MEMIG: SELF-EVOLVING MEMORY CONSTRUCTION FOR ANTIBODY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory has become central to self-evolving LLM agents, which store and retrieve experience from past rollouts to guide future decisions. Existing systems represent this experience as free-form natural language and maintain it by rewriting, which fits tasks whose experience is narratable. However, in scientific discovery and optimization, experience is often a quantity measured by an oracle under a specific context. Rewriting it as text leaves no index for where a record should be, replaces a measurement with a description, and cannot express how interventions interact. To address this, we argue that agent memory for biomolecule optimization should instead be indexable, quantitative, and structured, and we introduce MEMIG. This memory construction framework realizes these three properties. In detail, MEMIG addresses each insight with a domain-grounded identifier and accumulates the quantitative effect measured by a property oracle, so that updates remain local and retrieval becomes an exact lookup. Next, MEMIG connects insights whose interventions are applied together and measures their epistasis by decomposing each evaluated set. Our results show that MEMIG improves performance in most settings across in silico antibody optimization tasks.

1 INTRODUCTION

LLM agents can self-evolve by accumulating memories from previous rollouts, which guide future decisions (Shinn et al., 2023; Zhao et al., 2024; Suzgun et al., 2026; Zhang et al., 2026a). This offers a practical, economical alternative to fine-tuning, as the agent improves without expensive training while accumulated memory remains interpretable and editable. Nevertheless, the representation of this memory is free-form and unstructured, with experience from previous rollouts stored in natural language. This fits tasks whose experience is narratable, such as question answering and long-horizon conversation (Zhong et al., 2024; Packer et al., 2023; Chhikara et al., 2025), where experience can be summarized in a sentence without losing rich information.

However, directly applying such narrative memory systems to biomolecule optimization is not straightforward. For instance, in a computational oracle-guided protein optimization task (Notin et al., 2023; Stanton et al., 2022; Gruver et al., 2023), an agent proposes mutations to sequences, evaluates the candidate using oracles, and reuses measurements from the candidates to guide later proposals. Each evaluation is not merely a narratable experience, but a numerical measurement tied to a specific biological context, whose effect may depend on other co-applied mutations. Therefore, memory for biomolecule optimization should be *indexable* by biological context, *quantitative* in what it stores, and *structured* to capture interactions between mutations.

First, memory should be indexed by the context in which a measurement applies. Without such an index, there is no way to address a single entry, so incorporating a new experience requires an LLM to rewrite or summarize the entire memory. To address this, recent work (Zhang et al., 2026a; Xu et al., 2025) introduces identifiers for each memory entry so that updates can be applied locally, but these identifiers are arbitrary labels without information about the context, so retrieval still relies on textual similarity. In contrast, the effect of the same mutation can depend on its exact biological context (Notin et al., 2023; Olsen et al., 2024). Therefore, memory should address each entry by this context, so that updates and retrieval operate over the context rather than textual similarity.

Second, memory should preserve quantitative measurements rather than store them as free-form text. Existing memory methods store experience as natural language authored by an LLM (Zhao et al.,

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.

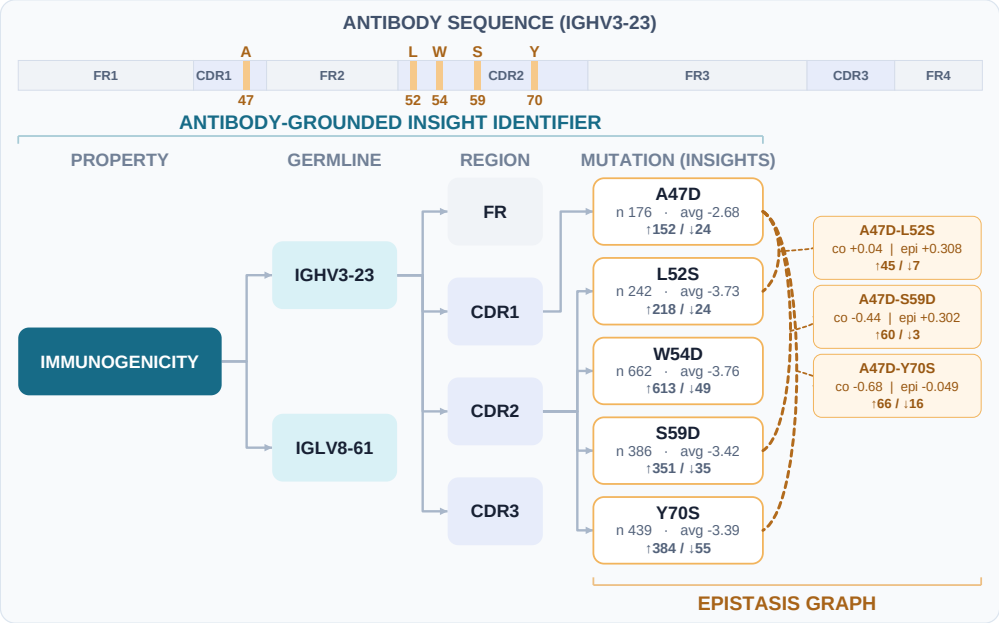

Figure 1: **Overview of MEMIG.** MEMIG constructs an epistasis graph with antibody-grounded identifiers. Directed edges organize property → germline → region → mutation hierarchies, while undirected edges connect co-occurring mutations and store their epistatic effects.

2024; Suzgun et al., 2026), where a reported number is written rather than computed and is therefore not tied to the measurement that produced it. This can discard the quantitative signal required for scientific optimization. Memory should instead retain oracle outputs as persistent quantitative records, while the LLM only converts the retrieved records into interpretable guidance for the agent.

Third, memory should capture the measured dependence between entities, rather than semantic similarity between the textual records. Graph-based memories organize records as linked notes rather than a flat list, but their edges represent semantic relatedness between texts (Chhikara et al., 2025; Xu et al., 2025). However, in biomolecule optimization, the effect of one mutation can depend on the presence of another, a phenomenon known as epistasis. Memory should therefore record these interactions explicitly rather than assume that mutation effects are additive.

To address this, we introduce MEMIG (**Mem**ory as **I**ndexable **G**raphs), a domain-grounded memory construction framework for protein optimization agents, as described in Figure 1. We focus on the domain of antibody optimization and design. To make memory indexable, we define an antibody-grounded identifier that specifies the target property, the germline gene, the antibody region, and the AHo-numbered (Honegger & PluÈckthun, 2001) mutation, so that each record is addressed by the biological context in which it applies. To make memory quantitative, each insight stores the effect computed by the oracle.

Moreover, to build structured memory, MEMIG creates a graph of co-occurring insights. Here, each edge in the graph connects mutations showing epistasis, quantified as the deviation of the joint effect from the sum of the individual effects. Next, before each rollout, MEMIG retrieves insights from the graph based on the context identifier, ranks them by their measured effects, and provides the selected insights with their graph neighborhoods to the agent. Then an LLM renders them as interpretable guidance while the quantitative memory remains. We empirically showed that incorporating MEMIG into frontier LLMs (GLM-5-Team et al., 2026; Anthropic, 2025) improves performance in the majority of settings in in silico antibody optimization tasks.

Our contributions are summarized as follows:

- We introduce MEMIG, an *indexable*, *quantitative*, and *structured* memory system which addresses each insight with an antibody-grounded identifier and stores the effect computed by a property oracle, so that updates remain local and retrieval becomes an exact lookup.

- We construct a graph-based memory over insights, where an edge stores the epistatic effect of mutations, so that memory precisely records how mutations combine.
- We demonstrate that MemIG outperforms existing memory systems on antibody optimization.

2 METHOD: MEMIG

In this section, we present MEMIG (**Mem**ory as **I**ndexable **G**raphs), an indexable, quantitative, and structured memory system for antibody optimization agents. In this work, we consider the following formulation of antibody optimization: given an antibody sequence, the model proposes a set of mutations to improve a defined set of properties under a maximum edit distance, as described in Figure 2.

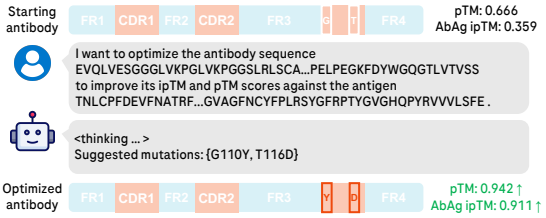

Figure 2: **Antibody optimization task.**

Throughout this section, an *observation* is a single measured effect of applying mutations, and an *insight* is the summary of the observations sharing the same identifier. *Memory* is then the graph whose nodes are these insights and whose edges store how mutations interact when applied together, as illustrated in Figure 1. Memory is self-evolving as the agent writes observations from its own evaluations, constructs memory from the observations, and retrieves the relevant insights.

Each component of MEMIG realizes three properties of memory: indexability, quantitativeness, and structured dependencies. Section 2.1 defines the indexable identifiers and quantitative records, Section 2.2 describes the epistasis graph, and Section 2.3 explains retrieval based on the constructed epistasis memory graph.

2.1 ANTIBODY-GROUNDED INSIGHT IDENTIFIER

Indexable identifier. The identifier makes memory indexable by addressing each insight with the antibody context where it was observed. In detail, an identifier is composed of a property, germline, region and mutation, where the region is a contiguous block of residues that define a framework or complementarity-determining region (e.g. FR1 or CDR2) and the mutation position comes from AHo alignment (Honegger & PluÈckthun, 2001), such as `immunogenicity/IGHV3-23/[cdr1]/H:A47D`. The AHo scheme enables a single insight to accumulate evidence across antibodies of different lengths. We use exact matching so that observations are aggregated only when their biological context and substitution match. Consequently, the same mutation observed under a different context forms a different insight. This design is inspired by Zhang et al. (2026a), which introduces identifiers to enable local memory updates.

Addressing insights in this way keeps the memory update local and efficient. Each new observation updates only the insight sharing its identifier, leaving the rest of memory untouched. Therefore, the cost scales with the number of identifiers affected by the new observation rather than the total memory size, whereas rewriting the entire memory grows more expensive as experience accumulates.

Quantitative record. Each insight records the quantitative effect of its mutation, so that memory reports exact quantities rather than free-form impressions. In detail, an insight accumulates the mean effect across turns on each property computed by the oracle, the number of observations, and the directional votes over every observation sharing the identifier. The directional vote records whether the mutation moved the property toward the objective, so that the insight reports not only the average magnitude of an effect but also how consistently that effect occurs. For example, the insight

```
immunogenicity/IGHV3-23/H:A47D [cdr1]:  avg -2.68, n=176, ↑152/↓24
```

records the substitution of alanine (A) with aspartate (D) at heavy-chain position 47 in the CDR1 of a sequence from the IGHV3-23 gene family. The mutation decreased (improved) immunogenicity by 2.68 on average over 176 observations, and did so in all but 24 of them.

