MGPO: Mutant Group Policy Optimization for Structure-Diverse Protein Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Proteins can admit multiple structurally distinct solutions for the same amino-acid sequence, making broad structural exploration important for protein generative models. Mutation offers a natural way to expand policy exploration beyond ordinary autoregressive sampling, but we identify a previously under-characterized failure mode of mutation-augmented reinforcement learning: evolutionary exploration produces heterogeneous structural neighborhoods with different reward offsets, so a single global baseline can assign negative advantages to mutants that are improving within their own neighborhood. We introduce Mutant Group Policy Optimization (MGPO), which couples structured evolutionary exploration with neighborhood-aware credit assignment in the protein-structure-token space. MGPO generates structurally diverse mutant populations through reward-blind mutation and quality-diversity search, organizes them using reward-independent structural geometry, and computes each mutant's advantage relative to its full structural neighborhood before representative compression. This preserves learning signals for promising mutant families that would otherwise be suppressed by global population statistics. On the CAMEO22 benchmark, MGPO substantially improves mean lDDT-CA and TM-score. Diversity is evaluated in both structure token space and decoded 3D space; the measurements show that MGPO can improve the diversity of generated protein structures. A matched same-layout comparison, together with local-only ablations, shows that neighborhood-local credit assignment accounts for most of the improvement, while the evolutionary search provides the diverse structural neighborhoods on which this mechanism operates. These results establish MGPO as a mutation-aware policy-optimization framework for jointly improving structural quality and maintaining population-level exploration.

1 Introduction

Protein folding—inferring a protein's three-dimensional structure from its amino-acid sequence—is among the hardest problems in biology. Despite rapid progress in structure prediction (Jumper et al., 2021; Abramson et al., 2024; Lin et al., 2023; Hayes et al., 2024; Wang et al., 2025b), autoregressive models still do not perform folding natively: even the strongest general-purpose language models currently delegate structure prediction to specialized external predictors rather than generating a structure end-to-end. We therefore study whether an autoregressive model can generate structure directly, using DPLM-2's discrete structure tokens as its output vocabulary (Wang et al., 2025a). Optimizing such a generator with reinforcement learning, however, surfaces a credit-assignment problem once exploration is augmented by mutation: mutants of the same sequence occupy separated structural neighborhoods with different reward offsets, so a candidate that improves within its own neighborhood ($R_i > b_g$) can still fall below the global mean ($R_i < b_{\text{global}}$) and receive a negative global advantage, suppressing entire promising mutant families.

MGPO builds on two lines of work. First, DPLM-2's structure tokenizer encodes each residue as a 13-bit binary vector via Lookup-Free Quantization (LFQ) (Yu et al., 2023),

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.

putting a structure token in one-to-one correspondence with its 13-bit code (Wang et al., 2025a); we adopt this representation so that an autoregressive model can emit structure tokens directly. On this code, bit-flip-style operations always yield a valid structure token, but simple single-bit flipping alone does not diversify structural exploration enough—which is why richer mutation and evolutionary search are needed. Second, PPO, GRPO, and TOPR provide the on-policy and off-policy RL foundations and training framework for language models (Schulman et al., 2017; Shao et al., 2024; Le Roux et al., 2025), which we adopt. The gap between these two lines is that their advantage baselines are global: once mutation spreads samples across structural neighborhoods with different reward offsets, a global baseline misreads a within-neighborhood improvement as a global regression. This is precisely the problem we address.

We introduce Mutant Group Policy Optimization (MGPO), to our knowledge the first systematic application of evolutionary mutation to the RL fine-tuning of protein structure generation. MGPO implements nine mutation/evolutionary search strategies—genetic algorithms, quality-diversity, DPP selection, island models, MAP-Elites, and others—in the DPLM-2 structure-token space, each supplying structurally diverse mutants under a matched evaluation budget. To keep mutation-driven training stable, MGPO then groups these mutants by structure rather than reward and scores each mutant against its own group mean, so that a family improving within its own neighborhood is not suppressed by the global baseline. Our contributions are: (1) introducing mutation/evolutionary search into protein-structure RL, with a systematic comparison of nine strategies; and (2) identifying the local/global advantage sign-failure in mutation-augmented training, together with structure-local credit assignment as its remedy. Overall, MGPO improves the exploration effectiveness, structural diversity, and prediction quality of protein structure models, offering a new RL solution to the most challenging folding task.

