seven metrics serve as pre-screening filters: only TCRs that pass both a binding-affinity threshold and an authenticity threshold are prioritized for downstream wet-lab functional assays. Further details on model architectures and the threshold-selection procedure appear in Section C.

3.3.2 Baselines

Since there is no prior work on generating TCRs for novel epitopes using in-context prompts, we design several baselines and an oracle to highlight the unique benefits of our approach. The first baseline is a pre-trained TCR generation model trained exclusively on TCR sequences (`GPT-LL`) without epitope-specific fine-tuning, which we refer to as epitope agnostic generator. This establishes a baseline performance and helps estimate an appropriate threshold for our binding affinity metrics, under the assumption that the rate of good TCRs will be very small. The second baseline is a model fine-tuned on Vanilla training samples, performed with 0-shot inference given only the target epitope (`Vanilla-0-shot`). This aims to evaluate the model's ability to generalize to unseen epitopes without any TCR information. The third and fourth baselines are models fine-tuned with ICT, and performed with few-shot inference given fake TCRs that are randomly generated from recombinations of amino acid tokens (`ICT-FSP-Fake`) and given randomly selected healthy TCRs known to not bind the target epitope (`ICT-FSP-Healthy`). Those aim to explore the impact of ICT with non-informative in-context TCRs. We also include an oracle model (ICT-FSP) that is fine-tuned with ICT and performed with few-shot inference given ground-truth TCRs. Although this scenario is not realistic for real-world applications where ground-truth is unknown, it provides a valuable estimate of the best-case few-shot performance achievable with our ICT approach.

Note that `GPT-LL` functions as an unconditional TCR generator—analogous to [11], while Vanilla-0-shot is a conditional generator without the ability to incorporate extra context, similar to ERtransformer [22] and GRATCR [23]. Although we do not directly benchmark against these prior works, we include our implementations of `GPT-LL` and Vanilla-0-shot as baselines to illustrate relative performance. It is important to note that our goal is to study the effects of in-context training and self-contemplating prompting on TCR generation, rather than to maximize raw performance scores.

4 Results

We first demonstrate that aligning training and inference through in-context training allows the model to effectively leverage contextual TCRs during inference. Furthermore, by integrating in-context training with self-contemplation prompting (SCP), we not only boost performance with contextual TCRs, but also enable the generation of high-quality TCRs without relying on ground-truth data. We also validate the reliability of our TCR evaluation metrics, ensuring robust and accurate assessments.

In our experiments, models can be exposed to data during both training and inference. For this, we classify these models based on whether each model is exposed to known binding TCRs of the query epitope E during training and/or inference (see Table A1 for details along with section numbers in which each model is discussed).

4.1 TCR generation for unseen epitopes are challenging in standard framework

When training and inference distributions diverge, standard 0-shot TCR generation struggles on epitopes never seen during training. To quantify this gap, we compare performance on seen versus unseen epitopes (Table 1). Across most metrics, performance scores on unseen epitopes drops markedly—for instance, BAP LSTM falls by 12.3 percent and BAP CNN by 14.4 percent. Even the combined score declines (–2.4 percent). Moreover, unseen epitopes exhibit much higher variance than seen epitopes (Figure A2) for nearly every metric. These findings confirm that generating high-quality TCRs for novel epitopes is more challenging.

4.2 In-context training is critical for few-shot TCR generation

To investigate the impact of aligning training and inference distributions, we compare vanilla training with ICT on few-shot TCR generation tasks. This approach is particularly relevant in real-world scenarios when dealing with epitopes that have very limited known binding TCRs. We hypothesize that leveraging the available in-context information

Table 1: Mean (std) "good TCR" rates for seen vs. unseen epitopes, and performance gap Δ.

Metric	Seen Epitopes	Unseen Epitopes	Δ
BAP LSTM	91.0 (8.2)	78.7 (29.8)	–12.3 %
BAP CNN	84.8 (14.8)	70.4 (30.6)	–14.4 %
BAP MLP	77.1 (16.0)	88.1 (7.3)	+11.0 %
TCRMatch	97.6 (14.2)	93.1 (22.7)	–4.5 %
GPT-LL	97.3 (3.3)	92.5 (21.6)	–4.8 %
BLOSUM62 Score	79.4 (18.2)	74.8 (21.3)	–4.6 %
Bit Score	78.2 (16.2)	70.2 (20.2)	–8.0 %
All metrics combined	45.9 (18.2)	43.5 (22.6)	–2.4 %

during the generation process can significantly enhance TCR design by providing the model with crucial binding data as context. To isolate the effect of in-context learning, known binding TCRs for *unseen epitope* (See Table A1 for definition) are provided during inference, instead of machine generated TCRs.

We find that vanilla training fails to leverage in-context TCRs: the average success rate drops from 43% to just 2% when extra context TCRs are included in the prompt. This outcome is expected, since vanilla training (prompt format: `epitope$tcr`) never exposes the model to multiple TCRs during training, whereas ICT training (prompt format: `epitope$tcr_1$tcr_2$...`) explicitly conditions on additional examples. As a result, the vanilla model cannot generate valid TCRs when given extra in-context sequences.

Meanwhile, ICT achieves equal or better performance in k-shot than 0-shot inference for 20 out of 29 novel epitopes. This emphasizes the crucial role of aligning the training input with how the model will be used, in order to fully unlock the potential of in-context learning for novel epitopes.

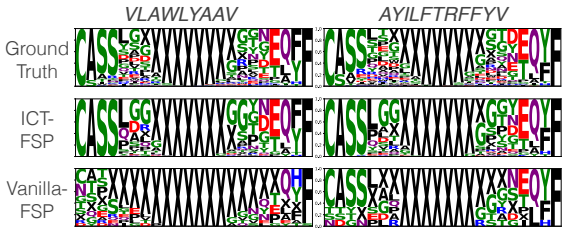

Figure 4: Visualization of ground-truth and generated TCRs by ICT-FSP and Vanilla-FSP using a 4-shot inference using ground-truth TCRs. Each TCR sequence is mid-padded with a special token `X` to make the lengths of all TCRs identical prior to generating the logos. Results of the two epitopes *VLAWLYAAV* and *AYILFTRFFYV* are presented as illustration examples, selected at random.

Visual assessment using SeqLogo also supports this observation. We select two epitopes that have shown suboptimal performance on zero-shot inference and examine their seqlogo plots generated by 4-shot inference by ICT-FSP and Vanilla-FSP against ground-truth (Figure 4). Each TCR sequence is mid-padded with a special token `X`, a standard method to analyze TCR sequences [24]. To evaluate differences in sequence generation patterns across models, we compared all generated sequences without filtering for "good" TCRs. This approach allows us to directly assess the intrinsic tendencies of each model's output, providing a clearer picture of generation patterns without the influence of evaluation metrics. Vanilla training struggles to replicate the ground-truth amino acid distribution when few-shot TCR prompts are used. In contrast, TCRs generated from ICT-FSP closely mirrored the ground-truth. This disparity highlights the benefits of aligning training with inference strategies, particularly when contextual TCRs are available to guide generation.

4.3 Self-contemplation prompting unlock in-context TCR generation for novel epitopes

Carefully generated TCR prompts enhances high-quality TCR generation. By leveraging self-generated TCR sequences as in-context examples, our model successfully generates candidate TCRs for novel epitopes without relying on any known binders. In particular, SCP with a larger context window achieves a substantially higher success rate than the Vanilla 0-shot baseline, which has no in-context TCRs (Tables 2, 3). Moreover, the sequences produced by both SCP-Random and SCP-Select closely resemble ground-truth TCRs (Figure A3). We compare and evaluate three SCP variants (SCP-Random, SCP-Select, and SCP-Chain). SCP-Random selects k in-context TCRs randomly from the model's 0-shot inferences. SCP-Select, on the other hand, identifies high-quality TCRs and selects the top k in-context TCRs based on BAP and GPT-LL evaluation metrics. SCP-Chain iteratively selects in-context TCRs from previous inference outputs as prompts for subsequent contexts. For iteration j, SCP-Chain selects the best TCR generated

Table 2: Good TCR rates for binding affinity measurement (BAP MLP / BAP LSTM / BAP CNN) of different approaches for novel epitopes. Each setting generates 300 TCRs per epitope.

