Natural-Language-Guided Generator-Agnostic Shortlisting for Protein Binder Design

Gyubok Lee [1] **Kiwoong Yoo** [2] **Jimin Seo** [3] **Kyunghoon Hur** [4] **Edward Choi** [1]

Abstract

Modern de novo design workflows generate many candidate protein binders, but wet-lab validation capacity remains limited, making shortlisting a major bottleneck. We study whether LLMs can generate multi-metric ranking policies from precomputed structural-confidence and interface-quality proxy scores. Rather than proposing a new protein binder design pipeline, we focus on *post-generation binder shortlisting*: selecting the final top-K candidates from already generated binder pools using a shared panel of precomputed proxy scores. On the 10-target held-out split, averaging performance over five sampled global iterative `gpt-4o` policies reaches 0.589 Recall@10, modestly improving over the strongest single-feature fixed baseline, Protenix binder ipTM, which reaches 0.571 Recall@10. On the 3-target held-out subset comprising Nipah, RBX1, and TREM2, target-conditioned iterative `gpt-5.4` policies reach the strongest LLM performance, with 0.519 Recall@10 and 0.583 NDCG@10. These results suggest that LLM-generated ranking policies can act as an interpretable post-generation decision layer for combining heterogeneous proxy metrics to prioritize binders from large candidate pools.

1. Introduction

Recent de novo protein binder pipelines couple generative backbone models such as RFdiffusion (Watson et al., 2023), ProteinMPNN-style sequence designers (Dauparas et al., 2022), and structure-prediction filters based on AlphaFold2 (Jumper et al., 2021; Evans et al., 2021) or Boltz-2 (Passaro et al., 2025). These advances have made large-scale candidate generation increasingly routine, but experimental validation capacity remains limited. As a result, shortlisting has become a central determinant of how efficiently generated binders are converted into validated hits (Bennett et al., 2023; Pacesa et al., 2025; Adaptyv Bio, 2026).

A common practice is to prioritize or filter candidates using fixed thresholds, single-score rankings, or hand-tuned combinations of structure-prediction confidence scores and interface-centric proxies (Bennett et al., 2023; Pacesa et al., 2025; Adaptyv Bio, 2026). Representative examples include BindCraft's fixed AF2/Rosetta filter set, Adaptyv's Boltz-2 ipSAE-based computational selection, and PXDesign's Protenix-based confidence filters (Pacesa et al., 2025; Adaptyv Bio, 2026; Team et al., 2025a;b). These workflow-specific filters are important for candidate curation, but they do not fully resolve the final selection problem: after designs have been generated, filtered, or collected from different workflows, only a small number can be experimentally tested. A recent meta-analysis of 3,766 experimentally tested de novo binders reports that interface-focused confidence metrics such as ipSAE and orthogonal physicochemical descriptors can improve binder selection, while predictive performance still varies substantially by target (Overath et al., 2025). Together, these observations motivate a post-generation shortlisting setting that combines complementary proxy scores and tests whether the ranking rule should be global or target-conditioned.

To study this setting, we separate shortlisting from candidate binder generation and treat it as a post-generation decision problem. For each target protein, we fix the generated candidate pool and a common 17-feature panel of precomputed proxy scores before shortlisting. The panel combines model-native confidence scores from AF2-Multimer, Boltz-2, and Protenix with interface-level proxy descriptors computed from predicted complexes. We use *policy* broadly to denote any deterministic shortlisting rule that maps candidate proxy scores to a ranked or selected subset. We compare fixed heuristics, supervised machine learning (ML) base-

[1]Kim Jaechul Graduate School of AI, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea [2]LG AI Research, Seoul, South Korea [3]Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea [4]Korea Electronics Technology Institute (KETI), Seongnam, South Korea. Correspondence to: Gyubok Lee <gyubok.lee@kaist.ac.kr>.

Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)

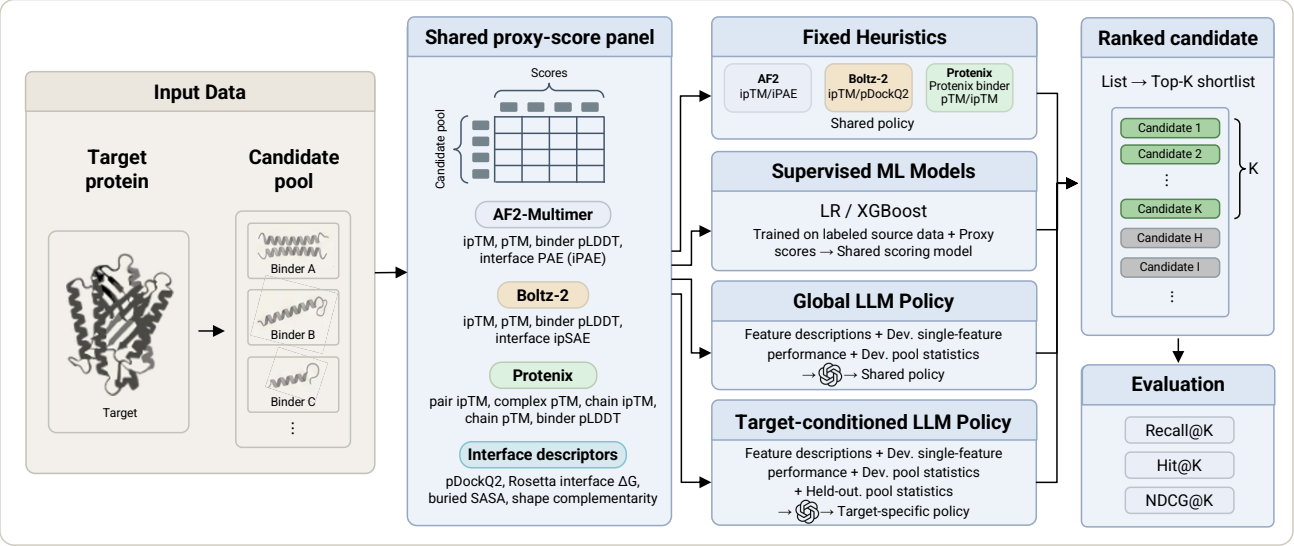

Figure 1. Overview of the post-generation binder shortlisting task. For each target protein, a fixed pool of candidate binders is generated in advance by upstream design workflows. A shared proxy-score panel is then computed for all candidates using AF2-Multimer, Boltz-2, Protenix/PXDesign, and interface descriptors such as Rosetta interface ΔG. Shortlisting methods, including fixed heuristics, supervised ML models, global LLM policies, and target-conditioned LLM policies, rank the candidates and return a top-K shortlist for experimental testing. Global LLM policies apply a single shared ranking rule to all held-out targets, whereas target-conditioned LLM policies generate a target-specific ranking rule.

lines, and large language model (LLM)-based shortlisting methods that generate either a single global policy for all held-out targets or a separate target-conditioned policy for each held-out target.

Our contributions are threefold:

- **A post-generation shortlisting task.** We formulate binder shortlisting as final candidate selection from fixed generated pools using a common 17-feature proxy panel and a shared top-K recall protocol.
- **A controlled comparison of shortlisting strategies.** We compare fixed single-feature ranking heuristics, supervised ML baselines, global LLM policies, and target-conditioned LLM policies on the same held-out candidate pools and top-K metrics.
- **LLM-generated ranking policies provide a competitive post-generation decision layer.** Across held-out binder pools, iterative LLM policies synthesize interpretable feature-weighted combinations of structural-confidence and interface-quality proxy scores. These policies are competitive with strong single-feature and supervised baselines in top-K recall, and in the best settings modestly improve Recall@10.

