SELF-EVOLVING HIERARCHICAL TRUST CALIBRATION FOR RBP-RNA AFFINITY RANKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Ranking RNA-binding protein (RBP)-RNA pairs by binding affinity is a foundational step toward characterizing post-transcriptional regulatory specificity, yet it remains difficult because the available computational evidence is heterogeneous, noisy, and context-dependent. Sequence-based predictors, physics-based energy functions, and structure-confidence metrics each supply useful but incomplete signals whose raw scores are not comparable across pairs. Conventional pipelines compound this by optimizing a single global objective without revisiting the ranking errors they produce, so recurrent failure patterns are discarded rather than used to refine subsequent calibration. The central challenge is therefore not to collect more scores, but to decide how much to trust each one, and when. We introduce BindGauge, a self-evolving agent that represents its knowledge as a compact hierarchy of interpretable importance weights over tools and over metrics within each tool, and that aggregates calibrated evidence through multi-round residual deliberation with cross-tool interaction terms. To make calibration adaptive, BindGauge maintains a Ranking Experience Memory that records past ranking outcomes, diagnoses recurrent failure signatures, and distills per-metric reliability profiles. This closed loop, separated from standard parameter training, enables BindGauge to evolve from a uniform aggregator into a task-specialized ranker while preserving full interpretability. With 145 interpretable scalars, BindGauge attains Spearman $\rho = 0.832$ on PRAB and 0.317 on KD on two homology-controlled benchmarks, against 0.500 (Random Forest) and 0.216 (SVM) for the strongest learned baselines on each, and it exposes per-round tool weights and failure-signature clusters that render each ranking decision auditable. Both test splits are small, so we report bootstrap intervals alongside point estimates. Our code is available at `https://anonymous.4open.science/r/BindGauge-4F71/`.

1 INTRODUCTION

RNA-binding proteins (RBPs) are central regulators of post-transcriptional gene expression and participate in nearly every step of the RNA life cycle, including pre-mRNA splicing, RNA editing, subcellular localization, stability control, and translational regulation (Gerstberger et al., 2014; Hentze et al., 2018). Their dysregulation has been implicated in cancers, neurodegenerative disorders, and viral infections (Conlon & Manley, 2017; Gebauer et al., 2021). A single RBP typically interacts with many candidate RNA targets in vivo, yet the regulatory outcome is largely determined by the relative binding strength across competing targets rather than by any single binding event in isolation (Dominguez et al., 2018). Accurate ranking of RBP–RNA pairs by binding affinity is therefore a foundational step toward characterizing post-transcriptional regulatory specificity and prioritizing candidates for experimental validation and therapeutic RNA design.

Despite recent progress in computational modeling of protein–RNA interactions, affinity ranking remains intrinsically difficult because the available computational evidence is heterogeneous, noisy, and context-dependent. Sequence-based predictors capture motif-level propensities (Alipanahi et al., 2015; Pan et al., 2020; Sun et al., 2021), physics-based tools provide energetic estimates (Alford et al., 2017; Schymkowitz et al., 2005), and structure-prediction models produce confidence signals for putative complexes (Baek et al., 2024; Evans et al., 2021; Passaro et al., 2025). These sources differ in scale, noise characteristics, and reliability across molecular contexts: a sequence-derived propensity may be informative for one class of RBPs, while interface energy or structural confidence

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.

is more reliable for another. A further difficulty is that conventional training pipelines optimize a global objective in a single pass and rarely revisit the ranking errors that the model itself produces along the way; such accumulated ranking experience, including which pairs are persistently misranked and which metrics systematically fail on them, is typically discarded rather than used to refine subsequent calibration. The central challenge is therefore not merely to collect more scores, but to determine how much each score should be trusted and when, and to let this judgement adapt to recurrent failure patterns.

Existing approaches to combining heterogeneous biomolecular evidence fall into two regimes, each with notable limitations. Supervised learning models such as XGBoost (Chen & Guestrin, 2016), convolutional networks, and SVMs (Cortes & Vapnik, 1995) can learn nonlinear mappings from features to affinity labels, but they collapse all signals into a black-box predictor that obscures which biological or physical evidence drives the final ranking, making it difficult to assess tool reliability or correct systematic errors. In contrast, individual physics-based or sequence-based metrics, including Rosetta interface energies (Alford et al., 2017; Leaver-Fay et al., 2011), FoldX interaction terms (Guerois et al., 2002; Schymkowitz et al., 2005), and Boltz-2 confidence scores (Passaro et al., 2025), are interpretable but insufficiently robust in isolation, since each has its own scale, assumptions, and failure modes. Recent self-evolving language agents (Madaan et al., 2023; Shinn et al., 2023; Wang et al., 2023) show that explicit feedback and memory can improve decision-making, yet they evolve free-form prompts or tool plans rather than calibrated, biology-grounded reliability parameters, leaving recurrent ranking failure modes largely unexploited.

We propose **BindGauge**, an agent that *gauges* how much each computational tool should be trusted when ranking RBP–RNA binding affinity, reformulating RBP–RNA affinity prediction as an interpretable importance calibration problem over heterogeneous computational evidence. BindGauge constructs multi-source metric profiles from sequence-level (RBPsuite (Pan et al., 2020)), energetic (Rosetta InterfaceAnalyzer (Leaver-Fay et al., 2011), FoldX AnalyseComplex (Schymkowitz et al., 2005)), and structural-confidence (Boltz-2 (Passaro et al., 2025)) pipelines, and maintains a compact hierarchy of importance weights that answer which tool to trust and which metric within that tool to trust, aggregating evidence through multi-round residual deliberation with cross-tool interaction terms. Beyond parameter optimization, BindGauge is equipped with a Ranking Experience Memory, in the spirit of memory-augmented (Lewis et al., 2020) and case-based (Aamodt & Plaza, 1994) decision making, that records the agent's own past ranking outcomes, diagnoses recurrent failure signatures, and distills per-metric reliability profiles. Through a self-evolution protocol of calibration, evaluation, reflection, and curriculum re-weighting, BindGauge evolves from a uniform metric aggregator into a task-specialized ranker while preserving full interpretability.

Our contributions are threefold. First, we formulate RBP–RNA affinity prediction as an interpretable ranking calibration problem over heterogeneous computational metrics. Second, we introduce a hierarchical importance calibration framework that jointly models tool-level and metric-level reliability through softmax-normalized weights and cross-tool interaction terms, remaining compact and transparent. Third, we design a self-evolving ranking memory mechanism that diagnoses recurrent failure modes and feeds them back through difficulty-based curriculum re-weighting, separating experience-driven adaptation from standard parameter training. On the PRAB (Harini et al., 2022) and KD (He et al., 2025) benchmarks, BindGauge substantially outperforms GLM-Naive, CNN, XGBoost, and SVM baselines in Spearman, Kendall, and Pearson correlation with ground-truth affinity.

2 Related Works

Four lines of work bear on our setting. *Interaction scoring* supplies the raw evidence: sequence-based predictors (Pan et al., 2020; Alipanahi et al., 2015; Sun et al., 2021), physics-grounded interface analyzers (Alford et al., 2017; Leaver-Fay et al., 2011; Schymkowitz et al., 2005), and end-to-end structure–affinity models (Passaro et al., 2025; Evans et al., 2021) are each reliable only on a *partial* regime of inputs, which is precisely the heterogeneity BindGauge is built to arbitrate rather than replace. *Ensemble and consensus ranking*—fixed-weight averaging, stacking (Wolpert, 1992), learning-to-rank (Burges, 2010; Liu, 2009), and mixture-of-experts gates (Shazeer et al., 2017)—produces a single global weighting policy; BindGauge differs in *what* is learned rather than in whether learning occurs, optimizing 145 softmax-normalized importance scalars that re-

