PeptideMTR: A Dual-Objective Approach to Building Foundation Models for Therapeutic Peptides

Anonymous Author(s)
Affiliation
Address
email

Abstract

Foundation models for molecular science have significantly impacted small-molecule and protein modeling, however there is a lack of models able to encode therapeutic peptides. Existing chemical language models often operate with short context windows, while protein language models are limited to canonical amino acids and struggle with nonnatural residues, modifications, or cyclizations. We present PeptideMTR, a SMILES-based foundation model with multimodal pretraining via descriptor alignment. PeptideMTR couples masked language modeling with an auxiliary regression objective to RDKit-derived physicochemical descriptors, aligning symbolic sequence representations with continuous chemical properties. Our contributions are threefold: (i) a kmer tokenizer tailored to chemically coherent fragments and peptide motifs, (ii) a dual-objective pretraining scheme that unifies symbolic and numeric modalities, and (iii) an empirical study of the impact scaling from 32M to 337M parameters has on predicting peptide permeability and aggregation. PeptideMTR consistently outperforms fingerprint baselines and MLM-only pretraining, demonstrating that multimodal pretraining yields richer peptide representations.

1 Introduction

Peptides are an increasingly important therapeutic class [1, 2, 3, 4, 5, 6], occupying an intermediate size range that falls between small molecules and proteins. Drug-like peptides frequently include noncanonical residues [7], cyclization [8, 9], and backbone/side-chain modifications [10]. These features challenge standard machine-learning toolkits: protein language models are tied to amino-acid alphabets and are unable to incorporate non-canonical amino acids [11], and chemical language models trained on small molecules require adaptation to peptide-specific motifs and long-range interactions beyond steric hindrance [12, 13].

Most prior work treats representation learning as single-modal, focused on either discrete sequence modeling or purely numeric descriptors. For peptides, however, useful representations must reconcile symbolic sequence context with continuous physicochemical behavior. We address this by coupling a SMILES tokenizer tailored to chemically coherent peptide fragments with a dual pretraining objective that aligns sequence predictions with molecule-level properties.

We introduce PeptideMTR, a BERT-style encoder [14] trained with a dual objective: masked language modeling (MLM) over kmer SMILES tokens and multi-task regression (MTR) [15] to RDKit [16] computed physicochemical descriptors. This blended pretraining ties local sequence syntax to global chemical attributes, yielding embeddings that transfer to peptide property prediction. We study scaling from 32M to 337M parameters and perform ablations over tokenizer, masking strategy, and the inclusion of MTR. Empirically, PeptideMTR outperforms fingerprint baselines on multiple peptide endpoints, improves over MLM-only pretraining, and achieves results that outclass prior literature. Pretrained models of small, base, and large are availabe at https://huggingface.co/subm-123abc.

2 Related work

2.1 Protein Language Models

Large-scale corpora such as UniProt [17] have enabled the training of protein language models (pLMs) at the scale of hundreds of millions to billions of sequences. Representative models include the ESM-family of models [18, 19, 20] and ProtTrans [21], demonstrating that self-supervised training objectives on large sequence databases can produce embeddings informative of protein structure and function. It is hypothesized that MLM provides a method to learn long-range dependencies and implicit biophysical constraints [22]. Beyond MLM, other pretraining strategies such as autoregressive objectives [23], contrastive learning [24, 25], multi-task pretraining [26], and multi-model inputs [20] have been explored. Finetuned pLMs achieve strong results on tasks including secondary-structure prediction [27], mutation-effect estimation [28, 29], protein-protein interaction [30], and protein structure prediction [19].

2.2 Chemical Language Models (CLMs)

Chemical language models (CLMs) apply sequence modeling to small molecules, with examples such as SMILES-BERT [31], ChemBERTa [32, 33], and MoLFormer [34]. Transformer-based CLMs leverage string notations (e.g., SMILES, SELFIES) to represent molecules in a tokenizable form [35, 36]. Pretraining adopts NLP-style objectives, MLM or auto-regressive modeling, applied to large molecular databases like PubChem [37]. Multi-task regression (MTR) for CLMs was introduced in ChemBERTa-2 [32], predicting molecular properties for compounds during pretraining, and resulting in improvements over MLM pretraining. A CLM pre-trained on both small molecules and peptides, PeptideCLM [38], demonstrated that SMILES-based CLMs can capture sequence–structure relationships relevant to membrane interactions for cyclic peptides. We acknowledge the large field of molecular graph representations, but do not cover it in this work.

3 Methods

3.1 Kmer Tokenization Strategy

Based on peptide tokenization methods in PeptideCLM [38] and the concepts from SmilesPE [39], we built a peptide tokenizer with a smaller vocabulary than related work and a higher compression ratio than a simple atomistic tokenizer. We first created a pretokenizer able to identify individual atoms. This included Br/Cl as separate from B/C and bracketed text, denoting chirality or ionic charge (e.g., [N+] or [C@@H]). We then evaluated all kmers of up to 6 characters from 200,000 small molecules from PubChem and ChEMBL and 200,000 peptides from SmProt [40]. We filtered this list to select for the highest occurring kmers that followed several rules (Table 1), resulting in 160 single atom tokens and 405 total tokens. Improved tokenization compression was achieved with our kmer tokenizer compared to the DeepChem tokenizer [41]. This method provides a 62% reduction in encoding length for a random sample of 10,000 small molecules from PubChem, and a 36% reduction for a random sample of 10,000 peptides from ESMAtlas (Figure S2).

Table 1: Filtering rules applied during construction of the kmer vocabulary.