2. Related Work

Binder pipelines and candidate selection. Modern binder-design workflows often combine RFdiffusion backbone generation (Watson et al., 2023), ProteinMPNN-style sequence design (Dauparas et al., 2022), and structure-based validation or filtering with predictors such as AF2 (Bennett et al., 2023). Existing systems typically implement candidate selection through workflow-specific filters or ranking rules. BindCraft (Pacesa et al., 2025) uses fixed AF2-confidence, Rosetta, and interface-quality filters; Adaptyv's Nipah release used Boltz-2 ipSAE ranking together with community voting and expert curation (Adaptyv Bio, 2026); and PXDesign (Team et al., 2025b) uses Protenix/AF2-based filtering and releases PXDesignBench for standardized monomer and binder evaluation. These pipelines show that predictor-derived confidence and interface metrics are useful for binder triage, but they leave open how to combine heterogeneous proxy scores once a fixed candidate pool must be shortlisted for experimental testing. Consistent with this gap, a 3,766-binder meta-analysis reports that ipSAE-based scores outperform common interface-confidence metrics and that Rosetta-derived descriptors provide complementary signal, while predictive performance remains target-dependent (Overath et al., 2025). Our work is complementary: we do not introduce a generator or reproduce a pipeline-specific filtering stack, but study a post-generation, generator-agnostic shortlisting layer over heterogeneous candidate pools using fixed proxy scores from multiple predictor families.

Target-aware binder design. Cao et al. (2022) generate binders from target structure alone. Gainza et al. (2023)

Table 1. Composition of the development and held-out evaluation datasets after target-level merging and exact sequence de-duplication. Counts report designs, experimentally confirmed binders, and targets for each source. For pMHC targets, the table reports short target names. The full peptide–HLA pairs are SLLMWITQC–HLA-A*02:01 and RVTDESILSY–HLA-A*01:01.

Split	Source	Candidate origin	Target names	Designs	Binders	Targets
Development	BoltzGen	BoltzGen	AMBP, HNMT, IDI2, IL-7Ra, Insulin Receptor, MZB1/PERP1, PDGFR Beta, PD-L1, PHYH, PMVK, RFK	327	103	11
Held-out eval.	BindCraft1 revalidation	BindCraft	IFNAR2, spCas9, Der f 21, Der f 7	76	31	4
	Merged EGFR	Mixed participant-submitted methods, including Adaptyv EGFR competition submissions and BindCraft EGFR revalidation	EGFR	605	68	1
	pMHC minibinders	RFdiffusion + ProteinMPNN + AlphaFold2 filtering	NY-ESO-1 pMHC; RVTDESILSY pMHC	137	3	2
	Nipah release	Mixed participant-submitted methods	Nipah Virus Glycoprotein G (NiV-G)	1,030	103	1
	GEM/Adaptyv	Mixed participant-submitted methods	RBX1	321	9	1
	BioArena/Adaptyv	Mixed human/agent submissions with heterogeneous methods	TREM2	100	37	1
			Held-out subtotal	**2,269**	**251**	**10**
			Overall total (development + held-out)	**2,596**	**354**	**21**

APPRAISE (Ding et al., 2024) ranks engineered proteins by target-binding propensity through structure modeling. These works focus on generation or pairwise compatibility scoring. Our work differs in task: given a generated pool and precomputed proxy scores, we synthesize a separate decision policy for selecting K candidates.

LLMs and agents for protein design. ProtAgents (Ghafarollahi & Buehler, 2024) frames protein discovery as a multi-agent collaboration among LLM-backed roles that can retrieve knowledge, analyze structures, and call physics or machine-learning tools. ProteinCrow (Ponnapati et al., 2025) similarly builds an agentic protein-design assistant around curated tools, structural inputs, literature, and biochemical context. More broadly, Lee et al. (2025) review language-model use in protein design, including sequence modeling, context-conditioned design, and structure integration. Our use of LLMs is complementary: we apply them at the post-generation decision layer, where they propose ranking policies for constructing final shortlists from already-generated candidate pools.

3. Problem Setup

Shortlisting as policy synthesis. For a target protein t, we are given a generated candidate binder pool $\mathcal{C}_t$ of n_t designs. Each candidate $c \in \mathcal{C}_t$ has a fixed proxy-score vector $\mathbf{x}_{t,c} \in \mathbb{R}^m$ computed before shortlisting, where m is the number of common features available for every candidate. The task is to choose a subset $S_t \subseteq \mathcal{C}_t$ of size K that maximizes recall of experimentally verified binders under the validation budget. A method therefore outputs a ranking policy $\pi_t \in \Pi$, and a deterministic executor scores every candidate by π_t and returns the top K. In this formulation, ordinary metric-based ranking is a special case: sorting by one score, such as Boltz-2 ipSAE, is a one-feature policy. Policy synthesis generalizes this by choosing which proxy scores to combine and how strongly to weight them for the target pool.

Policy space. We use the term *policy* for the structured triple (F, w, g) that specifies a ranking function. Here $F \subseteq \{f_1, \ldots, f_m\}$ selects a subset of features, $w \in \{1, 2, 3\}^{|F|}$ assigns integer weights, and each selected feature has a predefined higher-is-better or lower-is-better direction. In the main LLM policy space, g is a weighted normalized sum: the executor normalizes every selected feature within the target pool, computes the aggregate score, sorts candidates in descending order, and returns the top K.

4. Datasets

4.1. Sources and split

We collected labeled binder-design data from eight public sources or workflow releases. A source is included only when it reports candidate-level experimental outcomes for tested designs, so that each target defines a retrospective shortlisting episode: the candidate pool is fixed, and every candidate has a binder/non-binder label.

Table 1 summarizes the target-disjoint development and held-out splits. The **development split** contains 11 targets from BoltzGen, a de novo binder-generation workflow and validation dataset (Stark et al., 2025). The **10-target held-out split** combines independent workflow outputs and public validation releases, including BindCraft revalidation (Pacesa et al., 2025), pMHC minibinders, Nipah (Adaptyv Bio, 2026), RBX1 (GEM Workshop & Adaptyv Bio, 2026), TREM2 (bioArena & Adaptyv Bio, 2026), and merged EGFR challenge/revalidation pools; its experimental labels were released after the documented `gpt-4o-2024-11-20` cutoff date used for the knowledge-leakage audit (October 1, 2023). We also report a **3-target held-out subset** consisting of Nipah, RBX1, and TREM2, whose labels were released after the docu-

Table 2. Candidate-level proxy score panel (17 active features). Feature definitions, monotonic directions, and extraction details are in Appendix C.

Family	Features	Description
AF2-Multimer	ipTM, pTM, binder pLDDT, interface PAE	Model-native complex confidence, binder local confidence, and interface PAE from AF2-Multimer (Evans et al., 2021; Mirdita et al., 2022); lower interface PAE is better.
Boltz-2	ipTM, pTM, binder pLDDT, ipSAE, pDockQ2	Model-native confidence scores from Boltz-2 (Passaro et al., 2025), ipSAE-style interface confidence (Dunbrack Jr, 2025), and pDockQ2 (Zhu et al., 2023), reported as metric-wise best summaries across target-MSA diffusion samples.
Protenix/PXDesign	pair ipTM, complex pTM, binder ipTM, binder pTM, binder pLDDT	Confidence fields from Protenix/PXDesign (Team et al., 2025a;b) covering complex, binder-chain, and binder-target confidence under our two-chain target-binder convention.
Rosetta interface descriptors	interface ΔG, buried SASA, shape complementarity	Rosetta descriptors (Stranges & Kuhlman, 2013; Lawrence & Colman, 1993) are computed on Boltz-2 target-MSA predicted complexes, not experimental structures.

mented `gpt-5.4` cutoff date (August 31, 2025). When the same biological target appears in multiple releases, we merge the corresponding pools and de-duplicate exact candidate amino-acid sequences. EGFR, for example, combines Adaptyv R1, Adaptyv R2, and BindCraft1 revalidation into 605 unique designs. Full preprocessing details, including binding-outcome parsing, source-specific target assignment, zero-positive target handling, and sequence de-duplication, are provided in Appendix A.