2 Related Work

DPLM-2 represents protein backbone structure with discrete LFQ tokens and a frozen coordinate encoder/decoder (Wang et al., 2025a); MGPO uses this codec as a mutation and evaluation interface rather than using DPLM-2 as the trainable policy. Group-relative RL methods such as GRPO (Shao et al., 2024), sequence-level clipping/reweighting methods such as TOPR and related off-policy variants (Le Roux et al., 2025; Zheng et al., 2025; Shen et al., 2026), and large-scale RL systems (Yu et al., 2025) assume rollout or otherwise characterizable behavior distributions more directly than the evolutionary-mutant setting considered here. Diversity-aware protein optimization has also been pursued through set rewards or regularizers (Wang et al., 2025c; Ye et al., 2026). MGPO differs by changing the reference used for mutant credit assignment instead of making diversity itself the primary reward. Evolutionary backends such as novelty search, quality-diversity optimization, DPP selection, island models, and MAP-Elites (Lehman & Stanley, 2011; Pugh et al., 2016; Kulesza & Taskar, 2012; Whitley et al., 1999; Mouret & Clune, 2015) are treated as modular proposal mechanisms and are evaluated separately in the appendix.

3 preliminary

3.1 Structure-token representation

A prompt x fixes a single-chain amino-acid sequence of length L. The trainable autoregressive policy emits one DPLM-2 structure token per residue, $\mathbf{t}(y) = (t_1, \ldots, t_L)$ with $t_i \in \{0, \ldots, 8191\}$. DPLM-2 is frozen and used only to encode reference structures and decode generated token paths. Because $8192 = 2^{13}$, every token corresponds to a 13-bit LFQ code. We use normalized bit-Hamming distance

$$d(\mathbf{t}, \mathbf{t}') = \frac{1}{13L} \sum_{i=1}^{L} \mathrm{popcount}(t_i \oplus t'_i) \tag{1}$$

for reward-blind mutation, deduplication, and grouping. This is a computational geometry in token space, not a physical distance; all candidates that affect training are decoded and scored structurally.

The policy entering RL is initialized from Qwen3.5-4B, whose vocabulary is expanded with the 8192 structure tokens plus special tokens marking the start and end of a protein structure, and is then trained as the RL cold-start model on 200B tokens from the AlphaFold Protein Structure Database v3 (Varadi et al., 2022). We therefore call this model the SFT initialization throughout and measure every RL delta relative to it.

3.2 Mutation-induced baseline mismatch

For a mutant pool $\mathcal{M}$ partitioned by reward-blind structure into $G_1, \ldots, G_{n_G}$, let $b_g = |G_g|^{-1} \sum_{i \in G_g} R_i$. A global baseline can reverse local improvement whenever

$$R_i - b_g > 0 \quad \text{and} \quad R_i - b_{\text{global}} < 0.$$

This mismatch is specific to heterogeneous pooled data; it does not imply that global-baseline on-policy RL generally collapses. Our controls therefore separate the effect of mutation from the effect of the baseline.

Direct sign-reversal diagnostics. For each mutant we define the directional failure indicator

$$\chi_i^{+\to-} = \mathbb{I}[R_i - b_{g(i)} > 0 \ \wedge \ R_i - b_{\text{global}} < 0], \tag{2}$$

and report both its frequency and its local-advantage mass,

$$\text{FlipRate} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \chi_i^{+\to-}, \qquad \text{FlipMass} = \frac{\sum_i |R_i - b_{g(i)}| \chi_i^{+\to-}}{\sum_i |R_i - b_{g(i)}|}, \tag{3}$$

We also report the opposite-direction rate $\mathbb{I}[R_i - b_{g(i)} < 0 \wedge R_i - b_{\text{global}} > 0]$ so that the audit does not presuppose an asymmetric outcome. These quantities are descriptive mechanism diagnostics; the fixed-batch intervention in Section 5.3 tests whether the sign changes alter the actual update direction.

4 Mutant Group Policy Optimization

4.1 Grouping and structure-local advantages

For each prompt, the rollout policy produces $n_A = 8$ anchors. The default Batch-Cluster QD backend runs seven QD generations, each contributing $E = 8$ structurally evaluated mutants, for a total of $n_M = 56$ structurally evaluated mutants. Balanced reward-blind k-medoids partitions the mutants into $n_G = 3$ groups, and $n_R = 8$ representatives per group are retained, yielding $8 + 24 = 32$ training sequences. Crucially, the baseline is computed on the full cluster before representative selection:

$$b_g = \frac{1}{|G_g|} \sum_{j \in G_g} R_j, \qquad A_i^M = \frac{R_i - b_g}{s_M}, \quad s_M = \text{Std}_{j \in \mathcal{M}}(R_j). \tag{4}$$

Using the full cluster prevents the reward-stratified selector from moving the reference. A shared s_M keeps advantage magnitudes comparable across groups. Figure 1 summarizes the data flow; selection details are in Appendix G.

