أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Evaluating Binding Predictors for Aptamer Discovery: Construction Gaps in Interaction Benchmarks

Mona Yassine, Abeer Albakran, Kerven Durdymyradov

mona.yassine@kaust.edu.sa, abeer.albakran@kaust.edu.sa, kerven.durdymyradov@kaust.edu.sa

Abstract

Background. Aptamers are short nucleic acid molecules that bind protein targets with high affinity, making them promising theranostic agents. However, their discovery still relies on SELEX, an iterative wet-lab process that can take months per target. Computational binding predictors could accelerate candidate prioritization, but their value for discovery depends on whether they generalize to protein targets unseen during training.
Methods. Here, we evaluate AptaTrans, a published transformer-based aptamer–protein binding predictor reported to achieve a ROC-AUC of 0.921, as the binding component of a multi-objective aptamer design engine. In attempting to reproduce the model, we identified several evaluation and implementation issues: the released weights achieve a ROC-AUC of 1.000 on the reported test set, checkpoints are selected using test-set performance, and 99.2% of test proteins also occur in training. We therefore re-curated the dataset and introduced a protein-grouped split with no target overlap between partitions.
Results. Across controlled configurations, performance was strongly dependent on target overlap: near-total target overlap yielded ROC-AUCs of 0.689–0.753, whereas target-disjoint evaluation remained at or below chance. We further found that 90.3% of positive test pairs in the reference benchmark contain aptamers appearing in training only as negatives. Regenerating negatives within partitions removes this label inversion and yields a target-disjoint ROC-AUC of 0.515 ± 0.020. Replacing the protein representation with ESM-2 does not recover a useful predictor, while ablations indicate that remaining signal is largely attributable to aptamer sequence rather than the aptamer–protein interaction.
Conclusions. These results suggest that current benchmarks primarily measure candidate re-ranking for known targets rather than de novo binder discovery. We release a re-curated dataset and evaluation protocol to distinguish these settings.

Date: August 19, 2026

Contents

1 Introduction **2**

2 Related Work **3**

3 Methodology **3**

3.1 Method Details . . . 4

3.2 Evaluation Procedure . . . 5

4 Results **6**

4.1 Reproducibility Assessment of the AptaTrans Baseline . . . 6

4.2 Re-training from Clean Weights . . . 7

4.3 Evaluation Protocol . . . 7

4.4 Dataset Re-curation . . . 9

4.5 Negative Generation . . . 10

4.6 Retrain on Re-curated Data . . . 11

4.7 Control: Row-Level Split . . . 11

4.8 Cross-Model Evaluation on Target-Disjoint Data . . . 12

4.9 Reference Dataset Under a Target-Disjoint Split . . . 13

4.10 Negative-Sampling Order . . . 14

4.11 Is Protein Representation the Bottleneck? . . . 16

4.12 Results on the Corrected Dataset . . . 17

5 Discussion **18**

5.1 What the Reported Performance Supports . . . 18

5.2 Evaluation Practice in Related Models . . . 19

5.3 Consequences for Generation . . . 19

5.4 Why We Did Not Build the Scoring Function . . . 20

5.5 Threats to Validity . . . 20

6 Future Work **20**

7 Conclusion **21**

References **22**

1 Introduction

Aptamers are short, single-stranded DNA or RNA molecules that fold into three-dimensional shapes and bind specific targets with high affinity, making them valuable for therapeutics, biosensing, and diagnostics [1]. They are especially promising as theranostic agents, where a single molecule plays both a diagnostic and a therapeutic role: it locates diseased cells and delivers treatment to them [2]. Traditionally, aptamers are discovered through SELEX [1], a wet-lab process that is slow, costly, and optimises primarily for binding affinity, with no systematic mechanism to ensure the dual imaging-and-therapy suitability a theranostic agent requires.

This motivates computational design. Recent pipelines generate candidate sequences and score them with a learned aptamer–protein binding predictor, shortlisting a small number for wet-lab validation. Our original aim was to extend one such pipeline: AptaTrans [3], which pairs a Monte Carlo Tree Search generator with a transformer binding predictor, by replacing its single binding score with a multi-objective function co-optimising affinity, specificity, structural stability, functional-site engagement, and conjugation-site accessibility.

A design engine of this kind rests on an assumption worth stating explicitly: that the underlying binding predictor works for a protein the model has not seen before. That is the discovery setting. A researcher with a novel target and no known binders is precisely the user such a tool is built for, and the predictor must generalise across protein space for the search it guides to be meaningful.

Before extending AptaTrans we attempted to reproduce its reported performance. This did not succeed, and investigating why became the substance of the project. The released fine-tuned weights classify every pair in the paper's own test partition correctly, exceeding the reported figure. The training loop selects checkpoints by test-set score, with no validation partition anywhere in the fine-tuning path. And the distributed partitions are not disjoint at the level of the entities being predicted: 120 of the 121 test proteins also appear in training. The reported evaluation therefore measures performance on targets the model has already seen.

We consequently reframed the work as an evaluation of whether learned binding predictors support the discovery setting at all. This report contributes:

1. A reproducibility audit of AptaTrans and its released artifacts, identifying four independent defects, one of which prevents the code from running on current PyTorch at all.
2. A re-curated aptamer–protein dataset drawn from AptaDB and the UTexas Aptamer Database, with per-pair provenance retained, containing 1,185 positive pairs across 346 unique proteins against 725 across 144 in the reference set.
3. An evaluation protocol that partitions by protein so that no target spans partitions, holds out six proteins with clinically established aptamers by construction, and adds a validation partition so checkpoint selection never touches test.
4. A second benchmark defect, to our knowledge previously undocumented: generating negative pairs before splitting inverts the label association for the majority of test positives under target-disjoint evaluation, and this is present in the reference benchmark as distributed. Correcting the first defect exposes it.
5. Measurements across a series of controlled configurations showing that performance tracks target overlap rather than dataset, curation, or architecture, and that no configuration tested performs above chance on unseen targets, including one built on pretrained protein language model embeddings.

The multi-objective design engine remains well-motivated, and we describe its intended construction in Section 3. What this work establishes is that a prerequisite for it is currently

missing.

2 Related Work

Computational aptamer design. The field has progressed from physics-based methods toward learned generative models [1]. Early approaches such as MAWS and AptaLoop construct aptamers using thermodynamic or entropy-based scoring, and genetic-algorithm methods such as that of Torkamanian-Afshar evolve sequences against a binding-affinity fitness function. These are interpretable and, relevant to what follows, cannot exhibit target leakage: having no training set, they compute a score for a novel protein in the same way as for a familiar one. Their cost is computational intensity and limited optimisation capability [1].

More recent work applies deep learning to sequence generation. Latent-space generative models including RaptGen (VAE-based), AptaDiff (diffusion-based) and RNAGEN (GAN-based) learn to produce candidate sequences [1], while language-model approaches treat aptamer sequences as text, including the autoregressive AptaGPT, the latent-space Bayesian optimisation of InstructNA [4], and the conditional autoencoder of GRAPE-LM [5]. All three rely on experimental selection data such as SELEX or screening to guide design for a given target.

Interaction prediction. A second class of models scores aptamer–protein binding directly: AptaBERT [6], AptaTrans [3], AptaBLE [7] and SelfTrans-Ensemble [8] use transformer encoders, with several reporting strong predictive accuracy. Search-based methods such as Apta-MCTS [9] combine generation and scoring by exploring sequence space under the guidance of a binding predictor; the AptaTrans pipeline integrates this search with a transformer binding model [3].

These models share a common benchmark. The dataset distributed with AptaTrans derives from Li et al. [10], and AptaBLE combines it with the UTexas Aptamer Database [11]. Where evaluation protocols are described, they generally involve random partitioning of aptamer–protein pairs.

Evaluation practice. Splitting by row rather than by entity is a known source of optimistic bias in adjacent domains. Scaffold splitting in molecular property prediction and protein-identity-aware splitting in drug–target interaction prediction both emerged as corrections to random splitting once it became clear that random partitions place the same entity on both sides. To our knowledge no equivalent convention has been established for aptamer–protein interaction prediction, and the benchmark in common use has not been examined for this property.