	BAP MLP			BAP LSTM			BAP CNN		
N. of Training Contexts	0	5	10	0	5	10	0	5	10
N. of Inference Contexts	0	5	10	0	5	10	0	5	10
Epitope-agnostic Generator (Baseline)	21.28 (1.54)	–	–	48.74 (0.67)	–	–	47.63 (0.80)	–	–
Vanilla-0-shot (Baseline)	88.14 (1.35)	–	–	78.66 (5.53)	–	–	70.41 (5.69)	–	–
ICT-FSP-Fake (Baseline)	–	78.15 (1.64)	79.67 (1.61)	–	73.18 (5.50)	72.66 (5.19)	–	60.71 (6.27)	64.67 (5.46)
ICT-FSP-Healthy (Baseline)	–	77.80 (1.31)	84.63 (0.92)	–	83.13 (3.82)	85.26 (3.76)	–	75.89 (4.23)	75.30 (4.56)
ICT-FSP (Oracle)	–	79.67 (1.47)	87.17 (1.01)	–	**85.20 (3.71)**	87.49 (3.03)	–	74.25 (4.92)	**77.54 (4.11)**
ICT-SCP-Random	–	80.54 (1.44)	89.61 (1.02)	–	84.25 (3.80)	86.29 (3.91)	–	75.77 (4.61)	75.94 (4.91)
ICT-SCP-Chain	–	80.00 (1.45)	90.86 (0.96)	–	84.30 (3.73)	**87.64 (3.72)**	–	75.49 (4.26)	77.41 (4.61)
ICT-SCP-Select	–	**82.70 (1.22)**	**91.91 (0.92)**	–	84.75 (3.75)	83.71 (5.38)	–	**76.62 (4.39)**	75.29 (5.88)

from iteration $j-1$ based on BAP and GPT-LL evaluation metrics, and add it to the prompt used in iteration $j-1$. Overall, SCP methods offer flexible prompting and often match or exceed the performance of FSP-based approaches. Interestingly, for authenticity metrics, increasing the number of context prompts boosts the "good TCR" rates for TCRMatch and GPT-LL, while the bit score and BLOSUM62 scores decline slightly. This suggests that SCP-generated TCRs follow the learned distribution of real TCRs yet maintain biologically favorable sequence diversity.

Beyond reporting the good-TCR rate, we also examine the raw metric distributions for each in-context prompting method. Boxplots (Figure A5) summarize mean metric scores across all novel epitopes, showing that most prompting techniques consistently outperform the baselines. To reveal epitope-specific behavior, we present heatmaps of metric scores per epitope (Figure A6), which exhibit greater variability. These results demonstrate that our framework is highly flexible: users can select prompting strategies based on available data and personal preference. The heatmaps suggest that exploring and ensembling multiple prompting methods per epitope can yield even more robust performance.

Synthesized TCR can provide adequate contextual information better than or comparably to ground-truth. Given that SCP relies entirely on self-generated TCRs as in-context prompts, we compare the performance of synthesized context TCRs from SCP (ICT-SCP-Random) to ground-truth context TCRs (ICT-FSP). While ground-truth context TCRs are impractical to obtain for novel epitopes, ICT-FSP can serve as a performance upper bound. Conversely, we evaluate the performance of Fake (ICT-FSP-Fake) and healthy TCRs (ICT-FSP-Healthy) as non-binding inference contexts, providing a performance lower bound. Table 2 and 3 show that ICT-SCP performs comparably to or even better than ICT-FSP, suggesting that SCP, even with self-generated TCRs, is as effective as models utilizing ground-truth binding TCRs as inference context. This ability to iteratively generate its own prompts makes SCP independent of pre-existing TCR knowledge and expands its applicability to cases lacking known binding TCRs. Our findings signify SCP's effectiveness, particularly when combined with in-context training, for generating TCRs in scenarios where ground-truth TCR data is unavailable.

A minimal number of context TCRs are sufficient to achieve optimal performance. As we increase the number of in-context TCRs in the prompt, generation quality for novel epitopes improves and then plateaus after only a few shots (Figure 5). Specifically, BAP LSTM and BAP CNN reach their performance ceiling by 3-shot, whereas BAP MLP continues to climb with additional context. In contrast, TCRMatch and GPT-LL already saturate at 1-shot, likely because they excel at discriminating non-binders (Figure A4), so the model can produce non-random, epitope-specific sequences with even a single example. Notably, as we add more context, both BLOSUM62 and bit-score metrics decline by roughly 10%. This does not indicate poorer quality; rather, it reflects increased sequence diversity. In fact, even at 9-shot, the average BLOSUM62 and bit scores remain above the "good TCR" cutoffs determined via Youden's index. In other words, generated sequences continue to conform to biologically plausible substitution patterns while exploring a wider range of variants.

Table 3: Good TCR rates for authenticity measurement (TCRMatch / GPT-LL / Bit Score / BLOSUM62 Score) of different approaches for novel epitopes. Each setting generates 300 TCRs per epitope.

	TCRMatch			GPT-LL			Bit Score			BLOSUM62 Score		
N. of Training Contexts	0	5	10	0	5	10	0	5	10	0	5	10
N. of Inference Contexts	0	5	10	0	5	10	0	5	10	0	5	10
Epitope-agnostic Generator (Baseline)	97.30 (1.14)	–	–	96.40 (2.01)	–	–	1.62 (0.16)	–	–	15.32 (1.97)	–	–
Vanilla-0-shot (Baseline)	93.13 (4.22)	–	–	92.46 (4.02)	–	–	70.22 (3.74)	–	–	74.80 (3.96)	–	–
ICT-FSP-Fake (Baseline)	–	90.43 (4.27)	96.63 (1.96)	–	91.15 (3.40)	95.93 (1.91)	–	69.23 (4.31)	74.86 (2.22)	–	72.93 (4.78)	78.30 (2.68)
ICT-FSP-Healthy (Baseline)	–	99.61 (0.10)	99.53 (0.14)	–	98.56 (0.16)	98.63 (0.11)	–	80.62 (1.20)	**76.91 (1.31)**	–	**84.53 (0.96)**	**80.89 (1.03)**
ICT-FSP (Oracle)	–	96.26 (3.36)	99.48 (0.12)	–	96.47 (2.22)	98.72 (0.21)	–	**77.71 (2.94)**	74.91 (1.49)	–	81.43 (3.05)	78.40 (1.19)
ICT-SCP-Random	–	99.70 (0.09)	99.71 (0.11)	–	98.63 (0.16)	99.14 (0.16)	–	75.34 (1.42)	62.31 (2.30)	–	78.55 (1.24)	65.59 (2.29)
ICT-SCP-Chain	–	99.07 (0.46)	98.39 (0.85)	–	98.45 (0.14)	**99.48 (0.07)**	–	77.44 (1.33)	63.33 (2.13)	–	81.23 (1.26)	66.00 (2.17)
ICT-SCP-Select	–	**99.75 (0.07)**	**99.77 (0.08)**	–	**98.99 (0.09)**	99.17 (0.21)	–	73.16 (1.51)	58.11 (2.49)	–	76.66 (1.30)	61.32 (2.47)