Exact reward accounting. In the default MGPO learning rule, every member of the $n_M = 56$ mutant pool that enters Eq. equation 4 receives the same full structural reward R used for training. Cheap token-space or proxy scores may be used to construct or prune an unevaluated proposal pool, but they are not substituted for R_i when computing b_g, s_M, representative quality, or mutant advantages. An approximate survivor-only full-reward mode is evaluated separately in Appendix F.2.

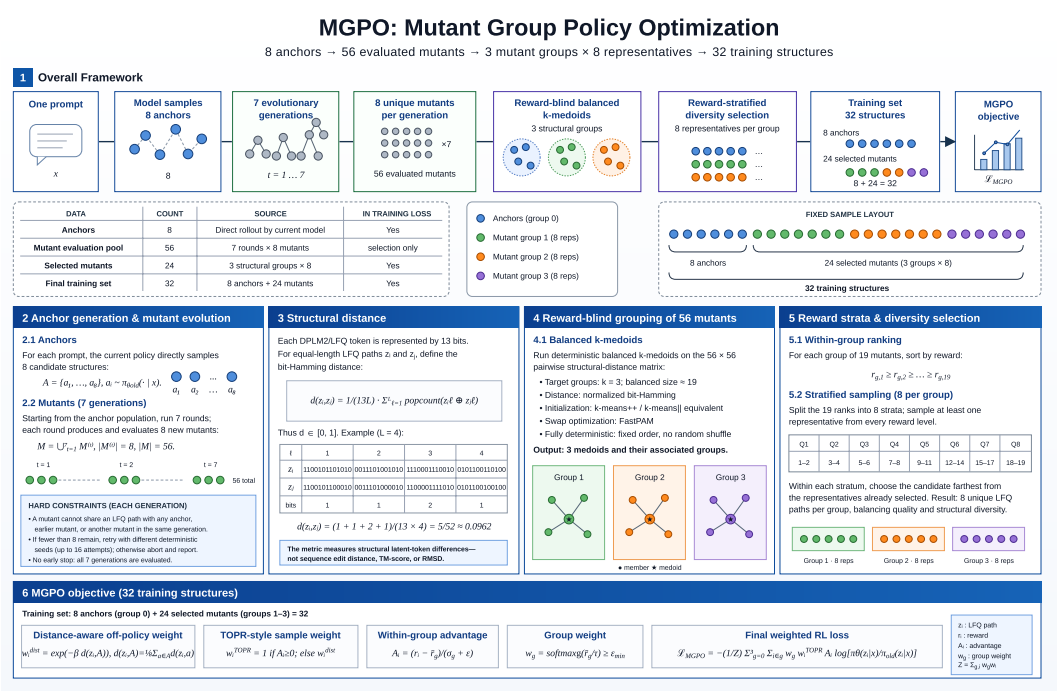

Figure 1: MGPO Framework. Eight rollout anchors seed 56 evaluated mutants; reward-blind structural grouping is followed by representative selection and structure-local mutant credit assignment. The key causal control keeps the entire layout fixed and replaces only b_g with b_{global}.

4.2 Secondary weighting and distance stabilization

The full implementation allocates every retained group nonzero learning mass,

$$W_g = w_{\min} + (1 - w_{\min})(n_G + 1)p_g, \tag{5}$$

where p_g is a numerically stable softmax over group means (Appendix G.4). The distance heuristic uses the mutant's LFQ shift δ_i from its root/current-policy anchors to construct $q_i = \exp(-\kappa\delta_i)$. Only negative mutant advantages are attenuated:

$$\omega_i = \begin{cases} 1, & A_i^M \geq 0, \\ q_i^a \exp\{b(1 - q_i)\}, & A_i^M < 0. \end{cases} \tag{6}$$

We use $a = b = 1$. More generally, for $0 < b \leq a$ and $q \in (0, 1]$, $\log \omega^- = a \log q + b(1 - q) \leq (b - a)(1 - q) \leq 0$, so the rule can only shrink, not amplify, a negative update. This does not remove off-policy bias: q_i is a structural-shift score rather than a probability ratio. We therefore treat this component as an auxiliary stabilization heuristic rather than as a theoretical off-policy correction.