#	**Remove tokens:**
1	with numbers
2	that start with `')'`
3	that end with `'('`
4	that contain `')*'` w/o a leading `'*('`
5	with more than 4 atom characters
6	with fewer than 1,000 occurrences

Table 2: Transformer encoder configurations for three sizes of models.

Model	**Small**	**Base**	**Large**
Layers (l)	14	24	32
Hidden dim. (d)	512	768	1024
FF dim.	768	1024	2048
Heads (h)	8	12	16
Max length	2048	2048	2048
Parameters	32M	114M	337M

3.2 BERT-style Model Architecture

Our model follows a BERT-style transformer encoder. SMILES strings are tokenized with the kmer vocabulary and embedded into d dimensions. Each of l layers contains multi-head self-attention with rotational embeddings, SwiGLU feed-forward layers, Pre-LN normalization, residuals, and dropout. We trained three model sizes: small (32M), base (114M), and large (337M), with hidden dimension d scaling with model size and h attention heads, each with a dimension of 64 (Table 2).

3.3 MLM and MTR Pretraining

Pretraining datasets (Table S1) included PubChem, ESMAtlas and LMSD (Figure S1). Pubchem was filtered to remove molecules shorter than 20 characters and molecules containing silicon chains. Peptides included ESMAtlas [19] clustered at 30% sequence similarity, > 0.7 pTM and pLDDT, and length under 100 AA. Lipids were included from LMSD [42]. Balanced sampling per epoch was introduced, with lipids upsampled to 250k, peptides included the full dataset of nearly 10M, and small molecules down sampled to 10M.

Masked language modeling (MLM) was performed with a 25% masking rate either randomly selected or in multiple spans drawn from a Gaussian distribution ($\mu = 3.5$, $\sigma = 1$), with all selected sites replaced by the `[MASK]` token [43]. For multi-task regression (MTR), on the same masked input in parallel, an MTR head (two fully-connected layers with SiLU [44] activation) using a mean-pooled sequence embedding to predict physicochemical properties. The properties were a set of 99 normalized physicochemical descriptors computed by RDKit (Table S3). We selected a subset of the total RDKit descriptors available based on run speed and features important for peptide chemistry. The total loss combined MLM and MTR as a weighted sum:

$$\mathcal{L} = \lambda_{\text{MLM}} \cdot \mathcal{L}_{\text{MLM}} + \lambda_{\text{MTR}} \cdot \mathcal{L}_{\text{MTR}},$$

with weights set as $\lambda_{\text{MLM}} = 0.6$ and $\lambda_{\text{MTR}} = 0.4$, to balance convergence speed and downstream performance. SMILES strings were randomly canonicalized for improved generalization [45].

3.4 Finetuning for Downstream Prediction of Labeled Data

Finetuning was conducted using nested cross-validation, where an outer loop established a fixed holdout test set and an inner loop trained ensembles by sequentially holding out each non-test fold, as previously described [38]. This approach mimics prediction on unseen data, allowing for early stopping to prevent overfitting to the test distribution. Model checkpoints were selected by validation loss, and final predictions were an average of outputs from the ensemble. Finetuning for each model was repeated three times with performance reported as the mean and standard error across replicates.

4 Experiments

4.1 Effect of Tokenization, Pretraining Objective, Masking Strategy, and Model Scale on Prediction of Membrane Permeation for Cyclic Peptides

Effects of pretraining on downstream finetuning were assessed with a dataset of measured membrane permeability for cyclic peptides [46]. Results show a marked increase in performance when using MTR in addition to MLM (Table 3). At small scale (32M parameters), atomistic tokenization using the DeepChem tokenizer matched the performance of kmer tokenization but had a higher computational cost due to an increased number of tokens. Span masking consistently outperformed random masking across larger model sizes (base/large), suggesting that masking longer contiguous fragments enables the model to better capture peptide motifs. Additionally, models trained with span masking had lower variance across multiple finetuning runs than models with random masking.

Scaling model parameters from 32M to 337M yielded steady improvements across all metrics (Table 3). While performance continued to increase with size on the cyclic peptide data, gains diminished at the largest scale, suggesting that dataset variance rather than model capacity becomes the limiting factor. The 337M span-masked model achieved the strongest results overall and outperformed PeptideCLM and ChemBERTa-2, which achieved AUROC of 0.78 and 0.74 on the same dataset in a previous study [38].

Table 3: Ablation studies of PeptideMTR models on cyclic peptide permeability prediction, in ascending order by AUROC. (* model randomly initialized prior to finetuning.)

Size	Tokenizer	Masking	MTR	$R^2 \uparrow$	RMSE $\downarrow$	AUROC $\uparrow$	AUPRC $\uparrow$
Small	Kmer	*	*	$-0.08 \pm .03$	$0.80 \pm .01$	$0.64 \pm .01$	$0.67 \pm .01$
Small	Kmer	Span	No	$0.13 \pm .09$	$0.72 \pm .03$	$0.68 \pm .01$	$0.69 \pm .01$
Small	Kmer	Random	Yes	$0.24 \pm .08$	$0.68 \pm .04$	$0.70 \pm .02$	$0.74 \pm .01$
Small	Kmer	Span	Yes	$0.23 \pm .05$	$0.68 \pm .02$	$0.70 \pm .01$	$0.74 \pm .01$
Small	Atom	Random	Yes	$0.25 \pm .06$	$0.67 \pm .03$	$0.71 \pm .01$	$0.74 \pm .01$
Base	Kmer	Random	Yes	$0.39 \pm .09$	$0.59 \pm .04$	$0.79 \pm .01$	$0.77 \pm .01$
Base	Kmer	Span	Yes	$0.52 \pm .02$	$0.54 \pm .01$	$0.81 \pm .01$	$0.78 \pm .01$
Large	Kmer	Random	Yes	$0.57 \pm .04$	$0.51 \pm .03$	$0.83 \pm .01$	$0.81 \pm .01$
Large	Kmer	Span	Yes	$\mathbf{0.58} \pm .01$	$\mathbf{0.50} \pm .01$	$\mathbf{0.83} \pm .00$	$\mathbf{0.81} \pm .00$