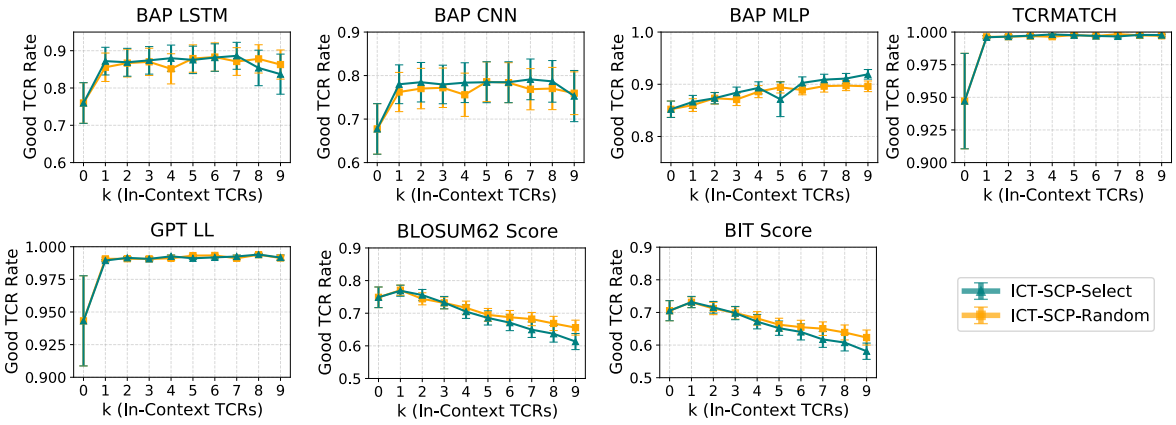

Figure 5: Good TCR rates of self-contemplation prompting (SCP) approaches for novel epitopes along with the number of inference shots (i.e., context TCRs). All inferences are performed by a model trained on in-context instances formed as formed as $\texttt{EPI\$TCR}_1\texttt{\$}\cdots\texttt{\$TCR}_{10}$. Each setting generates 300 TCRs for each epitopes.

4.4 Reliability of TCR quality metrics

We assess the reliability of seven TCR quality evaluation metrics—BAP LSTM, BAP CNN, BAP MLP, TCRMatch, GPT-LL, bit score, and BLOSUM62 score—by (1) testing whether metric scores for high-quality versus low-quality TCR groups originate from different distributions and (2) measuring each metric's ability to predict high-quality versus low-quality TCRs (Figure A4). Details about data curation and optimal threshold are provided in Section C.

5 Discussion and Conclusion

Our study is the first to delve into high-quality TCR generation especially for novel epitopes with in-context learning. The challenge is, due to scarce data and limited diversity, TCR sequence generation models exhibit limited generalizability. These models fail to take advantage of in-context learning when generating binding TCRs for novel epitopes, which typically come from a different data distribution. To address this, we shape training samples into a form of few-shot learning, reducing the discrepancy between training and inference distributions. Since no known binding TCRs exist for novel epitopes, we generate and select high-quality TCRs to serve as a few-shot prompt. This ultimately improves the model's performance in few-shot settings. We also introduce a reliable way for screening the generated TCRs based on two key criteria: binding affinity and authenticity. Complementary structural analyses on selected SCP-generated sequences illustrate the functional plausibility of our designed TCRs (see Fig. A7 and Section C). These screened TCRs can be candidates for further wet-lab validation to confirm their effectiveness.

This method of designing high-quality TCRs for novel epitopes (new targets) is crucial to accelerate the design of personalized immunotherapy strategies. Computational generating and evaluating TCRs significantly speed up the T cell engineering process. This results in substantial cost reductions by minimizing the need for labor-intensive wet-lab validation. Our findings, while specifically focused on the generation of TCR sequences targeting a particular epitope, may offer valuable insights for the broader field of biological sequence generation. The key finding is the importance of both context-rich training and inference. This includes training a model in the same way as inference and incorporating synthetic data to enrich the inference when real-world data is limited. This might be beneficial for many biological sequence generation tasks that often grapple with challenges such as limited dataset size, constrained diversity, and strong emphasis on targeted generation.

While we provide specific methods for determining cutoffs, alternative approaches, such as t-tests, could be explored. The thresholds used are illustrative and should be adjusted based on users' tolerance for false positives. Higher thresholds reduce false positives but may miss TCRs of interest, while lower thresholds capture more TCRs at the cost of increased false positives. Further, the evaluation metrics used in this study, while achieving high precision, are not without inherent error. Ultimately, wet-lab validation remains a definitive step in confirming the TCR effectiveness. However, it is important to note that this computational pipeline prioritizes curating a refined set of high-quality candidate TCRs for subsequent wet-lab validation, resulting in expediting overall TCR engineering process.

References

[1] Estelle Baulu, Célia Gardet, Nicolas Chuvin, and Stéphane Depil. TCR-engineered T cell therapy in solid tumors: State of the art and perspectives. *Science Advances*, 9(7):eadf3700, 2023.

[2] Muzamil Y Want, Zeenat Bashir, and Rauf A Najar. T cell based immunotherapy for cancer: approaches and strategies. *Vaccines*, 11(4):835, 2023.

[3] Meriem Attaf, Mateusz Legut, David K Cole, and Andrew K Sewell. The T cell antigen receptor: the Swiss army knife of the immune system. *Clinical & Experimental Immunology*, 181(1):1–18, 2015.

[4] Vanessa Isabell Jurtz, Leon Eyrich Jessen, Amalie Kai Bentzen, Martin Closter Jespersen, Swapnil Mahajan, Randi Vita, Kamilla Kjærgaard Jensen, Paolo Marcatili, Sine Reker Hadrup, Bjoern Peters, et al. NetTCR: sequence-based prediction of TCR binding to peptide-MHC complexes using convolutional neural networks. *BioRxiv*, page 433706, 2018.

[5] Ido Springer, Nili Tickotsky, and Yoram Louzoun. Contribution of T cell receptor alpha and beta CDR3, MHC typing, V and J genes to peptide binding prediction. *Frontiers in Immunology*, 12:664514, 2021.

[6] Michael Cai, Seojin Bang, Pengfei Zhang, and Heewook Lee. ATM-TCR: TCR-epitope binding affinity prediction using a multi-head self-attention model. *Frontiers in Immunology*, 13, 2022.

[7] Pengfei Zhang, Seojin Bang, and Heewook Lee. PiTE: TCR-epitope binding affinity prediction pipeline using Transformer-based sequence encoder. In *Pacific Symposium on Biocomputing 2023: Kohala Coast, Hawaii, USA, 3–7 January 2023*, pages 347–358. World Scientific, 2022.

[8] Pengfei Zhang, Seojin Bang, Michael Cai, and Heewook Lee. Context-aware amino acid embedding advances analysis of TCR-epitope interactions. *eLife*, April 2023.

[9] Pengfei Zhang, Seojin Bang, and Heewook Lee. Active learning framework for cost-effective TCR-epitope binding affinity prediction. In *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 988–993. IEEE, 2023.

[10] Ziqi Chen, Martin Renqiang Min, Hongyu Guo, Chao Cheng, Trevor Clancy, and Xia Ning. T-cell receptor optimization with reinforcement learning and mutation polices for precision immunotherapy. In *International Conference on Research in Computational Molecular Biology*, pages 174–191. Springer, 2023.

[11] Kristian Davidsen, Branden J Olson, William S DeWitt III, Jean Feng, Elias Harkins, Philip Bradley, and Frederick A Matsen IV. Deep generative models for t cell receptor protein sequences. *Elife*, 8:e46935, 2019.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[13] Nili Tickotsky, Tal Sagiv, Jaime Prilusky, Eric Shifrut, and Nir Friedman. McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. *Bioinformatics*, 33(18):2924–2929, 2017.

[14] 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.

[15] Mikhail Shugay, Dmitriy V Bagaev, Ivan V Zvyagin, Renske M Vroomans, Jeremy Chase Crawford, Garry Dolton, Ekaterina A Komech, Anastasiya L Sycheva, Anna E Koneva, Evgeniy S Egorov, et al. VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Research*, 46(D1):D419–D427, 2018.