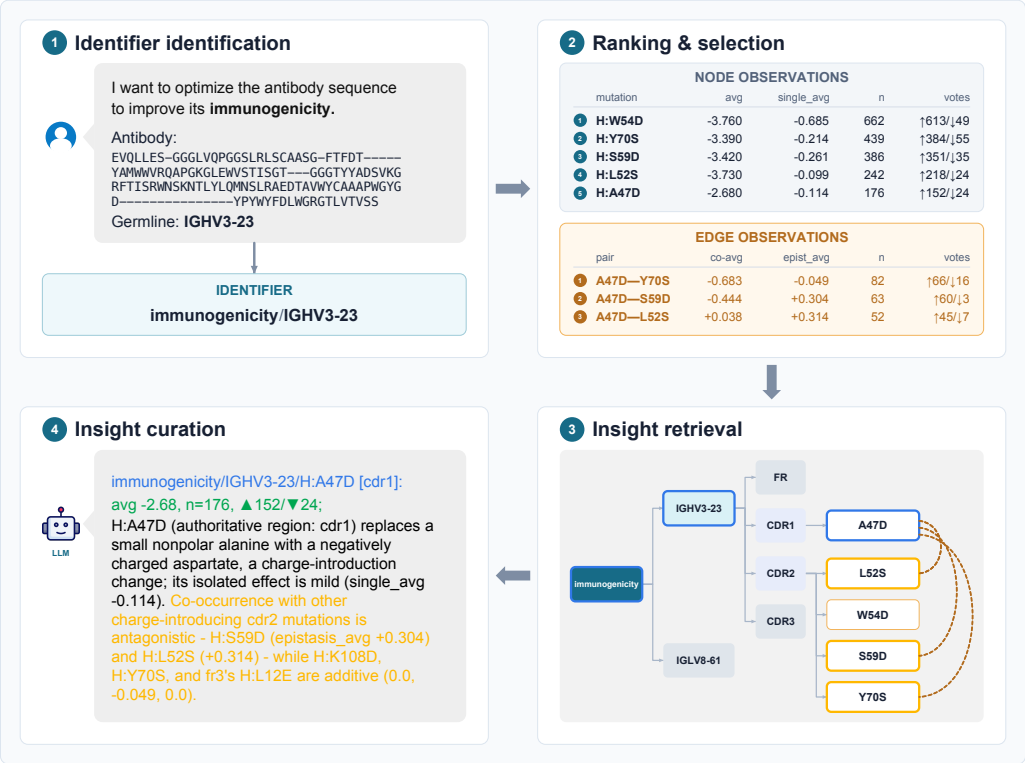

Figure 3: **Memory retrieval of MEMIG.** Given a starting antibody, MEMIG identifies the matching identifiers, retrieves the corresponding insights with their neighbors, and curates them into natural language. Blue denotes the antibody-grounded identifier and the corresponding node, green its quantitative record, and yellow the epistasis graph neighbors and the curation based on them.

2.2 EPISTASIS GRAPH

The epistasis graph makes memory structured by connecting insights whose mutations interact. The joint effects of mutations may be non-additive to other co-occurring mutations, a phenomenon known as epistasis, so a set of independent insights cannot represent it. To capture this in the epistasis graph, each node is a single mutation, and each edge accumulates the joint effect of a co-occurring pair. These edges connect mutations within a chain and across the heavy and light chains.

We measure the epistatic effect of a pair by decomposing each mutation set the agent proposes. For each pair of mutations a and b in the set, we additionally apply a alone, b alone, and a and b together to the starting sequence, and measure the resulting property changes Δ_a, Δ_b, and Δ_{ab}. Each node stores the mean isolated effect when the mutation is applied alone as `single_avg`, while `avg` stores the mean effect of the complete evaluated mutation sets containing that mutation to reflect the effects of co-applied mutations.

The epistatic effect is then the deviation of the pair from additivity, $\varepsilon_{ab} = \Delta_{ab} - \Delta_a - \Delta_b$. When $\varepsilon_{ab} = 0$, the joint effect equals the sum of the individual effects, indicating additivity and no epistasis. A nonzero ε_{ab} indicates an epistatic interaction: positive values mean that the joint property change exceeds the sum of the individual changes, whereas negative values mean that it falls below this sum. The edge between a and b stores the mean of Δ_{ab} as `co_avg` and the mean of ε_{ab} as `epistasis_avg` over every observation of that pair. Notably, this decomposition requires additional oracle calls, which is possible because property oracles are cheap and re-runnable on any sequence. An example of the stored nodes and edges with these values is shown in Figure 3.

2.3 MEMORY RETRIEVAL

Based on the constructed graph, memory is retrieved before each rollout and injected into the prompt. The retrieval process consists of four steps: identifier identification, ranking and selec-

tion, insight retrieval, and insight curation, as described in Figure 3. In detail, we collect the insights whose identifiers match the germline and the target properties of the current sequence, and whose mutations match its wild-type residues at the corresponding AHo positions. We rank them by their net directional votes breaking ties by the magnitude of the `avg`, and select up to K insights. In addition to these exact-match insights, the memory also includes global insights that aggregate mutation trends across germlines to capture broader mutation trends, as detailed in Section B.

The retrieved insights are curated into natural language so that the agent receives interpretable guidance. Specifically, we curate each insight with an LLM, which also receives up to N 1-hop neighbor insights and their edges from the epistasis graph, so that the text describes both the isolated and joint effects. Curation produces the guidance presented to the agent without modifying the stored quantitative statistics. For example, an insight `immunogenicity/IGHV3-23/H:A47D [cdr1]` is curated as:

```
immunogenicity/IGHV3-23/H:A47D [cdr1]: avg -2.68, n=176,
↑152/↓24; H:A47D (authoritative region: cdr1) replaces a small
nonpolar alanine with a negatively charged aspartate, a
charge-introduction change; its isolated effect is mild
(single_avg -0.114), so the -2.68 average (n=176) is mostly
confounded by companion mutations. Co-occurrence with other
charge-introducing cdr2 mutations is antagonistic | H:S59D
(epistasis_avg +0.304) and H:L52S (+0.314) | while H:K108D,
H:Y70S, and fr3's H:L12E are additive (0.0, -0.049, 0.0).
```

After curation, the curated insights are concatenated to form the memory prompt. Memory is initialized as empty and the memory of the sequence is written only after all of its rollouts have been scored. Therefore, the rollouts of a sequence share the same memory snapshot, and a sequence retrieves only the insights accumulated from the sequences optimized before it. We provide an example of the constructed memory in Section A.

3 Experiments

In this section, we evaluate MemIG's effectiveness on two antibody optimization tasks. We employ Haiku-4.5 (Anthropic, 2025) and GLM-5.3-Flash (GLM-5-Team et al., 2026) as our backbone models. We provide the detailed experimental settings in Section B.

Metrics. We employ two metrics throughout the experiments: pass@k (Chen et al., 2021) and improvement (Imp.). Here, we consider two types of objectives: (1) minimization and maximization, which move the property as far as possible in one direction, and (2) range, which moves the property into a target interval. Pass@k estimates the probability that at least one of k sampled rollouts passes, computed as $1 - \binom{n-c}{k} / \binom{n}{k}$, where n rollouts are sampled and c pass. We report $k \in \{1, 3\}$. A rollout passes when the optimized sequence improves upon the starting sequence for minimization and maximization tasks, or lies within the target interval for range tasks.

Next, improvement measures the magnitude of the gain. Specifically, it is computed as the signed difference between the optimized and the starting property values in the favorable direction for minimization and maximization tasks, and as the reduction in distance to the target interval for range tasks. For example, in the range task, when the starting sequence property is -0.6, and the optimized sequence property is -0.4, the distance to the target interval $[-0.5, -0.2]$ decreases from 0.1 to 0.0, giving an improvement of 0.1. Notably, both metrics are computed at the best turn of each rollout determined by the reward and are then averaged over all rollouts.

Baselines. We utilize six baselines, including two statistical and four LLM-based approaches. The statistical baselines include random and MCMC (Metropolis et al., 1953). The random baseline samples k mutations uniformly from $\{1, \ldots, B\}$ mutations, where B is the edit distance budget, and applies the mutations to the starting variable fragment (Fv). The MCMC baseline runs Metropolis-Hastings from the starting Fv. At each of T steps, matched to the turn budget of the agent, it proposes three random mutations that remain within the edit distance budget, scores them with the

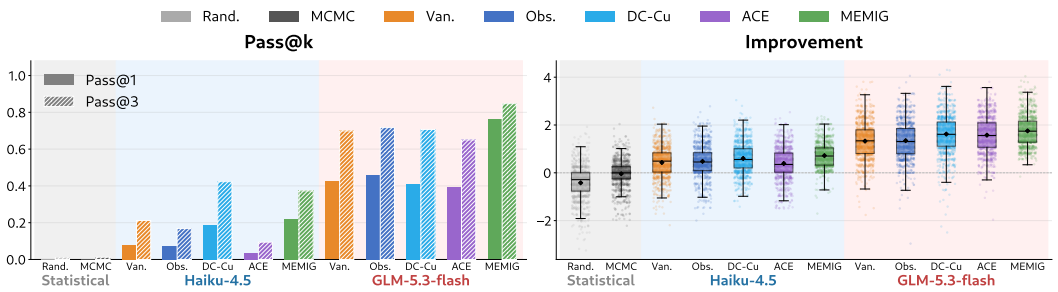

Figure 4: **Multi property optimization results.** Pass@1 and pass@3 (left) and the distribution of improvement over the starting sequence (right), where each colored dot is one rollout and the black diamond marks the mean.

same reward r used to evaluate the agent, and accepts the mutations under the Metropolis criterion $\min(1, \exp(\Delta r/\tau))$ with the temperature fixed at $\tau = 1.0$.

The LLM-based baselines include a vanilla LLM agent and three LLM agents with different memory frameworks, including observations-only, DC-Cu (Suzgun et al., 2026), and ACE (Zhang et al., 2026a). The vanilla LLM agent applies the agentic system without memory. Next, the LLM agent with observations receives the same oracle outputs that MEMIG uses to construct its memory, but in their raw form rather than aggregated under identifiers or curated. Lastly, DC-Cu and ACE are self-evolving memory baselines that accumulate experience across rollouts, where DC-Cu regenerates a compact cheatsheet after each query while ACE maintains an indexed playbook updated through incremental deltas. For fair comparison, we provide all LLM-based baselines with the same tool access, prompt, maximum input and output tokens, and turn budget, which isolates the contribution of the memory.

3.1 MULTIPLE PROPERTY OPTIMIZATION

Task description. The goal of multiple property optimization is to satisfy multiple property constraints at once by mutating the variable fragment (Fv) of a given antibody sequence. Specifically, we select four antibody properties: immunogenicity (Thrift et al., 2024), liability count (Cock et al., 2009), GRAVY (Kyte & Doolittle, 1982), and charge. Here, immunogenicity quantifies the risk that the antibody triggers an unwanted immune response and liability count measures chemically unstable sequence motifs (e.g., oxidation and deamidation sites). Next, GRAVY measures the average hydrophobicity over the residues and charge measures the net charge of the sequence at pH 5.5. Immunogenicity and liability count are minimization, and GRAVY and charge are range tasks, targeting $[-0.5, -0.2]$ and $[0.0, 8.0]$, respectively. We provide the details in Section B.

We randomly select 100 antibody sequences from the OAS dataset (Olsen et al., 2022) and generate 10 rollouts per sequence. We filter out the sequences that already satisfy the target constraint, so the pass@k of the starting sequences is zero by construction.

Results. We provide the results in Figure 4 and full values in Section C. We observe that MEMIG consistently improves multiple property optimization across both backbones. Specifically, it achieves the highest average improvement on Haiku-4.5 (0.718 vs. 0.602 for the strongest baseline) and GLM-5.3-Flash (1.754 vs. 1.621), while also attaining the best pass@1 on both backbones. The box plots show a broad upward shift in the improvement, with medians close to the corresponding means, indicating that the gains are systematic rather than driven by a few favorable rollouts. Across all baseline comparisons, these gains are significant under one-sided Mann–Whitney U tests (Mann & Whitney, 1947) with Holm–Bonferroni correction ($p < 0.005$).

3.2 IN SILICO ANTIBODY BINDING OPTIMIZATION

Task description. The goal of this task is to optimize the binding of an antibody to a given antigen by mutating CDR-H3. Specifically, we optimize two confidence scores of a structure prediction model, the predicted TM score (pTM) for the antibody structure and the interface predicted TM score (AbAg-ipTM) for the antibody-antigen interaction, where we target > 0.8 and > 0.7 respectively.