Practice is beginning to shift. AptaBLE [7] reports constructing benchmark datasets as largely non-overlapping with training and validation at both the aptamer and protein levels, and partitions training data by sequence homology clustering. This is the correct direction, and we discuss in Section 5 both what it achieves and what it leaves unaddressed.

Multi-objective design. Existing approaches optimise for binding or general functionality without jointly optimising the combination of properties a theranostic aptamer requires: dual suitability for diagnostic imaging and therapeutic action alongside affinity, specificity and stability. Our original contribution addressed this gap; Section 3 describes the intended design and Section 5 explains why we did not proceed to build it.

3 Methodology

3.1 Method Details

The system we set out to build takes a target protein as input and returns a ranked shortlist of candidate aptamers intended to act as theranostic agents. It builds on the AptaTrans pipeline [3], which combines Apta-MCTS, a Monte Carlo Tree Search generator, with a transformer binding predictor that guides the search. The intended contribution replaces the pipeline's single binding score with a multi-objective scoring function evaluating candidates across binding, specificity, stability, and dual therapeutic and imaging suitability at once. We describe the design here because it motivates the evaluation that follows and remains the direction this work is intended to support. As Section 4 establishes, the binding predictor on which the search depends does not generalise to targets outside its training data, so the scoring function was not built; Section 5 sets out why we consider this the correct decision rather than a deferral.

The system operates as follows:

- The user provides a target protein.
- Apta-MCTS generates candidate aptamer sequences, building each sequence nucleotide by nucleotide.
- Each candidate is scored across all objectives by the multi-objective scoring function.
- The combined score guides the search toward candidates that balance every objective.
- Candidates are ranked and the most promising are shortlisted.

3.1.1 Generator

Apta-MCTS builds sequences through Monte Carlo Tree Search, repeating four phases: selection (choosing a branch via Upper Confidence Bound), expansion (adding a candidate node), simulation (completing and scoring the sequence), and backpropagation (propagating the score up the tree). The scoring function is a modular component passed into the search, so it can be replaced without changing the algorithm.

3.1.2 Binding Predictor

Binding is predicted by AptaTrans, which encodes the aptamer and protein separately, computes an interaction map, and produces a binding score. We retain its pretrained encoders and fine-tune the predictor on our re-curated dataset.

3.1.3 Multi-Objective Scoring Function

Our core contribution scores each candidate across five objectives: binding affinity, specificity, structural stability, therapeutic suitability, and imaging suitability. Binding affinity is taken directly from the binding predictor, while specificity measures how strongly the aptamer binds the target relative to off-target decoy proteins. Structural stability is given by the minimum free energy of the predicted fold, computed with ViennaRNA. Therapeutic suitability assesses whether the aptamer binds at the target's functional site and thereby blocks its activity, determined from the docking pose together with functional-site annotations, while imaging suitability checks for an accessible conjugation site outside the binding interface where an imaging label can be attached without disrupting the fold. These sub-scores are normalized and combined into a single weighted score that guides generation. We adopt an aptamer-as-drug approach, in which therapeutic action arises from the aptamer binding and blocking the target directly rather than from carrying a separate drug payload.

3.2 Evaluation Procedure

Dataset: Re-curation
AptaTrans was trained on limited, older data whose partitions we found to be contaminated (Section 4.1), so we re-curate the training data from primary sources. We draw aptamer–protein pairs from two established databases: AptaDB (2024), which records targets as UniProt accessions and is therefore resolvable deterministically, and the UTexas Aptamer Database, which records targets as free-text names and requires a name-to-UniProt resolution step. We considered three further sources: Aptagen/Apta-Index, APTABASE, and aptamer–protein complexes from the PDB, but did not include them: the first two provide no programmatic access, and the third requires structure-level filtering to separate genuine aptamer targets from crystallization chaperones, which we leave to future work.

Following SelfTrans [8] we deduplicate across and within sources. Negatives are constructed at a 1:3 positive:negative ratio, matching the ratio in the AptaTrans reference dataset (725 positives to 2,175 negatives) rather than the 1:1 ratio used by SelfTrans, so that class balance remains comparable to the baseline we measure against. We depart from the SelfTrans methodology in three respects, each stated as a deliberate choice rather than adopted by default. First, sequence length is left unfiltered: SelfTrans applies a 200 nt / 1000 aa cap without stated justification, and the reference dataset we are matching contains aptamers of 6–274 nt and proteins of 52–1,676 aa, so the cap would exclude common aptamer targets including EGFR, HER2, and MUC1 while making our data incompatible with the reference schema. Second, we do not augment with general RNA–protein interaction data (RPI2825, NPInter); doing so addresses class imbalance at the cost of training the model partly on RNA–protein interaction in general rather than aptamer–protein interaction specifically. Third, we do not adopt the sequence-mutation negative strategy, for the reason given in Section 4.5. The resulting dataset is larger and more recent than AptaTrans's original dataset (see Section 4.4), and is used to fine-tune the binding predictor.

Tools for Scoring Functions
The new scoring objectives are computed using established tools: UniProt supplies target annotations and off-target proteins for the specificity and functional-site scores; RCSB PDB provides target structure and binding-site geometry, used for functional-site binding and conjugation-site availability; ViennaRNA computes the minimum free energy of the folded sequence for structural stability; RNA Composer generates the three-dimensional aptamer structure required by both the functional-site and conjugation-site scores; FreeSASA determines solvent accessibility to identify labelable conjugation sites; and ZDOCK/HDOCK docks the aptamer against the target for functional-site binding and independent re-validation.

Evaluation Metrics
Item 1 below is the protocol used throughout Section 4. Items 2 to 4 describe the intended evaluation of the generator and scoring function, which was not reached; they are retained here as the specification for that work rather than as procedures carried out.

1. Binding predictor accuracy: ROC-AUC, F1, MCC, and accuracy on a held-out test set.
2. Generator output quality: validity (% of sequences folding into stable structures, via ViennaRNA free energy), uniqueness (% distinct sequences), novelty (edit-distance from the nearest training sequence), and diversity (average pairwise edit-distance across the set). A generation check by positive control confirms quality: candidates match or exceed a known aptamer on ZDOCK score, binding site (PyMOL), and motifs (MEME), adopted from the AptaTrans paper.

3. Multi-objective vs. single-objective: per-objective scoring compares our candidates against the binding-only baseline across all objectives (expecting similar binding but higher specificity, stability, and therapeutic/imaging scores). A scorer check by positive control confirms a known aptamer scores highly. Candidates are also shown to beat randomly generated sequences, and a Pareto front with hypervolume quantifies the improved trade-offs.
4. Result validity: selected candidates are independently re-validated by converting each sequence to a three-dimensional structure with RNA Composer and docking it against the target using ZDOCK. Since ZDOCK is structure-based and independent of the sequence-based selection score, agreement is non-circular. This validation chain is adopted from the AptaTrans paper.

4 Results

4.1 Reproducibility Assessment of the AptaTrans Baseline

Before extending the AptaTrans pipeline with our multi-objective scoring function, the released pipeline was evaluated to establish an unbiased baseline. This evaluation identified four issues with the reference implementation:

Contamination of the released fine-tuned checkpoint. The released weights were evaluated on the paper's own designated test set and resulted in ROC-AUC 1.000, accuracy 1.000, MCC 1.000, and F1 1.000, correctly classifying all 580 pairs. This exceeds the paper's reported ROC-AUC of 0.921 on the same set. The dataset was not the cause: the train/test split is fixed within the distributed pickle file rather than generated by a random split, and overlap between the two partitions accounts for only 5 of 578 unique test pairs (0.86%), too small to explain a perfect score. This suggests the released fine-tuned weights are not the weights used to produce the paper's reported evaluation, and they cannot be used to reproduce it. The two pretrained encoders however, obtained via self-supervised masked-token and secondary-structure prediction and never exposed to interaction labels, are unaffected and were retained.

Checkpoint selection on the test set. Inspection of the reference training loop (_train_epoch in aptatrans_pipeline.py) showed that the model checkpoint is saved whenever ROC-AUC on the test set improves, with no independent validation set used for model selection. This constitutes selection-on-test and can inflate the reported test performance.