[16] Sean Nolan, Marissa Vignali, Mark Klinger, Jennifer N Dines, Ian M Kaplan, Emily Svejnoha, Tracy Craft, Katie Boland, Mitch Pesesky, Rachel M Gittelman, et al. A large-scale database of T-cell receptor beta (TCRβ) sequences and binding associations from natural and synthetic exposure to SARS-CoV-2. *Research Square*, 2020.

[17] Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. RITA: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*, 2022.

[18] Rui Li, Guoyin Wang, and Jiwei Li. Are human-generated demonstrations necessary for in-context learning? *International Conference on Learning Representations*, 2024.

[19] Jing Su, Chufeng Jiang, Xin Jin, Yuxin Qiao, Tingsong Xiao, Hongda Ma, Rong Wei, Zhi Jing, Jiajun Xu, and Junhong Lin. Large language models for forecasting and anomaly detection: A systematic literature review. *arXiv preprint arXiv:2402.10350*, 2024.

[20] William D Chronister, Austin Crinklaw, Swapnil Mahajan, Randi Vita, Zeynep Koşaloğlu-Yalçın, Zhen Yan, Jason A Greenbaum, Leon E Jessen, Morten Nielsen, Scott Christley, et al. TCRMatch: predicting T-cell

receptor specificity based on sequence similarity to previously characterized receptors. *Frontiers in Immunology*, 12:640725, 2021.

[21] William J Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, 1950.

[22] Jiannan Yang, Bing He, Yu Zhao, Feng Jiang, Zhonghuang Wang, Yixin Guo, Zhimeng Xu, Bo Yuan, Jiangning Song, Qingpeng Zhang, et al. De novo generation of t-cell receptors with desired epitope-binding property by leveraging a pre-trained large language model. *bioRxiv*, pages 2023–10, 2023.

[23] Zhenghong Zhou, Junwei Chen, Shenggeng Lin, Liang Hong, Dong-Qing Wei, and Yi Xiong. Gratcr: epitope-specific t cell receptor sequence generation with data-efficient pre-trained models. *IEEE Journal of Biomedical and Health Informatics*, 2025.

[24] Marie-Paule Lefranc, Christelle Pommié, Quentin Kaas, Elodie Duprat, Nathalie Bosc, Delphine Guiraudou, Christelle Jean, Manuel Ruiz, Isabelle Da Piédade, Mathieu Rouard, et al. IMGT unique numbering for immunoglobulin and T cell receptor constant domains and Ig superfamily C-like domains. *Developmental & Comparative Immunology*, 29(3):185–203, 2005.

[25] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1):4348, 2022.

[26] Yicheng Lin, Dandan Zhang, and Yun Liu. Tcr-gpt: Integrating autoregressive model and reinforcement learning for t-cell receptor repertoires generation. *arXiv preprint arXiv:2408.01156*, 2024.

[27] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[28] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR, 2021.

[29] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809. Association for Computational Linguistics, July 2022.

[30] Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730. Association for Computational Linguistics, 2022.

[31] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[32] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[33] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[34] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

[35] Julia Koehler Leman, Brian D Weitzner, Steven M Lewis, Jared Adolf-Bryfogle, Nawsad Alam, Rebecca F Alford, Melanie Aprahamian, David Baker, Kyle A Barlow, Patrick Barth, et al. Macromolecular modeling and design in rosetta: recent methods and frameworks. *Nature methods*, 17(7):665–680, 2020.

[36] Elaine C Meng, Thomas D Goddard, Eric F Pettersen, Greg S Couch, Zach J Pearson, John H Morris, and Thomas E Ferrin. Ucsf chimerax: Tools for structure building and analysis. *Protein Science*, 32(11):e4792, 2023.

[37] Edward N Baker and Roderick E Hubbard. Hydrogen bonding in globular proteins. *Progress in biophysics and molecular biology*, 44(2):97–179, 1984.

[38] Sofie Gielis, Pieter Moris, Wout Bittremieux, Nicolas De Neuter, Benson Ogunjimi, Kris Laukens, and Pieter Meysman. Detection of enriched T cell epitope specificity in full T cell receptor sequence repertoires. *Frontiers in Immunology*, 10:2820, 2019.

A Related Works

We first discuss recent efforts and challenges in computational generation of TCRs. Next, we briefly discuss the strengths and limitations of in-context learning and explicit meta-learning to enhance generalizability.

Computational generation of cognate TCRs. While generative models have been widely applied to the generation of biological sequences, including protein sequences [25, 17], the specific domain of the generation of TCR sequences has not been thoroughly explored. Existing models primarily focus on replicating the distribution of real TCR sequences, often overlooking the critical need for epitope specificity [11]. Early epitope-conditioned schemes—such as GRATCR [23] and ERTransformer [22]—produce candidate receptors for a given peptide but cannot leverage additional context TCRs when only a few binders are known. Reinforcement-learning approaches generate epitope-specific TCRs [10, 26], but they are limited to antigens seen during training and require per-epitope retraining. Crucially, no existing method supports out-of-sample epitopes with in-context learning to flexibly incorporate user-provided sequences.

The cost of wet-lab validation is another significant challenge in the development of TCR generation models. To address this, computational methods for predicting binding affinity have emerged as valuable proxies. Although not infallible, they expedite the development process by reducing the number of candidates requiring subsequent wet-lab validation. Various computational approaches have been proposed [6, 4, 5]. Recent works, such as catELMo embeddings with shallow linearly connected layers [8] and PiTE [7], which leverage pre-trained TCR embedding models, have achieved an average AUC score of over 94% for TCR-epitope binding prediction, even for unseen epitopes.

In-context learning. LLMs have been recognized as task-agnostic few-shot learners [12, 27]. They can infer outputs for test inputs based on a prompt consisting of a few input-output pairs demonstrating a desired task. It is particularly beneficial for novel tasks where fine-tuning data is scarce or unavailable. This capability is largely attributed to their extensive pre-training on diverse and large datasets, allowing them to meta-learn implicitly and generalize quickly from few examples [12]. However, when a model is not sufficiently scaled or applied to domains significantly different from their pre-training data, their generalization ability diminishes [28].

Explicit meta-learning, as opposed to the implicit methods, involves pre-training or fine-tuning the model on training data closely aligned with the testing data. Although it requires adjustments of the pre-training parameters, they can significantly improve performance on targeted tasks. A recent work [29] leverages meta-learning to train an LLM for few-shot learning tasks, which they called *meta-training for in-context learning*. Another team [30] proposed a similar approach, which they called *in-context tuning*, involves fine-tuning of LLMs on a small dataset of input-output pairs, enhancing its few-shot inference capabilities.

B In-Context Generative Models Training and Inference

B.1 Design and training

Our models are fine-tuned from the `RITA_m`, a family of pretrained protein language models (pLMs). It is a medium-sized model with 300 million parameters across 24 layers, selected for its optimal balance between computational efficiency and generative performance. A character-level tokenizer is used to process epitope and TCR sequences, treating each amino acid as a distinct token. Special tokens such as `$` (delimiter), `<EOS>` and `<PAD>` are included to manage sequence operations effectively within our model's architecture. We employ a cross-entropy loss function focused on predicting tokens following the delimiter.

B.2 Model inference

During inference, the model generates TCR sequences by extending beyond the delimiter token using beam search. To control the generation process, we set a maximum sequence length of 64 amino acids, a `top_k` sampling value of 8 to encourage diversity.