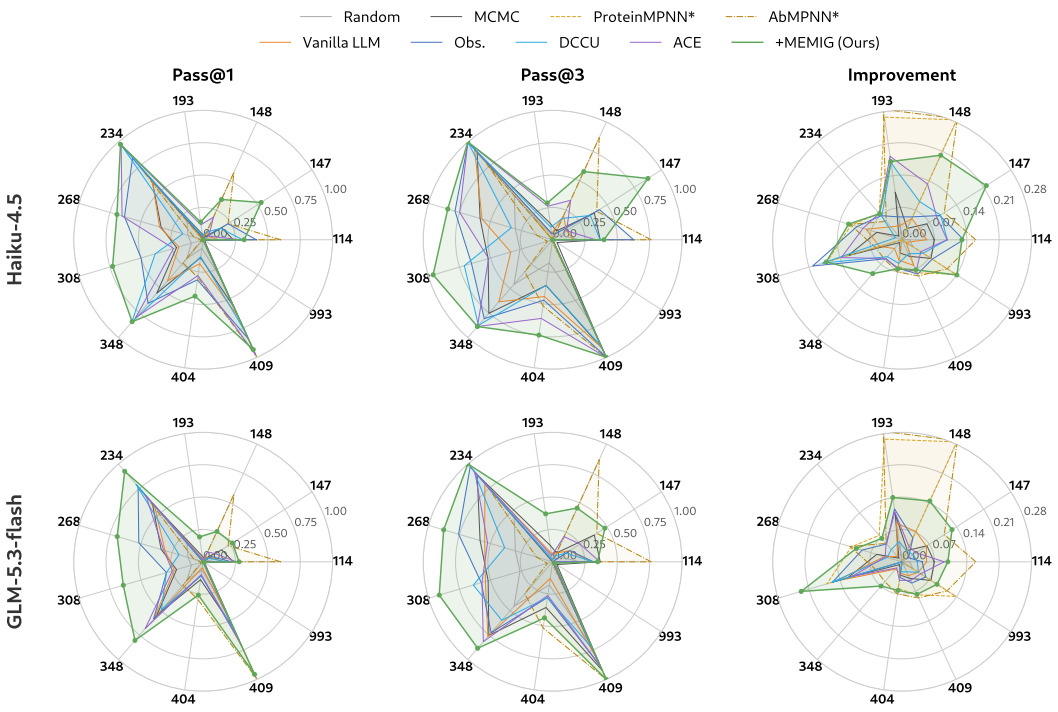

Figure 5: **In silico antibody binding optimization.** Each numeric label corresponds to an antibody-antigen cluster, clustered by CDR sequence identity. * denotes the structural baselines (dashed), which have access to the complex structure.

Such confidence scores discriminate binders from non-binders and therefore serve as useful proxies for in silico binding optimization (Abramson et al., 2024; Lin et al., 2022; Zhang et al., 2026b). We employ Protenix-v2 (Zhang et al., 2026b) for its predictive power and open availability.

We select antibody-antigen complexes from the SAbDab (Dunbar et al., 2014). We retain only complexes that are (i) of type fragment antigen-binding (Fab), (ii) bound to a single antigen chain, and (iii) whose CDR-H3 loop makes substantial contact with the antigen interface (H3 interface fraction > 0.3). These selected complexes are grouped into 11 clusters, where each cluster is treated as one optimization task, generating 30 rollouts per complex. We provide the details includng the number of complexes and their PDB IDs in each cluster in Section B.

Baselines. We add two structural baselines for in silico antibody optimization task, including ProteinMPNN (Dauparas et al., 2022) and AbMPNN (Dreyer et al., 2023). These baselines inverse-fold the native complex with every position fixed except CDR-H3. It is notable that they receive the backbone coordinates of the antibody-antigen complex, whereas MEMIG observes the sequence alone. We sample the same number of designs as the agent's turn budget and evaluate all designs with the same oracle. We treat these methods as structure-informed references because they receive complex backbone coordinates, whereas MEMIG receives only sequences. They could also be incorporated into the agent as design tools.

Results. We provide the results in Figure 5 and the detailed values in Section C. Among the non-structural methods, we observe that MEMIG attains the best pass@1 in 9 of the 11 clusters with GLM-5.3-Flash and in 7 of the 11 clusters with Haiku-4.5, and the highest improvement in every cluster with GLM- 5.3-Flash. Compared to all LLM-based and statistical baselines within each backbone, these gains are statistically significant under one-sided Mann–Whitney U tests (Mann & Whitney, 1947) with Holm–Bonferroni correction ($p < 0.005$). The structural baselines included as reference anchors lead on improvement, since improvement is averaged over all designs regardless of whether they pass the constraint. Notably, ProteinMPNN attains a pass rate of zero in 8 of the 11 clusters while leading on improvement in three of them, which shows that a large improvement does not imply that the designs reach the target.

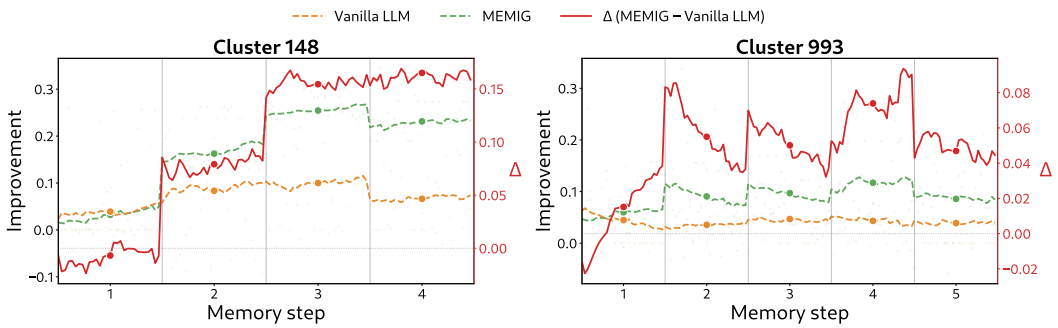

Figure 6: **Cumulative performance progression.** Vertical lines mark memory updates, where the rollouts between two lines are generated concurrently under the same memory. The curves report the centered rolling mean with a half-width of 30 rollouts, computed within each memory step. Notably, the ordering of rollouts within the same memory is random, so a downward slope does not imply performance decrease. Each dot marks the mean over all rollouts of that memory step.

3.3 Analysis

We conduct the following analyses on the in silico antibody binding optimization task with GLM-5.3-Flash on three randomly picked clusters (148, 193, 993). We provide additional analyses including same oracle call analysis and details in Section C.

Ablation study on component. We ablate the components of MemIG. In detail, the indexable identifier drops the germline+region field which keeps property and mutation only. Next, the epistasis graph removes the edges, so that memory stores the information of each mutation without access to the joint effects. As shown in Table 1, we observe that every ablation degrades the pass rate, which shows that all components contribute to satisfying the constraints. In contrast, the ablated epistasis attain a higher improvement, since improvement is averaged over all designs regardless of whether they satisfy the constraint.

Table 1: **Ablation study on component**.

Component	Pass@1	Pass@3	Imp.
MemIG (Ours)	0.149	0.278	0.125
Identifier	0.074	0.153	0.117
Epistasis	0.049	0.130	0.142
Vanilla LLM	0.015	0.047	0.067

Memory faithfulness. We next verify that the agent actually uses the retrieved insights stored in memory by MemIG. In detail, we take the insights that MemIG retrieves for each starting antibody as a fixed reference set, and compare MEMIG's proposals with this reference set. The adoption measures the fraction of proposed mutations that appear anywhere in the reference set, while helpful and harmful adoption further split this fraction by whether the recorded effect of the matched insight points toward or against the objective based on their directional votes. Since a mutation has a separate insight for each property, it can be helpful for one and harmful for another, so the two fractions sum to more than the total adoption. Additionally, we report the net directional alignment, a gap between helpful and harmful adoption (Δ), as the agent must not only read the memory but also distinguish its direction.

We provide the results in Table 2. We observe that 93.1% of the proposals by MemIG were present in the memory. These mutations were mostly from helpful insights, not from harmful insights, which shows that the agent reads and understands the recorded direction of an insight.

Table 2: **Memory faithfulness.** MemIG raises both adoption and net directional alignment.

Adoption	Helpful	Harmful	Δ
0.931	0.909	0.004	0.905

Cumulative performance progression. We analyze how performance changes as memory self-evolves in Figure 6 by tracking the improvement of the vanilla LLM and MemIG at each memory step, where the rollouts sharing a memory step are generated concurrently. We report clusters 148 and 993, whose germline remains constant across the starting complexes, so that the insights accu-

mulated from earlier rollouts apply directly to later ones. We observe that MEMIG improves over memory versions while the vanilla LLM stays near flat, so the gain comes from the accumulated memory rather than from the agent itself. Notably, the gap widens after the first memory update and remains positive, which shows that MEMIG improves through self-evolution. We additionally report the same analysis without the rolling average in Section C.

4 RELATED WORK

Self-evolving agent with memory. Self-evolving agents have recently gained substantial attention, where agents accumulate experience from past rollouts and store it into external memory without updating model weights (Shinn et al., 2023). ExpeL (Zhao et al., 2024) abstracts natural language insights from successful and failed experience, and maintains them through LLM-judged upvotes and downvotes. Dynamic Cheatsheet (Suzgun et al., 2026) generates the solution based on the current cheat sheet, curates the insight based on the generation, and updates the cheat sheet, while ACE (Zhang et al., 2026a) instead maintains an itemized playbook with helpful and harmful counters, updated with incremental deltas.

Graph-based memories keep their entries as linked notes rather than a flat list, where nodes and edges represent semantic relations between textual memories (Chhikara et al., 2025; Xu et al., 2025). Nevertheless, these approaches store experience as natural language written and judged by an LLM, address entries by opaque labels that do not encode applicability context, and connect entries by textual relatedness. To address these challenges, MEMIG stores oracle-measured quantities under the biological context-derived identifiers and connects them through oracle-measured interactions.

LLM agents for protein optimization. Protein sequence optimization has been studied as black-box optimization over a discrete space, where a model is trained on previously evaluated sequences and is used to propose the next candidates (Stanton et al., 2022; Gruver et al., 2023; Notin et al., 2023). Recently, LLM agents have entered this pipeline by orchestrating property predictors as tools and reasoning over the returned scores to select or revise candidates, with some accumulating design strategies across rollouts as natural language (Roohani et al., 2025; Tan et al., 2026; Liu et al., 2026). However, these approaches only accumulate narrative feedback without any quantitative record, which is not indexed and not structured in terms of biological context. To address this, MEMIG keeps oracle-measured effects under antibody-grounded insight identifiers and connects them through measured pairwise interactions, which keeps accumulated memory indexable, quantitative, and structured across rollouts.

5 CONCLUSION

In this paper, we introduce MEMIG, a memory construction framework for antibody optimization agents that keeps accumulated observations indexable, quantitative, and structured rather than narrated. Each insight is addressed by an antibody-grounded identifier over the property, germline, region, and mutation, and stores the effect measured by a property oracle. Next, edges between insights record the epistatic effect based on how mutations behave when applied together. Our findings suggest that the unit of memory should be defined by the domain it serves, and that grounding memory in measured quantities rather than generated text allows experience to accumulate without summarization. Although we instantiate MEMIG for antibody optimization, the framework can extend to other oracle-guided optimization problems when the domain provides a suitable identifier.