4.2. Feature extraction

Each candidate is represented by the 17 active proxy scores summarized in Table 2. Boltz-2 pDockQ2 and ipSAE are post-processed interface-quality or interface-confidence proxies computed from Boltz-2 predicted complexes and confidence outputs; pDockQ2 follows Zhu et al. (2023), and ipSAE follows the PAE-based interprotein scoring approach of Dunbrack Jr (2025). Rosetta InterfaceAnalyzer metrics are computed on Boltz-2 target-side-MSA complexes and used as geometry, burial, and energy proxy scores, not ground-truth binding energies. Protenix/PXDesign features provide complex, binder-chain, and pairwise binder-target confidence; Protenix ipTM-style scores are not assumed to be numerically calibrated to AF2-Multimer or Boltz-2 ipTM. All evaluated candidates have non-missing values for the 17 active features. Appendix C gives the active features, monotonic directions, and extraction summaries.

Inference settings. Feature extraction uses target-side-MSA complex predictions where available. For each target, we precompute one target-chain MSA and reuse it for all candidate binders; the de novo binder chain is kept single-sequence because designed binders have no natural homologs. We run AF2-Multimer with 3 models and 3 recycles, and Boltz-2 with 5 diffusion samples, 3 recycling steps, and 200 sampling steps. Protenix/PXDesign predictions likewise provide the target chain with the precomputed target MSA while keeping the binder chain single-sequence. Rosetta InterfaceAnalyzer is applied to Boltz-2 target-side-MSA predicted complexes. Templates are disabled in all complex-prediction runs.

5. Experimental Setup

5.1. Shortlisting methods

We use *global* to denote methods that use one rule unchanged across all held-out targets, and *target-conditioned* to denote methods that generate a separate rule for each held-out target. A target-conditioned method may choose different feature subsets or weights for different candidate pools.

Fixed ranking heuristics. We evaluate target-agnostic fixed rules as single-score references spanning the main structure-prediction signals used for binder triage: AF2-Multimer ipTM and interface PAE (lower is better) (Evans et al., 2021; Mirdita et al., 2022), Boltz-2 ipTM and a post-processed interface-quality estimate (pDockQ2 computed from Boltz-2 predicted complexes) (Passaro et al., 2025; Zhu et al., 2023), and Protenix/PXDesign binder-chain confidence (binder pTM and binder ipTM) (Team et al., 2025a;b). Each rule ranks candidates within a target pool by one score only, testing how far a commonly used single metric can go before any learned or LLM-composed policy is introduced. The Protenix binder metrics are the binder-chain confidence fields used by the PXDesign Protenix filters, evaluated here as single-score ranking heuristics rather than hard thresholds.

Logistic regression and XGBoost. We evaluate supervised ML baselines based on logistic regression and XGBoost (Chen & Guestrin, 2016). The first fits a logistic regression model with no regularization penalty and an XGBoost model on the 11-target BoltzGen development split using the same 17-feature panel as the LLM policies; these models test whether direct supervised learning over the feature panel is sufficient without target-conditioned policy synthesis. The second is a transfer baseline using Cao binder pools (Cao et al., 2022) with retrospective AF2 scores from Bennett et al. (2023). For this transfer setting, we use the available AF2 interaction pAE and binder pLDDT scores as the closest historical counterparts to our AF2 interface PAE and binder-chain pLDDT features. These transfer fea-

tures come from the AF2 scores previously computed by Bennett et al. for the Cao binder pools, rather than our AF2-Multimer target-side-MSA feature extraction, so they test cross-protocol transfer rather than a matched supervised retraining setting. Cao targets that overlap evaluation targets (EGFR, IL7Ra, and PDGFR) are removed before fitting. At evaluation time, each supervised model assigns every held-out candidate a fitted probability of being a binder, and candidates are ranked by this probability in descending order.

LLM policy sampling and averaging. We evaluate four LLM policy settings with `gpt-4o` on the 10-target held-out split and on the 3-target held-out subset, and we evaluate the corresponding `gpt-5.4` settings on the same 3-target held-out subset. All `gpt-4o` results use the `gpt-4o-2024-11-20` API snapshot. In the *global LLM* setting, the LLM sees natural-language feature descriptions, per-development-target pool distribution summaries, and development-set single-feature performance, emits one global policy, and that policy is fixed before held-out evaluation. The distribution summaries are reported separately for each development target pool rather than pooled across candidates, while the single-feature performance values are target-averaged Recall@10/Hit@10/NDCG@10 values from ranking each development target with one feature at a time. In the *global iterative LLM* setting, the final accepted policy from development-split iterative search is likewise fixed and applied unchanged to every held-out target.

The *target-conditioned LLM* setting is a label-free test-time adaptation setting: it instantiates one prompt per held-out target and includes the same development calibration context as the global LLM prompt, plus that held-out target pool's identifier and unlabeled score-distribution statistics computed only within the current candidate pool. Thus, the single-turn global and target-conditioned prompts share the development-side information, while target-conditioned prompting additionally exposes unlabeled current-target context and emits one rule per target. The *target-conditioned iterative LLM* setting additionally receives accepted/rejected development-search feedback from prior policy evaluations.

In all LLM settings, each sampled policy selects 3 to 5 features and assigns positive integer weights in $\{1, 2, 3\}$. Each LLM policy defines a weighted rank score over selected features: selected features are normalized to a 0–1 range within the target pool, lower-is-better features are direction-corrected so that larger normalized values are better, candidates are sorted by the resulting weighted score in descending order, and the top K candidates are selected. Appendix E summarizes the information available to each prompting setting and the shared policy constraints for the 17-feature panel. The reported LLM results average performance over five independently generated policies, which reduces run-to-run variability without treating policy averaging as a separate shortlisting method.

For proxy-score family ablations, we remove all selected terms belonging to one feature family from each sampled target-conditioned iterative policy, keep the remaining weights fixed, and re-evaluate the same deterministic executor. The LLM is not asked to regenerate or repair the policy after feature removal. Because proxy-score families are correlated and the remaining policy is not re-optimized, these ablations are interpreted as policy-dependence checks rather than monotonic feature-importance estimates. For model-to-model comparison, the main ablation table uses the shared 3-target held-out subset evaluated for both `gpt-4o` and `gpt-5.4`.

5.2. Evaluation protocol

Protocol. $K = 10$ is fixed before evaluation. The primary metric is Recall@10 over verified binders, with denominator $\min(K, n_{\text{binders}})$ so that targets with fewer than 10 verified binders can still attain a maximum score of 1.0 by recovering all positives. Secondary metrics are Precision@10 and normalized discounted cumulative gain (NDCG@10). Precision@10 is the wet-lab hit rate among the 10 selected designs, whereas NDCG@10 measures whether verified binders are concentrated near the top of the shortlist; its ideal DCG is computed with the same $\min(K, n_{\text{binders}})$ number of positives for each target. Statistics are averaged across target pools rather than pooled across individual candidates.

