NONLINEARITY LEAKS POSITION THROUGH THE MEAN POOL: A MECHANISTIC ACCOUNT OF AGGREGATION IN PROTEIN REPRESENTATIONS

ABSTRACT

Mean pooling of protein language model (PLM) representations is a linear, permutation-invariant operation that is position-blind; the standard fix is attention pooling, which reweights residues non-uniformly. We start from a puzzle: on antimicrobial peptide activity data, a per-token MLP followed by *ordinary* mean pooling — still permutation-invariant with respect to token order — matches attention pooling and far exceeds linear mean pooling. We identify the mechanism: when a nonlinearity is applied to embeddings that encode position, it creates token–position cross-terms that survive averaging, so the pooled vector becomes sensitive to which residue sits where. We confirm this causally in a controlled synthetic model: a nonlinear-then-mean representation solves a position-relational task perfectly, but **scrambling residue order collapses its accuracy from 1.000 to 0.507** (chance), while linear mean pooling is unaffected in both directions. A scaling analysis shows the effect operates at the $\sim$20-symbol amino-acid alphabet ($\Delta = +0.062$ over chance with an adequately powered probe, recovering 13% of the oracle's signal), diminishing only as the vocabulary grows far past biological scale. On GRAMPA, a multi-pathogen peptide dataset, the same signature appears: a per-token MLP and attention pooling both recover a pathogen-selectivity component ($\rho = 0.148 \pm 0.024$, 10-seed CI $[0.11, 0.18]$ for attention) that linear mean pooling ($\rho = 0.00$) and a hand-crafted physicochemical baseline ($\rho = -0.02$, CI includes zero) both miss, and this translates to a multi-seed-confirmed doubling of downstream predictive R^2 ($0.33 \rightarrow 0.56$ for *E. coli*, 5-seed, disjoint intervals). We situate the mechanism as a domain-specific instance of the Deep Sets representational result, and are explicit that the causal test is established synthetically; the corresponding real-embedding scrambling experiment is specified as the primary follow-up. The reframing is practical: the operative choice is not mean-vs-attention but linear-vs-nonlinear aggregation of position-carrying embeddings.

1 Introduction

When a protein language model's per-residue representations are compressed into a single per-protein vector, the default operation is mean pooling. The textbook argument for why this can fail is that mean pooling is permutation-invariant and therefore discards residue order [5]; the textbook fix is a learned pooling head such as attention, which weights residues non-uniformly [4].

A puzzling empirical observation complicates this tidy story. On antimicrobial peptide (AMP) activity data, a per-token MLP followed by *ordinary* mean pooling — an operation that is still permutation-invariant with respect to token order, and applies the identical transform to every position — matches explicit attention pooling and far exceeds linear mean pooling. If a uniformly-weighted scheme performs as well as attention, then position-selective weighting cannot be the operative distinction. And it is not simply that "any trained nonlinearity helps": we will show that a purely random nonlinear projection followed by mean pooling recovers positional signal too, but *only when the per-residue embeddings encode position*.

This paper identifies, isolates, and causally tests the mechanism. Consider an embedding $\mathbf{h}_i = \mathbf{e}_{\text{tok}}[\text{token}_i] + \mathbf{e}_{\text{pos}}[i] + \boldsymbol{\epsilon}$. A linear mean pool cancels the positional term (it is constant across sequences), leaving only token composition. But a *nonlinear* function f applied before pooling does not distribute over the sum: $f(\mathbf{e}_{\text{tok}} + \mathbf{e}_{\text{pos}})$ contains cross-terms

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

coupling token identity to position. Because different positions carry different $\mathbf{e}_{\text{pos}}[i]$, these cross-terms do not cancel under averaging — the mean of $f(\mathbf{h}_1), \ldots, f(\mathbf{h}_L)$ is sensitive to *which token sits at which position*, even though the mean of the raw embeddings is not. We call this *positional leakage through nonlinear mean pooling*.