Target-level overlap between the train and test partitions. Beyond the pair-level overlap noted above, the released partitions are not disjoint at the level of the entities being predicted. Of the 121 unique proteins in the test partition, 120 (99.2%) also appear in the training partition, as do 406 of 415 unique aptamers (97.8%). Measured across the dataset as a whole these figures are 120 of 144 proteins (83.3%) and 406 of 709 aptamers (57.3%); the test-side figures are the relevant ones for evaluating the reported result, since they describe the proportion of test entities the model has already been trained on. On this partition the reported evaluation measures performance on seen targets rather than generalization to unseen ones. The distributed file also contains 18 exactly duplicated rows and 23 duplicated (aptamer, protein) pairs.

The released code does not run on current PyTorch. The repository pins `torch==2.0.1` in its README but ships no environment specification, lock file, or container. Under current PyTorch builds the attention module fails with an opaque CUDA error (`CUBLAS_STATUS_INVALID_VALUE` raised from `cublasSgemmStridedBatched`) that gives no indication of its cause; we confirmed by reproducing the failure on synthetic tensors, independent of the model and data, that it originates

in the batched matrix multiply and not in the pipeline. An undeclared dependency (`einops`) is also absent from the stated requirements. Reproducing the pipeline therefore requires identifying a compatible toolchain by trial, which the released artifacts do not specify.

4.2 Re-training from Clean Weights

To obtain an unbiased baseline, the released fine-tuned weights were discarded and the API prediction head was re-trained from the clean pretrained encoders, introducing a validation split (85/15) carved from the training partition for checkpoint selection, leaving the test set untouched until final evaluation. This run achieved test ROC-AUC 0.739 (accuracy 0.738, F1 0.539, MCC 0.365), substantially below the paper's reported 0.921.

Because a single run cannot separate a real effect from run-to-run noise, this configuration was re-run as a seed sweep under the frozen protocol, giving ROC-AUC 0.753 ± 0.022 (mean ± std, $n = 3$ seeds). This replaces the single-run figure as the reference baseline against which subsequent configurations are measured. The ±0.022 is an indicative run-to-run noise floor at $n = 3$ rather than a confidence interval: a change counts as an improvement only if it clears the baseline by more than this spread.

To determine whether this gap was attributable to the validation-based selection reducing available training data or removing the inflating impact of selection-on-test, an exact replication of the reference setup was run, with fresh pretrained encoders, the full training partition with no validation split withheld, and checkpoint selection by test ROC-AUC exactly as implemented in the reference code. This run achieved test ROC-AUC 0.769 (accuracy 0.250, F1 0.400, MCC 0.000) at its selected epoch, closely matching the previous validation-selected run rather than the paper's reported value.

This result indicates that selection-on-test does not explain the discrepancy between the released weights and the paper's reported performance. We further note that the model selected as "best" in the replication run exhibits degenerate behavior at its default classification threshold (accuracy 0.250, MCC 0.000), consistent with the model predicting the positive class for all inputs despite a moderate ROC-AUC; test loss was also highly unstable across late epochs (ranging from 1.7 to over 15) against a near-zero training loss, consistent with overfitting on the training partition. We conclude that an honest re-training of this pipeline, using only publicly available code and data, yields ROC-AUC in the range of 0.74–0.77, and that the paper's reported 0.921 is not reproducible from the released artifacts alone, plausibly due to the contaminated checkpoint being distributed in place of the model actually evaluated, or to unreported implementation differences not captured in the public repository.

One qualification applies to the 0.753 baseline itself. It was measured on the distributed test partition, in which 120 of 144 proteins (and 120 of 121 test proteins) also appear in the training partition (Section 4.1). It is therefore inflated by the same target-level overlap that inflates the paper's reported figure, and should be read as a reproduction of the reference setup under honest checkpoint selection, not as a clean estimate of the pipeline's ability to generalize. Its purpose is to isolate the effect of selection-on-test, which it does: 0.753 against a reported 0.921 on identical data with identical overlap. It is not a measurement we can compare against results on a disjoint test set, and Section 4.3 accordingly constructs one.

4.3 Evaluation Protocol

Sections 4.1 and 4.2 establish that the original benchmark cannot support an honest evaluation: 120 of the 121 proteins in its test partition also appear in training, and its training loop selects

checkpoints on the test set. Continuing to evaluate against that test set would inherit both problems. We therefore discard the distributed partition and construct a new split over the re-curated data, then hold that protocol fixed across every subsequent arm so results remain mutually comparable.

Grouping by protein. The split is made over proteins, not rows; every pair involving a given protein sequence is assigned to exactly one partition. This follows directly from the failure mode identified in Section 4.1. Because the data is heavily skewed (1,407 unique aptamers against 346 unique proteins), a row-level random split would place the same target on both sides, allowing a model to score well by recognizing proteins rather than by learning interaction. Grouping by protein makes the test set a measurement of generalization to unseen targets.

Deliberate holdout of theranostic targets. Six proteins with clinically or experimentally established aptamers were assigned to the test partition by construction rather than by chance: VEGF-A (pegaptanib, FDA-approved 2004), complement C5 (avacincaptad pegol, FDA-approved 2023), nucleolin (AS1411, Phase I/II), CCL2 (NOX-E36), PTK7 (sgc8c), and thrombin (TBA/HD22). Together these account for 224 of the 947 test pairs. These are the targets for which the discovery setting matters most: proteins with clinically validated aptamers are the closest available analogue to a target a design pipeline would be asked to address, so a predictor that fails on them fails on the cases that motivate the field. Placing them deliberately rather than by chance also means the evaluation speaks to the theranostic design objective that motivated this work. A random assignment would have left their placement to the seed.

A validation partition. The reference pipeline has no validation set: `_train_epoch` scores the test set every epoch and saves whichever checkpoint maximises test ROC-AUC (Section 4.1). We carve a third partition so that checkpoint selection never touches test. The test set is scored exactly once, after training completes, using the checkpoint validation selected.

Resulting partitions. Proteins were assigned largest-group-first to approximate a 70/10/20 row split, the 20% test share matching the reference dataset's test fraction so the two remain comparable in size. The theranostic targets were placed first and excluded from subsequent assignment.

Table 1 Protein-grouped partitions. No protein sequence appears in more than one partition.

Partition	Rows	Positive	Negative	Proteins	Aptamers
Training	3,313	836	2,477	155	1,232
Validation	473	116	357	85	410
Test	947	233	714	106	689
Total	4,733	1,185	3,548	346	1,407

Protein overlap between every pair of partitions is zero by construction and was verified after assignment. Aptamer sequences are permitted to recur across partitions, since a given aptamer may legitimately be tested against several targets; among positive pairs this is rare, affecting 6 of 229 unique test aptamers (2.6%). Counting all rows rather than positives only, 571 of 689 unique test aptamers (82.9%) recur in training; this is an artifact of negative generation, which reuses each aptamer against multiple non-binding proteins, and does not constitute label leakage.

Comparability. Because the test set has changed, results here are not numerically comparable to the reference paper's reported 0.921, nor to our honest baseline of 0.753 ± 0.022, both of which were measured on the contaminated partition. The new figure is a measurement on a harder

and cleaner problem: 106 test proteins, none seen in training. A lower number than 0.753 is the expected outcome and is not evidence of a worse model. We report the earlier figures for context and state the change in test set wherever they appear alongside.

Frozen elements. The following are held constant across all runs: the split itself (assigned once with a fixed seed and stored as a protein-to-partition map); initialisation from the two pretrained encoders, never from the contaminated fine-tuned weights; checkpoint selection on validation ROC-AUC; a single test evaluation at the end; and ROC-AUC as the primary metric, being threshold-free. Accuracy, F1, and MCC are reported but depend on a threshold fitted at evaluation time and are therefore not used to rank runs.

One parameter is not constant across stages. The runs of Section 4.2 use batch size 8, which is what fits in 32 GB of VRAM; those from Section 4.6 onward use 16, the value in the reference configuration, on an A100. Batch size is held constant within each stage, so comparisons within a stage are unaffected, but figures from the two stages are not matched on this parameter.

4.4 Dataset Re-curation

We built a new positives set following the SelfTrans methodology [8], drawing from two sources: AptaDB, which is UniProt-keyed and therefore high-confidence, and the UTexas Aptamer Database, whose targets are name-matched to UniProt via a fuzzy lookup. From these, positive pairs pass through an ordered attrition ladder, each stage of which removes rows for one explicit reason (Table 2).