B.3 Hyperparameter tuning

We optimize the model with the following search space (bold indicates our final choices): number of contexutual TCRs (`T`) for ICT – $\{3, \mathbf{5}, 10\}$, batch size – $\{\mathbf{16}, 32\}$, learning rate – $\{2 \times 10^{-4}, \mathbf{2 \times 10^{-5}}, 1 \times 10^{-5}\}$, training epoch – $\{\mathbf{1}, 2, 4\}$, and temperature – $\{\mathbf{0.4}, 0.7, 1.0\}$. Training is stopped after the first epoch because validation error began to increase. We train the model via Adam optimizer with linear learning rate scheduler. The hyper-parameters are tuned via grid search.

B.4 Computing resources

The model is implemented using Pytorch with torch version 2.0.0 and CUDA version 11.7. It is trained on two NVIDIA GTX 2080 Ti GPUs in parallel (VRAM about 11GB) with training times of under 3 hours per epoch.

C Details of Metric Computation and Threshold Selection

We introduce two key aspects that are commonly used to evaluate quality of generated TCRs: binding affinity to the target epitope and similarity to naturally occurring TCRs. These metrics act as pre-screening filters, expediting the development process by reducing the number of candidates requiring subsequent wet-lab validation.

C.1 Binding affinity

- **BAP MLP**: BAP MLP [8] is a binding affinity prediction model that takes a pair of epitope and TCR sequences as input and outputs the predicted probability of binding. It employs a catELMo embedding [8] for TCR sequences and a traditional BLOSUM embedding [31] for epitope sequences. These embeddings are then fed into separate neural network layers with SiLU activation, batch normalization, and dropout. They are then concatenated and fed into another neural network with similar layers. Finally, a single neuron with a sigmoid activation function outputs a binding affinity score between 0 and 1 where a score of 0 indicates no predicted binding affinity, while 1 signifying a strong binding potential.
- **BAP LSTM**: BAP LSTM uses BLOSUM62 embeddings (20 dimensions per residue, padded to length 22) for both TCR and epitope inputs. Each input is passed through a bidirectional LSTM, producing two final states of size 64 each, which are concatenated into a 256-dimensional vector. This vector is fed into an MLP: a dense layer followed by a LeakyReLU activation and a dropout. The final output layer is a single neuron with sigmoid activation, yielding a binding probability in $[0, 1]$.
- **BAP CNN**: BAP CNN also uses BLOSUM62 embeddings (shape 22×20) for TCR and epitope sequences. For each input (TCR or epitope), five parallel 1D convolutional filters with kernel sizes $\{1, 3, 5, 7, 9\}$ and sigmoid activations extract features, followed by global max-pooling on each filter's output. The pooled vectors from TCR and from epitope are concatenated into a 160-dimensional vector. A dense layer with sigmoid activation is applied, followed by dropout. A final sigmoid neuron outputs a binding probability between 0 and 1.

C.2 Authenticity

- **GPT-LL**: GPT-LL is a GPT-style model fine-tuned on a protein sequence model [17] to learn patterns present in four million real TCR sequences from TCR repertoire data (ImmunoSeq, [16]). It comprises 24 transformer layers and a total of 300 million parameters, outputting a probability distribution over all possible amino acids in the vocabulary. Naturally occurring TCRs exhibit unique patterns due to their generation process: they are formed by recombining three different gene segments of multiple alleles and incorporating random base substitutions at the junctions. The log-likelihood score from a model trained on real TCRs can help identify these characteristic patterns. We calculate the GPT-LL log-likelikhood of each amino acid token in the generated TCR sequences and average these values to obtain a measure of authenticity. This score ranges from -Inf to Inf. Lower log-likelihood scores indicate a higher degree of anomaly, meaning the sequence is less likely to be a real TCR. Conversely, higher log-likelihood scores suggest a higher degree of authenticity.
- **TCRMatch**: TCRMatch [20] is a k-mer–based algorithm that quantifies similarity between two TCR sequences by decomposing them into overlapping k-mers and summing normalized similarity scores across all k values. To estimate the binding affinity of a generated TCR to a given epitope, we randomly sample 50 known binder TCRs for that epitope and compute the TCRMatch score between the generated TCR and each sampled binder. The final score is the maximum of these 50 pairwise scores, ranging from 0 (no similarity) to 1 (identical binding profile). Although originally designed for binding-prediction, we repurpose TCRMatch as an authenticity metric, since it effectively distinguishes true binder-like sequences from random or non-binding sequences.
- **BLOSUM62 Score**: We compute a BLOSUM62 alignment score for each generated TCR by performing a local pairwise alignment against a set of experimentally validated TCRs for the same epitope using Biopython's `PairwiseAligner`. The BLOSUM62 matrix assigns positive scores to biologically favorable (conservative) amino acid substitutions, negative scores to unlikely substitutions, and zero to neutral changes. For each generated TCR, we record the highest alignment score among all pairwise comparisons as its BLOSUM62 score. A higher value indicates a closer resemblance to known binding TCRs, providing a quantitative measure of biological plausibility and epitope specificity for the generated sequence.

- **Bit Score**: We measure similarity between generated and natural TCR sequences using BLAST [32]. For each epitope, we build a BLAST database from experimentally validated binding TCRs. Generated TCRs are formatted as FASTA queries and aligned against this database, with BLAST parameters set to report only high-confidence local alignments (E-value < 0.001). The resulting Bit score S' reflects both the raw alignment quality and its statistical significance:

$$S' = \frac{\lambda S - \ln K}{\ln 2} \tag{4}$$

Here, S is the raw alignment score, and λ and K are constants determined by the chosen scoring matrix (BLOSUM62) and gap penalties. A higher S' indicates a better alignment.

We also compute the E-value E to quantify the expected number of random matches:

$$E = \frac{m \cdot n}{2^{S'}} \tag{5}$$

where m is the query sequence length and n is the total length of the database. A smaller E-value signifies a more statistically significant alignment. Higher Bit scores combined with low E-values indicate greater sequence similarity and potential biological relevance.

- **VAE-loss**: We use a variational autoencoder (VAE) reconstruction loss as an authenticity indicator. A VAE encodes each TCR sequence into a compressed latent representation and then reconstructs it; authentic sequences will exhibit low reconstruction loss. We adapt the VAE architecture of Davidsen *et al.* [11], modifying it to accept only CDR3 sequences (the original model also used V and J gene segments). Each CDR3 is embedded via the BLOSUM62 matrix and processed by bidirectional LSTM layers. The training objective combines reconstruction loss (measuring the fidelity of sequence regeneration) with a Kullback–Leibler divergence term (ensuring the latent distribution matches a standard normal). We optimize with Adam and apply early stopping: training ends if validation loss fails to improve for 30 epochs or after 200 total epochs. We ultimately excluded VAE-loss from our main results, as it proved unable to distinguish real TCRs from random sequences.

C.3 Structural Binding Analysis

To provide structural context for generated CDR3β sequences, solved TCR:pMHC complexes known to bind GILGFVTL (PDB IDs: 1OGA, 2VLJ, 2VLK, 2VLR, 5EUO, and 5ISZ) were used as templates. During the evaluation, selected generated CDR3β sequences of interest were globally aligned to these reference CDR3β loops (Needleman–Wunsch [33], gap open penalty = –7, gap extension penalty = –1) and substituted into the closest-matching structure. Full models were then predicted with AlphaFold3 [34], followed by Rosetta [35] Relax minimization to remove steric clashes and optimize side-chain packing.

Interface energetics were evaluated with Rosetta's `InterfaceAnalyzer`, which reports the predicted Gibbs free energy of separation ($\Delta G_{\text{separated}}$), buried hydrophobic surface area (dSASA$_{\text{hphobic}}$), and interfacial hydrogen bonds (hbonds_int). Hydrogen bonds were defined using standard geometric constraints with a donor–acceptor distance cutoff of 3.5 Å, and hydrophobic interactions were defined by nonpolar contacts within 5 Å.