Held-out evaluation. Target-conditioned policies use unlabeled held-out pool statistics at test time, whereas fixed heuristics, supervised baselines, and global LLM variants are fixed before held-out evaluation and applied without held-out pool statistics. Additional provenance checks, historical dataset handling, and remaining leakage caveats are provided in Appendix B and Section 7.

6. Results

10-target held-out split. Table 3 reports the held-out evaluation with separate columns for the 10-target held-out split and the 3-target held-out subset. On the 10-target held-out split, the strongest single-feature fixed baseline is Protenix binder ipTM, with Recall@10 = 0.571, Hit@10 = 0.360, and NDCG@10 = 0.525. Averaging five global iterative `gpt-4o` policies gives the highest LLM Recall@10 and Hit@10, reaching Recall@10 = 0.589 and Hit@10 = 0.404, while target-conditioned iterative `gpt-4o` gives the highest LLM NDCG@10 (0.523). These results support the interpretation that LLM policies provide inspectable multi-feature ranking rules that can outperform strong predictor-native single-score baselines in Recall@10, while predictor-native single-score rankings remain strong NDCG baselines.

Table 3. Held-out evaluation ($K = 10$), grouped by method type. Recall@10 uses denominator $\min(K, n_{\text{binders}})$; Hit@10 is equivalent to Precision@10. Results are shown for the 10-target held-out split and for the 3-target held-out subset containing Nipah, RBX1, and TREM2. Dashes mark model/split combinations not reported. LLM results average five sampled policies.

	10-target held-out split			3-target held-out subset		
Method	Recall@10	Hit@10	NDCG@10	Recall@10	Hit@10	NDCG@10
Fixed ranking heuristics						
AF2 ipTM	0.437	0.360	0.417	0.433	0.433	0.487
AF2 interface PAE	0.417	0.290	0.337	0.300	0.300	0.298
Boltz-2 ipTM	0.403	0.290	0.377	0.133	0.133	0.166
Boltz-2 pDockQ2	0.515	0.340	0.383	0.400	0.400	0.416
Protenix binder pTM	0.455	0.280	0.327	0.233	0.233	0.222
Protenix binder ipTM	0.571	0.360	0.525	0.504	0.500	0.506
Supervised ML baselines						
LR-BG, no reg., 17 feat.	0.537	0.370	0.438	0.507	0.500	0.503
XGB-BG, 17 feat.	0.425	0.270	0.390	0.267	0.267	0.284
LR-Cao AF2	0.350	0.260	0.282	0.267	0.267	0.246
XGB-Cao AF2	0.397	0.280	0.337	0.267	0.267	0.272
LLM policies averaged over five samples: `gpt-4o`						
Global LLM	0.435	0.310	0.372	0.400	0.400	0.431
Target-conditioned LLM	0.498	0.378	0.445	0.493	0.493	0.526
Global iterative LLM	**0.589**	**0.404**	0.494	**0.504**	**0.500**	**0.529**
Target-conditioned iterative LLM	0.584	0.394	**0.523**	0.497	0.493	0.520
LLM policies averaged over five samples: `gpt-5.4`						
Global LLM	–	–	–	0.467	0.467	0.495
Target-conditioned LLM	–	–	–	0.502	0.500	0.514
Global iterative LLM	–	–	–	0.470	0.467	0.513
Target-conditioned iterative LLM	–	–	–	**0.519**	**0.513**	**0.583**

3-target held-out subset. The 3-target held-out subset includes Nipah, RBX1, and TREM2. On this subset, target-conditioned iterative `gpt-5.4` reaches the strongest LLM performance, with Recall@10 = 0.519, Hit@10 = 0.513, and NDCG@10 = 0.583. The strongest fixed and supervised baselines are Protenix binder ipTM (0.504/0.500/0.506) and LR-BG without regularization (0.507/0.500/0.503). XGBoost remains unstable in this small setting: it performs well on TREM2 alone but selected no verified binders for either Nipah or RBX1.

Generated LLM policy composition. The target-conditioned iterative `gpt-4o` policies concentrate weight on a small set of structure-confidence and interface-quality scores rather than spreading weight uniformly (Figure 2). Across the five policy samples for each of the 10 held-out targets, every policy selects Boltz-2 ipTM and Protenix pair ipTM. Rosetta shape complementarity appears in 40 of 50 policies, AF2 ipTM appears in 27, and Boltz-2 pDockQ2 appears once. By total selected weight, the policies allocate 38.6% to Boltz-2 features, 29.2% to Protenix features, 19.6% to AF2 features, and 12.6% to Rosetta interface descriptors. Thus, the generated policies do not appear to rely on arbitrary target-specific rules. Instead, they use a compact Boltz-2/Protenix confidence backbone and make target-conditioned adjustments through AF2 ipTM and Rosetta shape-complementarity inclusion.

Proxy-score group ablation of generated policies. Table 4 applies the same group leave-one-out procedure to target-conditioned iterative policies on the shared 3-target held-out subset, enabling a direct `gpt-4o` versus `gpt-5.4` comparison without changing the target set. The ablations are non-monotonic because the policy features are correlated and the remaining terms are not re-optimized after removal. For `gpt-4o`, removing Protenix causes the largest Recall@10 drop, while removing AF2 or Rosetta descriptors causes smaller drops and removing Boltz-2 slightly increases Recall@10. For `gpt-5.4`, removing Rosetta descriptors or Protenix is most harmful, consistent with its stronger use of target-specific interface-geometry and Protenix signals on the post-cutoff subset. These results should be read as policy-dependence checks, not monotonic feature-importance estimates.

Example generated policies. Table 5 shows RBX1 examples, including two fixed one-feature rules and the averaged iterative policies used by the reported LLM settings. Each sampled LLM policy is constrained to select 3 to 5 features, but an averaged example rule can contain more nonzero terms because it shows the union of features selected across five samples. The selected terms concentrate on strong predictor-native and interface-quality signals, such as Boltz-2 ipTM, Protenix pair ipTM, AF2 ipTM, Rosetta burial, and Rosetta shape complementarity, rather than introducing