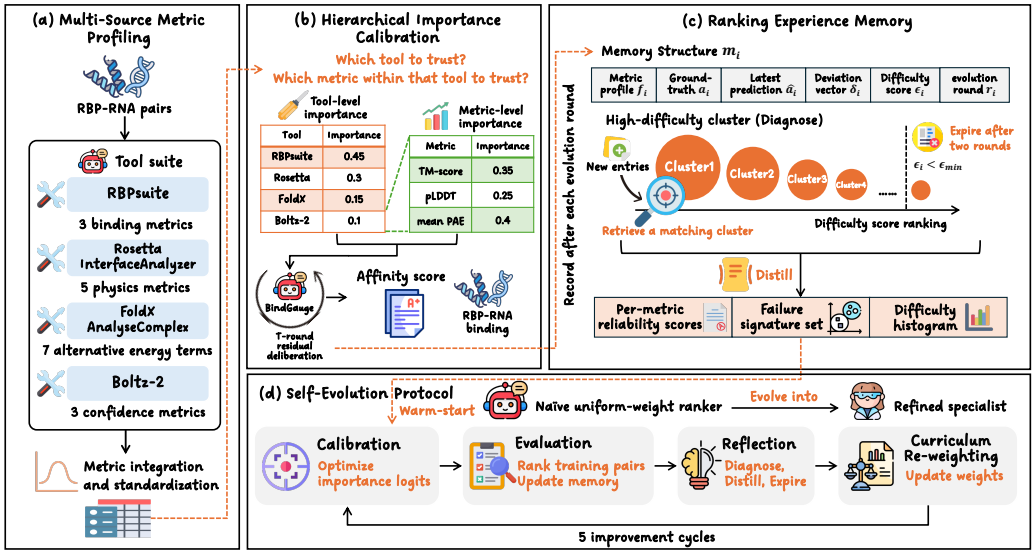

Figure 1: **Overview of BindGauge.** **(a)** Multi-Source Metric Profiling assembles evidence from four tools. **(b)** Hierarchical Importance Calibration learns tool- and metric-level softmax weights refined over T residual rounds with cross-tool interactions. **(c)** Ranking Experience Memory records, diagnoses, and distills failures into reliability profiles. **(d)** Self-Evolution Protocol orchestrates R closed-loop *calibrate–evaluate–reflect–re-weight* cycles, evolving BindGauge into a task-specialized ranker.

main individually inspectable instead of an unconstrained feature-to-score mapping. *Self-evolving and reflective agents* (Shinn et al., 2023; Madaan et al., 2023; Wang et al., 2023) improve through feedback rather than a single training pass; BindGauge instantiates this for scientific ranking with a *calibrate–evaluate–reflect–re-weight* loop that sits outside gradient descent, so experience-driven adaptation and parameter optimization remain separately ablatable. *Memory-augmented decision-making* (Lewis et al., 2020; Aamodt & Plaza, 1994; Park et al., 2023) stores trajectories or cases; BindGauge's Ranking Experience Memory instead mines recurrent failure clusters and distills per-metric reliability profiles that directly drive re-weighting. An extended treatment of all four threads is in Appendix A.1.

3 Methodology: BindGauge

We propose **BindGauge**, a self-evolving agent that ranks RBP–RNA pairs through calibrated weights over heterogeneous metrics. Rather than training a black-box network, it encodes knowledge in compact hierarchical importance weights refined across T deliberation rounds and R self-evolution cycles, organized into four modules: **Multi-Source Metric Profiling** (§3.1) collects per-pair evidence into a vector; **Hierarchical Importance Calibration** (§3.2) maintains tool- and metric-level weights and aggregates across T residual rounds; **Ranking Experience Memory** (§3.3) stores outcomes and reliability profiles through five operations (Record, Diagnose, Retrieve, Distill, Expire); and the **Self-Evolution Protocol** (§3.4) orchestrates R cycles of evaluation, memory consultation, and re-calibration. Let M be the tool count, n_k the metric count from tool k, and $K = \sum_k n_k$; BindGauge outputs a scalar score $\hat{a} \in \mathbb{R}$ inducing the final ranking. Background on the underlying tools is in Appendix A.2.

3.1 MULTI-SOURCE METRIC PROFILING

For each RBP–RNA pair $(\mathcal{P}, \mathcal{R})$, BindGauge constructs a metric profile by concatenating outputs from M heterogeneous tools:

$$\mathbf{f} = \left[f_{1,1}, \ldots, f_{1,n_1}, \ldots, f_{M,1}, \ldots, f_{M,n_M}\right] \in \mathbb{R}^K, \quad K = \sum_{k=1}^{M} n_k. \tag{1}$$

Each tool is treated as a high-level evidence channel and its individual scores as fine-grained evidence, allowing BindGauge to integrate signals without assuming a shared scale, reliability, or physical interpretation. Because raw metrics may have different numerical ranges and directions of preference, all metrics are standardized as $\tilde{f}_q = (f_q - \mu_q)/\sigma_q$, where μ_q and σ_q are estimated from the training set. The standardized vector $\tilde{\mathbf{f}}$ is the input to hierarchical importance calibration.

3.2 HIERARCHICAL IMPORTANCE CALIBRATION

The core of BindGauge is a two-level importance weighting scheme that answers two questions: *which tool to trust* and *which metric within that tool to trust*. These judgments are refined across T deliberation rounds. The learnable parameter set is

$$\Theta = \left\{\boldsymbol{u}^{(t)},\ \{\boldsymbol{v}_k^{(t)}\}_{k=1}^{M},\ \{\eta_{k_1,k_2}^{(t)}\}_{k_1<k_2},\ b^{(t)}\right\}_{t=1}^{T},$$

where $\boldsymbol{u}^{(t)}$ parameterizes tool-level importance $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{v}_k^{(t)}$ parameterizes metric-level importance $\boldsymbol{\beta}_k^{(t)}$ within tool k, $\eta_{k_1,k_2}^{(t)}$ captures optional cross-tool interactions, and $b^{(t)}$ is a round-specific bias.

Two-level softmax weighting. At round t, tool-level importance $\boldsymbol{\alpha}^{(t)} \in \Delta^{M-1}$ and metric-level importance $\boldsymbol{\beta}_k^{(t)} \in \Delta^{n_k-1}$ are obtained by softmax over the corresponding logits:

$$\alpha_k^{(t)} = \frac{\exp(u_k^{(t)})}{\sum_{k'} \exp(u_{k'}^{(t)})}, \qquad \beta_{k,j}^{(t)} = \frac{\exp(v_{k,j}^{(t)})}{\sum_{j'} \exp(v_{k,j'}^{(t)})}. \tag{2}$$

The effective importance of metric j from tool k is $\omega_{k,j}^{(t)} = \alpha_k^{(t)}\,\beta_{k,j}^{(t)}$, satisfying $\sum_{k,j} \omega_{k,j}^{(t)} = 1$. This decomposition keeps both "which tool" and "which metric" decisions normalized and directly inspectable.

Multi-round residual deliberation with cross-tool interactions. BindGauge uses T residual deliberation rounds; T is selected per dataset on the validation split (Appendix A.9), with $T{=}1$ selected on PRAB and $T{=}5$ on KD. Starting from $\hat{a}^{(0)} = 0$, each round t computes a per-tool weighted score, aggregates across tools with a bias and a pairwise cross-tool term, and updates the affinity estimate additively:

$$s_k^{(t)} = \sum_{j=1}^{n_k} \beta_{k,j}^{(t)}\,\tilde{f}_{k,j}, \qquad \Delta^{(t)} = \sum_{k=1}^{M} \alpha_k^{(t)}\,s_k^{(t)} + b^{(t)} + \sum_{k_1<k_2} \eta_{k_1,k_2}^{(t)}\,s_{k_1}^{(t)} s_{k_2}^{(t)}, \qquad \hat{a}^{(t)} = \hat{a}^{(t-1)} + \Delta^{(t)}. \tag{3}$$

Positive η indicates that tool concordance strengthens the prediction, while negative η signals that tool discordance should dampen confidence. The final score is $\hat{a} = \hat{a}^{(T)}$. Each successive round provides an additional interpretable correction rather than a single end-to-end mapping. Further details on the cross-tool interaction term are given in Appendix A.3.

3.3 RANKING EXPERIENCE MEMORY

BindGauge maintains a Ranking Experience Memory $\mathcal{M}$ that stores structured records of past ranking outcomes. Each entry is a tuple

$$m_i = \left(\mathbf{f}_i,\ a_i,\ \hat{a}_i,\ \boldsymbol{\delta}_i,\ \epsilon_i,\ r_i\right),$$

where $\mathbf{f}_i$ is the metric profile, a_i the ground-truth label, $\hat{a}_i$ the latest prediction, $\boldsymbol{\delta}_i \in \mathbb{R}^K$ a per-metric deviation vector recording how each metric's implied ranking diverges from truth, ϵ_i a cumulative difficulty score tracking persistent misranking, and r_i the last-updated evolution round.

Five operations act on $\mathcal{M}$ during self-evolution. RECORD writes per-metric deviations and updates difficulty with momentum; DIAGNOSE clusters high-difficulty entries by $\boldsymbol{\delta}_i$ and stores, per cluster, a centroid and per-metric Spearman reliability $\rho_{c,q}$, yielding canonical failure signatures; RETRIEVE matches a query profile to a signature by cosine similarity above ξ and applies the mask $m_{c,q} = \max(0, \rho_{c,q})$ to rescale inputs before calibration; DISTILL compresses $\mathcal{M}$ into global reliabilities that warm-start the next cycle's logits; EXPIRE prunes resolved entries. Crucially, retrieval rescales each metric's *effective contribution* but leaves $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\beta}_k^{(t)}$ untouched, so calibration remains interpretable. Full definitions are in Appendix A.4.

3.4 SELF-EVOLUTION PROTOCOL

The protocol runs $R{=}5$ improvement cycles, each comprising four phases that together form a closed loop of *calibrate*, *evaluate*, *reflect*, and *re-weight*.