Table 2 Attrition ladder from raw positives to the curated set.

Stage	Removed	Remaining
Raw positives	—	1,474
Drop low-confidence matches	−294	1,180
Cross- and within-source dedup	−29	1,151
Screen against the original test partition	−49	1,102
Length filter (protein token cap; aptamer >200 nt)	−50	1,052
Recovered on re-split (original test partition dissolved)	+145	1,197
Character filter (ambiguity codes, mixed T/U, duplicates)	−12	1,185

The screening stage was applied while the original test partition was still in use as a fixed benchmark; 49 curated positives shared an aptamer with it and were removed to prevent contamination. That partition has since been dissolved into the protein-grouped split (Section 4.3), so the stage is now vestigial and its 49 rows could in principle be recovered. We have not done so, since recovering them would require re-running curation for a gain of under 5%.

A final character filter was applied uniformly across both sources to remove sequences the model cannot meaningfully encode: 24 aptamers containing the ambiguity code `N`, 16 proteins containing `X`, 11 aptamers containing both thymine and uracil and therefore not resolvable as DNA or RNA, 2 aptamers containing gap characters, and 2 exactly duplicated rows. Sequence length was left unfiltered, matching the reference dataset, which despite a stated 200 nt / 1000 aa cap contains aptamers of 6–274 nt and proteins of 52–1,676 aa.

The result is 1,185 positive pairs against 725 in the original AptaTrans set, an increase of roughly 1.6×, and 346 unique proteins against 144.

The two sides of the dataset remain asymmetric: 1,407 unique aptamers against 346 unique proteins, with the aptamer side essentially flat and the protein side concentrated. This asymmetry drives two design decisions: the frequency-matched negative sampling described below, and the protein-grouped split of Section 4.3.

Table 3 Re-curated dataset against the AptaTrans reference. Aptamer overlap is comparable between the two datasets and is not the source of the difference: among positive pairs it is under 3% in both, and the much higher all-rows figure reflects negative generation reusing each aptamer against several non-binding proteins. The distinguishing property is protein overlap, which is near-total in the reference partition and zero here.

	AptaTrans	This work
Total rows	2,900	4,733
Positive pairs	725	1,185
Negative pairs	2,175	3,548
Unique aptamers	709	1,407
Unique proteins	144	346
Aptamer length (nt)	6–274	8–274
Protein length (aa)	52–1,676	50–1,738
Aptamer alphabet	ACGTU + BN-	ACGTU
Duplicate rows	18	0
Test proteins also in training	120 / 121 (99.2%)	0 / 106 (0%)
Test aptamers also in training, positive pairs	4 / 144 (2.8%)	6 / 229 (2.6%)
Test aptamers also in training, all rows	406 / 415 (97.8%)	571 / 689 (82.9%)
Validation partition	none	473 rows, 85 proteins

4.5 Negative Generation

The source databases list only binders, so a negative (non-binding) set must be constructed. We generated negatives at a 1:3 positive:negative ratio, matching the ratio observed in the AptaTrans reference dataset (725 positives to 2,175 negatives). The method is frequency-matched negative sampling: each protein is assigned exactly three times its positive count in negatives, so that protein identity carries no label signal and the model cannot shortcut the task by memorising which proteins are over-represented among binders. Multi-binder aptamers are excluded from the negative pool to avoid mislabelling known binders, and a per-aptamer reuse cap prevents any single aptamer from dominating the negative side. The generated set was validated for zero overlap with the positives, correct per-protein proportions, and native DNA/RNA bases preserved at storage.

We deliberately did not adopt the sequence-mutation strategy also used by SelfTrans, in which half the bases of a known binder are randomly mutated while the protein is held fixed. Aptamer binding is structure-dependent, so a heavily mutated sequence is not guaranteed to be a non-binder, and a model trained on such pairs may learn to detect mutation rather than to predict interaction.

Because negatives were generated before the partition assignment, following the reference procedure, frequency matching holds across the dataset as a whole rather than within each partition; the realised per-partition ratios were 1:2.96, 1:3.08, and 1:3.06. Section 4.10 shows that this ordering also introduces a systematic label inversion under target-disjoint evaluation, and reports the corrected dataset in which negatives are generated within each partition after splitting.

4.6 Retrain on Re-curated Data

The re-curated dataset was retrained under the protocol of Section 4.3: initialization from the clean pretrained encoders, checkpoint selection on validation ROC-AUC, and a single test evaluation at the end, for up to 200 epochs at learning rate 1×10^{-5} on an NVIDIA A100, across three seeds.

The result is negative. On the protein-grouped test partition the model reaches ROC-AUC 0.465 ± 0.010 ($n = 3$), below chance, with MCC exactly 0.000 in every run. The classification report shows why: the model assigns the positive label to all 947 test pairs, giving recall 1.00 on the positive class and 0.00 on the negative class. Accuracy of 0.246 is precisely the positive rate in the test partition (233/947), which is what a constant positive prediction yields.

The failure is identical across seeds. All three runs return accuracy 0.2460 and F1 0.3949 to four decimal places, differing only in ROC-AUC (0.4651, 0.4556, 0.4757). The model converges to the same degenerate output regardless of initialization, rather than failing in a way that varies run to run.

Training dynamics are consistent with this. Training loss falls to approximately 0.02 by epoch 40 and to below 0.01 by epoch 100, indicating the training pairs are substantially memorized, while validation loss oscillates between 2 and 75 across consecutive epochs and validation ROC-AUC remains within 0.46–0.56 throughout. The checkpoint saves during training reflect fluctuation around chance rather than improvement.

Two explanations are available, and they cannot be distinguished from these runs alone: either generalization to unseen protein targets is beyond what this architecture achieves at this data volume, or some aspect of the configuration is at fault independently of the split. Section 4.7 reports a controlled experiment that separates them.

The figures in this section were obtained before the correction described in Section 4.10. On the corrected dataset the same configuration reaches 0.515 ± 0.020; see Section 4.12. The below-chance result reported here is attributable to negative-sampling order rather than to the protein-grouped split.

4.7 Control: Row-Level Split

To determine whether the result above is attributable to the removal of target overlap or to a defect in the setup, we constructed a control differing from the grouped condition in exactly one respect. The same 4,733 rows, the same character filtering, the same 70/10/20 proportions, the same class balance within each partition, the same hyperparameters, and the same three seeds were used; only the assignment of rows to partitions changed, from protein-grouped to row-level random. This reintroduces target overlap: 248 of 254 test proteins (98%) also appear in training, comparable to the 99.2% test-side overlap in the distributed AptaTrans partition.

Under this control the model reaches test ROC-AUC 0.689 ± 0.019 ($n = 3$), against 0.465 ± 0.010 in the grouped condition. The difference of 0.224 is approximately eight times the combined seed-to-seed spread of the two conditions and is therefore not attributable to run-to-run variance.

Two of the three control runs produced a working classifier rather than a constant prediction. In the better of the two, the model correctly identifies 66% of non-binding pairs and 65% of binding pairs, and is right 85% of the time when it predicts non-binding. Performance is modest, but the model is making distinctions, unlike protein-grouped runs.

This isolates the cause. The pipeline is capable of learning from this data: given overlapping targets it produces a model with real if limited discriminative ability. Removing the overlap collapses it to a constant prediction in every seed. The failure in Section 4.6 is therefore a property of the task under target-disjoint evaluation, not a defect in the training setup.

Table 4 The two conditions differ only in how rows were assigned to partitions. All data, filtering, hyperparameters, and seeds are held constant. Figures are mean ± std over three seeds; per-run values are given for the metrics where seeds diverge.

	Protein-grouped	Row-level (control)
Test proteins also in training	0 / 106 (0%)	248 / 254 (98%)
Test ROC-AUC	0.465 ± 0.010	0.689 ± 0.019
Test accuracy	0.246	0.496
Test F1	0.395	0.449
Test MCC	0.000, 0.000, 0.000	0.269, 0.000, 0.221
Best validation ROC-AUC	0.554 ± 0.004	0.710 ± 0.030
Selected epoch	47, 88, 65	78, 36, 23

Threshold-dependent metrics are unreliable here. Across the three control seeds MCC was 0.269, 0.000, and 0.221 despite ROC-AUC of 0.708, 0.691, and 0.669 respectively. One run collapsed to a constant positive prediction while retaining a usable ranking of test pairs; the same degenerate behaviour observed in the exact replication of Section 4.2, where MCC was 0.000 alongside ROC-AUC 0.769. The architecture produces informative rankings but poorly calibrated probabilities, so accuracy, F1, and MCC vary sharply with initialization even when ranking quality does not. We therefore treat ROC-AUC as the only metric comparable across runs, and report the others for completeness.