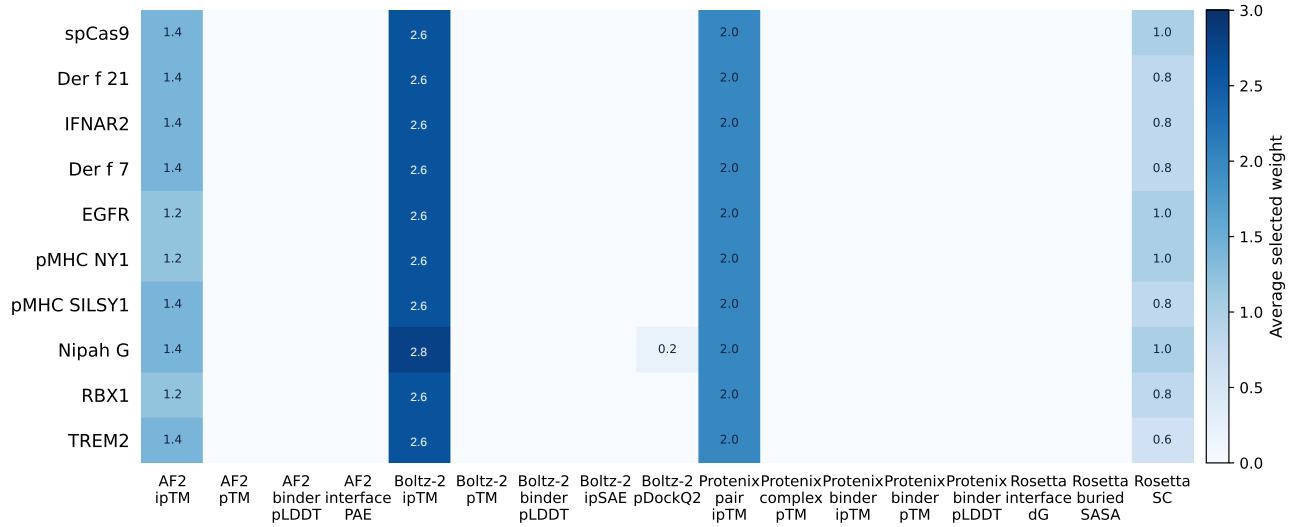

Figure 2. Feature weights in target-conditioned iterative `gpt-4o` ranking policies over the full 17-feature panel. The vertical axis lists held-out targets, and the horizontal axis lists all proxy-score features. Colored cells indicate features selected by the policies; the color scale shows the average selected weight across five sampled policies. Near-zero background cells indicate features not included in the ranking policy. For lower-is-better features, positive plotted weights apply to the direction-corrected normalized score used by the executor.

Table 4. Proxy-score group ablation of target-conditioned iterative policies on the shared 3-target held-out subset. Values are mean Recall@10 after removing all selected features from one group. No removal denotes the same ablation policy set before feature removal. Boltz-2 includes Boltz-2-derived pDockQ2; Rosetta denotes Rosetta interface descriptors.

Removed group	TC iter. `gpt-4o` 3-target	TC iter. `gpt-5.4` 3-target
No removal	0.497	0.519
AF2	0.436	0.499
Boltz-2	0.517	0.545
Protenix	0.340	0.453
Rosetta	0.455	0.423

unrelated proxy scores. For LLM policies, coefficients are average weights across five sampled policies, with unselected features contributing 0. Positive terms reward higher raw feature values, while negative terms reward lower raw feature values.

7. Analysis and Discussion

Global versus target-conditioned policies. The global LLM uses one rule inferred from development-target information only, then applies that rule unchanged to all held-out targets. Its Recall@10 on the 10-target held-out split is 0.435. Iterative feedback raises the global `gpt-4o` policy to 0.589 Recall@10, while target-conditioned iterative `gpt-4o` reaches a similar 0.584 Recall@10 and the highest LLM NDCG@10 on the same split (0.523). Thus, target conditioning does not uniformly dominate global policy synthesis: it can improve ranking quality without improving average top-K recovery. We interpret target conditioning as label-free adaptation to target-specific score distributions: the prompt can inspect the spread, skew, and relative behavior of proxy scores within a held-out candidate pool, then adjust which feature families to trust and how strongly to weight them, while the deterministic executor still applies the same fixed directions, within-target normalization, and top-K selection rule.

What iterative feedback adds. Iterative prompting adds development-search feedback by summarizing which feature-weighted policies were accepted or rejected on the development split and reporting their aggregate metrics. This changes the LLM's role from producing a rule from development summaries alone to producing a rule calibrated by prior policy search. On the 10-target held-out split, iterative feedback helps the global `gpt-4o` setting relative to the single global policy (0.589 vs. 0.435 Recall@10). The reflection records suggest that `gpt-4o` used this feedback mainly to converge on a compact, conservative Boltz-2/Protenix backbone; target conditioning then made only modest feature changes, which explains why target-conditioned iterative `gpt-4o` improves NDCG but does not exceed the global iterative policy in Recall@10. In contrast, `gpt-5.4` reflection records more explicitly discuss target-specific compression, spread, and redundancy in unlabeled score distributions. For RBX1, for example, the target-conditioned policies shift toward Rosetta burial/packing, Protenix pair ipTM, Boltz ipSAE, and AF2 interface PAE

Table 5. Example RBX1 ranking rules. Each $\hat{x}$ is a within-target 0 to 1 normalized feature; + rewards higher raw values and − rewards lower raw values. For LLM policies, coefficients are average policy weights across five sampled policies, with unselected features contributing weight 0 to the average; therefore, an averaged rule can include more nonzero terms than any single sampled policy. Global policies use one averaged rule for every target, whereas target-conditioned policies are averaged for RBX1 specifically. Candidates are sorted by the resulting rank score in descending order, and the top K candidates are selected.

Policy	Rank score
Fixed Protenix binder ipTM, RBX1	$\hat{x}_{\text{Protenix binder ipTM}}$
Fixed Boltz-2 pDockQ2, RBX1	$\hat{x}_{\text{Boltz pDockQ2}}$
Global iterative `gpt-4o`	$2.2\hat{x}_{\text{Boltz ipTM}} + 2.0\hat{x}_{\text{AF2 ipTM}} + 2.0\hat{x}_{\text{Protenix pair ipTM}} + 1.0\hat{x}_{\text{Rosetta SC}} + 0.2\hat{x}_{\text{Boltz pDockQ2}}$
Target-conditioned iterative `gpt-4o`, RBX1	$2.6\hat{x}_{\text{Boltz ipTM}} + 2.0\hat{x}_{\text{Protenix pair ipTM}} + 1.2\hat{x}_{\text{AF2 ipTM}} + 0.8\hat{x}_{\text{Rosetta SC}}$
Global iterative `gpt-5.4`	$2.4\hat{x}_{\text{Protenix pair ipTM}} + 1.8\hat{x}_{\text{AF2 ipTM}} + 1.2\hat{x}_{\text{Boltz pDockQ2}} + 1.2\hat{x}_{\text{Rosetta SC}} + 0.8\hat{x}_{\text{Rosetta buried SASA}} - 0.6\hat{x}_{\text{Rosetta interface dG}} + 0.4\hat{x}_{\text{AF2 pTM}} - 0.4\hat{x}_{\text{AF2 interface PAE}} + 0.4\hat{x}_{\text{Boltz ipTM}} + 0.2\hat{x}_{\text{Protenix binder pLDDT}} + 0.2\hat{x}_{\text{Protenix complex pTM}}$
Target-conditioned iterative `gpt-5.4`, RBX1	$2.0\hat{x}_{\text{Rosetta buried SASA}} + 1.8\hat{x}_{\text{Protenix pair ipTM}} + 1.6\hat{x}_{\text{Rosetta SC}} + 1.4\hat{x}_{\text{Boltz ipSAE}} + 0.6\hat{x}_{\text{Boltz pDockQ2}} - 0.6\hat{x}_{\text{AF2 interface PAE}} + 0.6\hat{x}_{\text{Boltz ipTM}} + 0.4\hat{x}_{\text{Boltz binder pLDDT}} + 0.4\hat{x}_{\text{AF2 ipTM}} + 0.2\hat{x}_{\text{AF2 pTM}}$

rather than simply reusing the global confidence backbone. This more selective adaptation is consistent with the 3-target held-out subset, where target-conditioned iterative `gpt-5.4` improves over single-turn target-conditioned `gpt-5.4` in Recall@10 (0.519 vs. 0.502) and NDCG@10 (0.583 vs. 0.514).

Strong single-score baselines remain important. Protenix binder ipTM is the strongest fixed one-feature baseline in the 17-feature panel, reaching 0.571 Recall@10 on the 10-target held-out split and 0.504 Recall@10 on the 3-target held-out subset. Boltz-2 pDockQ2 also remains competitive: it estimates interface quality from the predicted binder-target complex and confidence outputs, without using an experimental or designed reference structure. The generated policies do not replace these predictor-native signals. Instead, they combine them with complementary AF2 and Rosetta evidence; for example, all 50 target-conditioned iterative `gpt-4o` policies retain Boltz-2 ipTM and Protenix pair ipTM, while 40 include Rosetta shape complementarity and 27 include AF2 ipTM.