4.2 Predicting Peptide Aggregation

To further evaluate the applicability of the model to drug-like peptides, we finetuned the model to predict aggregation of chemically-modified linear peptides based on ThT assay data [47]. We compared the finetuned model against predictions from Morgan fingerprints [48] with a matching regression head (Table 4). ChemBERTa models will not work for these datasets as the length of the peptide SMILES is too large for the model context window. PeptideMTR outperformed the fingerprint baseline, with performance improving as model size increased (Figure S3). These results indicate that PeptideMTR learns transferable representations that generalize to important peptide endpoints, surpassing handcrafted descriptors.

Table 4: Performance of PeptideMTR compared to Morgan fingerprint baseline on prediction of peptide fibrillation. Results are mean of three runs on random seeds with standard error.

Dataset/Target	MFP	Small	Base	Large
Peptide Fibrillation (AUROC)	0.57 ± 0.003	0.71 ± 0.004	0.76 ± 0.004	0.83 ± 0.003

5 Limitations and Broader Impacts

While this work demonstrates the utility of the developed models, several limitations remain. First, the models rely on SMILES encodings, which lack direct higher-order structure often critical for peptide activity. Second, the regression head is trained against a restricted set of RDKit-derived physicochemical descriptors. These descriptors omit peptide-specific features such as conformational flexibility, secondary-structure propensity, or solvent accessibility. Finally, our models are modest in size and may show improved performance with an increased parameter count.

Despite these limitations, our results highlight the potential of foundation models focused on peptide chemistry. By jointly modeling SMILES sequences and continuous chemical descriptors, PeptideMTR demonstrates that multi-modal pretraining can improved representations for noncanonical peptide chemistry. More broadly, this work illustrates how foundation models can extend beyond canonical biomolecules, providing a roadmap for therapeutic peptide model development. While PeptideMTR has the potential for positive societal impacts in drug discovery and medicine, it is important to acknowledge the risk of dual-use concerns that may arise from its application.

6 Conclusions

We present PeptideMTR, a SMILES-based foundation model tailored for drug-like peptides that combines a chemically informed tokenizer with multi-objective pretraining. The model captures long-range dependencies, accommodates noncanonical residues, and produces transferable representations that outperform traditional molecular fingerprints after finetuning. These results demonstrate that peptide-specific language models can serve as a scalable and effective framework for advancing peptide drug discovery.

7 References

References

[1] Markus Muttenthaler, Glenn F King, David J Adams, and Paul F Alewood. Trends in peptide drug discovery. *Nature reviews Drug discovery*, 20(4):309–325, 2021.

[2] Bethany M Cooper, Jessica Iegre, Daniel H O'Donovan, Maria Ölwegård Halvarsson, and David R Spring. Peptides as a platform for targeted therapeutics for cancer: peptide–drug conjugates (pdcs). *Chemical society reviews*, 50(3):1480–1494, 2021.

[3] Lei Wang, Nanxi Wang, Wenping Zhang, Xurui Cheng, Zhibin Yan, Gang Shao, Xi Wang, Rui Wang, and Caiyun Fu. Therapeutic peptides: current applications and future directions. *Signal transduction and targeted therapy*, 7(1):48, 2022.

[4] John Fetse, Sashi Kandel, Umar-Farouk Mamani, and Kun Cheng. Recent advances in the development of therapeutic peptides. *Trends in pharmacological sciences*, 44(7):425–441, 2023.

[5] Panchali Barman, Shubhi Joshi, Sheetal Sharma, Simran Preet, Shweta Sharma, and Avneet Saini. Strategic approaches to improvise peptide drugs as next generation therapeutics. *International journal of peptide research and therapeutics*, 29(4):61, 2023.

[6] Komal Sharma, Krishna K Sharma, Anku Sharma, and Rahul Jain. Peptide-based drug discovery: Current status and recent advances. *Drug Discovery Today*, 28(2):103464, 2023.

[7] Jennifer L Hickey, Dan Sindhikara, Susan L Zultanski, and Danielle M Schultz. Beyond 20 in the 21st century: prospects and challenges of non-canonical amino acids in peptide drug discovery. *ACS Medicinal Chemistry Letters*, 14(5):557–565, 2023.

[8] Huiya Zhang and Shiyu Chen. Cyclic peptide drugs approved in the last two decades (2001–2021). *RSC Chemical Biology*, 3(1):18–31, 2022.

[9] Xinjian Ji, Alexander L Nielsen, and Christian Heinis. Cyclic peptides for drug development. *Angewandte Chemie*, 136(3):e202308251, 2024.

[10] Christina Lamers. Overcoming the shortcomings of peptide-based therapeutics. *Future Drug Discovery*, 4(2):FDD75, 2022.

[11] Zhenjiao Du, Doina Caragea, Xiaolong Guo, and Yonghui Li. Pepbert: Lightweight language models for bioactive peptide representation. *bioRxiv*, pages 2025–04, 2025.