4.8 Cross-Model Evaluation on Target-Disjoint Data

Sections 4.6 and 4.7 establish that our fine-tuned model fails under target-disjoint evaluation while succeeding when targets overlap. This section asks whether the released AptaTrans model behaves the same way, by evaluating it on the same test partition.

Constructing a fair comparison. The released weights were fine-tuned on the Li et al. data, and our protein-grouped test partition is not disjoint from it: 39 of its 106 proteins appear in the Li et al. training partition, though only 4 of 947 exact (aptamer, protein) pairs do. Evaluating both models on the full partition would therefore let the released weights predict on 37% of targets they have already seen, while ours has seen none. We construct a doubly-disjoint subset by removing every row whose protein appears in the Li et al. training partition, leaving 704 rows across 67 proteins (174 positive, 530 negative, ratio 1:3.05) that neither model was trained on.

Table 5 Both models evaluated on the protein-grouped test partition and on the doubly-disjoint subset. The released weights are those distributed with the AptaTrans repository; our model is the checkpoint from Section 4.6, seed 1004.

Model	Test partition	ROC-AUC	PR-AUC	Acc.	F1	MCC
Released AptaTrans	full grouped (947)	0.572	0.297	0.246	0.395	0.000
Released AptaTrans	disjoint subset (704)	0.444	0.218	0.247	0.396	0.000
Ours (Sec. 4.6)	full grouped (947)	0.465	0.231	0.246	0.395	0.000
Ours (Sec. 4.6)	disjoint subset (704)	0.467	0.233	0.247	0.396	0.000

Two observations follow.

The overlap effect is measurable within a single model. Removing the 39 previously seen proteins drops the released model from 0.572 to 0.444. Our model, which had seen none of those

proteins under either partition, is unchanged at 0.465 and 0.467. The 0.128 drop is therefore attributable to the removal of seen targets rather than to any difference in test-set difficulty, since both models face the identical change in test data. This is the same effect demonstrated across conditions in Section 4.7, here isolated within one set of weights.

Neither model discriminates on unseen targets. On the doubly-disjoint subset the released model reaches ROC-AUC 0.444 and ours 0.467, both below chance, and both return MCC exactly 0.000 with accuracy equal to the positive rate (174/704 = 0.247). Both assign the positive label to all 704 pairs. The model distributed as achieving ROC-AUC 1.000 on its own test partition has no discriminative behaviour on protein targets outside its training data.

The two models are not meaningfully separable here: 0.444 against 0.467 is within the seed-to-seed spread we measure elsewhere, and neither produces a usable decision boundary. The finding is that both fail, not that either is preferable.

4.9 Reference Dataset Under a Target-Disjoint Split

The comparisons above vary both the dataset and the split. To isolate the split alone, we re-partitioned the Li et al. dataset by protein, using it exactly as distributed with no filtering or deduplication, and retrained under the same protocol across three seeds. The resulting partitions are 2,030 / 290 / 580 rows across 71 / 31 / 42 proteins, with zero protein overlap and the 1:3 class ratio preserved in each.

Table 6 Li et al. data, unmodified, under its distributed random partition and under a protein-grouped partition. The underlying rows are identical in both conditions; only the assignment of proteins to partitions differs.

Split	Test-side target overlap	Test ROC-AUC
Random (as distributed)	120 / 121 (99.2%)	0.753 ± 0.022
Protein-grouped	0 / 42 (0%)	0.428 ± 0.019

Performance falls by 0.325, approximately eight times the combined seed-to-seed spread of the two conditions. This is the strictest form of the comparison: no difference in curation, dataset size, source databases, or negative-generation strategy can account for it, because the data is identical in both rows.

Early stopping terminates at arbitrary points. Selected epochs across the three runs were 6, 3, and 58. The variation does not reflect one run learning longer than another. Validation ROC-AUC oscillates between 0.42 and 0.46 throughout without trending upward, so a marginal new maximum (the epoch-58 checkpoint improved on epoch 57 by 0.0003) resets the patience counter and extends the run. Patience-based stopping on a metric fluctuating around chance therefore halts at a point determined by noise, and the selected checkpoint is closer to the initial state than to a converged one.

Degenerate output, with one partial exception. Two of the three runs return MCC exactly 0.000 with accuracy 0.2483, the positive rate in the test partition (144/580), and assign the positive label to all 580 pairs. The third (seed 2024) returns MCC 0.1023 and accuracy 0.2879, indicating a small number of negative predictions rather than a fully constant output. The collapse is therefore near-total but not uniform across initialisations, unlike the re-curated grouped condition of Section 4.6, where accuracy and F1 were identical to four decimal places in all three runs.

Performance tracks target overlap across all five conditions.

Data	Split	Negatives	Overlap	Test ROC-AUC
Li et al.	random (shipped)	before split	99.2%	0.753 ± 0.022
Re-curated	row-level	before split	98.0%	0.689 ± 0.019
Li et al.	protein-grouped	before split	0%	0.428 ± 0.019
Re-curated	protein-grouped	before split	0%	0.465 ± 0.010
Re-curated	protein-grouped	after split	0%	0.515 ± 0.020

The two conditions with near-total target overlap land within 0.065 of one another despite spanning different datasets, curation procedures, source databases, negative-generation strategies, and partition sizes, while every target-disjoint condition falls at or below chance. Correcting the negative-sampling order raises the disjoint result from below chance to chance but does not produce a working predictor. The variable that separates the conditions is target overlap; the dataset is close to irrelevant.

We take this as evidence that the reported performance of this pipeline reflects memorisation of protein targets seen during training rather than learned aptamer–protein interaction, and that the published figure of 0.921 does not measure the capability the model is presented as having.

This result is bounded by what we tested: one architecture, one training set size, one hyperparameter configuration. It shows that AptaTrans does not generalise to unseen protein targets under target-disjoint evaluation. Whether a larger training set or a different architecture would succeed remains open; Section 4.11 reports that substituting pretrained protein representations does not.

The Li et al. figures in this section precede the correction of Section 4.10. Their negatives were not regenerated, since doing so would alter the distributed dataset beyond re-partitioning it; the 92.4% inversion rate in that condition should be borne in mind when comparing 0.428 against the corrected 0.515 obtained on our own data.

4.10 Negative-Sampling Order

Sections 4.6 and 4.9 report ROC-AUC *below* 0.5 under target-disjoint evaluation (0.465 and 0.428 respectively), across six runs none of which exceeded 0.476. A model without signal is expected to score 0.5; scoring consistently below it indicates something learned that is anti-correlated with the labels. This section identifies the cause, which turns out to be a property of how negative pairs are constructed rather than of the model or the split.

The mechanism. Negatives are generated by pairing each aptamer with proteins it is not reported to bind. When this is done before the partition assignment, an aptamer's negatives and its positive can land on opposite sides of the split:

Aptamer A_1 binds protein P_0; negatives generated against P_1, P_2, P_3	
Training partition	(A_1, P_1) negative (A_1, P_2) negative (A_1, P_3) negative
Test partition	(A_1, P_0) **positive**

The model encounters A_1 three times during training, labelled negative each time, and never as a positive. At test time it meets the single pair where A_1 does bind. Any per-aptamer association it has learned is exactly wrong for this case. This is not a failure to generalise; it is having been trained on the opposite of the answer.

Prevalence. We define a test positive as *inverted* if its aptamer appears in the training partition only as a negative.

This definition is deliberately conservative. An aptamer appearing once as a positive and three times as a negative is still skewed toward the negative class, so the figures below count only the rows where the bias is total and understate the effect.

Measured across all four datasets:

Table 7 Inverted test positives, and the accuracy of predicting each test pair's label from its aptamer's majority training label, against the accuracy of predicting the negative class for every pair.