Limitation and challenge. First, our evaluation is conducted in silico using computational property oracles. This bounds the cost model of the epistasis decomposition, which assumes oracles that can be re-run freely. Although these oracles provide a controlled testbed, improvements may not directly translate to experimentally measured properties. Second, MEMIG currently represents mutation interactions through pairwise epistasis edges. Although each edge is estimated by evaluating the individual mutations and their pair, multi-mutation designs may exhibit higher-order interactions. Next, the curation step of MEMIG relies on the capability of the backbone model, as in prior work (Zhang et al., 2026a; Suzgun et al., 2026). If the model fails to describe a retrieved insight faithfully, the curated guidance may be misleading while the quantitative record mitigates this.

ETHICS STATEMENT

This work studies memory construction for antibody optimization agents, and the experiments are conducted in silico with computational property oracles. We use only publicly released data, including antibody sequences from OAS (Olsen et al., 2022) and antibody-antigen complexes from SAbDab (Dunbar et al., 2014), under their respective licenses; the OAS sequences derive from published immune repertoire studies and are not linked to identifiable individuals, and no human subjects are involved in our experiments. MEMIG optimizes developability and predicted interface confidence rather than any property related to toxicity or pathogenicity, and any resulting candidate would require experimental validation and human review before use.

REPRODUCIBILITY STATEMENT

We provide the information required to reproduce our experiments in the main text and appendix. Section 3 describes the evaluation setup and baselines, while Section B provides the property oracles, datasets, prompts, and hyperparameters. Complete numerical results and additional analyses are reported in Section C.

AI DISCLOSURE

We used generative AI tools to polish writing of the manuscript, create figures, and assist with analysis and plotting code. The research hypothesis, the method design, and all experiments were determined by the authors. The remaining tasks requiring disclosure, such as generating synthetic datasets and writing proofs, are not applicable.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Anthropic. Claude haiku 4.5, 2025. URL https://www.anthropic.com/claude/haiku. Large language model.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009. doi: 10.1093/bioinformatics/btp163. URL https://doi.org/10.1093/bioinformatics/btp163.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

Frédéric A Dreyer, Daniel Cutting, Constantin Schneider, Henry Kenlay, and Charlotte M Deane. Inverse folding for antibody sequence design using deep learning. *Workshop on Computational Biology at International Conference on Machine Learning*, 2023.

James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M. Deane. Sabdab: the structural antibody database. *Nucleic Acids Research*, 42(D1):D1140–D1146, 01 2014. ISSN 0305-1048. doi: 10.1093/nar/gkt1043. URL https://doi.org/10.1093/nar/gkt1043.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Nate Gruver, Samuel Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0547. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/29591f355702c3f4436991335784b503-Abstract.html.

Annemarie Honegger and Andreas PluÈckthun. Yet another numbering scheme for immunoglobulin variable domains: an automatic modeling and analysis tool. *Journal of molecular biology*, 309 (3):657–670, 2001.

J Kyte and RF Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of molecular biology*, 157(1):105—132, May 1982. ISSN 0022-2836. doi: 10.1016/0022-2836(82)90515-0. URL https://doi.org/10.1016/0022-2836(82)90515-0.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

Xingyue Liu, Zijie Xing, Runze Wang, Luoming Hu, and Yanming Shen. Language-guided expertise evolution for protein optimization. In *ICLR 2026 Workshop on AI with Recursive Self-Improvement*, 2026. URL https://openreview.net/forum?id=NeNWeyfUJj.

Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pp. 50–60, 1947.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *jcp*, 21(6):1087–1092, June 1953. doi: 10.1063/1.1699114.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. Proteingym: Large-scale benchmarks for protein fitness prediction and design. *Advances in neural information processing systems*, 36: 64331–64379, 2023.

Tobias H. Olsen, Fergus Boyles, and Charlotte M. Deane. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1):141–146, 2022. doi: https://doi.org/10.1002/pro.4205. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.4205.

Tobias H. Olsen, Iain H. Moal, and Charlotte M. Deane. Addressing the antibody germline bias and its effect on language models for improved antibody design. *Bioinformatics*, 40(11): btae618, 2024. doi: 10.1093/bioinformatics/btae618. URL https://doi.org/10.1093/bioinformatics/btae618.

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.

Yusuf Roohani, Andrew Lee, Qian Huang, Jian Vora, Zachary Steinhart, Kexin Huang, Alexander Marson, Percy Liang, and Jure Leskovec. BioDiscoveryAgent: An AI agent for designing genetic perturbation experiments. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/4252dc945318330290000f85dc5fac792-Abstract-Conference.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=vAElhFcKW6.

Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba

Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20459–20478. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/stanton22a.html.

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.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7080–7106. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.333. URL https://aclanthology.org/2026.eacl-long.333/.

Yang Tan, Yuanxi Yu, Can Wu, Bozitao Zhong, Mingchen Li, Guisheng Fan, Jiankang Zhu, Yafeng Liang, Nanqing Dong, and Liang Hong. Rank-and-reason: Multi-agent collaboration accelerates zero-shot protein mutation prediction, 2026. URL https://arxiv.org/abs/2602.00197.

William John Thrift, Quade Broadwell, Jason Perera, Nicolas W. Lounsbury, Jieming Chen, and Suchit Jhunjhunwala. Pep2vec: An interpretable model for peptide-mhc presentation prediction and contaminant identification in ligandome datasets. *bioRxiv*, 2024. doi: 10.1101/2024.10.14.618255. URL https://www.biorxiv.org/content/10.1101/2024.10.14.618255v1.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=FiM0M8gcct.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Y. Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8a94ff6f922d995d7d3f4ebf4143e442-Abstract-Conference.html.

Yuxuan Zhang, Chengyue Gong, Jinyuan Sun, Jiaqi Guan, Milong Ren, Song Xue, Hanyu Zhang, Wenzhi Ma, Zhenyu Liu, Xinshi Chen, and Wenzhi Xiao. Protenix-v2: Broadening the reach of structure prediction and biomolecular design. *bioRxiv*, 2026b. doi: 10.64898/2026.04.10.717613. URL https://www.biorxiv.org/content/early/2026/04/11/2026.04.10.717613.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

A EXAMPLES

A.1 EXAMPLE REASONING TRACE

Reasoning trace

Turn 1 | **Tool result (baseline)**


```
{
  "gravy": -0.1851, "charge_ph5_5": -0.154,
  "sequence_liabilities": 7, "immunogenicity": 5.0,
  "edit_distance": 0, "sequence_length": 228
}
```


Turn 2 | **Assistant**

Baseline: GRAVY -0.1851 (need ≤ -0.2), charge -0.154 (need ≥ 0), liabilities 7, immunogenicity 5.0, length 228 ✓. Need to lower gravy slightly and raise charge at pH 5.5. Prioritizing charge-raising mutations with big effects and gravy-lowering, drawing on **MEMORY** insights for IGHV3-23: H:V13K and H:D83K give the largest charge gains ($+5.1$, $+5.26$), while H:M41T, H:F78S, and H:M93L are top gravy-reducers with fully additive co-occurrence (epistasis_avg $= 0.0$ across all tested partners).

```
tool_call: evaluate_mutations
{
  "heavy_mutations": "{V13K, L12K, M41T, F78S, M93L, D83K, S7K,
     D100N, T88A}",
  "light_mutations": "{}"
}
```

Turn 2 | **Tool result**


```
{
  "gravy": -0.2776, "charge_ph5_5": 5.778,
  "sequence_liabilities": 5, "immunogenicity": 7.0,
  "edit_distance": 9
}
```


Turn 3 | **Assistant**

Gravy and charge are now in range, and liabilities dropped from 7 to 5, but immunogenicity rose from 5.0 to 7.0 — worse than baseline. This contradicts the **global insight** that "to_K" mutations are broadly disfavored for immunogenicity (-2.66 avg globally); apparently the four lysine substitutions (V13K, L12K, D83K, S7K) compound this penalty in combination even though their individual single-mutation deltas measured 0.0. Trimming to fewer K mutations and testing a leaner, more additive panel.

```
tool_call: evaluate_mutations
{
  "heavy_mutations": "{M41T, F78S, D83K, D100N}",
  "light_mutations": "{}"
}
```

Turn 3 | **Tool result**


```
{
  "gravy": -0.214, "charge_ph5_5": 2.778,
  "sequence_liabilities": 6, "immunogenicity": 4.0,
  "single_mutations": [
    {"token": "D100N", "immunogenicity_delta": -1.0},
    {"token": "M41T", "sequence_liabilities_delta": -1}
  ]
}
```


Turn 4 | **Assistant**

This 4-edit set improves every objective: gravy -0.214 ✓, charge 2.778 ✓, liabilities 6 (from 7), immunogenicity 4.0 (from 5.0). Per **MEMORY**, D100N is the only mutation

with a measured −1.0 direct immunogenicity effect, and it composes additively with D83K (epistasis_avg ≈ 0). Testing whether a light-chain liability fix can push further without disturbing this balance — the light FR1 has a methionine (M4) analogous to the heavy H:M41 oxidation-liability site the memory repeatedly flags.

```
tool_call: evaluate_mutations
{
  "heavy_mutations": "{M41T, F78S, D83K, D100N}",
  "light_mutations": "{M4L}"
}
```

Turn 4 | Tool result


```
{
  "gravy": -0.2057, "charge_ph5_5": 2.778,
  "sequence_liabilities": 5, "immunogenicity": 4.0,
  "single_mutations": [
    {"token": "M4L", "chain": "L", "sequence_liabilities_delta": -1
        }
  ]
}
```


Turn 5 | Assistant

The methionine-removal analogy paid off: L:M4L drops liabilities to 5 with zero cost to gravy, charge, or immunogenicity, all pairwise epistases measuring ≈0. Final design: heavy {M41T, F78S, D83K, D100N}, light {M4L} — all four objectives improved or in range, 5 of 10 edits used.