[12] Leyao Wang, Rishab Pulugurta, Pranay Vure, Yinuo Zhang, Aastha Pal, and Pranam Chatterjee. Pepdora: A unified peptide language model via weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2410.20667*, 2024.

[13] Raúl Fernández-Díaz, Rodrigo Ochoa, Thanh Lam Hoang, Vanessa Lopez, and Denis Shields. How to build machine learning models able to extrapolate from standard to modified peptides. *ChemRxiv*, 2025.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[15] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larranaga. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233, 2015.

[16] Greg Landrum. Rdkit documentation. *Release*, 1(1-79):4, 2013.

[17] The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2025. *Nucleic acids research*, 53(D1):D609–D617, 2025.

[18] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[19] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[20] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

[21] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: towards cracking the language of life's code through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:7112–7127, 2021.

[22] Zhidian Zhang, Hannah K Wayment-Steele, Garyk Brixi, Haobo Wang, Dorothee Kern, and Sergey Ovchinnikov. Protein language models learn evolutionary statistics of interacting sequence motifs. *Proceedings of the National Academy of Sciences*, 121(45):e2406285121, 2024.

[23] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. Protein design and variant prediction using autoregressive generative models. *Nature communications*, 12(1):2403, 2021.

[24] Michael Heinzinger, Maria Littmann, Ian Sillitoe, Nicola Bordin, Christine Orengo, and Burkhard Rost. Contrastive learning on protein embeddings enlightens midnight zone. *NAR genomics and bioinformatics*, 4(2):lqac043, 2022.

[25] Amy X Lu, Haoran Zhang, Marzyeh Ghassemi, and Alan Moses. Self-supervised contrastive learning of protein representations by mutual information maximization. *BioRxiv*, pages 2020–09, 2020.

[26] Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, and Yonghong Tian. Prollama: A protein large language model for multi-task protein language processing. *IEEE Transactions on Artificial Intelligence*, 2025.

[27] Wafa Alanazi, Di Meng, and Gianluca Pollastri. Porter 6: protein secondary structure prediction by leveraging pre-trained language models (plms). *International Journal of Molecular Sciences*, 26(1):130, 2024.

[28] Yuanfei Sun and Yang Shen. Structure-informed protein language models are robust predictors for variant effects. *Human Genetics*, 144(2):209–225, 2025.

[29] Sarah Gurev, Noor Youssef, Navami Jain, and Debora Marks. Variant effect prediction with reliability estimation across priority viruses. *bioRxiv*, pages 2025–08, 2025.

[30] Dan Liu, Francesca Young, Kieran D Lamb, Adalberto Claudio Quiros, Alexandrina Pancheva, Crispin Miller, Craig Macdonald, David L Robertson, and Ke Yuan. Plm-interact: extending protein language models to predict protein-protein interactions. *bioRxiv*, pages 2024–11, 2024.

[31] Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, pages 429–436, 2019.

[32] Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712*, 2022.

[33] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

[34] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022.

[35] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[36] Mario Krenn, Qianxiang Ai, Senja Barthel, Nessa Carson, Angelo Frei, Nathan C Frey, Pascal Friederich, Théophile Gaudin, Alberto Alexander Gayle, Kevin Maik Jablonka, et al. Selfies and the future of molecular string representations. *Patterns*, 3(10), 2022.

[37] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2025 update. *Nucleic acids research*, 53(D1):D1516–D1525, 2025.

[38] Aaron L Feller and Claus O Wilke. Peptide-aware chemical language model successfully predicts membrane diffusion of cyclic peptides. *Journal of Chemical Information and Modeling*, 65(2):571–579, 2025.

[39] Xinhao Li and Denis Fourches. Smiles pair encoding: a data-driven substructure tokenization algorithm for deep learning. *Journal of chemical information and modeling*, 61(4):1560–1569, 2021.

[40] Yanyan Li, Honghong Zhou, Xiaomin Chen, Yu Zheng, Quan Kang, Di Hao, Lili Zhang, Tingrui Song, Huaxia Luo, Yajing Hao, et al. Smprot: a reliable repository with comprehensive annotation of small proteins identified from ribosome profiling. *Genomics, proteomics & bioinformatics*, 19(4):602–610, 2021.

[41] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019.

[42] Manish Sud, Eoin Fahy, Dawn Cotter, Alex Brown, Edward A Dennis, Christopher K Glass, Alfred H Merrill Jr, Robert C Murphy, Christian RH Raetz, David W Russell, et al. Lmsd: Lipid maps structure database. *Nucleic acids research*, 35(suppl_1):D527–D532, 2007.

[43] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[44] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

[45] Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Randomized smiles strings improve the quality of molecular generative models. *Journal of cheminformatics*, 11(1):71, 2019.

[46] Jianan Li, Keisuke Yanagisawa, Masatake Sugita, Takuya Fujie, Masahito Ohue, and Yutaka Akiyama. Cycpeptmpdb: a comprehensive database of membrane permeability of cyclic peptides. *Journal of Chemical Information and Modeling*, 63(7):2240–2250, 2023.

[47] Christine Xue, Tiffany Yuwen Lin, Dennis Chang, and Zhefeng Guo. Thioflavin t as an amyloid dye: fibril quantification, optimal concentration and effect on aggregation. *Royal Society open science*, 4(1):160696, 2017.

[48] Harry L Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113, 1965.