Dataset and split	Inverted	Aptamer-history acc.	Always-negative acc.
Li et al., as distributed	131/145 (90.3%)	71.3%	76.1%
Re-curated, row-level	183/237 (77.2%)	72.6%	76.7%
Re-curated, protein-grouped	175/233 (75.1%)	72.2%	77.4%
Li et al., protein-grouped	133/144 (92.4%)	68.5%	76.1%

Two points follow. First, the inversion is present in the Li et al. dataset *as distributed*, at 90.3%. It is not introduced by our re-curation or by protein-grouped splitting; it is a property of the benchmark. Second, in every dataset, predicting a test pair's label from its aptamer's training history is *less* accurate than predicting the negative class unconditionally. The aptamer's training history does not merely fail to help, it points the wrong way.

Why random splitting conceals it. The inversion is equally present under random splitting (77.2% in our control, 90.3% in the reference partition), yet those conditions score 0.689 and 0.753. With near-total target overlap the model can identify the protein and has no need to rely on per-aptamer associations. Remove the overlap and the aptamer history is the only signal available, so the model follows it below chance. The two defects therefore compound: the first inflates reported performance, and correcting it exposes the second.

Correction. Negatives were regenerated after the split, drawing each partition's negatives only from aptamers and proteins already assigned to that partition. The original design decisions were preserved: 1:3 ratio, frequency-matched so each protein receives three times its positive count, multi-target aptamers excluded from the negative pool, and a per-aptamer reuse cap. All 1,185 positives are retained.

Table 8 Re-curated dataset before and after regenerating negatives within partitions. Positives are unchanged; only the negative pairs differ.

	Before	After
Rows	4,733	4,740
Positives	1,185	1,185
Rows, train / validation / test	3,313 / 473 / 947	3,344 / 464 / 932
Proteins, train / validation / test	155 / 85 / 106	142 / 74 / 93
Positive:negative ratio, each partition	1:2.96–3.08	1:3.00
Inverted test positives	175/233 (75.1%)	**0/233 (0%)**
Test negatives with positive-only history	47/714 (6.6%)	0/699 (0%)
Test aptamers unseen in training	52/233 (22.3%)	227/233 (97.4%)
Protein overlap between partitions	0	0

The proportion of test aptamers entirely unseen in training rises from 22.3% to 97.4%. This

follows from per-partition generation: an aptamer's negatives can no longer appear on the opposite side of the split from its positive. The test set is correspondingly harder, and legitimately so; the model has no per-aptamer memory to draw on, only the pairing.

The number of unique proteins falls from 346 to 309 (142 in training, 74 in validation, 93 in test). Regenerating negatives within partitions draws only from proteins already assigned to that partition, so proteins that previously appeared solely as negative partners are no longer represented. All 1,185 positives and every protein carrying a positive pair are retained; the reduction is confined to proteins that never had one. Figures reported elsewhere in this section as characterising the re-curated dataset, namely 346 unique proteins, 4,733 rows, describe the state before this correction, and Table 8 gives both.

4.11 Is Protein Representation the Bottleneck?

AptaTrans represents proteins with a six-layer transformer over a 715-token vocabulary of amino-acid 3-mers, pretrained on a PDB-derived corpus. Such an encoder has no principled representation of a protein outside its pretraining distribution, which is a candidate mechanical explanation for the collapse under target-disjoint evaluation. We tested this by substituting a protein language model and holding everything else fixed.

Setup. All 369 proteins were embedded with ESM-2 (650M parameters, pretrained on approximately 65M UniRef sequences), mean-pooled over residues to a 1280-dimensional vector per protein. Aptamers were represented by normalised 3-mer counts plus length, with thymine and uracil folded together so DNA and RNA aptamers share a feature space. The two are concatenated and passed to logistic regression and to a single-hidden-layer MLP. The split, test set, and metric are identical to the corrected condition of Section 4.12, so the figures below are directly comparable to AptaTrans on the same data.

The aptamer representation is deliberately crude. The intent is to vary the protein side alone: if pretrained protein embeddings recover performance, the encoder is the bottleneck.

Table 9 Models on ESM-2 protein embeddings, evaluated on the corrected protein-grouped split. Ablations zero one feature block at a time.

Model	ROC-AUC	PR-AUC	Accuracy	MCC
Logistic regression, ESM-2 + k-mers	0.531	0.269	0.258	0.032
MLP (256 hidden), ESM-2 + k-mers	0.561 ± 0.009	0.303	0.536	0.090
Ablation: aptamer k-mers only	0.529	0.272	0.278	0.044
Ablation: ESM-2 embeddings only	0.500	0.250	0.250	0.000
AptaTrans, same split	0.515 ± 0.020	—	0.254 ± 0.004	0.031 ± 0.028

Protein representation is not the bottleneck. Logistic regression on ESM-2 embeddings reaches 0.531 and the MLP 0.561 ± 0.009. Both are above chance, but neither approaches a usable predictor, and the substitution of a protein language model trained on five orders of magnitude more sequence data does not change the picture.

The ablations locate what signal remains. The protein-only model scores exactly 0.500 with MCC 0.000. This is the expected null: a model that cannot see the aptamer cannot represent an interaction, and it confirms the protein-grouped split leaks no protein-level label information.

The aptamer-only model scores 0.529, and the full logistic regression scores 0.531. Adding a 1280-dimensional protein embedding to the aptamer features improves ROC-AUC by 0.002.

Essentially all of the linear model's above-chance performance is attributable to properties of the aptamer alone, independent of which protein it is paired with. The MLP's additional 0.03 suggests a small non-linear interaction term, but the margin over chance is thin.

A model whose predictions depend on the aptamer alone is not performing interaction prediction. It is capturing that some sequences are more likely to be binders in general, rather than learning which protein a given aptamer binds. This is plausible, since every aptamer in the dataset was selected against something.

We note that the aptamer-only ablation scored 0.371 before negative regeneration, well below chance, and 0.529 after. The correction of Section 4.10 accounts for the difference, which is independent confirmation of that mechanism: with the inverted labels removed, aptamer features are no longer actively misleading.

4.12 Results on the Corrected Dataset

Retraining on the dataset of Section 4.10, with negatives generated within partitions after splitting and all else unchanged, gives test ROC-AUC 0.515 ± 0.020 over three seeds. The corresponding figure before correction was 0.465 ± 0.010.

Table 10 AptaTrans on the protein-grouped split, before and after regenerating negatives within partitions. Positives, split assignment, architecture, hyperparameters, and seeds are identical; only the negative pairs differ.

	Negatives before split	Negatives after split
Test ROC-AUC	0.465 ± 0.010	0.515 ± 0.020
Test accuracy	0.246 ± 0.000	0.254 ± 0.004
Test F1	0.395 ± 0.000	0.401 ± 0.001
Test MCC	0.000 ± 0.000	0.031 ± 0.028
Best validation ROC-AUC	0.554 ± 0.004	0.524 ± 0.012
Selected epoch	47, 88, 65	13, 1, 2
Inverted test positives	175/233 (75.1%)	0/233 (0%)

The correction accounts for the below-chance result. Performance rises by 0.050 and now sits at chance rather than beneath it. This confirms the mechanism of Section 4.10: the sub-0.5 scores were produced by aptamer-level label inversion, not by the model having learned something perverse about the interaction. With the inversion removed, the model neither helps nor hurts.

The corrected result is the substantive one. 0.515 ± 0.020 is indistinguishable from chance. Accuracy remains at 0.254, approximately the positive rate in the test partition (233/932 = 0.250), and MCC at 0.031 ± 0.028 is within one standard deviation of zero: the model still assigns nearly all test pairs to a single class. AptaTrans has no discriminative ability on protein targets absent from its training partition.

This is a stronger claim than the pre-correction figure supported. A below-chance result invites the objection that the dataset was malformed, and in this case that objection would have been correct. A chance result on a dataset verified to contain no label inversion, no protein overlap, and exact 1:3 class ratios in every partition does not admit the same reading.

Selected epochs collapse to the start of training. The three runs selected checkpoints at epochs 13, 1, and 2, against 47, 88, and 65 before correction. Validation ROC-AUC reaches its maximum almost immediately and never recovers it, so the retained checkpoint is close to the initial state.