Molecular visualizations were generated in ChimeraX [36] (v1.10rc). Hydrogen bonds are shown as red dashed lines, with the numbers indicating donor–acceptor distances in Ångström. Hydrogen bonds shorter than $\sim$2.5 Å are generally considered strong, whereas those longer than $\sim$3.5 Å are weak or negligible [37]. Labels such as VAL6 or THR8 correspond to residue identity and position within the specified chain (epitope = chain C; TCRβ = chain E).

C.4 Identification of optimal threshold

We identify the optimal thresholds for both criteria by maximizing Youden's Index (J) [21]. Detailed procedure is as follow.

- **Curation of binding and non-binding TCRs**: To determine the optimal threshold for the binding affinity measures, we curate sets of binding and non-binding TCRs specific to the target epitopes. For binding TCRs, we randomly source experimentally validated TCRs known to interact with the target epitopes from our test set. For non-binding TCRs, we follow a common practice [4, 38, 8], as there is a limited availability of experimentally validated non-binding TCRs. We source TCRs from healthy repertoires (ImmunoSEQ [16]) and randomly pair with target epitopes, generating non-binding TCR-epitope pairs. This resulted in 2,900 binding and 2,900 non-binding TCR-epitope pairs (100 binding and non-binding TCRs per each target epitope).

- **Curation of authentic and fake TCR**: To determine the optimal threshold for the authenticity measure, we assemble one set of *authentic* TCRs from human repertoires and three distinct sets of *synthetic fake* TCRs: random-sequence, Prior-based, and GPT-LL–generated. In total, we curate 2,900 authentic and 2,900 fake TCRs for threshold selection.

  Authentic TCRs We randomly sample 2,900 TCRβ CDR3 sequences (lengths ranging from 10 to 20 amino acids) from ImmunoSeq [16], representing bona fide human repertoires. These sequences serve as positive examples for authenticity.

  Random-sequence fakes (Naive) To create "easy" negatives, we generate 2,900 sequences by sampling each amino acid position independently and uniformly from the 20 standard residues. We preserve the length distribution of the authentic set by drawing each random TCR's length from the empirical length histogram of the 2,900 real CDR3s. Because these sequences ignore both positional biases and contextual dependencies, they are trivially distinguishable from authentic TCRs.

  Prior-based fakes (Intermediate) We next synthesize 2,900 "Prior" fake TCRs using a position-specific amino acid profile. First, we compute, for each CDR3 position $i \in \{1, \ldots, L_{\max}\}$, the empirical frequency $p_i(a)$ of amino acid a across four million human TCRs from ImmunoSeq. Then, for each authentic CDR3 of length L, we generate a fake CDR3 of length L by sampling each residue $r_i \sim \mathrm{Categorical}(p_i)$. These Prior fakes preserve realistic per-position amino acid frequencies but lack higher-order sequence correlations.

  GPT-LL fakes (Refined) Finally, we generate 2,900 "difficult" fake TCRs by sampling from our GPT-LL model (Section 4.4). Because GPT-LL was trained on four million real CDR3s, these sampled sequences can exhibit realistic sequence motifs and dependencies. However, they do not necessarily bind any known epitope, making them challenging negatives.

  Together, the three fake sets span a spectrum of difficulty: random-sequence fakes are easily filtered out, Prior fakes require a per-position profile check, and GPT-LL fakes closely resemble real TCRs but still differ in epitope specificity. We use these curated sets to (1) test whether authenticity metrics (GPT-LL, TCRMatch, BLOSUM62, bit score) yield significantly different scores on authentic versus fake TCRs, and (2) select optimal thresholds via Youden's index.
- **Identifying the best threshold via Youden's Index J**: We identify the optimal cut-off point for each evaluation metric by maximizing Youden's Index (J): $J = \text{sensitivity} + \text{specificity} - 1$. This is equivalent to maximizing the true positive rate while minimizing the false positive rate. The curated binding and non-binding TCR groups serve as positive and negative sets for determining binding affinity metrics thresholds. The curated authentic and fake TCR groups are used for establishing an authenticity metric threshold. A simple search iterating through a set of possible values between 0 and 1 identifies the threshold that yields the maximum value. Once these thresholds are established, a generated TCR is classified as 'high-quality' or 'good' TCR only if it surpasses all three metrics' established thresholds.

  The resulting optimal thresholds (by Youden's Index) for each metric are summarized in Table A2. We also computed thresholds by minimizing the Euclidean distance to the top-left corner of the ROC curve; these values closely match the Youden-based cutoffs, so we report only the latter. Distribution comparisons between high- and low-quality groups appear in Figure A4.

D Additional Figures and Tables

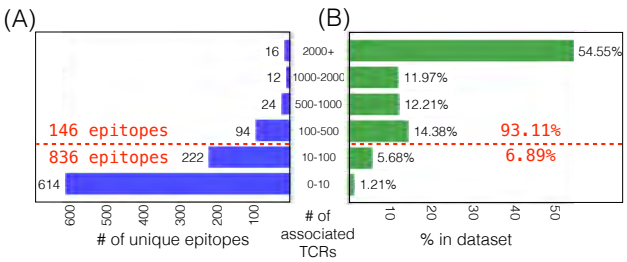

Figure A1: (A) The number of unique epitopes plotted against the number TCRs that recognize them and (B) percentage of each category in the dataset.

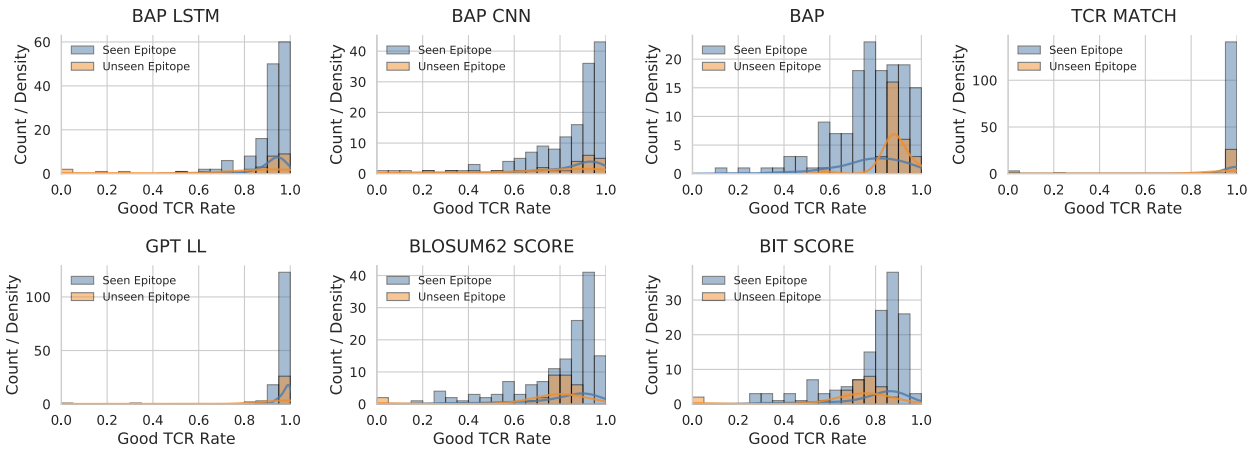

Figure A2: Distribution of "good" TCR rates for each evaluation metric, comparing seen versus unseen epitopes under standard 0-shot generation. Each subplot shows a histogram of per-epitope good-TCR rates (i.e., the fraction of generated TCRs passing both binding-affinity and authenticity thresholds) for seen epitopes (blue) and unseen epitopes (orange). Across all metrics (BAP LSTM, BAP CNN, BAP MLP, TCRMatch, GPT-LL, BLOSUM62 score, and bit score), the unseen-epitope distributions are shifted lower and exhibit greater variance, illustrating the increased difficulty of generating high-quality TCRs for novel epitopes.