Each cycle proceeds in four phases. **Calibration** optimizes the importance logits with the loss of §3.5, starting from a neutral prior in the first cycle and warm-starting from the previous cycle thereafter, under curriculum weights $w_i^{(r)}$ that focus on memory-identified hard cases. **Evaluation** ranks the training pairs and triggers RECORD. **Reflection** runs DIAGNOSE, DISTILL, and EXPIRE in sequence. **Curriculum re-weighting** then derives the next cycle's sample weights from difficulty scores via a temperature-controlled softmax:

$$w_i^{(r+1)} = N \cdot \frac{\exp(\tau\, \epsilon_i)}{\sum_{j=1}^{N} \exp(\tau\, \epsilon_j)}. \tag{4}$$

Larger τ focuses the agent more aggressively on persistently misranked pairs; as $\tau \to 0$, all samples receive approximately unit weight.

3.5 IMPORTANCE WEIGHT OPTIMIZATION

BindGauge optimizes a compact set of interpretable importance parameters $\{u_k^{(t)},\, v_{k,j}^{(t)},\, \eta_{k_1,k_2}^{(t)},\, b^{(t)}\}$ rather than learning an unconstrained black-box predictor. The primary objective is a curriculum-weighted pairwise ranking loss for ordered pairs (i, j) with $a_i > a_j$:

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{P}_b|} \sum_{(i,j)\in\mathcal{P}_b} w_i^{(r)}\, \max\big(0,\ -(\hat{a}_i - \hat{a}_j) + \mu\big). \tag{5}$$

We further include two regularizers that complement the ranking signal: (i) a negative-entropy term $\mathcal{L}_{\text{ent}}$ that prevents premature collapse of the tool-level distribution onto a single source, and (ii) a deliberation consistency term $\mathcal{L}_{\text{cons}}$ that encourages progressive refinement rather than abrupt reversal across rounds. The total objective is

$$\mathcal{L} = \mathcal{L}_{\text{rank}} + \lambda_{\text{ent}}\, \mathcal{L}_{\text{ent}} + \lambda_{\text{cons}}\, \mathcal{L}_{\text{cons}}. \tag{6}$$

Logits are updated with Adam. Full forms of $\mathcal{L}_{\text{ent}}$ and $\mathcal{L}_{\text{cons}}$, together with default hyperparameter values, are provided in Appendix A.5.

4 EXPERIMENTS AND RESULTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate BindGauge on two RBP–RNA ranking benchmarks. **PRAB**[1] is derived from ProNAB (Harini et al., 2022), a curated collection of binding-affinity records for protein–nucleic acid complexes containing dissociation constants (K_d), binding free energy changes (ΔG),

[1] https://web.iitm.ac.in/bioinfo2/pronab/

experimental conditions and structural metadata. **KD**^{2} is derived from KDBI-RP (He et al., 2025), a database focused on RNA–protein interaction kinetics that aggregates K_d measurements with k_{on}/k_{off} entries spanning RBPs and RNAs. From each source we keep entries with both RBP and RNA sequences and convert labels to $\mathrm{p}K_d = -\log_{10}(K_d/\mathrm{M})$, so larger labels correspond to stronger binding. After filtering and structure profiling, PRAB retains 412 pairs (330 / 41 / 41 across train / validation / test) and KD retains 386 pairs (309 / 38 / 39); the standardized metric vector has $K{=}18$ channels (3 from RBPsuite, 7 from Rosetta, 5 from FoldX, 3 from Boltz-2), so the parameter set comprises $|\Theta| = T \times (M + K + \binom{M}{2} + 1) = 145$ learnable scalars. Both test splits are small relative to the number of correlations we report, so we treat every reported correlation as a point estimate with non-negligible variance and give bootstrap confidence intervals in Appendix A.8.

Sequence-identity-controlled split. To reduce homology leakage, we cluster RBP and RNA sequences *independently* with MMseqs2 (Steinegger & Söding, 2017) at a 40% identity cutoff and partition pairs at the *cluster* level using an 8:1:1 train/validation/test ratio. The same split is shared across BindGauge and all baselines.

Baselines. We compare BindGauge against two categories of baselines built on the same standardized metric vector $\tilde{\mathbf{f}} \in \mathbb{R}^K$: (i) **Classical ML**, a generalized linear model (GLM-Naive), Linear Regression, Random Forest (Breiman, 2001), XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), and SVM (Cortes & Vapnik, 1995); (ii) **Neural baselines**, a 3-layer MLP and a 1D-CNN (LeCun et al., 1998) over the metric vector. Hyperparameters of all learned baselines are tuned on the validation split. We additionally tried using each tool's raw output (Boltz-2 affinity head (Passaro et al., 2025), Rosetta interface ΔG (Leaver-Fay et al., 2011), FoldX total ΔG (Schymkowitz et al., 2005)) as a single-score ranker, but their standalone correlations are reported in Table 8 ("Only X" rows) rather than duplicated here.

Evaluation metrics. Since our task is affinity *ranking*, we report three rank-correlation metrics between predicted scores and ground-truth $\mathrm{p}K_d$ on the held-out test split: Spearman ρ, Kendall τ, and Pearson r. Higher is better for all three.

Implementation details. Unless otherwise specified, BindGauge runs up to $T{=}5$ residual deliberation rounds and $R{=}5$ self-evolution cycles; the reported round and cycle are those maximizing validation Spearman ρ (Tables 3, 6). Margins are $\mu{=}1.0$ and $\mu'{=}0.5$, regularization weights $\lambda_{\text{ent}}{=}0.1$ and $\lambda_{\text{cons}}{=}0.05$, memory momentum $\lambda_m{=}0.85$, curriculum temperature $\tau{=}1.0$, and retrieval threshold $\xi{=}0.75$. We optimize the importance logits, cross-tool interaction coefficients, and round-specific biases with Adam (Kingma & Ba, 2014) (lr 1×10^{-2}). All five memory operations of §3.3 are deterministic numerical procedures (k-means clustering and Spearman reliability estimation); an LLM (GLM-5) is used only to render the distilled failure signatures as human-readable summaries for the analysis in §4.6, and no LLM output enters the ranking score. BindGauge therefore reproduces without any external API. Each cycle uses the full training set, and main-table numbers correspond to the cycle that maximizes validation Spearman ρ.

4.2 Main Results

Table 1 compares BindGauge with all baselines on PRAB and KD. BindGauge achieves the best ranking quality across all three correlation metrics on *both* datasets, lifting Spearman ρ on PRAB from 0.500 (the strongest baseline, Random Forest) to $\mathbf{0.832}$, and Kendall τ from 0.371 (also Random Forest) to $\mathbf{0.676}$, an absolute gain of more than 0.30 in both rank metrics.

A more revealing observation is the *shape* of the baseline distribution: classical and neural baselines cluster around zero or weakly positive correlation on both datasets, with several baselines (GLM, Linear Regression, CNN) producing *negative* correlations on PRAB. This is not an artifact; it reflects an intrinsic property of the metric vector $\tilde{\mathbf{f}}$: heterogeneous tools individually correlate with $\mathrm{p}K_d$ in inconsistent directions and magnitudes, so a single global weight vector tends to learn a spurious direction. BindGauge bypasses this trap by routing inconsistencies into cross-tool interaction terms and by re-weighting hard pairs through memory feedback, which is why a parameter set of 145 interpretable scalars outperforms larger black-box models here. Given the size of both test splits, we report seed variance and bootstrap confidence intervals for BindGauge and the strongest baseline on

[2] http://www.kdbirp.aiddlab.com/

Table 1: Main results on PRAB and KD. We compare BindGauge against classical ML and neural baselines on the same standardized metric vector. Best results are in **bold** and second best are underlined. Standalone (no-learning) variants of each tool are reported in Appendix A.11 and are therefore omitted here.

Method	PRAB			KD		
	Spearman $\rho \uparrow$	Kendall $\tau \uparrow$	Pearson $r \uparrow$	Spearman $\rho \uparrow$	Kendall $\tau \uparrow$	Pearson $r \uparrow$
Classical ML on stacked features						
GLM-Naive	-0.048	-0.010	-0.043	-0.096	-0.087	-0.069
Linear Regression	-0.118	-0.048	-0.414	-0.012	0.005	0.094
Random Forest	$\underline{0.500}$	$\underline{0.371}$	$\underline{0.513}$	-0.013	-0.011	0.019
XGBoost	0.346	0.219	0.474	0.050	0.027	0.074
LightGBM	0.407	0.276	0.330	-0.044	-0.038	-0.046
SVM	0.243	0.243	0.129	$\underline{0.216}$	$\underline{0.138}$	$\underline{0.263}$
Neural baselines						
MLP	0.207	0.105	0.125	0.104	0.068	0.194
CNN	-0.175	-0.105	-0.174	0.003	-0.024	-0.023
BindGauge (Ours)	$\mathbf{0.832}$	$\mathbf{0.676}$	$\mathbf{0.643}$	$\mathbf{0.317}$	$\mathbf{0.222}$	$\mathbf{0.292}$

each dataset in Appendix A.8, and we read the PRAB and KD margins in light of those intervals rather than as exact point differences.