8. Conclusion

We study post-generation binder shortlisting: selecting final top-K candidates from fixed generated binder pools using precomputed structural-confidence and interface-quality proxy scores. We cast this task as policy synthesis and compare fixed heuristics, supervised baselines, and LLM-generated ranking policies. On the 10-target held-out split, averaging performance over five sampled global iterative `gpt-4o` policies modestly improves over the strongest single-feature baseline, Protenix binder ipTM, in Recall@10. The generated policies do not replace structure-prediction and interface-quality metrics. Instead, they combine AF2, Boltz-2, Rosetta interface, and occasional Protenix confidence signals into interpretable feature-weighted ranking rules that can be adjusted to each target pool. This suggests that LLM-generated ranking policies can serve as an interpretable post-generation decision layer for prioritizing binders from heterogeneous candidate pools, while strong predictor-native single-score rankings should remain explicit baselines.

Acknowledgment

This work was supported by the Institute for Information & communications Technology Planning & Evaluation (IITP) grant (RS-2019-II190075) and the National Research Foundation of Korea (NRF) grant (NRF-2020H1D3A2A03100945), supported by the Korea Government (MSIT).

References

Adaptyv Bio. Nipah competition results. Proteinbase collection, 2026. URL https://proteinbase.com/collections/nipah-binder-competition-results. Experimental validation results released January 21, 2026.

Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., O'Meara, M. J., DiMaio, F. P., Park, H., Shapovalov, M. V., Renfrew, P. D., Mulligan, V. K., Kappel, K., et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13 (6):3031–3048, 2017.

Bennett, N. R., Coventry, B., Goreshnik, I., Huang, B., Allen, A., Vafeados, D., Peng, Y. P., Dauparas, J., Baek, M., Stewart, L., et al. Improving de novo protein binder design with deep learning. *Nature Communications*, 14 (1):2625, 2023.

bioArena and Adaptyv Bio. BioArena x Adaptyv: TREM2 binder design competition. Proteinbase competition page, 2026. URL https://proteinbase.com/competitions/bioarena-adaptyv-trem2. Experimental validation results released March 28, 2026.

Cao, L., Coventry, B., Goreshnik, I., Huang, B., Sheffler, W., Park, J. S., Jude, K. M., Marković, I., Kadam, R. U., Verschueren, K. H., et al. Design of protein-binding proteins from the target structure alone. *Nature*, 605 (7910):551–560, 2022.

Chen, T. and Guestrin, C. 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.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning-based protein sequence design using proteinmpnn. *Science*, 378(6615): 49–56, 2022.

Ding, X., Chen, X., Sullivan, E. E., Shay, T. F., and Gradinaru, V. Fast, accurate ranking of engineered proteins by target-binding propensity using structure modeling. *Molecular Therapy*, 32(6):1687–1700, 2024.

Dunbrack Jr, R. L. Rēs ipsae loquunt: What's wrong with alphafold's iptm score and how to fix it. *bioRxiv*, 2025.

Evans, R., O'neill, M., Pritzel, A., Antropova, N., Senior, A., Green, T., Žídek, A., Bates, R., Blackwell, S., Yim, J., et al. Protein complex prediction with alphafold-multimer. *biorxiv*, pp. 2021–10, 2021.

Gainza, P., Wehrle, S., Van Hall-Beauvais, A., Marchand, A., Scheck, A., Harteveld, Z., Buckley, S., Ni, D., Tan, S., Sverrisson, F., et al. De novo design of protein interactions with learned surface fingerprints. *Nature*, 617 (7959):176–184, 2023.

GEM Workshop and Adaptyv Bio. GEM x Adaptyv: RBX1 binder design competition. Proteinbase competition page, 2026. URL https://proteinbase.com/competitions/gem-adaptyv-rbx1. Experimental validation results released April 26, 2026.

Ghafarollahi, A. and Buehler, M. J. Protagents: protein discovery via large language model multi-agent collaborations combining physics and machine learning. *Digital Discovery*, 3(7):1389–1409, 2024.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Lawrence, M. C. and Colman, P. M. Shape complementarity at protein/protein interfaces, 1993.

Lee, J. S., Abdin, O., and Kim, P. M. Language models for protein design. *Current Opinion in Structural Biology*, 92:103027, 2025.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S., and Steinegger, M. Colabfold: making protein folding accessible to all. *Nature methods*, 19(6):679–682, 2022.

Overath, M. D., Rygaard, A. S., Jacobsen, C. P., Brasas, V., Morell, O., Sormanni, P., and Jenkins, T. P. Predicting experimental success in de novo binder design: a meta-analysis of 3,766 experimentally characterised binders. *BioRxiv*, pp. 2025–08, 2025.

Pacesa, M., Nickel, L., Schellhaas, C., Schmidt, J., Pyatova, E., Kissling, L., Barendse, P., Choudhury, J., Kapoor, S., Alcaraz-Serna, A., et al. One-shot design of functional protein binders with bindcraft. *Nature*, 646(8084):483–492, 2025.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Ponnapati, M., Cox, S., Gordon, C. W., Hammerling, M. J., Narayanan, S., Laurent, J. M., Braza, J. D., Hinks, M. M., Skarlinski, M. D., Rodriques, S. G., et al. Proteincrow: A language model agent that can design proteins. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025.

Stark, H., Faltings, F., Choi, M., Xie, Y., Hur, E., O'Donnell, T., Bushuiev, A., Uçar, T., Passaro, S., Mao, W., et al. Boltzgen: Toward universal binder design. *bioRxiv*, pp. 2025–11, 2025.

Stranges, P. B. and Kuhlman, B. A comparison of successful and failed protein interface designs highlights the challenges of designing buried hydrogen bonds. *Protein Science*, 22(1):74–82, 2013.

Team, B. A. A., Chen, X., Zhang, Y., Lu, C., Ma, W., Guan, J., Gong, C., Yang, J., Zhang, H., Zhang, K., et al. Protenix-advancing structure prediction through a comprehensive alphafold3 reproduction. *BioRxiv*, pp. 2025–01, 2025a.

Team, P., Ren, M., Sun, J., Guan, J., Liu, C., Gong, C., Wang, Y., Wang, L., Cai, Q., Ma, W., et al. Pxdesign: Fast, modular, and accurate de novo design of protein binders. *bioRxiv*, pp. 2025–08, 2025b.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976): 1089–1100, 2023.

Zhu, W., Shenoy, A., Kundrotas, P., and Elofsson, A. Evaluation of alphafold-multimer prediction on multi-chain protein complexes. *Bioinformatics*, 39(7):btad424, 2023.

A. Dataset Preprocessing

Candidate pools are defined at the target level before feature extraction. A design is included only when the public release provides a candidate amino-acid sequence, a target sequence, and a candidate-level experimental binding outcome. Entries without an explicit binding measurement are treated as unlabeled rather than as negatives and are excluded from recall-based evaluation; this criterion removes 7 Adaptyv EGFR round-1 submissions. For ProteinBase-style releases, binding outcomes are read from the release-provided evaluation records. A design is labeled positive if any intended-target binding record is positive and negative if all intended-target binding records are negative. Expression measurements and binding-strength annotations are retained as metadata, but they do not define the binary label.