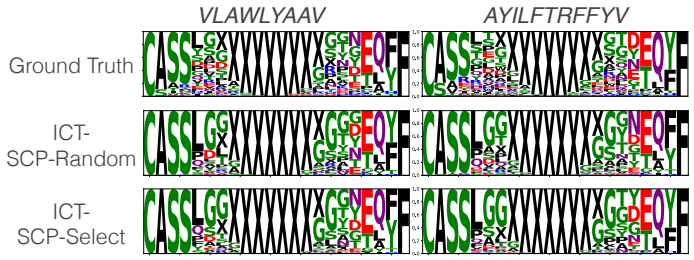

Figure A3: Visualization of ground-truth and generated TCRs by ICT-SCP-Random and ICT-SCP-Select using a 4-shot inference. Results of the two epitopes *VLAWLYAAV* and *AYILFTRFFYV* are presented. Notably, the SCP method operates without requiring known binding TCRs as part of the in-context prompts, relying instead on self-generated TCRs.

Table A1: Classification of models used in all experiments based on data used in training and at the time of prompting respect to query epitope E.

Q: Do E and its known binding TCRs appear in training?	Yes (*Seen E*)		No (*Unseen E*)	
Q: Any known binding TCRs for E provided at the time of prompting?	Yes	No	Yes	No (***Novel E***)
Sect 4.1: Vanilla-0-shot (Seen E)		✓		
Sect 4.2: Vanilla-FSP Sect 4.2: ICT-FSP			✓	
Sect 4.1, 4.2: Vanilla-0-shot Sect 4.2: ICT-0-shot Sect 4.3: ICT-SCP-Select/Random/Chain				✓

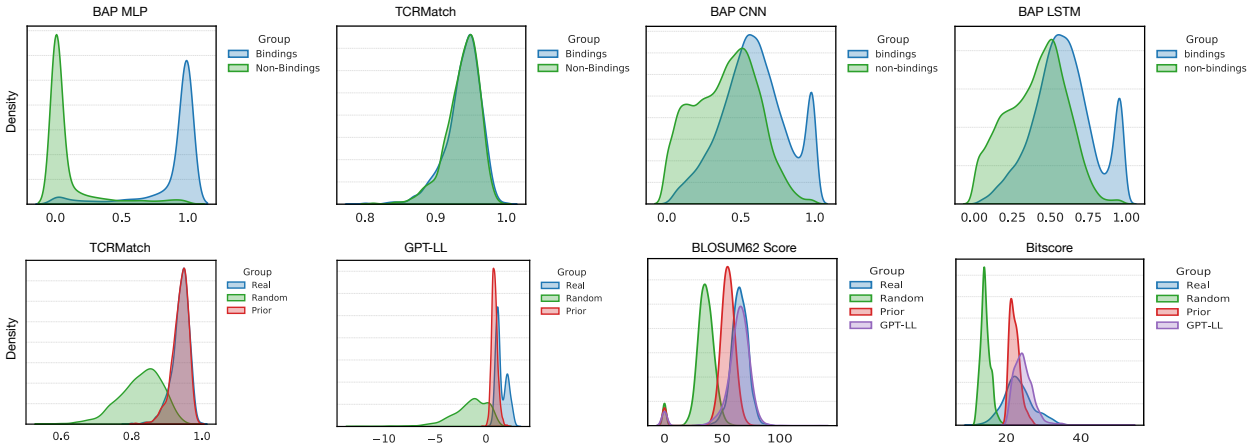

Figure A4: Density distributions of evaluation metrics on curated TCR sets. *Top row*: For binding-affinity metrics (BAP MLP, TCRMatch, BAP CNN, BAP LSTM), we plot score densities for known binding TCRs (blue) versus non-binding TCRs (green). In each panel, binding TCRs concentrate at higher scores while non-binding TCRs cluster at lower values, demonstrating clear separation. *Bottom row*: For authenticity metrics, we compare real (authentic) TCRs (blue) against three synthetic negative sets: random sequences (green), Prior-based fakes (purple), and GPT-LL–generated fakes (red). Each metric (TCRMatch, GPT-LL, BLOSUM62 Score, Bit Score) shows distinct density peaks for real versus fake sequences, confirming their ability to discriminate between authentic TCRs. GPT-LL fake was excluded in the computational models TCRMatch and GPT-LL.

Table A2: Optimal cutoffs, false positive rates (FPR), and true positive rates (TPR) for each metric, computed by Youden's index and by minimum-distance.

Metric	**Youden's J**			**Min-Distance**		
	Cutoff	**FPR**	**TPR**	**Cutoff**	**FPR**	**TPR**
BAP MLP	0.473	0.078	0.916	0.473	0.078	0.916
TCRMatch	0.941	0.508	0.537	0.941	0.506	0.534
BAP CNN	0.515	0.325	0.639	0.503	0.347	0.660
BAP LSTM	0.523	0.304	0.630	0.511	0.331	0.656
GPT-LL	1.063	0.205	0.776	1.061	0.207	0.778
BLOSUM62 Score	60.000	0.203	0.807	60.000	0.203	0.807
Bit Score	17.300	0.035	0.948	17.300	0.035	0.948