A.2 Observations

Observations


```
{
  "applied_mutations": [
    {"bullet_id": "H:M41L", "chain": "H", "germline": "IGHV3-21*08"
        ,
     "group_id": "H:41L", "pos": 41, "region": "cdr1", "target": "L
         ",
     "target_group": "nonpolar", "token": "M41L", "wt": "M", "
         wt_group": "nonpolar"},
    {"bullet_id": "H:N75T", "chain": "H", "germline": "IGHV3-21*08"
        ,
     "group_id": "H:75T", "pos": 75, "region": "cdr2", "target": "T
         ",
     "target_group": "polar", "token": "N75T", "wt": "N", "wt_group
         ": "polar"},
    {"bullet_id": "H:E92K", "chain": "H", "germline": "IGHV3-21*08"
        ,
     "group_id": "H:92K", "pos": 92, "region": "fr3", "target": "K"
         ,
     "target_group": "positive", "token": "E92K", "wt": "E", "
         wt_group": "negative"},
    {"bullet_id": "H:F31S", "chain": "H", "germline": "IGHV3-21*08"
        ,
     "group_id": "H:31S", "pos": 31, "region": "cdr1", "target": "S
         ",
     "target_group": "polar", "token": "F31S", "wt": "F", "wt_group
         ": "aromatic"},
    {"bullet_id": "H:L91T", "chain": "H", "germline": "IGHV3-21*08"
        ,
     "group_id": "H:91T", "pos": 91, "region": "fr3", "target": "T"
         ,
```

```
      "target_group": "polar", "token": "L91T", "wt": "L", "wt_group
          ": "nonpolar"},
    {"bullet_id": "L:L54S", "chain": "L", "germline": "IGLV1-44*01"
        ,
     "group_id": "L:54S", "pos": 54, "region": "cdr2", "target": "S
          ",
     "target_group": "polar", "token": "L54S", "wt": "L", "wt_group
          ": "nonpolar"},
    {"bullet_id": "L:Y57S", "chain": "L", "germline": "IGLV1-44*01"
        ,
     "group_id": "L:57S", "pos": 57, "region": "cdr2", "target": "S
          ",
     "target_group": "polar", "token": "Y57S", "wt": "Y", "wt_group
          ": "aromatic"},
    {"bullet_id": "L:D99N", "chain": "L", "germline": "IGLV1-44*01"
        ,
     "group_id": "L:99N", "pos": 99, "region": "fr3", "target": "N"
        ,
     "target_group": "polar", "token": "D99N", "wt": "D", "wt_group
          ": "negative"},
    {"bullet_id": "L:F78S", "chain": "L", "germline": "IGLV1-44*01"
        ,
     "group_id": "L:78S", "pos": 78, "region": "cdr2", "target": "S
          ",
     "target_group": "polar", "token": "F78S", "wt": "F", "wt_group
          ": "aromatic"},
    {"bullet_id": "L:W109Y", "chain": "L", "germline": "IGLV1-44*01
          ",
     "group_id": "L:109Y", "pos": 109, "region": "cdr3", "target":
          "Y",
     "target_group": "aromatic", "token": "W109Y", "wt": "W", "
          wt_group": "aromatic"}
  ],
  "example_id": 1,
  "germline_heavy": "IGHV3-21*08",
  "germline_light": "IGLV1-44*01",
  "memory_version": 1,
  "n_edits": 10,
  "passed": 1.0,
  "properties": {
    "charge_ph5_5": {"beneficial": true, "delta": 2.884, "
        improvement": 1.3, "start": -1.3, "value": 1.584},
    "gravy": {"beneficial": true, "delta": -0.052, "improvement":
        0.0092, "start": -0.1908, "value": -0.2428},
    "immunogenicity": {"beneficial": true, "delta": -4.0, "
        improvement": 4.0, "start": 18.0, "value": 14.0},
    "sequence_liabilities": {"beneficial": true, "delta": -2.0, "
        improvement": 2.0, "start": 7.0, "value": 5.0}
  },
  "pair_mutations (sample)": [
    {"token_a": "H:M41L", "token_b": "H:N75T",
     "immunogenicity": {"beneficial": true, "delta": -1.0, "
         epistasis": 0.0},
     "sequence_liabilities": {"beneficial": true, "delta": -2.0, "
         epistasis": 0.0}},
    {"token_a": "H:E92K", "token_b": "H:L91T",
     "immunogenicity": {"beneficial": true, "delta": -1.0, "
         epistasis": 1.0},
     "charge_ph5_5": {"beneficial": false, "delta": 1.918, "
         epistasis": 0.0}},
    {"token_a": "L:D99N", "token_b": "L:F78S",
     "immunogenicity": {"beneficial": true, "delta": -1.0, "
         epistasis": -1.0},
```

```
      "gravy": {"beneficial": true, "delta": -0.0158, "epistasis":
          0.0}}
  ],
  "proposed_seq": "--QLVES-GGGLVQPGGSLRLSCVASG-FTSSP-----
      FCLTWVRQAPGKGLEWISSISYF--
      DATYIYYTHSLTGRITVSRDNAQNSLFTKINSLRAEDTAVYYCARSRGIG
      ------------------NYYFDLWGQGTLVTVSS|QPVLTQP-
      PSASEAPGQGVRISCSGGI-SNIGT-----TAVQWYRQIPGAAPQSLISE--------
      NNQRPSGVPDRSSGSRSG--TSASLDISRLQSNDEADYYCATYDDS
      ----------------------LQGYVFGTGTKVTVL",
  "reward": 0.9810355955754664,
  "rollout_idx": 0,
  "run": "bfa94dd6",
  "start_seq": "--QLVES-GGGLVQPGGSLRLSCVASG-FTFSP-----
      FCMTWVRQAPGKGLEWISSISYF--
      DATYIYYTHSLNGRITVSRDNAQNSLFLEINSLRAEDTAVYYCARSRGIG
      ------------------NYYFDLWGQGTLVTVSS|QPVLTQP-
      PSASEAPGQGVRISCSGGI-SNIGT-----TAVQWYRQIPGAAPQLLIYE--------
      NNQRPSGVPDRFSGSRSG--TSASLDISRLQSDDEADYYCATWDDS
      ----------------------LQGYVFGTGTKVTVL",
  "turn_idx": 0
}
```


A.3 EPISTASIS GRAPH

Epistasis graph


```
{
  "property": "immunogenicity",
  "germline_heavy": "IGHV3-23",
  "germline_light": "IGKV3-15",

  "nodes (H:cdr1)": {
    "H:25T": {
      "avg": -3.83, "support": 6, "upvotes": 6, "downvotes": 0,
      "group_note": "H:25T lowers immunogenicity on average (-3.83,
          n=6).",
      "bullets": {
        "H:A25T": {
          "avg": -3.83, "support": 6, "single_avg": 0.0,
          "insight": "H:A25T lowered immunogenicity by -3.83 on
              average (n=6)."
        }
      }
    },
    "H:41L": {
      "avg": -2.02, "support": 194, "upvotes": 158, "downvotes": 36
          ,
      "group_note": "H:41L lowers immunogenicity on average (-2.02,
          n=194).",
      "bullets": {
        "H:M41L": {
          "avg": -2.02, "support": 194, "single_avg": 0.043,
          "insight": "H:M41L lowered immunogenicity by -2.02 on
              average (n=194)."
        }
      }
    },
    "H:43F": {
      "avg": 3.54, "support": 13, "upvotes": 2, "downvotes": 11,
```

```
      "group_note": "H:43F raises immunogenicity on average (+3.54,
          n=13).",
      "bullets": {
        "H:W43F": {
          "avg": 3.54, "support": 13, "single_avg": 1.25,
          "insight": "H:W43F raised immunogenicity by +3.54 on
              average (n=13)."
        }
      }
    },
    "H:45D": {
      "avg": -3.0, "support": 3, "upvotes": 3, "downvotes": 0,
      "group_note": "H:45D lowers immunogenicity on average (-3, n
         =3).",
      "bullets": {
        "H:R45D": {
          "avg": -3.0, "support": 3, "single_avg": 0.0,
          "insight": "H:R45D lowered immunogenicity by -3 on
              average (n=3)."
        }
      }
    }
  },

  "edges (intra-chain, H-H)": {
    "H:F90K__H:N87T": {
      "co_avg": -3.132, "co_support": 106, "co_up": 106, "co_down":
          0,
      "epistasis_avg": 1.009,
      "note": "Large-support pair; positive epistasis means the
          combo underperforms the sum of single benefits."
    },
    "H:F90K__H:S88K": {
      "co_avg": -3.206, "co_support": 107, "co_up": 107, "co_down":
          0,
      "epistasis_avg": 0.617,
      "note": "High-confidence pair (n=107), mild positive
          epistasis but still strongly beneficial."
    },
    "H:D83K__H:F90K": {
      "co_avg": -3.0, "co_support": 8, "co_up": 8, "co_down": 0,
      "epistasis_avg": 0.0,
      "note": "Additive charge-reversal + aromatic-to-K pairing."
    },
    "H:I58S__H:M93L": {
      "co_avg": 0.632, "co_support": 57, "co_up": 36, "co_down": 21
          ,
      "epistasis_avg": 0.316,
      "note": "Net-harmful pair with synergistic (positive)
          epistasis; avoid combining."
    }
  },

  "edges (cross-chain, H-L)": {
    "H:F78S__L:I92T": {
      "co_avg": -3.0, "co_support": 24, "co_up": 24, "co_down": 0,
      "epistasis_avg": 0.0,
      "note": "Strong beneficial cross-chain pair, fully additive."
    },
    "H:D83K__L:M4L": {
      "co_avg": 0.0, "co_support": 36, "co_up": 36, "co_down": 0,
      "epistasis_avg": 0.0,
```

```
      "note": "Neutral, high-support cross-chain pairing (FR3
          charge reversal + light N-term liability fix)."
    },
    "H:M41T__L:M4T": {
      "co_avg": 0.173, "co_support": 52, "co_up": 43, "co_down": 9,
      "epistasis_avg": 0.0,
      "note": "Largest-support cross-chain pair in the set; net
          mildly harmful but additive."
    },
    "H:R97D__L:M4T": {
      "co_avg": -3.0, "co_support": 14, "co_up": 14, "co_down": 0,
      "epistasis_avg": 0.0,
      "note": "Strong, additive cross-chain deimmunizing pair."
    }
  }
}
```


A.4 RENDERED MEMORY

Rendered memory

example_id: 98
germline_heavy: IGHV3-23*01
germline_light: IGKV1-8*01
props: gravy, charge_ph5_5, sequence_liabilities, immunogenicity

MEMORY
Prior rollout insights. Use them to guide mutation choices, but still verify with tools.
Node single_avg = isolated single-mutation effect. Edge co_avg = isolated pair effect (only those two mutations). epistasis_avg = co_avg - (single_avg_A + single_avg_B), the deviation from additivity in the raw property: positive means co_avg is greater than the sum of the singles, negative means less, near zero is additive. Favorability depends on the objective direction (raise / lower / target band), not the raw sign of the deviation. Section grouping reflects the objective direction; ↑/↓ votes encode favorability.

[LOCAL BENEFICIAL INSIGHTS (sequence_liabilities/immunogenicity)]
Mutations whose average effect helps the objective
(avg = mean property change; ↑/↓ = rollouts it helped/hurt).
Prefer these.

- sequence_liabilities/IGHV3-23/H:M41T [cdr1]: avg −3.09, n=1196, ↑1195/↓1; In the IGHV3-23 cdr1, H:M41T replaces a nonpolar methionine with polar threonine, removing a methionine oxidation liability and lowering sequence_liabilities by −3.09 on average (n=1196). CDR residues can contact antigen, so swapping hydrophobic bulk for a small polar side chain should be checked against paratope geometry. Co-occurrence data show purely additive behavior with H:F78S, H:M93L, H:V13K, H:L12K, and H:D83K (all epistasis_avg = 0.0), so M41T can be combined freely with these liability-reducing mutations.
- immunogenicity/IGHV3-23/H:F78S [cdr2]: avg −1.7, n=894, ↑772/↓122; H:F78S (cdr2) swaps an aromatic residue for a polar one, eliminating a bulky hydrophobic side chain that likely contributes T-cell receptor/MHC-facing hydrophobic patches, and yields the largest immunogenicity reduction in this set (deterministic effect −1.70, n=894). Because F78 sits in CDR2, verify antigen binding after the aromatic-to-polar swap, since removing the F side chain could perturb paratope packing. Combinations are favorable: synergistic with H:I80T (epistasis_avg +0.200, co_avg −0.438 versus single_avg −0.274) and with H:M93L (epis-

tasis_avg +0.069), while V13K, L12K, S7K, and M41T are additive (epistasis_avg 0.0), supporting F78S as a stackable core of a deimmunization design. ...

[LOCAL INCREASING INSIGHTS (charge_ph5_5)]
Mutations that raise the property on average (avg = mean change; ↑/↓ = times it helped/hurt reaching the target band — e.g. if already in/above the band, raising it usually hurts). Use when you need it higher.