Target identifiers are assigned according to the experimental assay target reported by each source. Single-target competitions, including Adaptyv EGFR R1/R2 and Nipah, use the competition target, and off-target or control assay records are not used for the binary intended-target label. BindCraft1 revalidation candidates are assigned by their assay target in `evaluations`. For BoltzGen, PDB-like structural seed identifiers are mapped to the released biological assay target rather than used directly as target names; for example, `1g13`, `3apu`, `2a1x`, and `3qkg` correspond to GM2A, ORM2, PHYH, and AMBP, respectively.

Source-level pools are retained for audit and feature-extraction checks, including BoltzGen targets with no released positives. Targets with zero positives (GM2A, ORM2, and TNF-α) are excluded from recall-based evaluation because Recall@10 is undefined when the target-level positive denominator is zero. When multiple releases contain the same biological target, the evaluation pool merges those releases and de-duplicates exact candidate amino-acid sequences. If duplicate sequences have discordant labels, the merged label is positive if any duplicate record is positive. For EGFR, this merge combines Adaptyv R1, Adaptyv R2, and BindCraft1 revalidation into 605 unique candidate sequences from 615 labeled candidate records, with 68 positives after positive-if-any label aggregation.

B. Temporal Leakage Audit

Table 6 records the documented model cutoffs used for the temporal-leakage audit, and Table 7 records the public-release dates used for the dataset-level argument. Development entries are included for auditability because their labels are used in supervised fitting, global-policy construction, or iterative policy-search feedback; they are not held-out evaluation labels. For `gpt-4o-2024-11-20`, the documented cutoff is 2023-10-01, so the 10-target held-out split and the 3-target held-out subset both use labels released after the cutoff. The `gpt-5.4` model has a later 2025-08-31 cutoff; Nipah, RBX1, and TREM2 are the held-out targets in Table 3 whose source pools and experimental labels became public after this later date. ProteinBase competition pages sometimes retain stale stage-detail text after release, so we use the overview "Results released" date when available. BindCraft1 revalidation is public on ProteinBase; collection-level asset timestamps are consistent with 2025-10-01, but we use the 2025-10-06 ProteinBase launch as the conservative public web-availability date. BoltzGen's manuscript was posted on bioRxiv on 2025-11-24, but validated-label summary files were already present in the Hugging Face `boltzgen/adaptyv_data1` upload on 2025-10-27, with a stable re-upload on 2025-10-31. The BoltzGen labels are explicitly supplied only as development information rather than used as held-out outcomes.

Model / run	Documented cut-off	Use in paper
`gpt-4o-2024-11-20`	2023-10-01	Reported on the 10-target held-out split and the 3-target held-out subset
`gpt-5.4`	2025-08-31	Reported on the 3-target held-out subset: Nipah, RBX1, and TREM2

Table 6. Model cutoff dates used for the temporal-leakage audit.

Source / pool	Targets used	Public date	After `gpt-4o`?	After `gpt-5.4`?
BindCraft1 revalidation	spCas9, Der f 21, IFNAR2, Der f 7, EGFR entries	2025-10-06	yes	yes
Adaptyv EGFR R1	EGFR	2024-10-18	yes	no
pMHC minibinders	NY-ESO-1 pMHC, RVTDESILSY pMHC	2024-12-03	yes	no
Adaptyv EGFR R2	EGFR	2025-01-15	yes	no
BoltzGen validated-label dataset	development targets only	2025-10-27	yes	yes
GEM/Adaptyv RBX1	RBX1	2026-04-26	yes	yes
Nipah release	Nipah G	2026-01-21	yes	yes
BioArena/Adaptyv TREM2	TREM2	2026-03-28	yes	yes
Cao/Bennett historical	transfer baseline only	2022 to 2023	no	no

Table 7. Dataset-level temporal-leakage audit. Dates report conservative public availability of candidate-level experimental labels, or stable public data-package availability where explicitly noted. Development and transfer-baseline entries are shown for provenance but are not held-out evaluation labels. For the `gpt-5.4` comparison, Nipah, TREM2, and RBX1 are the held-out targets whose source pools and experimental labels became public after the documented cutoff.

C. Feature Glossary

The main policy panel contains 17 reproducible candidate-level features, listed with monotonic directions and extraction summaries in Table 8.

Inference settings. AF2-Multimer and AF2-monomer runs use ColabFold v1.5.5 with three models, three recycles, and no custom templates. The single-sequence complex wrapper uses single-sequence MSA mode. The target-MSA complex wrapper supplies a precomputed multimer

Feature	Direction	Extraction summary
AF2-Multimer ipTM	higher	Mean AF2-Multimer ipTM across aggregated target-MSA complex models.
AF2-Multimer pTM	higher	Mean AF2-Multimer pTM across aggregated target-MSA complex models.
AF2 binder pLDDT	higher	Mean binder-chain AF2-Multimer pLDDT across aggregated target-MSA complex models.
AF2 interface PAE	lower	Mean cross-chain AF2-Multimer PAE over target-binder interface residue pairs, averaged across models.
Boltz-2 ipTM	higher	Best Boltz-2 ipTM summary across target-MSA diffusion samples.
Boltz-2 pTM	higher	Best Boltz-2 pTM summary across target-MSA diffusion samples.
Boltz-2 binder pLDDT	higher	Best binder-chain pLDDT summary across Boltz-2 target-MSA diffusion samples.
Boltz-2 ipSAE	higher	Best conservative ipSAE-style interface-confidence summary across Boltz-2 target-MSA diffusion samples.
Boltz-2 pDockQ2	higher	Best pDockQ2 summary across Boltz-2 target-MSA diffusion samples.
Protenix pair ipTM	higher	Protenix/PXDesign pairwise binder-target confidence under the two-chain target-binder convention.
Protenix complex pTM	higher	Protenix/PXDesign complex-level pTM for the predicted target-binder complex.
Protenix binder ipTM	higher	Protenix/PXDesign binder-chain ipTM under the two-chain target-binder convention.
Protenix binder pTM	higher	Protenix/PXDesign binder-chain pTM under the two-chain target-binder convention.
Protenix binder pLDDT	higher	Protenix/PXDesign binder-chain pLDDT under the two-chain target-binder convention.
Rosetta interface ΔG	lower	Rosetta InterfaceAnalyzer interface ΔG on the Boltz-2 target-MSA predicted complex selected for Rosetta analysis.
Rosetta buried SASA	higher	Rosetta buried solvent-accessible surface area on the Boltz-2 target-MSA predicted complex selected for Rosetta analysis.
Rosetta shape complementarity	higher	Rosetta InterfaceAnalyzer shape complementarity (Lawrence & Colman, 1993) on the Boltz-2 target-MSA predicted complex selected for Rosetta analysis.

Table 8. Active features in the 17-feature policy panel. Directions indicate the monotonic orientation used by the deterministic executor. Target-MSA complex predictions use the target-chain MSA while designed binder chains remain single-sequence.

A3M for the target chain, while the binder chain remains single-sequence because no homologs exist for a de novo binder. Target-chain A3M files are generated once with the Protenix/PXDesign-compatible MMseqs2 MSA service and cached before feature extraction.

Boltz-2 runs use 5 diffusion samples, 3 recycling steps, 200 sampling steps, and the `boltz2` model, matching BoltzGen's de novo binder refolding configuration (Stark et al., 2025). In the single-sequence baseline, both chains are modeled without MSA information. In the target-MSA variant, the target chain receives the precomputed target MSA and the binder chain remains single-sequence.