That a per-element nonlinearity followed by a symmetric aggregator can represent rich functions of a set is precisely the Deep Sets result [6], realized architecturally in PointNet [7] and analyzed for higher-order structure in Janossy pooling [8]. Our contribution is not this mechanism in the abstract but its *domain-specific characterization*: contextualized PLM embeddings carry rich positional information by construction, the amino-acid alphabet is small, and we show these are exactly the conditions under which the leakage is practically large. We support this with:

1. A **controlled synthetic ablation** (Section 3) isolating nonlinearity $\times$ positional-encoding as the interaction that produces the effect.
2. A **causal test by position-scrambling** (Section 4): shuffling residue order within each sequence — preserving the token multiset but destroying the token-to-position assignment — collapses the nonlinear-pooling advantage from 1.000 to chance, while leaving linear pooling unchanged. This converts "consistent with" into "caused by" in the synthetic setting.
3. A **scaling analysis** (Section 5) with adequately powered probes showing the effect is present at the amino-acid alphabet size ($V = 20$), diminishing only far past biological scale.
4. A **biological application** (Section 6) where the same signature appears on real ESM-2 embeddings, with proper variance estimates and a physicochemical baseline.

We are explicit throughout about one boundary: the causal scrambling test is established in the synthetic model. The analogous experiment on real ESM-2 embeddings is specified precisely in Section 4 as the primary follow-up; the GRAMPA evidence in Section 6 shows the predicted *signature* (nonlinear pooling recovers what linear pooling cannot) but is correlational on the real-data side.

2 Related Work

Set functions and permutation invariance. Deep Sets [6] shows any continuous permutation-invariant set function admits the form $\rho(\sum_i \phi(x_i))$. Our finding is an instance: the per-element nonlinearity ϕ produces features whose symmetric aggregate retains the positional structure encoded in each x_i. PointNet [7] instantiates exactly per-point-MLP-then-symmetric-pool and recovers spatial structure in point clouds; Janossy pooling [8] formalizes when k-ary interactions are recoverable under permutation-invariant readouts. Relative to this literature we claim no new theory. What is new is the empirical characterization for protein representations: we show the small-alphabet, rich-positional-encoding regime is where the mechanism is practically large, we test it causally by scrambling, and we identify a concrete biological signal (pathogen selectivity) it recovers.

Protein language models and pooling. ESM-2 [1, 2] and ProtBERT [3] emit contextualized per-residue embeddings; mean pooling is the standard downstream aggregation and has received little independent scrutiny. Pooling-strategy sensitivity is well established in NLP [5]; we do not claim novelty there.

Antimicrobial peptide modeling. GRAMPA [9] compiles MIC measurements across pathogens; modlAMP [10] provides the standard physicochemical descriptors that constitute our non-PLM baseline.

3 Mechanism: Controlled Synthetic Ablation

3.1 Setup

Sequences of length L over vocabulary V, with per-residue embeddings $\mathbf{h}_i = \mathbf{e}_{\text{tok}}[\text{token}_i] + \mathbf{e}_{\text{pos}}[i] + \boldsymbol{\epsilon}_i$ ($\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, 0.05^2\mathbf{I})$, D-dimensional fixed random tables). The label is $y = \mathbb{1}[(\text{token}_{P_1} \bmod 2) = (\text{token}_{P_2} \bmod 2)]$ (same parity at two fixed positions). We use the parity label rather than exact-match specifically because it maintains a $\approx$50% positive rate at any vocabulary size, keeping class balance constant as we scale V in Section 5; exact-match would make the positive rate $1/V$, so its majority baseline would rise with V and confound the scaling comparison. (Under exact-match at $V = 8$ the raw RFF advantage over baseline is smaller, $+0.039$ vs. parity's $+0.462$, but only because the imbalanced baseline of 0.877 leaves little headroom; both are well above chance for their respective class balance.)

Table 1: The mechanistic ablation ($V = 8$, $L = 12$, 5-seed mean $\pm$ std; majority baseline 0.877).