- charge_ph5_5/IGHV3-23/H:D83K [fr3]: avg +5.26, n=698, ↑697/↓1; H:D83K in fr3 reverses a negative Asp to a positive Lys, removing a −1 group and adding a +1 group at pH 5.5, making it the strongest and cleanest charge lever here (+5.26, n=698, just 1 downvote). Its pair with H:D109K in cdr3 is additive — co_avg = 3.93 versus D109K single_avg = 1.97, epistasis_avg = +0.0002 — and other pairs (H:M114L, H:V89T, H:N111S, H:M113L, H:M41L) all show epistasis_avg = 0.0, so Asp-to-Lys reversals stack without antagonism. Favor D-to-K reversals for fr3 charge engineering, but check that the eliminated acidic side chain does not participate in a needed salt bridge before committing.
- charge_ph5_5/IGHV3-23/H:V13K [fr1]: avg +5.1, n=764, ↑761/↓3; H:V13K in fr1 replaces a nonpolar Val with a positively charged Lys, which is directly protonated at pH 5.5 and delivers the largest charge_ph5_5 gain in this set (+5.1, n=764, only 3 downvotes) — a mechanistically consistent direct-charge mechanism. Pairing with H:D100A in fr3 is additive: co_avg = 1.97 versus D100A's single_avg = 0.97, epistasis_avg ≈ 0, and all other neighbors also show epistasis_avg = 0.0. Use V13K as a first-line fr1 lever for raising acidic-pH charge, stacking it with other charge mutations without expecting antagonism. ...

[LOCAL DECREASING INSIGHTS (gravy)]
Mutations that lower the property on average (↑/↓ = times it helped/hurt reaching the target band — e.g. if already in/below the band, lowering it usually hurts). Use when you need it lower.

- gravy/IGHV3-23/H:L91K [fr3]: avg −0.133, n=273, ↑272/↓1; H:L91K in fr3 substitutes the nonpolar leucine with a positively charged lysine, reliably lowering hydrophobicity (single_avg −0.133, n=273, 272 upvotes), making it a strong lever for reducing overall GRAVY when the framework can tolerate added positive charge. All six measured pairings are essentially additive (epistasis_avg within ±4e−5), e.g. with H:M41T in cdr1 (epistasis +1.7e−5) and H:D83K in fr3 (−4.1e−5), so its effect should transfer predictably into multi-mutation designs.
- gravy/IGHV3-23/H:M41T [cdr1]: avg −0.102, n=1196, ↑1181/↓15; H:M41T (cdr1) replaces a nonpolar methionine with polar threonine, lowering gravy by −0.102 on average (n=1196) — a strong shift toward surface hydrophilicity that typically aids solubility when CDR1 is solvent-exposed, but may hurt if M41 packs against a hydrophobic binding interface. The effect is essentially additive with co-mutations: epistasis with H:F78S is −7.6e−06, with H:I80T 2.1e−05, and with H:V13K −1.2e−05, so no pair-level risk of antagonism. ...

[GLOBAL BENEFICIAL INSIGHTS (sequence_liabilities/immunogenicity)]
Mutations whose average effect helps the objective (avg = mean property change; ↑/↓ = rollouts it helped/hurt). Prefer these.

- global/sequence_liabilities/to_K: Substituting to K has a helpful average effect on sequence liabilities (−2.85), with 6244 of 6272 aggregated observations agreeing on help. Because this rule aggregates across all germlines with strongly consistent votes, it is likely generalizable rather than germline-specific. Apply this rule broadly as a default avoid-mutation for sequence liabilities, and defer to a germline-specific local insight only if that germline's local data contradict the global direction.

- global/immunogenicity/to_T: Substituting to T has a helpful average effect on immunogenicity (-3.31), with 2247 of 2456 aggregated observations voting helpful. The large negative average and consistent votes across all germlines indicate a generalizable, germline-independent penalty, and it is the largest immunogenicity help among these targets. Apply this rule broadly by default; defer only when a germline-specific local insight shows the opposite trend in that germline. ...

[GLOBAL INCREASING INSIGHTS (charge_ph5_5)]
Mutations that raise the property on average (avg = mean change; ↑/↓ = times it helped/hurt reaching the target band). Use when you need it higher.

- global/charge_ph5_5/to_K: Mutation to lysine raises charge at pH 5.5 strongly and consistently, averaging $+5.81$ across 6272 observations with a $\sim$99.5% upvote rate, consistent with lysine's protonation and positive charge at acidic pH. This effect reflects the direct physicochemical consequence of adding a protonatable amine, so it is expected to generalize across germlines with little germline-specific variance. Apply this rule broadly whenever the goal is increasing charge at pH 5.5. ...

[GLOBAL DECREASING INSIGHTS (gravy/charge_ph5_5)]
Mutations that lower the property on average (↑/↓ = times it helped/hurt reaching the target band). Use when you need it lower.

- global/gravy/to_K: Mutating to K lowers GRAVY (hydropathy) by ~ -0.135 on average, a harmful shift for hydrophobicity with 6272 observations and 99% agreement. The huge support base makes this direction highly generalizable across germlines, since K's charged side chain consistently reduces hydropathy regardless of framework context. Apply this rule broadly as a default expectation; defer to germline-specific insights mainly when magnitude, not direction, matters.
- global/charge_ph5_5/to_D: Mutating to D reduces charge at pH 5.5 by an average of -1.13, a large negative charge shift. The effect is nearly universal: 389 of 398 instances (98%) agree, with only 9 counterexamples, reflecting the consistent electrostatic consequence of introducing an acidic residue. Treat this as a broadly applicable rule across germlines; deviate only when a local insight documents an exception. ...

B EXPERIMENTAL DETAILS

B.1 DETAILS OF OPTIMIZED PROPERTIES

We provide the details of the optimized properties, including the oracle and the description of their significance in Table A1.

Table A1: **Optimized properties.** Six antibody properties the agent optimizes, the oracle or function measuring each, and why each are significant.

Property	Objective	Oracle	Why it matters
Immunogenicity	Minimize	Pep2Vec (Thrift et al., 2024)	Predicted MHC class II epitope cores that can provoke an anti-drug antibody response.
Liabilities	Minimize	Biopython (Cock et al., 2009)	Motifs that degrade during manufacture and storage, costing potency and shelf life.
GRAVY	Range	Biopython (Kyte & Doolittle, 1982)	Exposed hydrophobicity drives aggregation, but an overly hydrophilic domain loses stability.
Charge	Range	Biopython (Cock et al., 2009)	Endosomal charge at pH 5.5 governs clearance rate and formulation viscosity at both extremes.
pTM	Range	Protenix (Zhang et al., 2026b)	Predicted TM-score between the predicted and the true complex.
AbAg iPTM	Range	Protenix (Zhang et al., 2026b)	Accuracy of the predicted relative placement of the antibody and the antigen.

B.2 DETAILS OF IN SILICO BINDING OPTIMIZATION DATASET

We provide the details of in silico binding optimization dataset in Table A2.Notably, an example is identified by a complex identifier of the form `{index}_{pdb}_{antibody chains}_{antigen chain}`, so two entries from the same PDB with different chain combinations are two distinct examples.

Table A2: **Details of the in silico binding optimization dataset.** Antibodies were clustered by CDR sequence identity. Each antibody's Heavy and Light chain CDR loops are concatenated into one string. Mmseqs2 (Steinegger & Söding, 2017) is run with a 70% sequence identity cutoff.

Cluster	# Complexes	PDBs
993	5	7l7e, 8d8q, 8ekd, 8suo
114	6	6kyz, 8f2j
147	7	7ben, 7beo, 7orb, 7tbf
148	4	7jmw, 7lq7, 7s5q, 7s5r
193	6	4fp8, 5umn, 6d0u
234	4	4mwf, 6bkb, 6uyd, 6uym
268	24	4g6f, 4u6g, 5ghw, 5iq7, 5iq9, 5sy8, 5t29, 5t5b, 5t6l, 5t80, 5t85, 5tfw, 5wdf, 6vpx, 6x58, 8sx3, 8tzw, 8u08, 8v2e
308	7	6p3r, 6p3s
348	8	7won, 7pqz, 7swp, 7wp0, 7wp1, 7wp2, 7wp5, 8siq
404	5	6vi1, 6xmi, 8djg
409	3	2hrp, 8gkl, 8v8f

B.3 Details of global insight construction

Global insights capture mutation trends that are shared across germlines in addition to the context-specific insights retrieved by exact matching. In detail, we aggregate observations for each property by their mutation target, such as `to K`, and record the mean property effect, number of observations, and directional votes. These insights are stored separately from the exact-match insights and are included in the memory prompt together with the retrieved local insights. Global insights provide broader evidence when a mutation trend transfers across biological contexts. Notably, they complement rather than replace the context-specific insights defined by the antibody-grounded identifier.

B.4 Prompts

Shared prompt for multi property and in silico antibody binding optimization

Your task is to optimize an antibody Fv by proposing AHO-coordinate mutations to its heavy and light variable domains, while staying within a maximum number of de-gapped sequence edits from the starting molecule.

TURN BUDGET
You have {max_turns} turns total. Each turn is either a mutation-set proposal OR a tool call — both count toward the limit. A mutation-set proposal does NOT end the episode. After every proposal, you must call evaluate_mutations in a separate message to evaluate it, then use the measured properties to propose an improved mutation set. Propose a different mutation set on every non-tool-call turn.

COORDINATES
Positions use AHO (Honegger-Plueckthun) antibody numbering: a fixed coordinate system in which a given column maps to the same structural site across antibodies. A position means the canonical AHO column, not the raw de-gapped offset. A "-" is a canonical AHO position that is absent in this antibody.
Count columns INCLUDING the "-" gaps in the displayed sequence — every "-" still consumes a column number. To place a token, read the residue printed at that column in the chain shown below and use it as the token's first letter; if that letter is not what you intended to edit, your column index is wrong (you have almost certainly counted the de-gapped offset instead of the AHO column).

Example: for "...G A - S G..." at columns "...24 25 26 27 28...", the S sits at column 27, so editing it is S27X — not S26X, because the gap at column 26 still counts.

OUTPUT FORMAT
Each proposal must contain all three tag blocks:
<|reasoning_summary_start|>
[Your reasoning: what mutations you made, why, how you expect the target property to change]
<|reasoning_summary_end|>
<|mutation_answer_heavy_start|>
{A27D, G31S, T58A}
<|mutation_answer_heavy_end|>
<|mutation_answer_light_start|>
{S27T, Y50F}
<|mutation_answer_light_end|>
Mutation token syntax on AHO columns:
- A27D changes AHO column 27 from A to D (a substitution — the common case).
- -50G fills an absent AHO column 50 with G (insertion).
- A95- empties AHO column 95, deleting that residue from the de-gapped sequence.
- {} makes no change.

Most edits are simple substitutions; reach for the gap operations only when an insertion or deletion is actually needed.
Before proposing edits, call get_antibody_residue_index(heavy_reference, light_reference) once. Use the returned {wt}{pos} prefixes verbatim and append only your target residue ('A27' ->'A27D') so your wildtype letter and AHO column are always correct.
Positions never renumber because AHO chain length is fixed. Your edit budget is max_edits de-gapped edits across both chains combined: residue-to-residue substitutions count as 1, residue-to-gap counts as 1, and gap-to-residue counts as 1. Gap-to-gap changes do nothing and should not be proposed.

VALID MUTATION RULES
1. Match the wild-type letter. The first character of every token MUST be the residue currently at that AHO column in the chain shown below, using "-" if the column is a gap. A mismatch is skipped.
2. Stay in range. Heavy positions must be between 1 and {heavy_aho_length}; light positions must be between 1 and {light_aho_length}. Out-of-range tokens are skipped.
3. No duplicates. Do not target the same AHO column twice within one chain; duplicate-position tokens are skipped.

evaluate_mutations reports any mutation it had to skip and why, such as expected V at column 27. When you see this, re-read the displayed chain at that exact column, set your token's first letter to the residue printed there, and resubmit — do not keep the rejected token. A skipped mutation is one that simply did not happen.

When a separate MEMORY message is present, it contains prior rollout insights for related germlines and objectives; use it to prioritize mutations, but still verify every proposal with evaluate_mutations.