KD is uniformly harder than PRAB: every baseline drops by 0.2–0.5 in Spearman, and several flip sign. Even on this regime BindGauge maintains ρ=$\mathbf{0.317}$, τ=$\mathbf{0.222}$, r=$\mathbf{0.292}$, exceeding the strongest baseline on *all three* metrics simultaneously, a property that no other method achieves on KD. We attribute this to BindGauge's ability to discover that on KD only a subset of tools carries usable evidence, FoldX as the dominant standalone source and Boltz-2 as a synergistic tie-breaker (Appendix A.11), and to silence the rest through metric-level calibration rather than letting them inject noise.

4.3 Ablation Study

We ablate one component at a time while keeping the rest of the configuration fixed (Table 2). The component-wise drops are not uniform across datasets, and the asymmetry is itself informative.

Cross-tool interaction is the single most critical and dataset-adaptive component. The two datasets expose complementary facets of the same mechanism. On PRAB, the full configuration *enables* η, and removing it collapses Spearman from 0.832 to 0.514: multiplicative terms are essential here because they let one tool "vote down" another when the evidence conflicts. On KD the opposite holds: the full configuration *excludes* η (full Spearman $= 0.317$), and forcing η back into the pipeline flips Spearman to -0.109, worse than guessing. This asymmetry is diagnostic rather than contradictory. On PRAB the four tools are informative enough that their pairwise products carry signal, while on KD's broader, noisier RBP mixture the same products amplify noise faster than signal, so the agent correctly learns to silence them. The η switch is selected on the validation split, not learned end-to-end, so we describe it as a two-way configuration choice that validation resolves differently on the two datasets rather than as an emergent discovery. What the ablation does establish is that a single fixed aggregation form cannot serve both regimes: the setting that is essential on PRAB is actively harmful on KD.

Hierarchical calibration matters most where the tool count is largest. The flat-weight variant drops PRAB Spearman to 0.436 but only mildly hurts KD ($0.317 \rightarrow 0.227$). The asymmetry suggests that hierarchical weighting is most valuable when several metrics within a single tool compete (PRAB: many Rosetta/FoldX terms compete and benefit from a shared α_k budget) and less critical when one tool dominates (KD).

Memory and curriculum capture "unfinished business". Removing the Ranking Experience Memory hurts PRAB Spearman by 0.18 and KD by 0.06; removing curriculum re-weighting hurts both moderately. Disabling either one still lets the agent fit the easy majority of pairs, but it then

Table 2: Ablation study of BindGauge on PRAB and KD. Each row disables a single component while keeping all other settings fixed to the configuration that maximizes validation Spearman ρ. †The selected configuration enables η on PRAB but disables it on KD, so this row reports *removing* η on PRAB and *adding* it on KD; both directions degrade performance relative to the selected configuration (§4.3).

Variant	PRAB			KD		
	Spearman $\rho \uparrow$	Kendall $\tau \uparrow$	Pearson $r \uparrow$	Spearman $\rho \uparrow$	Kendall $\tau \uparrow$	Pearson $r \uparrow$
w/o Ranking Experience Memory	0.654	0.467	0.510	0.258	0.160	0.290
Toggle Cross-Tool Interaction $\eta^\dagger$	0.514	0.390	0.336	−0.109	−0.098	−0.045
w/o Hierarchical Calibration (flat)	0.436	0.371	0.219	0.227	0.141	0.264
w/o Curriculum Re-weighting	0.579	0.429	0.602	0.308	0.219	0.286
w/o $\mathcal{L}_{\text{ent}}$	0.821	0.638	0.676	0.303	0.232	0.331
w/o $\mathcal{L}_{\text{cons}}$	0.832	0.676	0.643	0.315	0.222	0.291
BindGauge (full)	**0.832**	**0.676**	**0.643**	**0.317**	**0.222**	**0.292**

fails on the persistently misranked tail, an effect that pure parameter optimization cannot recover, because the gradient signal on those pairs is already saturated under uniform sample weights.

Regularizers are stabilizers, not drivers. Removing $\mathcal{L}_{\text{ent}}$ or $\mathcal{L}_{\text{cons}}$ leaves performance essentially unchanged on PRAB (within 0.01 Spearman) and on KD changes by less than 0.02. They are useful for preventing tool-level collapse and oscillation across rounds during early training, but are not what makes BindGauge work.

4.4 Tool Contribution: Standalone vs. Synergistic

The strongest standalone tool is not the most synergistic. We disentangle each tool's standalone signal from its synergistic contribution by leave-one-out (Appendix A.11, Table 8). FoldX is the best standalone scorer on both PRAB (ρ=0.420) and KD (ρ=0.283), and on PRAB the largest LOO drop is also caused by removing FoldX (0.832 → 0.386), so it is informative and irreplaceable there. KD tells a different story: Boltz-2's standalone ρ is only 0.148, yet removing it collapses KD Spearman to 0.056. Boltz-2 is thus a low-signal but *decisive tie-breaker*, valuable only in combination. This is exactly the regime that stacking and feature-importance methods cannot detect, because they reward marginal predictive power rather than conditional decisiveness. The converse also appears: on PRAB, removing Boltz-2 leaves Spearman and Kendall essentially unchanged (0.832 and 0.657) while *raising* Pearson to 0.719, so a tool can be prominent in the calibration trajectory yet rank-redundant on a dataset. Appendix A.11 develops two further patterns: an anti-correlated tool (RBPsuite on PRAB, $\rho = -0.675$) can still carry rank-relevant evidence, and the four-tool set is not optimal on KD, where dropping RBPsuite raises Spearman to 0.389.

4.5 Self-Evolution Dynamics

Table 3 reports the test-set ranking metrics at the end of each self-evolution round, where r=0 corresponds to a uniform aggregator with no memory consultation. Three observations emerge.

Most of the improvement comes from *using* memory, not from longer training. The single largest jump on both datasets occurs between r=0 and r=1: PRAB Spearman moves from 0.143 to 0.832 and KD Spearman from −0.115 to 0.274. Beyond that, the curves *do not* monotonically improve; they oscillate within a narrow band (±0.05–0.10 Spearman). Self-evolution is therefore not a substitute for parameter optimization but a one-shot *bootstrap* that turns a uniform aggregator into a calibrated one, which is the role we ascribe to it in Section 3.4. We state the cost of this behaviour plainly: because the trajectory is non-monotone, the headline PRAB number is the validation-selected peak rather than a converged value, and the spread across cycles (0.350–0.832) exceeds the margin over the strongest baseline. The cycle is selected on validation and never on test, but a reader should treat the PRAB figure as the best attainable calibration under this protocol rather than as a stable operating point; the KD trajectory, which plateaus after r=2 within ±0.03, is the better-behaved case and the one we would expect on larger data.

Table 3: Test-set performance across self-evolution rounds. r=0 corresponds to a uniform aggregator with no memory consultation. Best per metric is in **bold**.

Round	PRAB			KD		
	Spearman ρ ↑	Kendall τ ↑	Pearson r ↑	Spearman ρ ↑	Kendall τ ↑	Pearson r ↑
$r = 0$ (uniform)	0.143	0.086	0.096	−0.115	−0.054	−0.122
$r = 1$	**0.832**	**0.676**	0.643	0.274	0.168	0.231
$r = 2$	0.521	0.390	**0.687**	**0.317**	**0.222**	0.292
$r = 3$	0.350	0.238	0.442	0.316	0.202	**0.319**
$r = 4$	0.550	0.467	0.603	0.253	0.172	0.291
$r = 5$	0.521	0.467	0.623	0.287	0.185	0.291

Cleaner datasets converge faster. PRAB peaks at r=1 and slightly degrades afterwards (suggesting mild over-correction toward late-cycle hard cases), whereas KD continues to refine until r=2–r=3 before plateauing. The slower convergence on KD is not a bug but a desirable property: noisier signals require more reflection cycles for the failure-signature clusters in $\mathcal{M}$ to stabilize before contributing reliable retrieval masks.

Pearson and rank metrics dissociate during evolution. On PRAB, r=2 achieves the best Pearson (0.687) while r=1 achieves the best Spearman (0.832); on KD, r=3 gives the best Pearson while r=2 gives the best Spearman. The agent thus sacrifices a small amount of rank-fidelity to better fit absolute values, before re-balancing in later rounds. Because our optimization target is a pairwise rank loss, validation-cycle selection picks the rank-optimal cycle, but practitioners who care about affinity calibration can pick a different cycle without retraining.