Appendix

S1. Datasets and Curation

All datasets were downloaded from the links in table S1. All databases were passed through RDKit, converted into a mol object, and then to a SMILES string. Any that failed were removed. Molecules already represented in the database as SMILES strings were converted using `Chem.MolFromSmiles(Chem.MolToSmiles(SMILES))`. Peptides represented as amino acid characters were converted using `Chem.MolFromSequence(Chem.MolToSmiles(AMINO_ACIDS))`.

PubChem was filtered to remove any molecules that contained a SMILES string length of less than 20. A second filter was done to remove anything following a '.' wholly contained in brackets (i.e. `CCO.[Br]`). Leading or trailing Br and Cl (salts) were removed from all molecules. All remaining lines with a '.' were split into two lines. Then, any duplicates were removed. Any molecules with a four silicon oxide repeat "[Si](=O)[Si](=O)[Si](=O)[Si](=O)" were removed. The resulting total dataset size was 108,583,157

ESMAtlas was downloaded with the following prefilters: MGnify90 is clustered down to 30% sequence similarity with `mmseqs easy-linclust -kmer-per-seq 100 -cluster-mode 2 -cov-mode 1 -c 0.8`. Structures are filtered to > 0.7 pTM and pLDDT. Structures are sorted by pTM * pLDDT and the best from each cluster is chosen as the representative.

We conducted further filtering to select sequences with 100 amino acids or less.

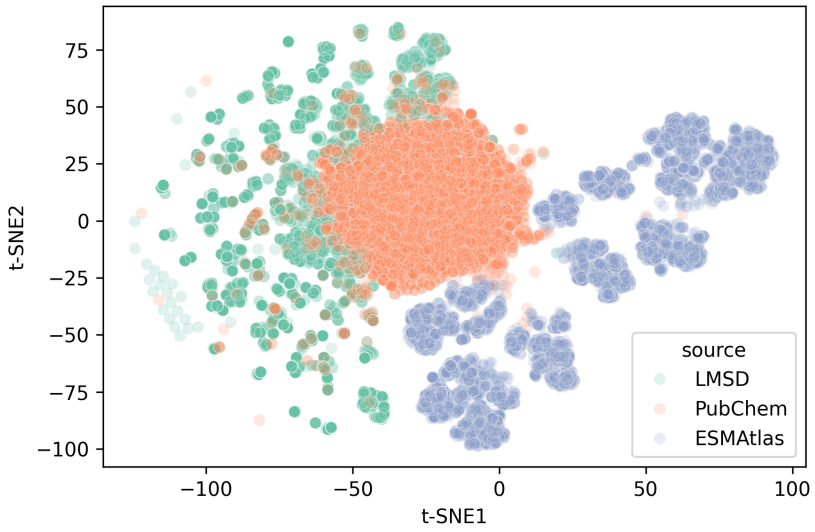

Figure S1: Scatter plot of 10,000 sampled molecules from the three datasets used in pretraining.

Table S1: Datasets used in pretraining.

Dataset	Modality	Count	Link
PubChem	Small molecules	108,583,157	pubchem.ncbi.nlm.nih.gov
ESMAtlas	Proteins / peptides	9,634,945	ESMAtlas Link
LMSD	Lipids	50k	lipidmaps.org/databases/lmsd

S2. Tokenizer

S2.1 Vocabulary and rules

Full description of tokenization methods are in Table 1. Full token list can be seen by loading the tokenizer from https://huggingface.co/subm-123abc. Example tokens to highlight multi-atom tokenization:

Table S2: Sample tokens from kmer tokenizer.

Tokens
=NC, #Cc, c(CO), n(C)c, n(C), CN=C, (CCO), CCO), C(C)N, C#C, C(CC), CS(=O), NC(C), CSc, NN, CC(O), CNc, CCn

S2.2 Compression and efficiency

To measure the differences in tokenized length between a naive atomistic tokenizer and our kmer strategy, we tokenized a random 10,000 peptides and a random 10,000 small molecules from the pretraining datasets. Compression rate between atomistic and kmer tokenizer for peptides was 0.36 and for small molecules was 0.62.

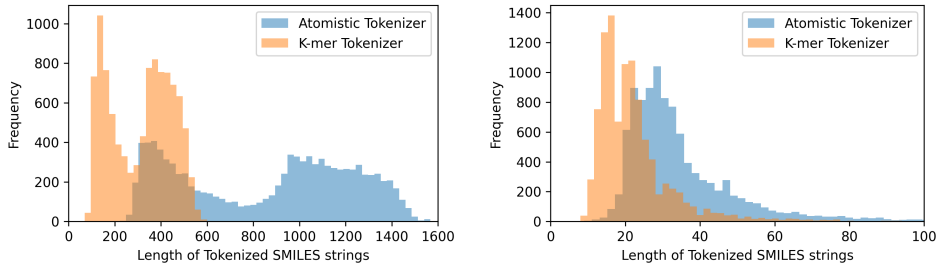

Figure S2: Comparison of tokenizers: (left) peptide length distribution, (b) small molecule length distribution.

S3. Pretraining Objectives

S3.1 Masking distributions

The masking procedure employs a Gaussian distribution to determine the lengths of token spans to be masked within sequences. Each sequence's length is used to calculate the total number of tokens to mask, based on a specified masking percentage.

For each sequence, the procedure samples span lengths with a mean of 3.5 and a standard deviation of 1.0, ensuring a minimum span of 1 token. A random starting position is selected for each span, and the end position is adjusted to prevent exceeding the sequence length. The process includes checking for overlapping positions to avoid masking the same token multiple times.