To evaluate a mutation set, first propose it in one message, then call evaluate_mutations in a separate message, passing your mutation set and the original starting chains as heavy_reference and light_reference. It returns both the reconstructed sequence and its properties.

ITERATION RULE
For range objectives, keep proposing and evaluating until the target value is inside the requested range and all constraints hold, or until you exhaust the turn budget. For maximize/minimize objectives, keep improving and evaluating until the turn budget is exhausted.

AVAILABLE PROPERTIES
{available_properties}

REQUIREMENTS
1. Mutation-set proposals and tool calls must be in separate messages.
2. Your FIRST message must be a tool-call message that calls get_antibody_residue_index and measures the starting Fv baseline.
3. The reasoning summary must be substantive.
4. The heavy and light mutation answer tags must contain ONLY edit sets — no labels.
5. Your LAST message must include both heavy and light mutation answer tags.

GOAL
Given the antibody Fv AHO-numbered heavy and light chains below, propose designs within {max_edits} combined de-gapped edits that {objective_text}.

SEQUENCE REGION MAP
Heavy chain (AHO-numbered, {heavy_aho_length} columns):
Full Fv: {heavy_aho_sequence}
FR1: {h_fr1}

CDR1: {h_cdr1}
FR2: {h_fr2}
CDR2: {h_cdr2}
FR3: {h_fr3}
CDR3: {h_cdr3}
FR4: {h_fr4}
Germline: {h_v_gene} / {h_j_gene} ({h_species})

Light chain (AHO-numbered, {light_aho_length} columns):
Full Fv: {light_aho_sequence}
FR1: {l_fr1}
CDR1: {l_cdr1}
FR2: {l_fr2}
CDR2: {l_cdr2}
FR3: {l_fr3}
CDR3: {l_cdr3}
FR4: {l_fr4}
Germline: {l_v_gene} / {l_j_gene} ({l_species})

Objective: {objective_detail}
Constraints: {constraints_text}

WORKFLOW
1. In the first tool-call message, call get_antibody_residue_index on the starting AHO chains and call evaluate_mutations with {} for heavy_mutations and {} for light_mutations to measure the baseline.
2. Identify heavy and/or light AHO-column substitutions likely to move the target property.
3. Propose heavy and light mutation sets in a separate message.
4. Call evaluate_mutations on your heavy and light mutation sets plus the starting AHO chains to evaluate them.
5. Iterate — evaluate every proposal, then propose an improved mutation set. For range objectives, stop early only after the value is in range and constraints hold. For maximize/minimize objectives, keep improving until you run out of turns.
6. Your last message must contain both mutation answer tag blocks with your best edit sets.

Property optimization. The task prompt is combined with the shared prompt.

`Task prompt for in silico antibody binding optimization`

Your task is to improve the properties of an antibody Fab by editing only Fv, while staying within a maximum number of de-gapped sequence edits from the starting molecule.

Properties are computed on the de-gapped Fv. Charge, GRAVY, liability, immunogenicity, edit_distance, and sequence_length use de-gapped heavy plus de-gapped light. AbLang2 naturalness is the mean of the separate de-gapped heavy-chain and light-chain scores.

SCORING
The objective properties are:
{property_description}

In silico antibody binding optimization. The task prompt is combined with the shared prompt.

Task prompt for in silico antibody binding optimization

Your task is to improve binding of an antibody Fab to its target protein by editing only heavy-chain CDR-H3, while staying within a maximum number of de-gapped sequence edits from the starting molecule.

SCORING
The two objective properties are:
- pTM: Protenix global fold confidence (0–1).
- abag_ipTM: antibody - antigen interface confidence (0–1), the primary binding signal.

Curator. The insight curator gets the retrieved subgraph as input and generates the natural language insight.

Prompt for insight curator

You curate antibody-optimization memory insights for mutations within one germline. Return a JSON list with one object per input item, each containing exactly the string fields id and insight. Each insight must be 2-3 tightly written sentences covering the mechanism, when the mutation helps or hurts, and actionable guidance. The insight must add mechanistic, region-aware reasoning (framework vs CDR, biophysical rationale for the property) and actionable mutation guidance. Stay grounded in the supplied evidence. The region field, when present, is the authoritative AHo region (e.g. fr1, fr2, cdr2) for this position; state it verbatim and never infer the region from the position number. When amino_acid_groups is present, use the original-to-mutated residue R-group change for biophysical rationale.

The item has two effect fields: single_avg is the isolated single-mutation effect (only this mutation applied); avg is the mean full-design effect when this mutation appeared alongside others (confounded). Prefer single_avg for mechanistic claims. The item may include co_occurrence: each neighbor entry has single_avg (neighbor's isolated effect), co_avg (true isolated pair effect — only these two mutations applied), and epistasis_avg (= co_avg - this single_avg- neighbor single_avg), the deviation from additivity in the raw property: positive means co_avg is greater than the sum of the singles, negative means less, near zero is additive. In your insight, when co_occurrence is present: cite the neighbor token, give the numeric epistasis_avg and its sign, and say whether it is favorable given the objective direction: positive for maximize, negative for minimize; for range, judge movement toward the target band. If the objective or target-band evidence is unavailable, describe only the deviation. Make combination claims ONLY from co_occurrence; omit if absent.

B.5 HYPERPARAMETERS

Table A3: **Hyperparameters used in all experiments.**

Task	Hyperparameter	Value
Multiple property	Rollouts per example	10
	Number of examples	100
	Maximum turns per rollout	15
In silico binding	Rollouts per example	30
	Number of examples	# complexes per cluster
	Maximum turns per rollout	10
Both	Local memory top-k	60
	Global memory top-k	30
	# of Neighbors N	6

For in silico antibody binding optimization, the number of examples per cluster is determined by the number of antigen–antibody complexes available in each cluster, as shown in Table A2.

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 CUMULATIVE PERFORMANCE PROGRESSION WITHOUT A ROLLING AVERAGE

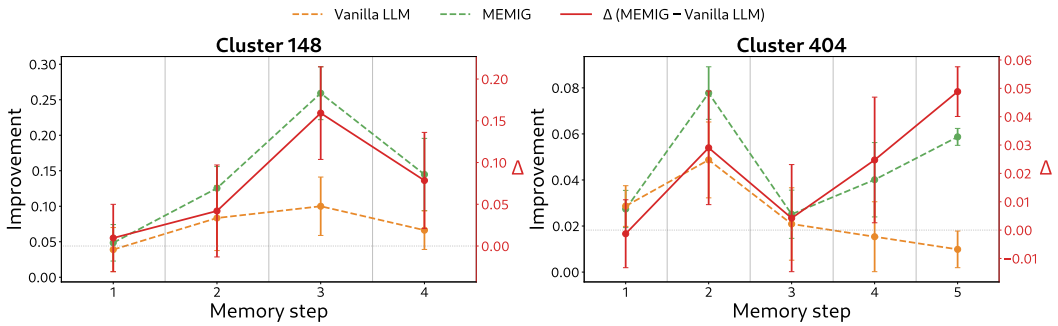

Figure 7: **Cumulative performance progression without a rolling average.** Vertical lines mark memory updates, where the rollouts between two lines are generated concurrently under the same memory. Each point is the mean improvement over all rollouts sharing a memory step, and error bars indicate the standard error across those rollouts.

The rolling average in Figure 6 smooths the rollouts within each memory step, which can obscure the value of an individual step. To remove this smoothing, we report the mean improvement of every rollout sharing a memory step as a single point in Figure 7, with error bars indicating the standard error across those rollouts. We observe the same trend as in Figure 6, where MEMIG improves over successive memory steps while the vanilla LLM does not, which confirms that the progression is not an artifact of the smoothing.

C.2 SAME ORACLE CALL ANALYSIS

Table A4: **Comparison under the same oracle calls. Bold** highlights the best score in a cluster and * denotes additional decomposition oracle call mode.

Cluster	Model	Pass@1	Pass@3	Avg
148	DC-Cu*	0.067	0.182	0.109
	ACE*	0.100	0.215	**0.169**
	MEMIG (Ours)	**0.258**	**0.456**	0.144
193	DC-Cu*	0.056	0.159	0.110
	ACE*	0.028	0.081	0.107
	MEMIG (Ours)	**0.194**	**0.374**	**0.141**
993	DC-Cu*	0.000	0.000	0.070
	ACE*	0.000	0.000	0.070
	MEMIG (Ours)	**0.007**	**0.020**	**0.090**

We compare MEMIG against the memory baselines under a matched oracle budget, since the decomposition of MEMIG consumes additional oracle calls. Here, single-mutation and pairwise decomposition is an epistasis graph component of MEMIG rather than a shared experimental setting, since ACE (Zhang et al., 2026a) and DC-Cu (Suzgun et al., 2026) accumulate experience from the trajectories an agent produces. Therefore, modifying them is not required for a fair comparison, as the decomposition is part of what we propose rather than an advantage granted by the setup. Nevertheless, we report a setting in which ACE and DC-Cu receive the same decomposed measurements, which removes any difference in the oracle call budget between the methods.

We provide the results in Table A4 under GLM-5.3-flash backbone. We observe that MEMIG attains the highest pass@1 and pass@3 on all three clusters even when the baselines share the same oracle calls, which shows that the gain comes from how the measurements are stored rather than from access to them. Here, ACE attains a higher average improvement on one cluster while satisfying the constraints less often, which is consistent with the observation that improvement is averaged over all designs regardless of whether they satisfy the constraint.

C.3 COST ANALYSIS

Table A5: **Token cost analysis.** Average token consumption per rollout. Input and output tokens are reported in thousands. Tokens per pass include tokens consumed by unsuccessful rollouts.

Method	Input	Output	Pass@1	Tokens / pass
Vanilla LLM	98.1	6.8	0.022	4.719M
Obs.	204.3	7.2	0.000	–
ACE	103.8	6.6	0.017	6.623M
DC-Cu	120.3	7.7	0.006	23.028M
MEMIG	270.1	9.2	**0.194**	**1.436M**

Table A6: **Oracle evaluation cost.** Average number of oracle evaluations per rollout. Single and pair evaluations are additional evaluations used by MEMIG to construct the epistasis graph.

Method	Ref.	Candidate	Single	Pair	Total
Vanilla LLM	1.00	6.03	0	0	7.03
Obs.	1.00	5.88	0	0	6.88
ACE	1.00	5.44	0	0	6.44
DC-Cu	1.00	5.76	0	0	6.76
MEMIG	1.00	4.95	8.46	22.97	37.38

We report the computational cost of GLM-5.3-Flash on the in silico antibody binding optimization task (cluster 193). As shown in Table A5, MEMIG uses more input tokens because the retrieved memory is injected before each rollout, while achieving the highest Pass@1 and the lowest token cost per successful rollout. While the absolute token usage is highest, the token cost per successful rollout reduces to 1.436M, compared with >4M for all baselines.

Table A6 reports the mean number of unique oracle inputs per rollout. The additional single- and pair-mutation inputs introduced by MEMIG's epistasis decomposition increase this for $\binom{n}{2} + n$ where n denotes the number of proposed mutations. This increase is by design, as MEMIG explicitly evaluates isolated mutations and pairs to measure epistasis, and these additional evaluations remain practical in our in silico setting because the computational oracles are easily re-runnable without experimental measurements. Moreover, the decomposition budget is a tunable knob rather than a structural requirement. For example, pairs can be subsampled instead of exhaustively decomposed, so the design does not depend on oracles being free.

C.4 DETAILED RESULTS

Multi property optimization. We provide the detailed results for Section 3.1.

Table A7: **Multiple property optimization.** Teal indicates improvement over all baselines and **bold** indicates the best results.