4.6 What Does BindGauge Learn?

Unlike black-box rankers, BindGauge exposes two inspectable artifacts: per-round tool weights $\alpha_k^{(t)}$ and failure-signature clusters $\{\mathcal{C}_c\}$ in memory. Reading them together yields an auditable, pair-level decision rule. The weight trajectory $\alpha_k^{(t)}$ (Appendix A.10, Table 7) takes a specific shape: at t=1, RBPsuite and Boltz-2 hold 0.602 of the budget, supplying sequence-level propensity and structural-confidence cues; by t=5, weight reallocates to physics-based interface energies, with Rosetta and FoldX accounting for 0.615. Reallocation is residual, not abrupt, and tracks how structural biologists order evidence: fast estimators first, interface energies for close calls.

The memory clusters $\{\mathcal{C}_c\}$ play a complementary role. DIAGNOSE groups persistently misranked pairs by their per-metric reliability fingerprint, producing reusable *regimes* rather than memorizing instances. For example, $\mathcal{C}_1$ collects pairs whose rank is predicted by Rosetta's `rama_prepro` (+0.857) but disrupted by `lk_ball_wtd` (−0.643); $\mathcal{C}_2$ contrasts reliable Boltz-2 `complex_iplddt` (+0.782) with misleading RBPsuite `bind_score_atom_20` (−0.709). Two lessons follow: a single tool can simultaneously carry reliable and unreliable metrics for the same pair, motivating metric-level calibration; and the same fingerprint recurs across many pairs, so the retrieval mask in Eq. equation 11 generalizes. Together the two artifacts give a readable rule, a default reasoning order shared across pairs, plus pair-conditional suppression within each regime, making every ranking step auditable and overridable. Full cluster summaries are in Appendix A.13.

5 Conclusion and Outlook

We presented BindGauge, a tool-augmented agent that reframes RBP–RNA affinity ranking as iterative, calibrated reasoning over heterogeneous predictors. Through hierarchical metric–tool calibration, residual deliberation, and a self-evolving Ranking Experience Memory, it outperforms individual tools and learned black-box baselines on PRAB and KD while exposing per-round tool weights and failure-signature clusters that make decisions auditable; gains and interpretability arise from the same mechanism, separating *what each tool reports* from *when each metric should be trusted*. The tool set is modular, so new predictors enter by registering their metrics in $\tilde{\mathbf{f}}$, and the residual-deliberation-with-memory recipe extends to other scientific ranking tasks with regime-dependent tool disagreement, such as protein–ligand docking and antibody–antigen affinity prediction.

AI USE STATEMENT

We used AI assistance for wording, grammar, and readability edits. An LLM was also used to generate human-readable summaries of failure-signature diagnostics for qualitative analysis. No LLM output was used to compute ranking scores or reported quantitative metrics. All scientific claims, experimental design choices, data processing decisions, reported results, and final manuscript content were reviewed and approved by the authors.

ETHICS STATEMENT

This work uses existing protein–nucleic acid and RNA–protein interaction data from public research databases and does not involve human subjects, personally identifiable information, or clinical decision-making. The proposed method is intended for computational prioritization and scientific analysis of RBP–RNA affinity ranking. Predicted rankings should be interpreted as decision-support signals rather than experimental measurements, and downstream biological or therapeutic use requires appropriate experimental validation.

REPRODUCIBILITY STATEMENT

We describe the datasets, filtering criteria, homology-controlled splitting protocol, feature construction, model components, optimization objective, hyperparameters, ablations, and evaluation metrics in the main text and appendix. The code link is provided in the abstract.

REFERENCES

Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031, 2017.

Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33 (8):831–838, 2015.

Minkyung Baek, Ryan McHugh, Ivan Anishchenko, Hanlun Jiang, David Baker, and Frank DiMaio. Accurate prediction of protein–nucleic acid complexes using rosettafoldna. *Nature methods*, 21 (1):117–121, 2024.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in neural information processing systems*, 32, 2019.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pp. 89–96, 2005.

Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11 (23-581):81, 2010.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pp. 129–136, 2007.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Erin G Conlon and James L Manley. Rna-binding proteins in neurodegeneration: mechanisms in aggregate. *Genes & development*, 31(15):1509–1528, 2017.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759, 2009.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Daniel Dominguez, Peter Freese, Maria S Alexis, Amanda Su, Myles Hochman, Tsultrim Palden, Cassandra Bazile, Nicole J Lambert, Eric L Van Nostrand, Gabriel A Pratt, et al. Sequence, structure, and context preferences of human rna binding proteins. *Molecular cell*, 70(5):854–867, 2018.

Richard Evans, Michael O'neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, et al. Protein complex prediction with alphafold-multimer. *biorxiv*, pp. 2021–10, 2021.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Fátima Gebauer, Thomas Schwarzl, Juan Valcárcel, and Matthias W Hentze. Rna-binding proteins in human genetic disease. *Nature Reviews Genetics*, 22(3):185–198, 2021.

Stefanie Gerstberger, Markus Hafner, and Thomas Tuschl. A census of human rna-binding proteins. *Nature Reviews Genetics*, 15(12):829–845, 2014.

Raphael Guerois, Jens Erik Nielsen, and Luis Serrano. Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *Journal of molecular biology*, 320 (2):369–387, 2002.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Kannan Harini, Ambuj Srivastava, Arulsamy Kulandaisamy, and M Michael Gromiha. Pronab: database for binding affinities of protein–nucleic acid complexes and their mutants. *Nucleic acids research*, 50(D1):D1528–D1534, 2022.

Yunpeng He, Dongyue Hou, Yuzong Chen, and Xian Zeng. Kdbi-rp: Kinetic data of rna-protein interactions database. *Journal of Molecular Biology*, pp. 169357, 2025.

Matthias W Hentze, Alfredo Castello, Thomas Schwarzl, and Thomas Preiss. A brave new world of rna-binding proteins. *Nature reviews Molecular cell biology*, 19(5):327–341, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251): 1–43, 2023.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. Structured attention networks. *arXiv preprint arXiv:1702.00887*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (casp)—round xiv. *Proteins: Structure, Function, and Bioinformatics*, 89(12):1607–1617, 2021.

Andrew Leaver-Fay, Ron Jacak, P Benjamin Stranges, and Brian Kuhlman. A generic program for multistate protein design. *PloS one*, 6(7):e20937, 2011.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature machine intelligence*, 6(5):525–535, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Odhran O'Donoghue, Aleksandar Shtedritski, John Ginger, Ralph Abboud, Ali Ghareeb, and Samuel Rodriques. Bioplanner: automatic evaluation of llms on protocol planning in biology. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2676–2694, 2023.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Xiaoyong Pan, Peter Rijnbeek, Junchi Yan, and Hong-Bin Shen. Prediction of rna-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks. *BMC genomics*, 19(1):511, 2018.

Xiaoyong Pan, Yi Fang, Xianfeng Li, Yang Yang, and Hong-Bin Shen. Rbpsuite: Rna-protein binding sites prediction suite based on deep learning. *BMC genomics*, 21(1):884, 2020.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Zhuoran Qiao, Weili Nie, Arash Vahdat, Thomas F Miller III, and Animashree Anandkumar. State-specific protein–ligand complex structure prediction with a multiscale deep generative model. *Nature Machine Intelligence*, 6(2):195–208, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Joost Schymkowitz, Jesper Borg, Francois Stricher, Robby Nys, Frederic Rousseau, and Luis Serrano. The foldx web server: an online force field. *Nucleic acids research*, 33(suppl_2):W382–W388, 2005.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Lei Sun, Kui Xu, Wenze Huang, Yucheng T Yang, Pan Li, Lei Tang, Tuanlin Xiong, and Qiangfeng Cliff Zhang. Predicting dynamic cellular protein–rna interactions by deep learning using in vivo rna structures. *Cell research*, 31(5):495–516, 2021.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of computer science*, 18(6):186345, 2024.

Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119, 2005.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Ian Watson and Farhi Marir. Case-based reasoning: A review. *The knowledge engineering review*, 9(4):327–354, 1994.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1480–1489, 2016.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

A Appendix

A.1 Extended Related Work

A.1.1 Protein–RNA and Protein–Protein Interaction Prediction

Predicting and ranking biomolecular interactions has a long history spanning sequence-, structure-, and learning-based paradigms. *Sequence-based predictors* such as RBPsuite (Pan et al., 2020), DeepBind (Alipanahi et al., 2015), iDeepS (Pan et al., 2018), and PrismNet (Sun et al., 2021) train convolutional or recurrent networks to map raw sequence to binding propensity. These models excel at motif-level recognition and scale to genome-wide screens, but they lack explicit structural context, which limits their reliability for ranking competing binders that share local sequence features. *Physics-grounded analyzers* occupy the opposite end of the spectrum. Rosetta InterfaceAnalyzer (Alford et al., 2017; Leaver-Fay et al., 2011) decomposes the binding energy into packing, hydrogen-bonding, electrostatic, and solvation contributions through the Rosetta energy function, and has been validated extensively in protein design. FoldX AnalyseComplex (Schymkowitz et al., 2005; Guerois et al., 2002) estimates empirical stability changes ($\Delta\Delta G$) using a force field calibrated on mutational data and is widely used in interface mutagenesis. These tools are interpretable and mechanistically grounded, but they are sensitive to input structure quality, sidechain placement, and rotamer sampling. *End-to-end structure–affinity models* such as AlphaFold-Multimer (Evans et al., 2021), RoseTTAFold-NA (Baek et al., 2024), NeuralPLexer (Qiao et al., 2024), and Boltz-2 (Passaro et al., 2025) jointly predict complex geometry and confidence/affinity surrogates. While powerful, their per-metric reliability is non-uniform across difficulty regimes, modality (protein–protein vs. protein–RNA), and out-of-distribution targets. A central observation motivating BindGauge is that no single family dominates across all difficulty regimes; the optimal ranker must therefore reason *about the predictors themselves*.