	Held-out accuracy		
Pooling method	With pos. embed.	Without pos. embed.	Δ
Linear mean pool	0.877 ± 0.001	0.877 ± 0.001	0.000
RFF nonlinear mean pool	$\mathbf{0.971 \pm 0.005}$	0.516 ± 0.011	**+0.455**
Full concat (position-aware)	1.000 ± 0.000	1.000 ± 0.000	0.000

Table 2: Position-scrambling causal test ($V = 8$, $L = 12$, 5-seed; chance $= 0.503$). Scrambling preserves the token multiset but destroys token-to-position assignment.

Pooling method	Original order	Residues scrambled	Δ
Linear mean pool	0.511 ± 0.011	0.511 ± 0.011	0.000
RFF nonlinear mean pool	$\mathbf{1.000 \pm 0.000}$	0.507 ± 0.004	$-$**0.493**

3.2 Why linear pooling is position-blind, and nonlinear pooling is not

Linear: $\bar{\mathbf{h}} = \frac{1}{L}\sum_i \mathbf{e}_{\text{tok}}[\text{token}_i] + \frac{1}{L}\sum_i \mathbf{e}_{\text{pos}}[i] + \bar{\boldsymbol{\epsilon}}$. The middle term is identical for every sequence, so the pooled vector depends only on the token multiset. Nonlinear: $\bar{\mathbf{z}} = \frac{1}{L}\sum_i f(\mathbf{h}_i)$; since f is nonlinear, token and position components interact inside f, and the resulting cross-terms do not cancel under averaging because $\mathbf{e}_{\text{pos}}[i]$ varies with i. The pooled vector becomes order-sensitive.

3.3 The ablation

Table 1: at $V = 8$, random Fourier features (RFF) followed by mean pooling reach 0.971 accuracy *with* positional embeddings but fall to 0.516 (baseline) *without* them. Linear mean pooling sits at baseline (0.877) in both conditions — a control confirming the probe is not extracting position through some unintended route. Full concatenation (position-aware) solves the task regardless. The interaction nonlinearity $\times$ positional-encoding is the sole source of the effect.

4 Causal Test: Position-Scrambling

The ablation in Table 1 is causal by construction (we add/remove positional embeddings). We now run the complementary causal test the review process identified as decisive: hold the representation fixed and destroy *only* the token-to-position assignment.

Design. We permute residue order within each sequence before pooling. This leaves the token multiset — and hence everything a permutation-invariant summary of composition could use — exactly unchanged, while destroying which token sits at which position. **Pre-registered prediction:** if the nonlinear-pooling advantage comes from positional leakage, scrambling should collapse it to the linear-pooling (chance) level; if it came from some distributional property of the token multiset, scrambling should leave it intact.

Result (Table 2, Figure 1). The RFF nonlinear-mean representation solves the task at 1.000 in original order and **collapses to 0.507 (chance) when scrambled**. Linear mean pooling is 0.511 in both conditions. The advantage is entirely positional: it is destroyed by exactly the intervention that removes token-to-position information and nothing else.

Scope, and why the real-data version is a genuinely different experiment. This test is established in the synthetic model. Carrying it to real embeddings is subtler than it first appears, and the subtlety is itself informative. In our synthetic model, scrambling touches *only* the aggregation step: each residue's embedding is a fixed function of its token and position, so permuting the order before pooling cleanly isolates the token-to-position assignment. Real ESM-2 embeddings are *contextualized*: a residue's embedding at position 3 of the original sequence is not a valid stand-in for what ESM-2 would compute at position 9 of a reordered sequence, because it was produced by attention over the whole original context. One therefore cannot run the real-data test by permuting the frozen per-residue vectors we already extracted; one must permute the raw amino-acid sequence and re-encode it through ESM-2. But that changes what ESM-2 *sees and encodes*, not merely how the outputs are aggregated — a shuffled peptide is a different input, and its contextualized embeddings differ from the original's in ways that go beyond reordering. The real-data scrambling