Masked positions are updated with a specified mask token ID, and the operation continues until the desired number of tokens is masked across all sequences.

S3.2 Descriptor head and targets

To produce logits for masked language modeling (MLM), a standard sequence head is applied to the last layer of the transformer.

For the descriptor output, mean pooling is performed on the embedded representations. The embeddings corresponding to valid (non-padded) tokens are summed, and the total count of these tokens is calculated. The mean of the embeddings is then computed by dividing the summed embeddings by the count of valid tokens, ensuring that only meaningful tokens contribute to the descriptor representation.

The descriptors are predicted with an MTR head that is made of two full-connected layers. The first is connected to a hidden vector of length that matches the embedding dimension. The second layer outputs a vector of length 99, matching that of the RDKit descriptors.

Table S3: RDKit Descriptors Organized by Type

Descriptor Type	Descriptors
1. **Topological Indices**	Chi0, Chi0n, Chi0v, Chi1, Chi1n,Chi1v, Chi2n, Chi2v
2. **Morgan Fingerprints**	FpDensityMorgan1, FpDensityMorgan2, FpDensityMorgan3
3. **Kappa Indices**	Kappa1, Kappa2, Kappa3
4. **Molecular Properties**	ExactMolWt, MaxAbsPartialCharge, MaxPartialCharge, MinAbsPartialCharge, MinPartialCharge, MolLogP, MolMR, MolWt
5. **Structural Counts**	RingCount, HeavyAtomCount, HeavyAtomMolWt, FractionCSP3
6. **SA Descriptor**	HallKierAlpha, LabuteASA, TPSA
7. **Atom Count Descriptors**	NHOHCount, NOCount, NumHAcceptors, NumHDonors, NumHeteroatoms
8. **Ring Counts**	NumAliphaticCarbocycles, NumAliphaticHeterocycles, NumAliphaticRings, NumAromaticCarbocycles, NumAromaticHeterocycles, NumAromaticRings
9. **Electronic Descriptors**	NumRadicalElectrons, NumValenceElectrons
10. **Rotatable Bonds**	NumRotatableBonds
11. **Saturated Structure Counts**	NumSaturatedCarbocycles, NumSaturatedHeterocycles, NumSaturatedRings
12. **PEOE Descriptors**	PEOE_VSA1, PEOE_VSA2, PEOE_VSA3, PEOE_VSA4, PEOE_VSA5, PEOE_VSA6, PEOE_VSA7, PEOE_VSA8, PEOE_VSA9, PEOE_VSA10, PEOE_VSA11, PEOE_VSA12, PEOE_VSA13, PEOE_VSA14
13. **SMR Descriptors**	SMR_VSA1, SMR_VSA2, SMR_VSA3, SMR_VSA4, SMR_VSA5, SMR_VSA6, SMR_VSA7, SMR_VSA8, SMR_VSA9, SMR_VSA10
14. **SlogP Descriptors**	SlogP_VSA1, SlogP_VSA2, SlogP_VSA3, SlogP_VSA4, SlogP_VSA5, SlogP_VSA6, SlogP_VSA7, SlogP_VSA8, SlogP_VSA9, SlogP_VSA10, SlogP_VSA11, SlogP_VSA12
15. **Functional Groups**	fr_amide, fr_NH0, fr_NH1, fr_NH2, fr_COO, fr_priamide, fr_guanido, fr_imidazole, fr_phenol, fr_Al_OH, fr_C_O, fr_ether, fr_alkyl_halide, fr_unbrch_alkane, fr_aryl_methyl, fr_benzene, fr_ester, fr_ketone, fr_methoxy, fr_sulfide, fr_sulfonamd

S4. Training Setup and Efficiency

Pretraining hyperparameters are outlined in Table S4.

S5. Downstream Finetuning Protocol

S5.1 Avoiding data leakage

The pretraining data used in this study only contained linear, natural peptides. All finetuning data was either cyclic or modified peptides. This avoids any chance of data leakage from pretraining to evaluation datasets.

Table S4: Pretraining hyperparameters.

Setting	Value
Architecture	BERT-style encoder (Small/Base/Large)
Parameters	32M, 114M, 337M
Tokenizer	kmer SMILES (vocab: 405)
Max sequence length	2048
Batch size	512 seqs (global)
Optimizer	AdamW
AdamW $(\beta_1, \beta_2, \epsilon)$	$(0.9, 0.999, 1\text{e}{-8})$
Weight decay	0.01
Learning rate (peak)	$3\text{e}{-4}$
LR schedule	Cosine decay with linear warmup
Warmup steps	5,000
Training steps	100k
Dropout (attn/ffn)	0.1 / 0.1
Masking rate (MLM)	25%
Span masking	$\mathcal{N}(\mu=3.5, \sigma=1)$ spans
MTR heads	2-layer MLP, SiLU, mean-pooled embedding
Regression targets	99 RDKit descriptors (normalized)
Loss weights	$\lambda_{\text{MLM}} = 0.6,\ \lambda_{\text{MTR}} = 0.4$
Precision	bfloat16
Gradient clipping	0.1
Gradient accumulation	1–8 (as needed for memory)
Data aug.	RDKit randomized SMILES

S5.2 Nested CV & ensembling diagram

The fine-tuning process utilized a nested cross-validation approach to enhance the model's performance on downstream tasks involving labeled data. This method entails two levels of cross-validation, an outer loop and an inner loop.

Outer Loop: In the outer loop, a specified portion of the dataset is designated as a fixed holdout test set. This test set remains untouched throughout the training process to provide an unbiased evaluation of the model's performance on unseen data.