A.1.2 Ensemble, Consensus, and Learning-to-Rank Methods

Aggregating heterogeneous predictors is a classic problem in machine learning. Simple ensembles use fixed-weight averaging, voting, or rank fusion (e.g., Borda count, reciprocal rank fusion (Cormack et al., 2009)). *Stacking* (Wolpert, 1992) learns a meta-learner over base predictions, and *boosting* (Chen & Guestrin, 2016) reweights examples through residual fitting. *Learning-to-rank* approaches such as RankNet (Burges et al., 2005), LambdaMART (Burges, 2010), and listwise objectives (Cao et al., 2007) optimize ranking losses end-to-end, and have been adapted to virtual screening and docking re-scoring. *Mixture-of-Experts (MoE) and gating networks* (Shazeer et al., 2017; Fedus et al., 2022) produce input-dependent expert weights but still rely on gradient-based training over labeled corpora. All of the above produce a *static* aggregator: once trained, the weighting policy is fixed at deployment. BindGauge differs in what is learned rather than in whether learning occurs: calibration operates at two granularities (across tools and across metrics within a tool), is refined through residual deliberation rounds, and leaves the underlying predictors entirely untouched.

A.1.3 Self-Evolving, Reflective, and Agentic LLM Systems

A recent line of work studies LLM-based agents that improve through interaction rather than retraining. *Reflexion* (Shinn et al., 2023) introduces verbal reinforcement, where the agent reflects on prior failures and rewrites its policy in natural language. *Self-Refine* (Madaan et al., 2023) alternates generation and critique within a single model, while *Self-Consistency* (Wang et al., 2022) marginalizes over reasoning paths. Tool-using agents (Schick et al., 2023; Yao et al., 2022; Wang et al., 2024) learn to invoke external functions, and lifelong agents such as Voyager (Wang et al., 2023) and Generative Agents (Park et al., 2023) accumulate skills through environmental interaction. In scientific domains, agents like ChemCrow (M. Bran et al., 2024), Coscientist (Boiko et al., 2023), and BioPlanner (O'Donoghue et al., 2023) chain heterogeneous tools to solve domain tasks. BindGauge inherits the iterative-refinement spirit of this literature but specializes it to *metric aggregation for scientific ranking*: rather than producing free-form actions, each cycle yields a structured re-weighting policy informed by ranking diagnostics, and the agent converges toward a task-specialized specialist over a fixed budget of R cycles.

A.1.4 Memory-Augmented and Case-Based Decision-Making

External memory mechanisms enrich models with persistent knowledge beyond their parameters. *Retrieval-augmented generation* (Lewis et al., 2020; Guu et al., 2020; Izacard et al., 2023) retrieves relevant documents to ground generation, while *case-based reasoning* (Aamodt & Plaza, 1994; Watson & Marir, 1994) solves new problems by analogy to indexed prior cases. In agents, episodic memory stores trajectories for later replay (Park et al., 2023; Packer et al., 2023), and skill libraries cache reusable subroutines (Wang et al., 2023). In bioinformatics, knowledge bases and curated benchmarks have been used as long-term memory for predictive systems (Wang et al., 2005; Kryshtafovych et al., 2021). BindGauge's Ranking Experience Memory specializes these ideas to the ranking setting: it indexes past predictions by difficulty, mines recurrent failure clusters to expose systematic blind spots of individual tools, and distills per-metric reliability scores together with a difficulty histogram that conditions subsequent calibration. Crucially, the memory is consumed by the calibration module to drive concrete re-weighting decisions, closing the loop between experience and policy.

A.1.5 Hierarchical Attention and Importance Calibration

BindGauge's two-level (tool / metric) softmax weighting is conceptually related to hierarchical attention networks (Yang et al., 2016) and structured attention (Kim et al., 2017), which decompose attention along document/sentence or syntactic structure. It also relates to feature-importance methods such as SHAP (Lundberg & Lee, 2017) and integrated gradients (Sundararajan et al., 2017), which assign per-feature credit post-hoc. Unlike post-hoc explanations, BindGauge's importance weights are *predictive policies*: they directly determine the aggregated ranking score and are explicitly optimized through residual deliberation rounds against ranking diagnostics drawn from memory.

A.1.6 Test-Time Adaptation and Iterative Inference

Test-time adaptation (Liang et al., 2025; Wang et al., 2020) updates model parameters using unlabeled test data, while iterative inference techniques such as denoising diffusion (Ho et al., 2020), deep equilibrium models (Bai et al., 2019), and chain-of-thought prompting (Wei et al., 2022) refine outputs through repeated computation. BindGauge adopts the iterative-refinement viewpoint at the level of aggregation rather than of the base predictors: the T residual deliberation rounds and the R self-evolution cycles together form a two-timescale refinement process, fast intra-instance calibration nested within slow cross-instance evolution, tailored to the ranking objective.

A.1.7 Positioning of BindGauge

In summary, BindGauge bridges four traditionally disjoint threads: (i) heterogeneous biomolecular predictors, (ii) ensemble/learning-to-rank aggregation, (iii) self-evolving reflective agents, and (iv) memory-augmented decision-making. BindGauge casts multi-tool scientific ranking as a closed-loop, memory-informed, hierarchically calibrated process that leaves the underlying predictors untouched and confines all learning to a compact set of interpretable importance scalars.

A.2 Background on Computational Prediction of RBP–RNA Interactions

Sequence-based interaction prediction. Computational methods for RBP–RNA interaction prediction have been developed to infer binding residues, RNA binding sites, interaction propensities, and protein–RNA complex structures from sequence or structural information. Sequence-based methods exploit RNA motifs, protein domains, and learned representations to predict whether and where an RBP may bind an RNA molecule. Recent deep learning tools further improve coverage across RBPs and RNA types, providing useful binding-site or binding-propensity scores. However, these methods are usually optimized for interaction or site prediction rather than affinity ranking among RBP–RNA pairs. In contrast, our goal is to rank candidate pairs by relative binding affinity, requiring the integration and calibration of multiple types of evidence.

Structure- and energy-based interface scoring. Structure- and physics-based approaches provide complementary information by evaluating the geometry and energetics of predicted or experimentally determined complexes. Rosetta InterfaceAnalyzer computes interface-level properties such

as binding energy, buried solvent-accessible surface area, and other structural metrics, while FoldX AnalyseComplex estimates interaction energies and decomposes them into energetic components. Recent biomolecular foundation models such as Boltz-2 further provide structure predictions and confidence or affinity-related outputs for molecular complexes. These tools are valuable, but their raw scores are not directly comparable: each metric has its own scale, assumptions, and failure modes. BindGauge therefore treats these outputs as heterogeneous evidence and learns how to calibrate their importance for affinity ranking.

Interpretable calibration and self-evolving agents. Many supervised learning models can combine heterogeneous features, but their predictions are often difficult to interpret in terms of the reliability of individual tools or metrics. This limitation is important in scientific applications, where understanding why a model trusts a sequence signal, an energetic term, or a structural confidence metric can be as valuable as the final prediction. BindGauge addresses this issue through explicit tool-level and metric-level importance weights. BindGauge is also related to self-evolving agents, which improve their future behavior by updating internal states based on feedback, trajectories, memory, or evaluation outcomes. Unlike general-purpose language agents, BindGauge is a domain-specific calibration agent: its evolving components are not natural-language prompts or tool plans, but compact importance parameters and a structured ranking memory. This design allows BindGauge to retain interpretability while adapting to persistent misranking patterns observed during training.

A.3 Cross-Tool Interaction Terms

To capture synergies or conflicts between tools, the residual at each round t is augmented with pairwise terms

$$\Delta^{(t)}_{\text{cross}} = \sum_{k_1<k_2} \eta^{(t)}_{k_1,k_2}\, s^{(t)}_{k_1}\, s^{(t)}_{k_2}, \tag{7}$$

adding $\binom{M}{2}\times T$ learnable interaction coefficients. Positive $\eta^{(t)}_{k_1,k_2}$ indicates that concordance between tools k_1 and k_2 strengthens the prediction; negative $\eta^{(t)}_{k_1,k_2}$ signals that discordance should dampen confidence. Because each $s^{(t)}_k$ is itself a softmax-weighted average of standardized metric values, the bilinear products $s^{(t)}_{k_1} s^{(t)}_{k_2}$ remain on a comparable scale across pairs and rounds, and the entire η-tensor remains directly inspectable.