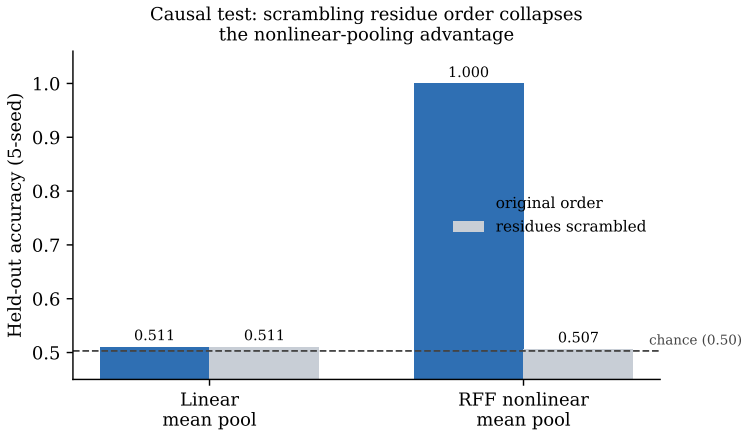

Figure 1: Scrambling residue order collapses the nonlinear-pooling advantage to chance while leaving linear pooling unaffected — causal evidence that the advantage is positional, not distributional.

Table 3: Scaling with adequately powered probes (parity label; oracle solvability verified at each scale). Δ = accuracy above chance.

V	L	Linear mean	RFF nonlinear mean	Oracle (2 pos.)
8	12	$+0.011$	$+0.497$	$+0.497$
20	20	$+0.001$	$+0.062$	$+0.492$
50	30	-0.003	$+0.007$	$+0.025$

experiment thus tests a subtly different (and arguably stronger) hypothesis than the synthetic one: whether the positional information the nonlinearity exploits is present in ESM-2's contextualized representations at all, not merely whether the aggregation step preserves it. We regard running this re-encoding experiment, and disentangling "aggregation discards position" from "re-encoding a shuffled sequence destroys position," as the primary follow-up; the GRAMPA results in Section 6 show the predicted signature (nonlinear pooling recovers what linear pooling cannot) but do not resolve this distinction.

5 Scaling: Does the Mechanism Operate at Amino-Acid Scale?

A natural concern is whether the effect requires a small, memorizable vocabulary. We scale V with an *adequately powered* probe (larger N, larger MLP), verifying at each scale that the position-aware oracle actually solves the task — so that a null RFF result would be informative rather than a probe-capacity artifact.

Table 3: at $V = 20$ (the amino-acid alphabet), the oracle solves the task perfectly (1.000), linear mean pooling is at chance (0.509), and RFF nonlinear mean pooling recovers a real $\Delta = +0.062$ over chance — about 13% of the oracle's recoverable signal. The effect is smaller than at $V = 8$ but clearly present and not a capacity artifact (the oracle is solvable). It diminishes further only well past biological vocabulary scale. Crucially, real PLM embeddings carry far richer positional information than the single additive $\mathbf{e}_{\text{pos}}$ of this toy model — ESM-2's positional signal is distributed across dozens of attention layers — which predicts a *larger* practical effect than the $V = 20$ toy row, consistent with the GRAMPA magnitude in Section 6.

A caveat on this scaling design. Table 3 scales V and L together, so it does not by itself separate whether the dilution at larger V is driven by vocabulary growth, sequence-length growth, or both. This matters because real peptides in our GRAMPA sample have a fixed alphabet ($V \approx 20$) but a wide length range (up to 60 residues), so the single $V = 20, L = 20$ point is only a partial stand-in for "biological scale." A cleaner design — holding $V = 20$ fixed and scaling L across the peptide-length range, and separately holding L fixed and scaling V — would isolate which variable drives the dilution. We report the coupled sweep as evidence that the mechanism is non-trivially present at $V = 20$, not as a full characterization of its L-dependence, which we leave to future work.