Inner Loop: The inner loop focuses on training the model by sequentially holding out each non-test fold of the data. This entails splitting the remaining data into multiple partitions, training the model on a combination of these partitions, and validating its performance on the held-out partition. The inner loop's design permits the systematic assessment of model performance across various training subsets, ensuring that the model is robust and generalizes well. This nested structure of cross-validation not only allows for effective validation of model performance but also aids in early stopping to mitigate overfitting. Early stopping was conducted by monitoring validation loss and terminating training when performance ceases to improve for 5 validation steps (or 50% of an epoch).

Model checkpoints are established for each fold based on validation loss, ensuring that the best-performing model configuration is retained for subsequent predictions. The final predictions for each test instance are generated by averaging outputs from the ensemble of models trained across all folds.

To ensure consistency and robustness in results, the fine-tuning procedure for each model is repeated three times. Performance metrics, including mean and standard error, are reported across these replicates to provide a comprehensive evaluation of the model's performance and variability.

S5.3 Finetuning Hyperparameters

Finetuning hyperparameters are detailed in Table S5.

Table S5: Finetuning hyperparameters (nested CV + ensembling).

Setting	Value *(Small, Base, Large)
Task heads	Regression / Binary classification
Loss	MSE (regression), Binary CEL (classification)
Batch size	16, 16, 32*
Learning rate	$3e{-4}$, $1e{-4}$, $5e{-5}$*
Optimizer	AdamW
Weight decay	0.01
LR schedule	No decay; no warmup
Dropout (head)	0.1
Max epochs (per fold)	10
Evaluation	20% epoch (early stopping patience 3)
CV scheme	Outer holdout + inner K-fold (K=5)
Ensembling	Mean of checkpoints across inner folds
Class imbalance	Equal sampling across bins
Input length	No truncations required
Replicates	3 (report mean $\pm$ std)

S6. Ablations

We performed ablations to isolate the effects of tokenizer choice, masking strategy, and model size on downstream performance. All experiments used identical training protocols and nested cross-validation.

S6.1 Tokenization Strategy

We compared the kmer tokenizer against a standard atomistic tokenizer. At small scale, both approaches achieved comparable accuracy, but atomistic tokenization resulted in significantly longer sequences, increasing computational cost. The kmer tokenizer offered more compact representations that enabled faster training and better scaling at larger model sizes.

S6.2 Masking Strategy

Random token masking was compared with span masking drawn from a Gaussian distribution ($\mu = 3.5$, $\sigma = 1$). Span masking consistently produced higher mean performance and reduced variance across replicates. The improvement suggests that masking chemically coherent spans helps the model learn peptide motifs more effectively than masking isolated tokens.

S6.3 Model Size

Scaling model parameters from 32M (Small) to 337M (Large) yielded steady improvements across regression and classification metrics. While performance continued to increase with size, gains diminished at the largest scale, indicating dataset variability rather than model capacity may be the limiting factor.

S7. Reproducibility

Three model weights for small, base, and large are released on huggingface at https://huggingface.co/subm-123abc. Required compute for training is outlined in Table S6.

S8. Broader Impacts (Extended Discussion)

S8.1 Positive Applications

Peptide-based therapeutics are an important and growing drug class, with applications in areas such as oncology, metabolic disease, and antimicrobial design. By developing a foundation model that accommodates noncanonical residues and chemical modifications, PEPTIDEMTR has the potential to accelerate early-stage drug discovery. In particular, improved predictive models can help reduce

Table S6: Compute resources, all used in cloud, by stage. (* = including unsuccessful runs/hyperparam sweeps/ablation grid)

Stage	GPU Hardware	VRAM	GPU hours/model	Precision
Pretraining (Small)	8xH100	80GB	40	bf16-mixed
Pretraining (Base)	8xH100	80GB	96	bf16-mixed
Pretraining (Large)	8xH100	80GB	192	bf16-mixed
Finetune: Permeability	1xH100	80GB	<1	bf16-mixed
Finetune: Fibrillation	1xH200	120GB	<1	bf16-mixed
Finetune: Albumin binding	1xH200	120GB	<1	bf16-mixed
Est. Total GPU hours*			10,000	

experimental costs, prioritize promising candidates, and decrease animal use in preclinical testing. Beyond drug discovery, peptide-specific embeddings may also enable advances in materials science (e.g., peptide-based biomaterials) and fundamental research into sequence–structure relationships.

S8.2 Limitations and Risks

While the model captures symbolic and physicochemical features of peptides, it does not incorporate explicit 3D structural information. As a result, its predictions are limited to coarse molecular properties rather than detailed biophysical mechanisms. Moreover, the training data are drawn from publicly available corpora that may contain errors, imbalances, or biases; these limitations can propagate into downstream predictions.

S8.3 Dual-Use Considerations

As with any generative or predictive model for biomolecules, there is some risk of misuse, such as the design of harmful peptides. Several mitigating factors reduce this concern: (i) the model is trained only on public corpora, (ii) it focuses on coarse physicochemical descriptors rather than activity labels, and (iii) it is intended for transfer learning rather than end-to-end generation of bioactive sequences. Nevertheless, careful consideration should be given to responsible release, including dataset documentation, usage terms, and community guidelines.

S8.4 Equity and Access

Foundation models require substantial computational resources to train. This raises concerns about equitable access, as smaller labs or groups without cloud resources may face barriers to adoption. To mitigate this, we are releasing trained checkpoints, tokenizers, and code to enable broader use by the community without the need for large-scale compute.