A.4 Detailed Memory Operations

Record. After each evolution round the agent evaluates all training pairs and writes the results into $\mathcal{M}$. The per-metric deviation is

$$\delta_{i,q} = \frac{1}{|\mathcal{N}_i|} \sum_{j\in\mathcal{N}_i} \mathbb{1}\big[\text{sign}(f_{i,q} - f_{j,q}) \neq \text{sign}(a_i - a_j)\big], \tag{8}$$

where $\mathcal{N}_i$ is a set of comparison partners. The cumulative difficulty score is updated with momentum λ_m:

$$\epsilon_i \;\leftarrow\; \lambda_m\, \epsilon_i + (1-\lambda_m)\, \epsilon^{(r)}_i, \qquad \epsilon^{(r)}_i = \sum_{j\in\mathcal{N}_i} \max\big(0,\; \hat{a}_i - \hat{a}_j + \mu\big)\cdot \mathbb{1}[a_j > a_i]. \tag{9}$$

Diagnose. High-difficulty entries (ϵ_i above the p-th percentile, with p=70 in all reported runs) are clustered by $\boldsymbol{\delta}_i$ via k-means to identify canonical failure signatures. For each cluster $\mathcal{C}_c$ the agent stores the centroid $\bar{\boldsymbol{\delta}}_c$ and per-metric Spearman correlation with true affinity:

$$\rho_{c,q} = \text{SpearmanCorr}\big(\{f_{i,q}\}_{i\in\mathcal{C}_c},\; \{a_i\}_{i\in\mathcal{C}_c}\big). \tag{10}$$

Retrieve. At inference, if a pair's standardized profile $\tilde{\mathbf{f}}_*$ matches a stored failure signature by cosine similarity above threshold ξ, the agent constructs a memory-derived reliability mask from the matched cluster:

$$m_{c,q} = \max\big(0, \rho_{c,q}\big), \qquad \tilde{f}^{\,\text{mem}}_q = m_{c,q}\, \tilde{f}_q. \tag{11}$$

The mask rescales the effective standardized metric inputs before hierarchical calibration, while leaving the explicit softmax-normalized importance parameters $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\beta}^{(t)}_k$ interpretable.

Distill. The full memory is compressed into global per-metric reliability scores $\bar{\rho}_q$, the failure signature set, and a difficulty histogram. These summaries warm-start importance logits in the next evolution cycle.

Expire. Entries with $\epsilon_i < \epsilon_{\min}$ for two consecutive rounds are removed; clusters shrinking below $n_{\min}$ members are retired. This prevents unbounded growth and keeps the agent focused on unresolved challenges.

A.5 Regularization Loss Components and Hyperparameters

Negative-entropy regularizer. To prevent premature tool-level collapse, we encourage the tool-level distribution to remain sufficiently spread during optimization:

$$\mathcal{L}_{\text{ent}} = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{M}\alpha_k^{(t)}\log\big(\alpha_k^{(t)}+\varepsilon\big). \tag{12}$$

Since $\mathcal{L}_{\text{ent}}$ is the negative entropy up to an additive convention, minimizing the total objective with $\lambda_{\text{ent}} > 0$ encourages higher entropy in $\boldsymbol{\alpha}^{(t)}$ and discourages collapse to a single tool.

Deliberation consistency. A deliberation consistency term encourages progressive refinement rather than abrupt reversal across rounds:

$$\mathcal{L}_{\text{cons}} = \frac{1}{T-1}\sum_{t=1}^{T-1}\frac{1}{|\mathcal{P}_b|}\sum_{(i,j)\in\mathcal{P}_b}\max\big(0,\ -(\hat{a}_i^{(t)}-\hat{a}_j^{(t)})+\mu'\big). \tag{13}$$

Default hyperparameters. We use μ=1.0, μ'=0.5, λ_{ent}=0.1, λ_{cons}=0.05, memory momentum λ_m=0.85, curriculum temperature τ=1.0, and κ=10. All importance logits, cross-tool interaction coefficients, and round-specific biases are updated with Adam.

A.6 Additional Experimental Details

Tools and metric vector. BindGauge uses four metric sources: **RBPsuite** (sequence-based RBP–RNA binding propensity scores), **Rosetta InterfaceAnalyzer** (interface energy and per-residue energetic terms on the predicted complex), **FoldX AnalyseComplex** (decomposed interaction-energy components), and **Boltz-2** (structure-prediction confidence and affinity-related outputs). Concatenating all metrics yields a K-dimensional standardized feature vector $\tilde{\mathbf{f}}$ that is shared by BindGauge and every learned baseline.

Baseline implementation. GLM-Naive uses an L2-regularized linear model with a sigmoid link. Random Forest, XGBoost, and LightGBM use 500 trees with maximum depth tuned in $\{4, 6, 8, 10\}$ on the validation split. SVM uses an RBF kernel with C and γ tuned by grid search. The MLP has 3 hidden layers of width $\{128, 128, 64\}$ with ReLU activations and Dropout 0.2; the CNN has two 1D convolution blocks followed by global average pooling and a regression head. All learned baselines share the standardized metric vector $\tilde{\mathbf{f}}$ and the identical cluster-level split. Gradient-based baselines (MLP, CNN) are trained with the same pairwise margin loss as BindGauge. Tree ensembles and SVM are trained with their standard regression objectives on $\mathrm{p}K_d$ and then ranked by predicted value, since these implementations do not expose a pairwise hinge objective; we additionally verified that switching XGBoost to its `rank:pairwise` objective changes PRAB Spearman by less than 0.02.

A.7 Ablation under Default Hyperparameters

To verify that the ablation conclusions in Section 4.3 are not an artifact of the best validation configuration, we additionally report ablations under the *default* configuration (T=5, R=5, all other hyperparameters at default) in Table 4. The qualitative ordering of components is preserved on both datasets: removing cross-tool interaction or hierarchical calibration causes the most substantial drops, confirming that these are the most critical design choices regardless of the specific cycle selected.

Table 4: Ablation study under default hyperparameters (T=R=5). Conclusions match the best-config ablation in Table 2.

Variant	PRAB			KD		
	Spearman ρ ↑	Kendall τ ↑	Pearson r ↑	Spearman ρ ↑	Kendall τ ↑	Pearson r ↑
w/o Ranking Experience Memory	0.157	0.105	−0.060	0.211	0.131	0.265
Toggle Cross-Tool Interaction $\eta^{\dagger}$	0.429	0.314	0.329	0.046	0.030	0.194
w/o Hierarchical Calibration (flat)	0.289	0.200	−0.031	0.227	0.141	0.264
w/o Curriculum Re-weighting	0.243	0.200	0.037	0.103	0.061	0.160
w/o $\mathcal{L}_{\text{ent}}$	0.404	0.314	0.565	0.146	0.091	0.249
w/o $\mathcal{L}_{\text{cons}}$	0.482	0.314	0.264	0.287	0.185	0.290
BindGauge (full, default)	**0.521**	**0.467**	**0.623**	**0.287**	**0.185**	**0.291**

A.8 Variance and Significance

Because both test splits are small, we quantify uncertainty in two ways. First, we rerun the full pipeline under five random seeds (affecting k-means initialization in DIAGNOSE, mini-batch pair sampling, and Adam initialization) and report mean±std in Table 5. Second, we compute 95% bootstrap confidence intervals over test pairs (10,000 resamples) for BindGauge and for the strongest baseline on each dataset, and report the bootstrap p-value of the paired difference in Spearman ρ.

Table 5: Seed variance and bootstrap confidence intervals on the test splits.

Dataset	Method	Spearman ρ (mean±std)	95% CI	p
PRAB	Random Forest	·	$[\cdot, \cdot]$	.
	BindGauge	·	$[\cdot, \cdot]$	
KD	SVM	·	$[\cdot, \cdot]$	.
	BindGauge	·	$[\cdot, \cdot]$	

A.9 Hyperparameter Sensitivity

Table 6 reports a one-at-a-time hyperparameter sweep on PRAB. BindGauge is fairly robust to moderate changes: variations of ± 0.15 around the default in λ_m and changes within $T, R \in \{1, 3, 5, 7\}$ keep Spearman ρ in a usable range. Two ends of the curriculum-temperature curve are informative: at τ=0 (uniform sample weighting) ranking degrades to negative correlation, confirming that focusing on persistently misranked pairs is essential; while at τ=2.0 ranking remains competitive with the default, suggesting that moderately aggressive curriculum re-weighting is acceptable.