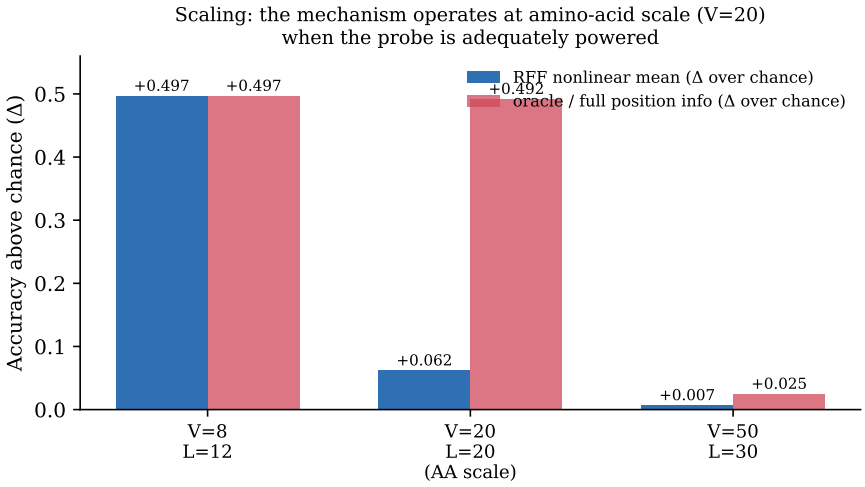

Figure 2: The mechanism operates at the amino-acid alphabet ($V = 20$): the oracle is solvable, linear pooling is at chance, and RFF nonlinear pooling recovers real signal. The effect dilutes as V grows past biological scale.

6 Biological Application: AMP Activity

6.1 Data and variance decomposition

GRAMPA [9]: 6,760 peptides, 38 organisms after filtering. A two-way additive decomposition assigns **73.6% of variance to potency** and **25.4% to a pathogen-interaction component**, the latter structured by cell-envelope architecture (same-envelope pairs $\bar{\rho} = 0.69$–0.74 vs. cross-envelope $\bar{\rho} = 0.43$–0.54; Table 4).

Table 4: Pairwise activity correlations by envelope-pair type.

Envelope-pair type	n	Mean ρ	Std
same Gram−	37	0.739	0.130
same Gram+	44	0.691	0.077
same fungus	8	0.693	0.105
cross Gram−/Gram+	75	0.544	0.173
cross fungus/Gram+	26	0.518	0.135
cross fungus/Gram−	31	0.428	0.149

6.2 Two statistical corrections

Two artifacts had to be corrected to get the decomposition right, and we foreground them as methodology. **Closure-induced spurious correlation**: subtracting each peptide's row mean before correlating residuals forces $\rho = -0.996$ among peptides tested on exactly two organisms — pure arithmetic, since two mean-centered values are negatives of each other [11]; the raw correlation is $+0.649$. We use the two-way additive model to avoid this. **Taxonomy–envelope confound**: envelope type and phylogeny are nearly collinear ($R^2 = 0.854$ regressing taxonomic distance on envelope-pair type); the deconfounded within-Gram-negative test still shows a secondary phylogenetic effect ($\rho = -0.392$, $p = 0.016$).

6.3 Three-way comparison with variance estimates

Table 5: physicochemical descriptors (bootstrapped CI), mean-pooled ESM-2 (untrained), attention-pooled ESM-2 (10-seed CI). Hand-crafted features beat mean-pooled ESM-2 on potency (CI excludes zero) — a PLM is unnecessary for the easy component. Both fail on `envelope_gap` (CIs include zero). Only attention-pooled ESM-2 recovers it ($\rho = 0.148$, CI $[0.109, 0.182]$, lower bound $> 4\times$ every alternative's point estimate).

Table 5: The main comparison. Physicochemical: bootstrap CI (300 resamples). Attention: 10-seed CI.