Training loss continues to fall throughout, so the model does fit the training partition, but nothing it learns transfers to unseen targets.

Table 11 All configurations. Target overlap is the proportion of test-partition proteins also appearing in the corresponding training partition. Figures are mean ± standard deviation where a seed count is given; others are single runs. The final block is the only one measured on a dataset free of both defects, and is the result the conclusions rest on.

Configuration	Test ROC-AUC	Accuracy	F1	MCC
Li et al. test partition, as distributed — 99.2% target overlap				
Paper (reported)	0.921	0.876	0.761	0.679
Released weights (contaminated)	1.000	1.000	1.000	1.000
Honest run, single seed (val-selected)	0.739	0.738	0.539	0.365
Honest baseline (val-selected, $n = 3$)	0.753 ± 0.022	—	—	—
Exact replication (test-selected, fresh)	0.769	0.250	0.400	0.000
Re-curated data, row-level partition — 98.0% target overlap				
AptaTrans ($n = 3$)	0.689 ± 0.019	0.496 ± 0.216	0.449 ± 0.044	0.163 ± 0.143
Protein-grouped, negatives generated before splitting — 0% target overlap				
AptaTrans, re-curated data ($n = 3$)	0.465 ± 0.010	0.246 ± 0.000	0.395 ± 0.000	0.000 ± 0.000
AptaTrans, Li et al. data ($n = 3$)	0.428 ± 0.019	0.262 ± 0.023	0.402 ± 0.007	0.034 ± 0.059
Released weights, re-curated test set	0.572	0.246	0.395	0.000
Released weights, doubly-disjoint subset	0.444	0.247	0.396	0.000
Ours, doubly-disjoint subset	0.467	0.247	0.396	0.000
Protein-grouped, negatives generated after splitting — 0% target overlap, no inversion				
AptaTrans ($n = 3$)	0.515 ± 0.020	0.254 ± 0.004	0.401 ± 0.001	0.031 ± 0.028
ESM-2 + k-mers, logistic regression	0.531	0.258	0.401	0.032
ESM-2 + k-mers, MLP ($n = 3$)	0.561 ± 0.009	0.536	0.384	0.090
ablation: aptamer k-mers only	0.529	0.278	0.403	0.044
ablation: ESM-2 embeddings only	0.500	0.250	0.400	0.000

5 Discussion

5.1 What the Reported Performance Supports

The measurements in Section 4 do not show that binding predictors are useless. They show that reported performance answers a narrower question than it appears to.

Two use cases must be separated. In *re-ranking*, the target is known and often has hundreds or thousands of measured sequences; the task is to prioritise among candidates for that target. A model that recognises the protein and orders its known aptamers serves this purpose, and target overlap between training and evaluation is not a defect in that setting; it reflects the deployment condition. AptaBLE's own workflows describe exactly this: screening a SELEX round-9 pool and re-ranking clustered candidates for CD117.

In *discovery*, the target is novel and has no known binders. This is the setting in which a generative design engine operates, and the one in which predictors are presented as accelerating the field. Our results speak to this case, and they are negative: on protein targets absent from training, every configuration we tested performs at or below chance.

It is worth allowing that the field may not regard target overlap as a defect at all. From an experimental standpoint, knowing what a given protein already binds is useful information for proposing further candidates against it, and a model that exploits this is doing something

reasonable rather than cheating. Our objection is not that such models are trained on seen targets, but that performance measured under overlap is reported without qualification alongside claims of generalisation to unseen ones. The measurement and the claim describe different capabilities.

The distinction matters because published claims generally cover both. If the intended contribution is re-ranking, current evaluation practice is defensible and the reported figures are informative for that purpose. If it is discovery, the reported figures do not measure the relevant capability, and the evaluation protocol needs to change.

5.2 Evaluation Practice in Related Models

AptaBLE [7] is instructive, both because it explicitly addresses target overlap and because its own reported results bear on our finding.

It constructs benchmark datasets as largely non-overlapping with training and validation at both the aptamer and protein levels, and partitions training data by sequence homology clustering. This is a substantive improvement over the practice we document here, and indicates that evaluation norms in the field are shifting.

Three points nonetheless deserve note. First, AptaBLE reports selecting among several checkpoints across several trained models by benchmark dataset performance. This is the same selection-on-test we identify in AptaTrans (Section 4.1), and it means the reported figures are a maximum over models and checkpoints chosen by the score being reported. Second, its negative examples are generated by shuffling each positive aptamer's sequence and pairing it with the same protein, four per positive. This is the sequence-mutation strategy we decline in Section 4.5 for the same reason: a model can satisfy the objective by detecting sequence scrambling rather than by predicting interaction. It is not the inversion we document in Section 4.10, since the shuffled sequence differs from the original, but it is a related construction artifact. Third, the comparison against AptaTrans appears to be performed on the Li et al. dataset with the target overlap we characterise, so the margin reported between the two models is measured under conditions that inflate both.

AptaBLE's CD117 case study is the closest published analogue to a target-disjoint evaluation, and its reported outcome is consistent with ours: on a target absent from training, all validated non-binding aptamers scored highly and model score showed no observable correlation with experimentally determined dissociation constants. Both were resolved only after fine-tuning on aptamers from that same target. We read this as independent evidence that generalisation to novel targets remains unsolved, reported candidly by the authors.

5.3 Consequences for Generation

Our measurements concern the binding predictor. The pipeline it belongs to uses that predictor as the scoring function for Monte Carlo Tree Search: Apta-MCTS builds sequences nucleotide by nucleotide, querying the predictor at each simulation to estimate whether a partial or completed sequence binds the target [3, 9]. The search is guided entirely by that signal.

If the predictor is at chance for a target it has not seen, the search is optimising against noise for precisely the case a design engine addresses. We did not test the generator and cannot claim it fails; the dependency, however, is structural rather than incidental, and evaluating generator output under target-disjoint conditions is a necessary check that we are not aware has been performed. This is the most direct consequence of our finding for applied aptamer design, and we note it as the priority item in Section 6.

5.4 Why We Did Not Build the Scoring Function

The multi-objective scoring function described in Section 3 takes binding affinity from the predictor as one of five sub-scores, and the search that consumes the combined score is driven by the same predictor. Both uses assume the predictor discriminates on the target in question.

Building the remaining four sub-scores would have produced a system whose generation is steered by a component that, for novel targets, provides no signal. The measurements would have been uninterpretable: any observed improvement in specificity, stability or conjugation-site accessibility would reflect the behaviour of those sub-scores under a random binding term rather than a working multi-objective search. We considered establishing whether the foundation holds to be a precondition, and report that it does not.

5.5 Threats to Validity

Single architecture. All AptaTrans measurements use one architecture at one data volume with one hyperparameter configuration. The ESM-2 experiment varies the protein representation but not the interaction model. A different architecture may succeed where these do not.

Fixed splits. Each partition assignment was generated once with a fixed seed and held constant. The reported spreads capture training randomness, not variance across split assignments. A different protein partition might yield a different figure, though the consistency across four datasets and two splitting strategies argues against this being large.

Dataset scale. The corrected dataset contains 1,185 positive pairs across 309 proteins. Generalisation across protein space from this quantity of data may be infeasible regardless of method, which would make our result a statement about the field's data availability rather than about its models.

Aptamer representation held crude. We varied the protein side and left the aptamer side as k-mer counts throughout. If aptamer representation is the binding constraint, our ESM-2 experiment would not detect it.

Reference negatives uncorrected. We regenerated negatives for our own dataset but not for the Li et al. data, since doing so would modify the distributed benchmark beyond re-partitioning. The 0.428 figure for that condition therefore retains a 92.4% inversion rate and is not directly comparable to our corrected 0.515.

6 Future Work

Threats to the validity of what we measured are given in Section 5; this section concerns what should be done next.

Evaluate the generator under target-disjoint conditions. The most direct consequence of our finding is that Apta-MCTS, and any search guided by a learned binding predictor, may be optimising against noise for novel targets. Generating candidates for held-out proteins and assessing them by criteria independent of the predictor like docking score, secondary-structure stability, and motif recovery against known binders would establish whether generated sequences carry any target specificity. This requires no new model and is the natural next experiment.