		Pass		Improvement				
Backbone	**Model**	@1	@3	Avg.	Charge	GRAVY	Immu.	Liab.
Random		0.003	0.009	-0.419	-0.071	0.000	-1.243	-0.363
MCMC		0.004	0.012	-0.043	0.308	0.008	-0.465	-0.115
Haiku-4.5	Vanilla	0.077	0.212	0.431	1.223	0.004	-0.037	0.535
	Obs.	0.073	0.168	0.477	1.225	0.019	0.167	0.499
	DC-Cu	0.188	0.423	0.602	1.240	0.040	0.431	0.696
	ACE	0.036	0.094	0.390	1.255	0.032	0.197	0.078
	MEMIG (Ours)	0.221	0.373	0.718	1.246	0.039	1.036	0.551
GLM-5.3-Flash	Vanilla	0.424	0.699	1.327	1.273	0.044	0.287	3.708
	Obs.	0.461	0.717	1.344	1.256	0.044	0.560	3.515
	DC-Cu	0.409	0.707	1.621	1.262	0.044	-0.165	**5.343**
	ACE	0.395	0.650	1.577	1.270	0.044	-0.049	5.043
	MEMIG (Ours)	**0.761**	**0.847**	**1.754**	**1.279**	**0.045**	**2.276**	3.416

In silico antibody binding optimization. We provide the detailed results for Section 3.2.

Table A8: **In silico antibody binindg optimization. Bold** indicates the best per column across all methods and teal indicates improvement over all non-structural methods within the same backbone. * indicates structural baselines included as reference.

		993					114					147				
		Pass		Improvement			Pass		Improvement			Pass		Improvement		
Category	**Model**	@1	@3	Avg	ipTM	pTM	@1	@3	Avg	ipTM	pTM	@1	@3	Avg	ipTM	pTM
Statistical	Random	0.000	0.000	0.028	0.043	0.013	0.089	0.206	0.016	0.036	-0.003	0.024	0.070	-0.035	-0.041	-0.030
	MCMC	**0.013**	**0.040**	0.075	0.110	0.039	0.233	0.368	0.069	0.115	0.024	0.171	0.383	0.064	0.101	0.027
Structure-based	ProteinMPNN*	0.000	0.000	0.138	**0.196**	0.081	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	AbMPNN*	0.000	0.000	0.116	0.164	0.068	**0.600**	**0.762**	**0.160**	**0.242**	**0.077**	0.229	0.406	0.109	0.158	0.059
Haiku-4.5	Vanilla	0.000	0.000	0.010	0.014	0.007	0.122	0.164	0.053	0.081	0.025	0.048	0.128	0.044	0.065	0.024
	+ Obs.	0.000	0.000	0.077	0.105	0.050	0.411	0.626	0.133	0.204	0.062	0.229	0.431	0.097	0.141	0.053
	+ DC-Cu	0.000	0.000	0.054	0.076	0.032	0.306	0.366	0.098	0.151	0.045	0.162	0.347	0.099	0.145	0.053
	+ ACE	0.000	0.000	0.048	0.068	0.028	0.283	0.333	0.098	0.149	0.046	0.052	0.129	0.090	0.129	0.050
	+ MEMIG (Ours)	0.000	0.000	**0.141**	0.193	**0.089**	0.317	0.397	0.130	0.197	0.062	**0.533**	**0.876**	**0.217**	**0.324**	**0.109**
GLM-5.3-Flash	Vanilla	0.000	0.000	0.042	0.056	0.028	0.156	0.291	0.053	0.082	0.024	0.067	0.159	0.063	0.090	0.035
	+ Obs.	0.000	0.000	0.061	0.083	0.039	0.217	0.316	0.044	0.068	0.019	0.076	0.159	0.017	0.026	0.007
	+ DC-Cu	0.000	0.000	0.042	0.058	0.026	0.178	0.313	0.033	0.051	0.015	0.052	0.124	0.028	0.041	0.016
	+ ACE	0.000	0.000	0.048	0.064	0.032	0.256	0.365	0.091	0.141	0.041	0.114	0.252	0.036	0.055	0.017
	+ MEMIG (Ours)	0.007	0.020	0.090	0.124	0.056	0.278	0.348	0.099	0.155	0.043	0.269	0.480	0.128	0.191	0.066

		148					193					234				
		Pass		Improvement			Pass		Improvement			Pass		Improvement		
Category	**Model**	@1	@3	Avg	ipTM	pTM	@1	@3	Avg	ipTM	pTM	@1	@3	Avg	ipTM	pTM
Statistical	Random	0.008	0.025	-0.020	-0.026	-0.014	0.011	0.032	0.041	0.055	0.027	0.192	0.453	-0.115	-0.174	-0.055
	MCMC	0.025	0.075	0.027	0.045	0.009	0.044	0.128	0.104	0.141	0.067	0.600	0.890	0.011	0.012	0.009
Structure-based	ProteinMPNN*	0.000	0.000	0.285	0.417	0.153	0.000	0.000	0.268	0.386	0.150	0.108	0.208	**0.075**	**0.109**	**0.042**
	AbMPNN*	**0.575**	**0.866**	**0.288**	**0.421**	**0.155**	0.000	0.000	**0.285**	**0.401**	**0.169**	0.733	0.775	0.052	0.075	0.030
Haiku-4.5	Vanilla	0.075	0.203	0.051	0.075	0.026	0.017	0.049	0.031	0.043	0.018	0.600	0.920	0.037	0.054	0.020
	+ Obs.	0.017	0.050	0.055	0.078	0.031	0.017	0.049	0.050	0.070	0.030	0.833	**0.997**	0.069	0.103	0.035
	+ DC-Cu	0.067	0.182	0.091	0.132	0.050	0.033	0.098	0.168	0.229	0.107	0.958	**1.000**	0.067	0.100	0.034
	+ ACE	0.192	0.335	0.132	0.193	0.072	0.117	0.262	0.183	0.253	0.112	0.967	**1.000**	**0.075**	**0.109**	**0.042**
	+ MEMIG (Ours)	0.342	0.579	0.201	0.293	0.109	**0.139**	**0.287**	0.171	0.237	0.105	**0.975**	**1.000**	0.074	0.108	0.041
GLM-5.3-Flash	Vanilla	0.025	0.075	0.072	0.103	0.041	0.022	0.067	0.085	0.114	0.055	0.575	0.799	0.015	0.023	0.007
	+ Obs.	0.000	0.000	0.057	0.073	0.040	0.000	0.000	0.116	0.158	0.073	0.767	0.972	0.051	0.075	0.027
	+ DC-Cu	0.017	0.050	0.025	0.036	0.014	0.006	0.017	0.044	0.059	0.029	0.792	0.930	0.035	0.053	0.018
	+ ACE	0.083	0.214	0.036	0.050	0.022	0.017	0.049	0.108	0.146	0.070	0.675	0.917	0.053	0.072	0.034
	+ MEMIG (Ours)	0.258	0.456	0.144	0.079	0.210	**0.194**	**0.374**	0.141	0.197	0.084	0.925	**1.000**	0.067	0.101	0.033

		268					308					348				
		Pass		Improvement			Pass		Improvement			Pass		Improvement		
Category	**Model**	@1	@3	Avg	ipTM	pTM	@1	@3	Avg	ipTM	pTM	@1	@3	Avg	ipTM	pTM
Statistical	Random	0.122	0.300	-0.037	-0.047	-0.026	0.095	0.263	0.007	0.012	0.001	0.192	0.455	-0.101	-0.144	-0.058
	MCMC	0.338	0.584	0.057	0.095	0.019	0.214	0.519	0.122	0.174	0.070	0.546	0.752	0.008	0.012	0.004
Structure-based	ProteinMPNN*	0.000	0.000	0.115	0.191	0.038	0.000	0.000	0.170	0.235	0.104	0.000	0.000	-0.022	-0.041	-0.002
	AbMPNN*	0.074	0.130	0.117	0.198	0.037	0.014	0.042	0.059	0.081	0.038	0.237	0.328	0.057	0.081	0.033
Haiku-4.5	Vanilla	0.351	0.568	0.081	0.130	0.032	0.195	0.337	0.051	0.073	0.029	0.421	0.635	0.025	0.037	0.013
	+ Obs.	0.633	0.786	0.102	0.168	0.035	0.500	0.653	0.202	0.287	0.116	0.650	0.806	0.054	0.079	0.030
	+ DC-Cu	0.163	0.332	0.015	0.024	0.006	0.386	0.711	0.179	0.255	0.103	**0.842**	0.875	0.048	0.071	0.026
	+ ACE	0.653	0.748	0.088	0.146	0.030	0.238	0.515	0.137	0.193	0.081	0.796	**0.887**	0.054	0.078	0.029
	+ MEMIG (Ours)	**0.694**	0.844	**0.121**	**0.199**	**0.042**	**0.729**	**0.962**	0.171	0.246	0.096	0.838	**0.887**	**0.097**	**0.148**	**0.046**
GLM-5.3-Flash	Vanilla	0.307	0.547	0.040	0.068	0.012	0.233	0.541	0.148	0.210	0.086	0.575	0.764	0.021	0.033	0.009
	+ Obs.	0.518	0.757	0.090	0.148	0.032	0.295	0.574	0.157	0.225	0.089	0.583	0.725	-0.012	-0.020	-0.005
	+ DC-Cu	0.194	0.386	0.033	0.055	0.012	0.295	0.638	0.158	0.225	0.092	0.492	0.599	0.000	-0.001	0.002
	+ ACE	0.318	0.524	0.038	0.064	0.012	0.248	0.569	0.113	0.160	0.065	0.683	0.817	0.017	0.029	0.006
	+ MEMIG (Ours)	0.692	**0.877**	0.103	0.170	0.036	0.643	0.913	**0.228**	**0.327**	**0.128**	0.804	0.884	0.070	0.103	0.036

		404					409				
		Pass		Improvement			Pass		Improvement		
Category	**Model**	@1	@3	Avg	ipTM	pTM	@1	@3	Avg	ipTM	pTM
Statistical	Random	0.067	0.181	-0.021	-0.039	-0.004	0.322	0.690	0.002	-0.002	0.007
	MCMC	0.140	0.357	0.029	0.058	0.000	0.567	0.922	0.040	0.066	0.013
Structure-based	ProteinMPNN*	0.007	0.020	0.058	0.117	0.000	0.244	0.388	0.079	0.138	0.019
	AbMPNN*	0.293	0.517	**0.070**	**0.141**	0.000	**0.989**	**1.000**	**0.086**	**0.149**	**0.023**
Haiku-4.5	Vanilla	0.187	0.443	0.045	0.090	0.000	0.656	0.917	0.061	0.105	0.017
	+ Obs.	0.313	0.472	0.062	0.124	0.000	0.844	0.995	0.081	0.140	**0.023**
	+ DC-Cu	0.133	0.356	0.051	0.103	0.000	0.778	0.906	0.033	0.057	0.010
	+ ACE	0.280	0.614	0.064	0.128	0.000	0.978	**1.000**	0.074	0.131	0.017
	+ MEMIG (Ours)	**0.440**	**0.744**	0.063	0.127	0.000	0.933	0.998	0.072	0.124	0.019
GLM-5.3-Flash	Vanilla	0.047	0.131	0.025	0.050	0.000	0.522	0.816	0.033	0.055	0.011
	+ Obs.	0.107	0.279	0.034	0.069	-0.001	0.667	0.890	0.051	0.083	0.018
	+ DC-Cu	0.107	0.279	0.022	0.045	-0.001	0.611	0.729	0.024	0.037	0.011
	+ ACE	0.100	0.263	0.040	0.081	0.000	0.711	0.890	0.045	0.074	0.015
	+ MEMIG (Ours)	0.260	0.438	0.062	0.125	0.000	0.956	**1.000**	0.077	0.136	0.019