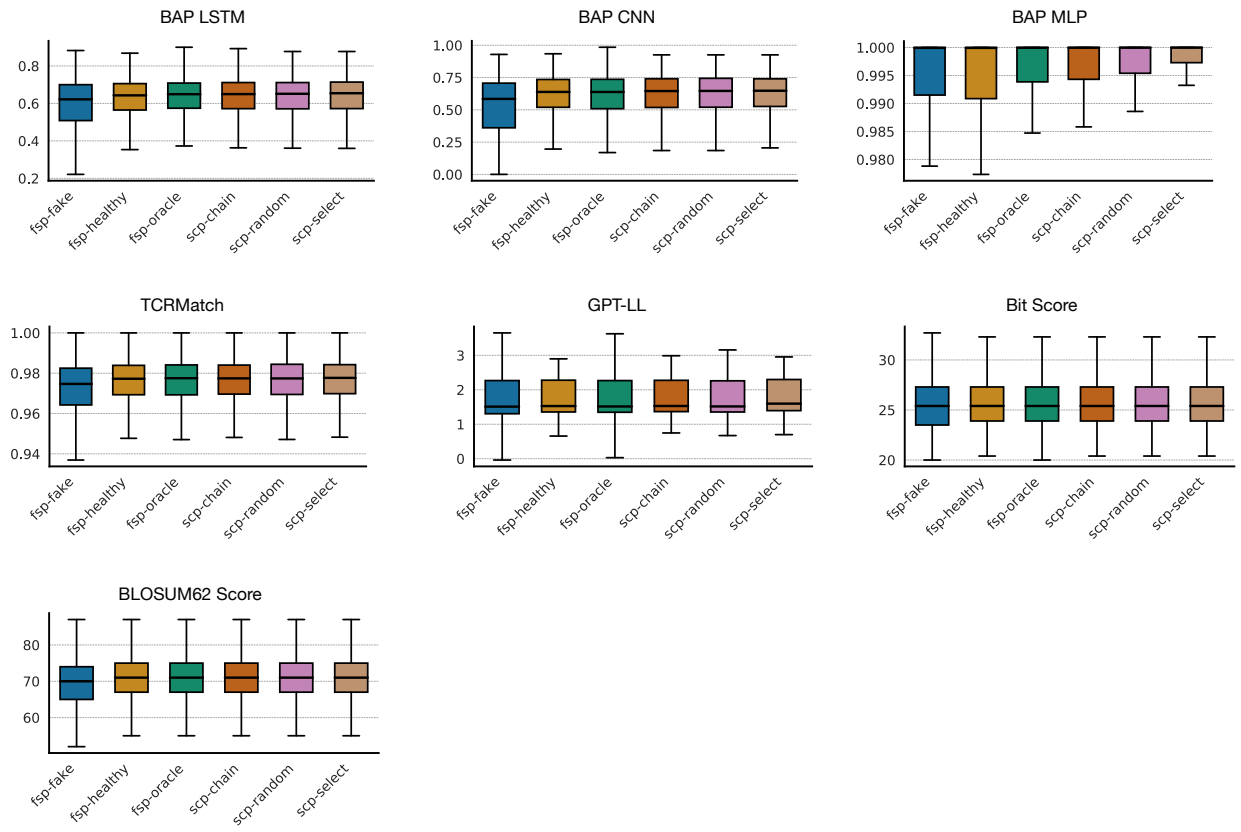

Figure A5: Comparison of metric distributions under FSP (Few-Shot Prompting) and SCP (Self-Contemplation Prompting) for novel epitopes. Each boxplot shows the distribution of "good TCR" rates across epitopes for one metric and one prompting method. The first three methods (FSP-Fake, FSP-Healthy, FSP-Oracle) use ground-truth or non-binding in-context examples; the latter three (SCP-Chain, SCP-Random, SCP-Select) use self-generated TCRs. Results are shown for seven evaluation metrics: BAP LSTM, BAP CNN, BAP MLP, TCRMatch, GPT-LL, Bit Score, and BLOSUM62 Score. Most prompting methods achieve similar performance and consistently outperform FSP-Fake. This diverse range of prompting techniques in our framework allows users to choose a flexible prompting strategy based on their available data and preferences.

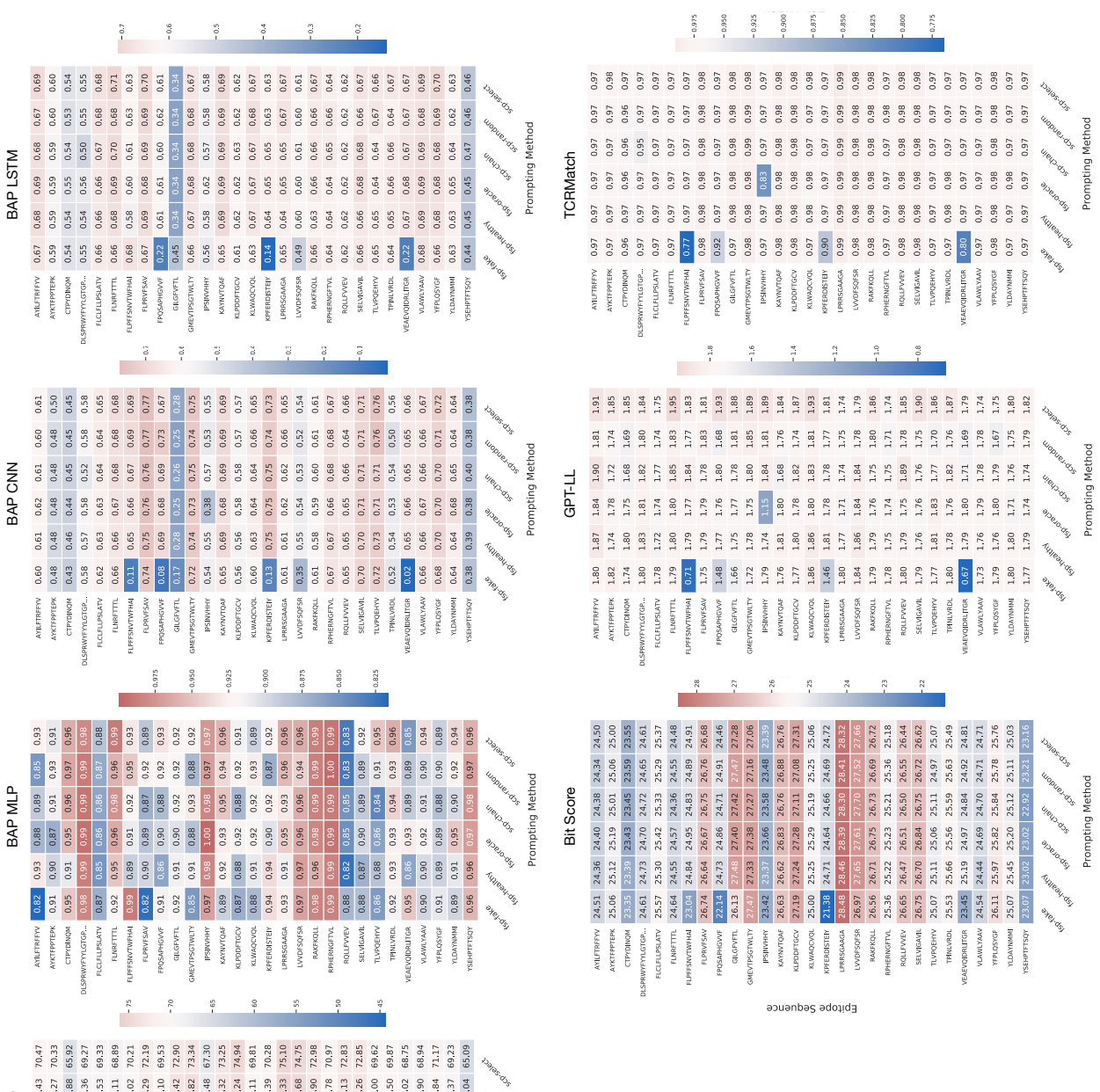

Figure A6: Per-epitope performance of different prompting methods for each evaluation metric. Each subplot (organized by metric) is a heatmap where rows correspond to individual epitopes and columns correspond to prompting methods: FSP-Fake, FSP-Healthy, FSP-Oracle, SCP-Chain, SCP-Random, and SCP-Select. Cell color represents the "good TCR" rate (higher values in red, lower values in blue). These heatmaps reveal epitope-specific variability—some epitopes benefit more from certain prompting strategies than others.

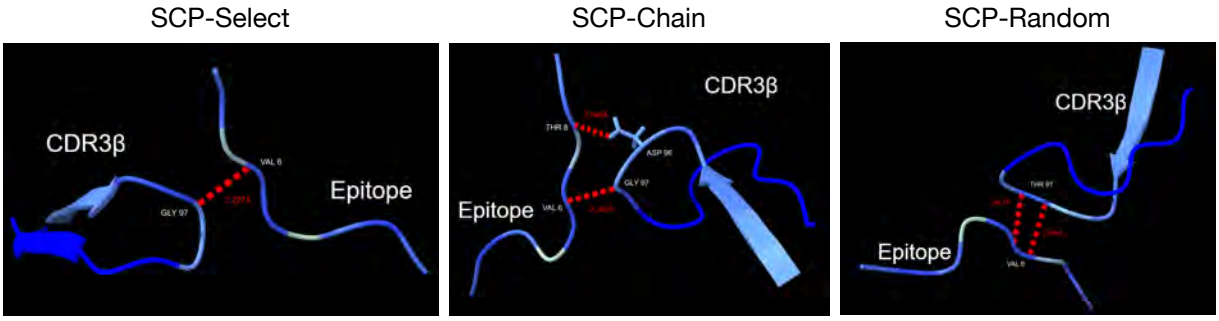

Figure A7: Examples of hydrogen-bond–mediated interactions between the epitope `GILGFVFTL` and generated TCR CDR3β loops under different self-contemplating prompting strategies (SCP-Select: `CASSLVGGGEQYF`, SCP-Chain: `CASSLDGLGLNTEAFF`, SCP-Random: `CASSPGTGGPGELFF`). Red dashed lines indicate hydrogen bonds, with numbers denoting bond lengths in Å. Hydrogen bonds shorter than $\sim$2.5 Å are generally considered strong, whereas those longer than $\sim$3.5 Å are weak or negligible. Accordingly, shorter distances indicate stronger interactions, and a greater number of red dashed lines reflects more extensive intermolecular contacts. The observed short distances and multiple contacts highlight that SCP-based designs can yield structurally plausible TCR–epitope interactions.