Protenix/PXDesign features follow the public PXDesign Protenix inference setting with one sample, two diffusion steps, four recycles, templates disabled, and MSA enabled. The target chain receives the precomputed target MSA and the binder chain remains single-sequence. Complexes are represented as target-binder pairs, and binder-chain confidence fields are extracted according to the PXDesign two-chain convention. The active Protenix/PXDesign policy features are pairwise binder-target ipTM, complex pTM, binder-chain ipTM, binder-chain pTM, and binder-chain pLDDT.

The main 17-feature panel uses target-MSA Boltz-2, AF2-Multimer, Protenix, and Rosetta complex outputs. Single-sequence complex predictions are retained for ablation but are not part of the main panel. ESMFold (Lin et al., 2023), using the HuggingFace `facebook/esmfold_v1` weights, is deterministic and run once per binder. Rosetta `InterfaceAnalyzerMover` (Stranges & Kuhlman, 2013) with the `ref2015` scorefunction (Alford et al., 2017) is applied to Boltz-2 predicted complexes after coordinate-constrained `FastRelax` pre-relaxation for side-chain packing and local relaxation.

D. Cao/Bennett AF2 Transfer Baseline

The historical Cao/Bennett transfer baseline is trained on the AF2 confidence measurements that have compatible counterparts in the current held-out pools: binder-target interface PAE and binder-chain pLDDT. On the historical training pools, these are the AF2 interaction PAE and binder pLDDT scores reported with the Cao/Bennett retrospective scoring data. On the current held-out pools, we recompute the analogous quantities using our AF2-Multimer target-MSA protocol: mean cross-chain PAE over target-binder interface residue pairs and mean binder-chain pLDDT, each averaged across AF2-Multimer model runs. RMSD-based historical features are excluded because the current held-out candidates do not generally include the original designed-complex reference structures needed to compute the same designed-versus-predicted RMSD terms. Thus, this baseline tests transfer across compatible AF2 confidence feature types, not an identically matched AF2 scoring protocol.

E. Prompt Information Conditions

This appendix records the prompt information conditions used to audit label exposure and distinguish global from target-conditioned policies. All settings use the same feature panel, policy class, and deterministic executor described in Section 5.1; held-out labels are never included in any prompt. Table 9 contrasts the single-turn global and target-conditioned settings, and iterative variants add aggregate development-search feedback without changing the held-out-label restriction.

Prompt aspect	Global LLM	Target-conditioned LLM
Policy emitted	One policy shared by all held-out targets	One policy generated separately for each held-out target
Feature panel	Same 17 feature descriptions and fixed directions	Same 17 feature descriptions and fixed directions
Development information	Per-development-target score distributions and target-averaged single-feature performance	Same development calibration context as the global prompt
Held-out information	No held-out target identifiers, source context, score distributions, or labels	Current held-out target identifier and unlabeled score distributions for that target pool
Labels available to prompt	Development labels only through aggregate single-feature performance	Development labels only through aggregate single-feature performance; no held-out labels
Executor	Same deterministic executor: fixed directions, within-target normalization, weighted sum, top 10	Same deterministic executor: fixed directions, within-target normalization, weighted sum, top 10

Table 9. Information contrast between global and target-conditioned single-turn LLM policies. Iterative variants keep the same held-out-label restriction and executor, but add aggregate development-search feedback.

Prompt inputs and constraints. Rather than relying on free-form natural-language recommendations, each prompt asks the model to emit a structured ranking policy. The common prompt inputs are: (i) natural-language feature descriptions and fixed monotonic directions for the 17 active proxy scores, (ii) development-set single-feature performance summarized as target-averaged Recall@10, Hit@10, and NDCG@10, and (iii) per-development-target score-distribution summaries. Target-conditioned prompts additionally include the current held-out target identifier and unlabeled score-distribution summaries for that held-out target pool. Global prompts omit held-out target identifiers and held-out score distributions, and ask for one policy that is later applied unchanged to every held-out target.

All prompts enforce the same policy constraints. A valid policy selects 3 to 5 features, assigns each selected feature an integer weight in $\{1, 2, 3\}$, and returns only the structured policy object. The prompt explicitly disallows hard thresholds, feature-specific filters, treating any proxy as a direct affinity measurement, or overriding feature directions and aggregation. The deterministic executor, not the LLM, applies the fixed feature directions, normalizes selected features within the target pool, computes the weighted normalized sum, sorts candidates in descending order, and selects the top 10 designs.

Iterative feedback. Iterative settings use the same information restrictions as their single-turn counterparts. They add only aggregate development-search feedback from previously evaluated policies, including accepted or rejected feature-weight combinations and development-set summary metrics. This feedback is computed on the development split and does not include held-out labels.

Representative output schema. The saved runs used a JSON-like structured output with one list of selected terms:

```
{
  "score_terms": [
    {"feature": "<feature_name>", "weight": 1|2|3}
  ],
  "rationale": "<brief policy rationale>"
}
```

F. Per-target Performance

Table 10 reports per-target Recall@10 for the strongest single-feature fixed heuristic and the matched-prompt `gpt-4o` policy settings. Table 11 reports the corresponding `gpt-5.4` detailed metrics only on the 3-target held-out subset used for `gpt-5.4` evaluation.

Target	n	n_b	Fixed	Global	TC	Iter. global	Iter. TC
spCas9	20	14	0.700	0.800	0.900	0.860	0.840
Der f 21	16	4	0.500	0.250	0.500	0.550	0.600
Nipah G	1030	103	0.700	0.600	0.800	0.760	0.720
IFNAR2	20	8	0.500	0.500	0.500	0.725	0.725
NY-ESO-1 pMHC	41	1	1.000	1.000	0.200	1.000	1.000
Der f 7	20	5	0.800	0.200	0.600	0.760	0.800
EGFR	605	68	0.200	0.400	0.400	0.380	0.280
RVTDESILSY pMHC	96	2	0.500	0.000	0.400	0.100	0.100
RBX1	321	9	0.111	0.000	0.000	0.111	0.111
TREM2	100	37	0.700	0.600	0.680	0.640	0.660
mean			0.571	0.435	0.498	**0.589**	0.584

Table 10. Per-target Recall@10 on the 10-target held-out split for `gpt-4o`. Fixed is Protenix binder ipTM, the strongest single-feature fixed baseline in Table 3. Global and TC denote single-turn global and target-conditioned matched-prompt policies; Iter. global and Iter. TC denote the corresponding iterative policies with development-feedback memory. LLM entries are sampled-policy averages where applicable.

Metric	Target	Global	TC	Iter. global	Iter. TC
Recall@10	Nipah G	0.800	0.850	0.580	0.640
Recall@10	RBX1	0.000	0.056	0.111	0.156
Recall@10	TREM2	0.600	0.600	0.720	0.760
Recall@10	mean	0.467	0.502	0.470	**0.519**
Hit@10	Nipah G	0.800	0.850	0.580	0.640
Hit@10	RBX1	0.000	0.050	0.100	0.140
Hit@10	TREM2	0.600	0.600	0.720	0.760
Hit@10	mean	0.467	0.500	0.467	**0.513**
NDCG@10	Nipah G	0.849	0.869	0.620	0.734
NDCG@10	RBX1	0.000	0.037	0.141	0.222
NDCG@10	TREM2	0.637	0.635	0.777	0.792
NDCG@10	mean	0.495	0.514	0.513	**0.583**

Table 11. Per-target `gpt-5.4` performance on the 3-target held-out subset. Global and TC denote single-turn global and target-conditioned matched-prompt policies; Iter. global and Iter. TC denote the corresponding iterative policies with development-feedback memory. Values are reported only for Nipah, RBX1, and TREM2, the post-cutoff subset used for `gpt-5.4` evaluation.