S8.5 Conclusion

Overall, we believe that the potential benefits of PEPTIDEMTR outweigh the risks. We encourage the community to adopt careful release practices and to consider downstream applications responsibly.

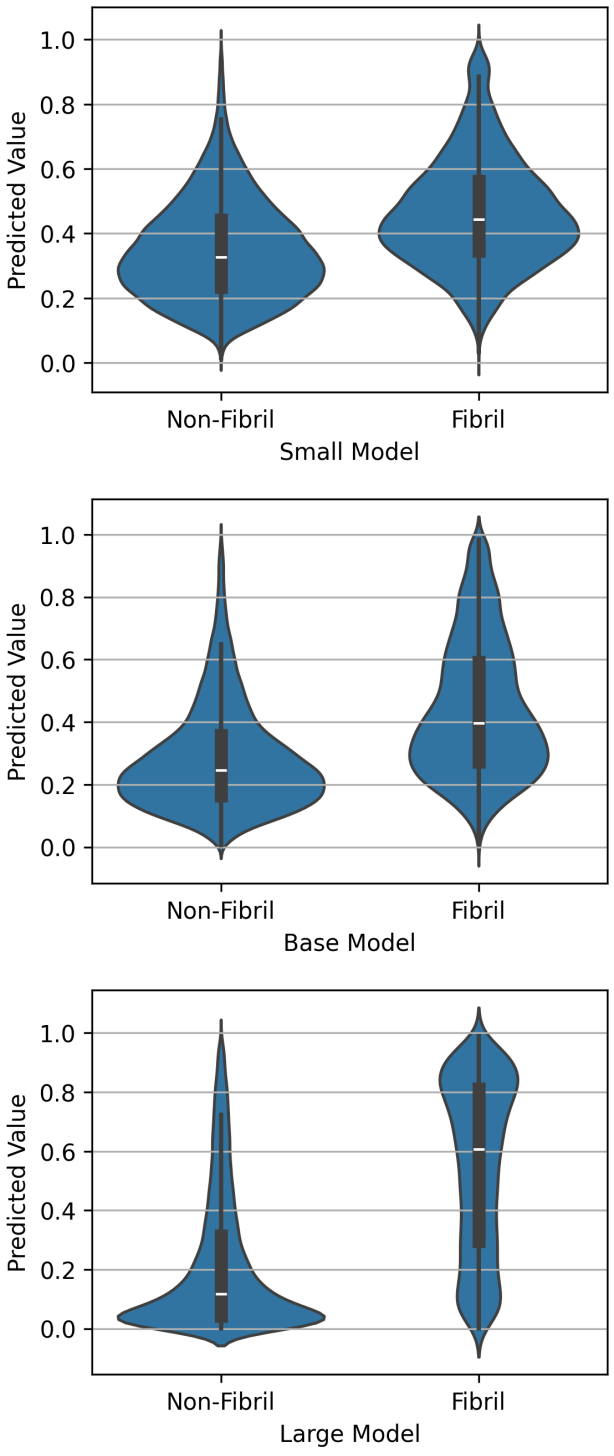

Figure S3: Violin plots depict the density of predicted values for fibril-forming versus non-fibril-forming peptides across three models: Small, Base, and Large. Each subplot shows distinct distribution profiles with varying overlap. The width of each violin indicates the density of predicted values, illustrating significant differences in prediction behavior across model sizes.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims, denoted i, ii, and iii in the abstract are reflect the main contributions of this work to the field. Evaluations of these contributions are in the experiments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are outlined in the conclusion of the paper and highlight future work that could aid in solving some of the limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results; therefore, a full set of assumptions and proofs are not applicable.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All model architectures, pretraining hyperparameters & data, and model weights are released. Hyperparameters and methods for finetuning are also outlined in detail in both the main text and in the supplement.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We made every effort to release all available resources to facilitate reproduction of the main experimental results. Model weights and open datasets, specifically the cyclic peptide PAMPA data, are made available. However, one dataset used for further evaluation of the model is unavailable due to proprietary intellectual property restrictions. We believe the resources to reproduce the main evaluation on PAMPA data are sufficient for evaluating model performance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, the paper specifies all relevant training and testing details, including data splits, hyperparameters, the method for their selection, and the type of optimizer used. This information is presented to ensure a comprehensive understanding of the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In our experiments, we conducted triplicate fine-tuning with random seed to ensure the reliability of our results. We report the standard error of the mean (SEM) for each model evaluated, providing a clear depiction of variability within our data.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, the supplementary table provides detailed information on the computer resources required for each experiment, including the type of compute resources used and memory specifications, supporting reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: No human subjects; all datasets are public and used under their licenses; we considered dual-use and document mitigations in appendix.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the potential positive societal impacts of our chemical language model for peptides, such as advancements in drug discovery and personalized medicine, while also acknowledging the ethical concerns and negative societal impacts, particularly regarding dual-use risks.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not involve personal information or datasets that pose significant risks of misuse. Given the very low likelihood of dangerous applications arising from our model, no additional safeguards are necessary for its release.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All credit for databases was given through citation. All data that is made available was previously released open source. Examples are PubChem, ESMAtlas, LMSD, and SmProt. All code is newly developed.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Yes, appendix contains pretraining information, links to datasets used in pretraining, and license will be released as MIT open source with final released under the (currently anonymous) user https://huggingface.co/subm-123abc. Current huggingface release is anonymized and temporary for double-blind submission.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No human subjects were used in this work.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subjects were used in this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Use of LLMs were not an important/original component of this work.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.