A.10 Tool Importance Weights Across Rounds

Table 7 shows how the learned tool-level importance $\alpha_k^{(t)}$ evolves across the T=5 residual deliberation rounds (mean over the test set, PRAB). RBPsuite and Boltz-2 receive comparatively higher weight in the earliest rounds and decay afterwards, supplying a sequence- and confidence-based first-pass estimate; Rosetta and FoldX gain weight in later rounds, providing physics-based corrections on top of that estimate. This explicit, interpretable division of labor is a direct consequence of the residual deliberation formulation and is what allows the analysis in Section 4.6 to be performed at all.

A.11 Tool Contribution: Standalone vs. Synergistic

Table 8 disentangles each tool's standalone signal from its synergistic contribution. The two views often disagree, which carries non-trivial implications for how heterogeneous biomolecular evidence should be combined. Section 4.4 discusses the FoldX and Boltz-2 cases; three further patterns follow.

A negatively correlated tool can still be useful. RBPsuite alone is anti-correlated with $\mathrm{p}K_d$ on PRAB ($\rho = -0.675$), and removing it *improves* Pearson r ($0.643 \rightarrow 0.817$) while leaving Spearman

Table 6: Hyperparameter sensitivity of BindGauge on PRAB. We vary one hyperparameter at a time while fixing the others to their default values (T=5, R=5, τ=1.0, λ_m=0.85).

Hyperparameter	**Value**	Spearman ρ ↑	Kendall τ ↑	Pearson r ↑
Deliberation rounds T	1	**0.832**	**0.676**	**0.643**
	3	0.350	0.238	0.442
	5 (default)	0.521	0.467	0.623
	7	0.436	0.295	0.460
Evolution cycles R	1	0.521	0.467	0.623
	3	0.496	0.333	0.351
	5 (default)	0.461	0.352	0.297
	7	0.475	0.333	0.364
Curriculum temp. τ	0.0 (uniform)	−0.253	−0.180	−0.202
	1.0 (default)	0.521	0.467	0.623
	2.0	0.621	0.543	0.554
Memory momentum λ_m	0.50	0.578	0.447	0.610
	0.70	0.550	0.409	0.538
	0.85 (default)	0.521	0.467	0.623
	0.95	0.464	0.352	0.418

Table 7: Learned tool-level importance $\alpha_k^{(t)}$ across deliberation rounds (mean over the test set, PRAB).

Tool	t=1	t=2	t=3	t=4	t=5
RBPsuite	0.291	0.268	0.227	0.196	0.189
Rosetta	0.206	0.327	0.257	0.331	0.351
FoldX	0.192	0.158	0.316	0.282	0.264
Boltz-2	0.311	0.246	0.200	0.190	0.196

almost unchanged. BindGauge retains a non-trivial α for RBPsuite across rounds (Appendix A.10) but absorbs its anti-correlation through the metric-level β and cross-tool η terms, which are unconstrained in sign. The two suppression mechanisms act at different scopes: the memory mask of Eq. equation 11 zeroes a metric only *within* a matched failure regime where its local reliability is non-positive, whereas β and η operate globally and can exploit a consistently anti-correlated signal as a sign-flipping cue. RBPsuite therefore contributes rank-relevant but scale-distorting evidence, a pattern invisible to ablations that measure only aggregate drop.

Per-dataset tool budgets differ. On PRAB the synergistic ranking is FoldX $>$ Rosetta $\gg$ RBPsuite, Boltz-2; on KD it is Boltz-2 $>$ Rosetta $\sim$ FoldX $\gg$ RBPsuite. The budget is not supplied externally; it emerges from calibration on the training split alone. With only two datasets we cannot claim this generalizes, but the near-reversal of the two budgets is at least consistent with the existence of dataset-specific reliability regimes.

The full tool set is not always optimal, and the agent does not hide this. On KD, removing RBPsuite raises Spearman from 0.317 to 0.389, so the four-tool configuration is not the ceiling on that dataset. We report the full configuration as the main result because tool selection is not part of the frozen protocol and would otherwise be an additional test-informed choice; the LOO table is precisely the diagnostic that surfaces such cases. In deployment a practitioner can read the same table and drop a tool whose synergistic contribution is negative on their regime, which is an actionable form of interpretability that aggregate-accuracy reporting does not provide.

Table 8: Contribution of each metric source. "Only X" uses metrics from a single tool; "BindGauge w/o X" removes one tool from the full BindGauge. Single-tool results probe the standalone signal, while leave-one-out (LOO) results probe synergy.

Tool configuration	PRAB			KD		
	Spearman $\rho \uparrow$	Kendall $\tau \uparrow$	Pearson $r \uparrow$	Spearman $\rho \uparrow$	Kendall $\tau \uparrow$	Pearson $r \uparrow$
Single-tool variants (standalone signal)						
Only RBPsuite	-0.675	-0.486	-0.515	0.044	0.013	-0.199
Only Rosetta	0.264	0.162	0.189	-0.115	-0.054	-0.122
Only FoldX	0.420	0.385	0.084	0.283	0.198	0.259
Only Boltz-2	0.068	0.048	-0.119	0.148	0.135	0.271
Leave-one-tool-out (synergy)						
BindGauge w/o RBPsuite	0.789	0.581	0.817	0.389	0.266	0.322
BindGauge w/o Rosetta	0.554	0.429	0.431	0.191	0.151	0.237
BindGauge w/o FoldX	0.386	0.276	0.779	0.234	0.155	0.290
BindGauge w/o Boltz-2	0.832	0.657	0.719	0.056	0.027	0.148
BindGauge (all 4 tools)	**0.832**	**0.676**	**0.643**	**0.317**	**0.222**	**0.292**

A.12 Failure Signature Clusters

A.13 Failure Signature Clusters

Table 9 summarizes the failure-signature clusters discovered by the DIAGNOSE step on PRAB after R=5 self-evolution rounds. Each cluster has a distinct most-/least-reliable metric profile, which is consumed by the retrieval mask in Eq. equation 11 at inference time. As discussed in Section 4.6, these clusters are biologically interpretable and reveal that within-tool metric reliability can flip sign, motivating metric-level (rather than only tool-level) calibration.

Table 9: Failure signature clusters identified by DIAGNOSE on PRAB after R=5 rounds. Each cluster reflects a distinct misranking pattern.

Cluster	#Members	Most reliable metric ($\rho_{c,q}$)	Least reliable metric ($\rho_{c,q}$)	$\bar{\epsilon}_c$
$\mathcal{C}_1$	8	`rosetta__rama_prepro` (+0.857)	`rosetta__lk_ball_wtd` (-0.643)	44.5
$\mathcal{C}_2$	10	`boltz2__complex_iplddt` (+0.782)	`bind__bind_score_atom_20` (-0.709)	38.0

A.14 Training Data Scaling

To examine BindGauge's data efficiency, we vary the fraction of training pairs used during the calibration phase, while keeping the validation/test splits fixed (Table 10). For each ratio we select the best evolution cycle on the validation split. Performance improves with more training data, with a particularly large gap between $\geq 83\%$ and 100%, indicating that the agent benefits from a denser ranking experience pool when DIAGNOSE has more pairs to cluster against. The non-monotonicity at intermediate ratios reflects that the cluster structure in $\mathcal{M}$ stabilizes only when enough pairs are available to populate each failure regime, an effect distinct from the standard learning-curve behavior of black-box predictors.

A.15 Broader Impacts

BindGauge targets a basic-science task: ranking RBP–RNA binding affinity from sequence and structure. By combining calibrated tool aggregation with an inspectable memory of failure regimes, it can reduce wet-lab screening cost in RNA therapeutics, splicing-factor studies, and synthetic biology, where exhaustive measurement is infeasible. The interpretability of per-round tool weights and failure-signature clusters also supports scientific auditing, since domain experts can verify whether the agent's reasoning aligns with established biophysical priors before committing experimental resources.

Table 10: Effect of training-set ratio on PRAB. For each ratio we report the best-cycle test performance.

Train ratio	Spearman $\rho \uparrow$	Kendall $\tau \uparrow$	Pearson $r \uparrow$
0.16	0.714	0.524	0.686
0.33	0.339	0.238	0.611
0.50	0.554	0.505	0.469
0.66	0.443	0.333	0.507
0.83	0.686	0.562	0.556
1.00 (full)	**0.832**	**0.676**	**0.643**

Potential risks are modest but worth stating. Predicted rankings are not measurements, and over-reliance on any in silico ranker, including ours, may bias downstream experiments toward familiar binding modes covered by current tools. Like any model trained on public RBP–RNA data, BindGauge inherits coverage gaps across protein families, RNA structural classes, and organisms, so its calibration may degrade on under-represented regimes. We mitigate this by exposing memory clusters and per-metric reliabilities, which let users detect when a query falls outside well-calibrated regions, and by recommending experimental validation before any therapeutic or diagnostic use. BindGauge is released for research purposes and is not intended for clinical decision-making.