Vary the aptamer representation. Our ESM-2 experiment substituted a pretrained protein encoder while leaving aptamers as k-mer counts. RNA-FM or nucleotide-transformer embeddings, or a secondary-structure-aware representation, would test the other half. AptaBLE pairs ESM-2 with nucleotide-transformer or RNA-FM [7], so this combination is known to be viable.

Extend the evaluation to other predictors. We evaluated AptaTrans and a linear and MLP baseline. Whether AptaBERT, AptaBLE and SelfTrans-Ensemble behave similarly under target-disjoint evaluation with corrected negative sampling is untested, and would establish whether the finding is a property of one architecture or of the task. AptaBLE is available as a hosted application rather than as downloadable weights, which constrains how it can be evaluated.

Physics-based scoring as a contrast class. Thermodynamic and docking-based methods cannot exhibit target leakage, having no training set. Benchmarking them on the same protein-grouped partition would establish a leakage-free reference point, at the cost of requiring three-dimensional structures for both partners and substantially more compute per pair. Even a small demonstration on the six held-out theranostic targets would be informative.

Data. 1,185 positive pairs across 346 proteins is small for a task requiring generalisation across protein space. Aptagen, APTABASE and PDB aptamer–protein complexes were identified but not incorporated, the last requiring structure-level filtering to separate genuine targets from crystallisation chaperones. Whether additional data changes the picture is the most basic open question, and the cheapest to answer as new databases are released.

Complete the multi-objective scoring function. The design in Section 3 remains well-motivated and its four non-binding sub-scores (specificity, structural stability via ViennaRNA, functional-site engagement via docking, and conjugation-site accessibility via solvent-accessible surface area) do not depend on the binding predictor. They could be developed and validated independently against known theranostic aptamers, so that they are ready when a predictor that generalises becomes available. Two design decisions remain open: the sub-scores are currently combined by a weighted sum with manually chosen weights, which embeds our priorities into the search and should be replaced by Pareto-based optimisation; and the therapeutic and imaging proxies are mechanistically motivated but not validated against experimental outcomes.

Scope. The intended system targets soluble proteins, since membrane and cell-surface targets have sparse aptamer data and less reliable structures. Extending to cell-surface targets, which include several of the most clinically relevant aptamer targets, depends on data availability rather than on method.

7 Conclusion

We set out to build a multi-objective design engine for theranostic aptamers on top of a published binding predictor, and began by reproducing that predictor's reported performance. It did not reproduce, and establishing why became the work.

The reference implementation has four independent defects: the distributed fine-tuned weights classify every pair in the paper's own test partition correctly and cannot be the weights that produced the reported figure; checkpoints are selected by test-set score with no validation partition; 120 of the 121 test proteins also appear in training; and the released code does not run on current PyTorch, with an undeclared dependency and no environment specification.

Across a series of controlled configurations spanning two datasets, two curation procedures, two splitting strategies and two negative-generation strategies, target overlap accounts for most of the variation in performance. Conditions with near-total overlap reach 0.689 to 0.753; every target-disjoint condition falls to chance or below. Evaluating the released weights on our own partition and then removing the 39 proteins they had already seen drops them from 0.572 to 0.444, isolating the effect within a single set of weights.

Correcting target overlap also exposes a second property of the benchmark. For the majority of

test positives, the aptamer appears in the training partition only as a negative, so any per-aptamer association the model learns is wrong for exactly those pairs. This affects 90.3% of test positives in the reference benchmark as distributed, and it is what produces below-chance rather than chance-level scores. Regenerating negatives within partitions removes the inversion entirely and raises the target-disjoint figure from 0.465 ± 0.010 to 0.515 ± 0.020. This remains indistinguishable from chance, but is now measured on a dataset verified free of both defects.

Substituting ESM-2 protein embeddings does not recover performance, and ablations show the protein contributes essentially nothing: a model given only the aptamer scores 0.529 and adding a 1280-dimensional protein embedding improves this by 0.002. What signal exists reflects properties of individual aptamer sequences rather than aptamer–protein pairing.

We read this as showing that reported performance supports re-ranking candidates for a target with known binders, which is a real and useful application, but does not support discovery against a novel target, which is what these tools are presented as enabling. The distinction has a direct consequence for generative design: Apta-MCTS is steered by the binding predictor at every simulation step, so for a novel target the search is guided by a signal we measure at chance. We did not test the generator and do not claim it fails, but the dependency is structural and evaluating it is the priority next step.

These conclusions are bounded by what we tested: learned sequence-based predictors, one interaction architecture, and approximately a thousand positive pairs. They do not establish that the task is infeasible. Physics-based scoring methods are unaffected by construction, having no training set through which leakage could occur. Whether a larger dataset, a different interaction architecture, or a richer aptamer representation would succeed remains open, and we release the corrected dataset and evaluation protocol so that such questions can be asked on a benchmark that measures what it appears to.

References

[1] Ilemobayo Victor Fasogbon et al. “Recent focus in non-SELEX-computational approach for de novo aptamer design: A mini review”. In: *Analytical Biochemistry* 699 (Apr. 2025), p. 115756. ISSN: 0003-2697. DOI: 10.1016/j.ab.2024.115756. URL: http://dx.doi.org/10.1016/j.ab.2024.115756.

[2] Lennart Bohrmann et al. “Aptamers used for molecular imaging and theranostics - recent developments”. In: *Theranostics* 12.9 (2022), pp. 4010–4050. ISSN: 1838-7640. DOI: 10.7150/thno.72949. URL: http://dx.doi.org/10.7150/thno.72949.

[3] Incheol Shin et al. “AptaTrans: a deep neural network for predicting aptamer-protein interaction using pretrained encoders”. In: *BMC Bioinformatics* 24.1 (Nov. 2023). ISSN: 1471-2105. DOI: 10.1186/s12859-023-05577-6. URL: http://dx.doi.org/10.1186/s12859-023-05577-6.

[4] Zhiming Zhang et al. “De novo design of functional nucleic acids of aptamers”. In: *Nature Computational Science* 6.4 (Mar. 2026), pp. 341–349. ISSN: 2662-8457. DOI: 10.1038/s43588-026-00965-3. URL: http://dx.doi.org/10.1038/s43588-026-00965-3.

[5] Jun Zhang et al. “Single-round evolution of RNA aptamers with GRAPE-LM”. In: *Nature Biotechnology* (Feb. 2026). ISSN: 1546-1696. DOI: 10.1038/s41587-026-03007-5. URL: http://dx.doi.org/10.1038/s41587-026-03007-5.

[6] Flemming Morsch et al. "AptaBERT: Predicting aptamer binding interactions". In: (Nov. 2023). DOI: 10.1101/2023.11.24.568626. URL: http://dx.doi.org/10.1101/2023.11.24.568626.

[7] Sawan Patel et al. "AptaBLE: A Deep Learning Platform for Aptamer Generation and Analysis". In: (Jan. 2026). DOI: 10.64898/2026.01.06.698056. URL: http://dx.doi.org/10.64898/2026.01.06.698056.

[8] Zhichao Yan, Yue Kang, and Buyong Ma. *Aptamer-protein interaction prediction model based on transformer*. 2025. DOI: 10.48550/ARXIV.2506.16084. URL: https://arxiv.org/abs/2506.16084.

[9] Gwangho Lee et al. "Predicting aptamer sequences that interact with target proteins using an aptamer-protein interaction classifier and a Monte Carlo tree search approach". In: *PLOS ONE* 16.6 (2021). Ed. by Jean-Christophe Nebel, e0253760. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0253760. URL: http://dx.doi.org/10.1371/journal.pone.0253760.

[10] Bi-Qing Li et al. "Prediction of Aptamer-Target Interacting Pairs with Pseudo-Amino Acid Composition". In: *PLoS ONE* 9.1 (Jan. 2014). Ed. by Anna Tramontano, e86729. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0086729. URL: http://dx.doi.org/10.1371/journal.pone.0086729.

[11] Ali Askari et al. "UTexas Aptamer Database: the collection and long-term preservation of aptamer sequence information". In: *Nucleic Acids Research* 52.D1 (Oct. 2023), pp. D351–D359. ISSN: 1362-4962. DOI: 10.1093/nar/gkad959. URL: http://dx.doi.org/10.1093/nar/gkad959.