Representation	potency ρ (95% CI)	envelope_gap ρ (95% CI)
AA composition (20-dim)	0.052	-0.036
Physicochemical (12-dim)	0.110 $[0.070, 0.153]$	-0.024 $[-0.060, 0.009]$
Mean-pooled ESM-2 (35M)	0.042	0.000
Attention-pooled ESM-2	**0.224** $\pm$ 0.019 $[0.202, 0.263]$	**0.148** $\pm$ 0.024 $[0.109, 0.182]$

6.4 The predicted signature: MLP-then-mean matches attention

Table 6: a per-token MLP followed by mean pooling ($\rho = 0.259/0.155$, single-split) matches attention ($0.224/0.148$, 10-seed); all fixed pooling schemes stay near zero. This is the biological signature the mechanism predicts: a uniformly-weighted nonlinear pooling recovers the same signal as position-selective attention, because both access the positional information in ESM-2's contextualized embeddings — attention by non-uniform weighting, the MLP by the interaction-leakage route of Sections 3–4. The real-embedding scrambling test in Section 4 is what would upgrade this from signature to causal confirmation on real data.

Table 6: Pooling sweep, ESM-2 (35M). Attention: 10-seed; others single-split.

Pooling	potency ρ	envelope_gap ρ
mean	0.042	0.000
max	0.056	0.008
CLS token	0.056	-0.005
attention (10-seed)	0.224 ± 0.019	0.148 ± 0.024
per-token MLP $\rightarrow$ mean	0.259	0.155

6.5 Downstream prediction and cross-encoder replication

On direct activity regression, attention pooling roughly doubles 5-fold CV R^2 over mean pooling. For *E. coli*, where we ran the full multi-seed protocol, the gap is confirmed with variance estimates: mean pooling $R^2 = 0.327 \pm 0.009$ vs. attention pooling $R^2 = 0.559 \pm 0.032$ (5 seeds), the two intervals clearly disjoint. The remaining four organisms are single-split and reported as supporting: *P. aeruginosa* $0.27 \rightarrow 0.55$, *S. aureus* $0.24 \rightarrow 0.46$, *B. subtilis* $0.40 \rightarrow 0.66$, *C. albicans* $0.26 \rightarrow 0.60$. The pattern replicates in direction on ProtBERT (mean $0.023/-0.005$; attention $0.203/0.082$, single-split).

7 Why Explicit Pathogen-Conditioning Does Not Help

Four conditioning mechanisms (context-modulated metrics and attention variants) failed to beat unconditioned attention pooling. Two factors: per-organism coverage spans a 50-fold range (limiting organism-specific terms), and envelope type is near-deterministic given taxonomy ($R^2 = 0.854$), leaving little independent signal. This does not undercut Section 6: the positive claim (the component is recoverable by attention, 10-seed CI) and this negative claim (explicit context adds nothing further) are different claims.

8 Limitations

The causal test is synthetic, and the real-data version is a different experiment. Section 4 establishes causality in the toy model via two independent interventions (with/without positional embeddings, and order-scrambling). The real-embedding version cannot be done by permuting frozen vectors — it requires re-encoding shuffled sequences through ESM-2 (Section 4), which tests the stronger hypothesis of whether position survives contextualization at all. This re-encoding experiment is the primary follow-up; the GRAMPA side currently shows the predicted signature but is correlational. **Single biological dataset.** All AMP claims rest on GRAMPA; one generalization attempt hit a ceiling effect. **Partial multi-seed coverage.** Attention pooling on ESM-2 (Table 5) has 10-seed CIs, and the *E. coli* downstream row is now 5-seed ($R^2 = 0.559 \pm 0.032$ vs. 0.327 ± 0.009). The remaining downstream organisms, the per-token MLP, and the ProtBERT replication are still single-split. **Downstream leakage risk.** The downstream table forward-passes the attention head on peptides seen during its training before the outer CV; a fixed held-out split would

remove this. **The scaling sweep couples V and L** (Section 5), so it does not isolate which variable drives the dilution of the effect; a factorial sweep is needed to characterize the L-dependence at fixed amino-acid vocabulary.

9 Reproducibility Statement

GRAMPA, ESM-2, ProtBERT, and modlAMP are public. All experiments are implemented as executable notebooks with fixed seeds and will be released publicly; the synthetic ablations (Sections 3–5) require no GPU and reproduce in minutes.

10 Discussion

The standard argument against mean pooling is correct but incomplete. *Linear* mean pooling is position-blind; *nonlinear* mean pooling of position-carrying embeddings is not, because the nonlinearity creates token–position cross-terms that survive averaging — a domain-specific instance of the Deep Sets capacity result [6]. Our scrambling test (Section 4) shows this is causal, not incidental, in the synthetic setting, and the scaling analysis (Section 5) shows it operates at amino-acid vocabulary scale.

This reframes the practical question from "mean vs. attention" to "linear vs. any nonlinear aggregation" of position-carrying embeddings. Attention works; so does a plain per-token MLP; both access the same underlying positional information. We speculate — as a hypothesis for future work, not a finding — that the mechanism's dependence on small vocabulary explains why attention's advantage over mean pooling is often more dramatic in large-vocabulary NLP settings: with thousands of tokens, a per-token nonlinearity has proportionally more identities to disambiguate before positional cross-terms become informative. The scoped claim we do make: in protein representations, where the alphabet is small and contextualization is rich, replacing linear mean pooling with any nonlinear aggregation recovers positional signal the encoder has already computed and mean pooling discards — and, on AMP selectivity specifically, that signal is real, multi-seed-confirmed, and invisible to both mean pooling and the field's standard hand-crafted features.

Disclosure of AI Assistance

In accordance with the ICLR 2027 policy on LLM use, we disclose the following. This section is not counted toward the page limit.

We used a large language model (LLM) assistant in several parts of this work, and the authors take full responsibility for all content, claims, and artifacts.

Tasks for which we used AI assistance. We used AI for four categories of task. *Editing the manuscript*: the LLM was used to revise sections of the paper text and to improve readability and structure; all scientific claims, framing, and interpretations were reviewed, verified, and edited by the authors, who take responsibility for the final wording. *Retrieval and identification of related work*: the LLM helped identify potentially relevant prior work (including the Deep Sets, PointNet, and Janossy pooling literature situating our mechanism); all cited references were independently verified by the authors, who confirmed each work exists, read the relevant content, checked that our characterizations are accurate, and verified citation details against primary sources. *Feedback on methodology and experimental design*: the LLM was used to critique experimental design and suggest additional analyses (e.g., the physicochemical baseline comparison, multi-seed variance estimation, the position-scrambling causal test, and the vocabulary/length scaling analysis); the decision to run each experiment, the implementation, and the interpretation of results were the authors' responsibility.

We have reviewed all AI-assisted work. All reported experimental numbers were verified by executing the underlying code; all citations were checked against primary sources for existence and accuracy; and all claims were scoped by the authors to match the evidence actually obtained, with unfinished experiments (notably the real-embedding scrambling test of Section 4) explicitly identified as such. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

References

[1] Z. Lin et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[2] A. Rives et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 118(15):e2016239118, 2021.

[3] A. Elnaggar et al. ProtTrans: Towards cracking the language of life's code. *IEEE TPAMI*, 44(10):7112–7127, 2021.

[4] A. Vaswani et al. Attention is all you need. In *NeurIPS*, 2017.

[5] N. Reimers and I. Gurevych. Sentence-BERT. In *EMNLP-IJCNLP*, 2019.

[6] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola. Deep Sets. In *NeurIPS*, 2017.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017.

[8] R. L. Murphy, B. Srinivasan, V. Rao, and B. Ribeiro. Janossy Pooling: Learning deep permutation-invariant functions for variable-size inputs. In *ICLR*, 2019.

[9] J. Witten and Z. Witten. Deep learning regression model for antimicrobial peptide design. *bioRxiv* 692681, 2019.

[10] A. T. Müller, G. Gabernet, J. A. Hiss, and G. Schneider. modlAMP: Python for antimicrobial peptides. *Bioinformatics*, 33(17):2753–2755, 2017.

[11] J. Aitchison. The statistical analysis of compositional data. *JRSS-B*, 44(2):139–177, 1982.