4.3 Training objective

Anchors use the rollout-based TOPR sequence weight (Le Roux et al., 2025); mutants use the local advantage, group weight, and optional attenuation above. With $\bar{\ell}_\theta(y \mid x) = T_y^{-1} \sum_t \log \pi_\theta(y_t \mid x, y_{<t})$,

$$\ell_a = -W_0 \omega_a^{\text{TOPR}} A_a \bar{\ell}_\theta(a \mid x), \qquad \ell_i = -W_{g(i)} \omega_i A_i^M \bar{\ell}_\theta(i \mid x),$$

$$\mathcal{L}_{\text{MGPO}} = \frac{\sum_{a \in \mathcal{A}} \ell_a + \sum_{g=1}^{n_G} \sum_{i \in S_g} \ell_i}{n_A + n_G n_R} + \beta \mathcal{L}_{\text{KL}}^{\text{anchor}}. \tag{7}$$

Mutant likelihoods are computed from the mutant paths themselves rather than from the anchor paths.

Table 1: Evolutionary backend strategies. Batch-Cluster QD is the default; the downstream MGPO grouping and loss are unchanged.

Backend	Varied factor	Proposal and survivor mechanism
Standard GA	Fitness selection	CRF, LFQ, and random proposals; global elitist survivor selection.
Sequential QD	Local competition	Nearest-resident replacement plus a bounded novelty archive.
Batch-Cluster QD	Proposal compression	A cheap pool is reduced by k-medoids; survivors combine quality and max–min distance slots.
CAMP	Mutation location	Separated disagreement centers receive paired local and broad perturbations.
QDPP	Set diversity	Quality-weighted DPP selection for cheap proposals and survivors.
ISLE	Population topology	Four lineage islands use neighbor crossover and retain quality/diversity specialists.
AOPS	Operator schedule	Mutation operators are selected by UCB from past quality–novelty utility.
ELEX	Many-case selection	ϵ-lexicase filters reward components and novelty cases.
MAPX	Structured archive	MAP-Elites cells index distance from the initial best and bit-density shift.

4.4 Evolutionary backend strategies

The evolutionary backend is a modular search component: it supplies the 56 evaluated mutants and is otherwise independent of the credit-assignment mechanism. We implement nine mutation and survivor strategies that differ only in their proposal and survivor mechanism; the downstream grouping, weighting, and loss of Section 4.1 are identical. Batch-Cluster QD is the default. All nine are matched in expensive structural evaluations (not necessarily in total CPU or proposal compute), so the comparisons below attribute differences to the search mechanism rather than to a larger structure budget. Table 1 summarizes the nine strategies and the single factor each one varies; full specifications are in Appendix E.

5 Experiments

5.1 Setup, metrics, and causal controls

The native-reference test set is the CAMEO22 benchmark (Haas et al., 2018), a held-out set of single-chain targets. RL uses the SFT initialization above. End-to-end evaluation samples $K = 8$ completions directly from the final policy; no mutation or reward-based selection is used at test time.

Metrics. We report two quality metrics judged against the native backbone—lDDT-CA and TM-score—and two population-diversity metrics. The LFQ diversity is the mean normalized bit-Hamming distance over all pairs of the $K = 8$ sampled structures,

$$\text{LFQ } \bar{d} = \frac{2}{K(K-1)} \sum_{1 \le i < j \le K} d(\mathbf{t}_i, \mathbf{t}_j), \tag{8}$$

with d defined in Eq. equation 1; larger $\bar{d}$ means the sampled structures are farther apart in token space. The decoded 3D diversity is the analogous mean pairwise 1−TM-score,

$$\text{3D div} = \frac{2}{K(K-1)} \sum_{1 \le i < j \le K} \big(1 - \text{TM}(\mathbf{t}_i, \mathbf{t}_j)\big), \tag{9}$$

where $\text{TM}(\cdot, \cdot)$ is the TM-score between two decoded structures; it is reported alongside $\bar{d}$ so that conclusions do not rely on LFQ geometry alone. Finally, collapse is defined per target: for target j, let $\bar{d}_j^{\text{init}}$ be the LFQ diversity of the SFT initialization and $\bar{d}_j^{\text{final}}$ that of the trained policy under matched sampling; target j collapses if $\bar{d}_j^{\text{final}} < 0.5\bar{d}_j^{\text{init}}$, and the collapse rate is the mean of this indicator over targets.

An online control we call Same-layout global matches full MGPO in the number of anchors and evaluated mutants, reward-blind grouping algorithm, representative-selection rule, group weighting, distance rule, and 32-sequence training layout, but replaces every mutant b_g by one global mutant baseline $b_{\text{global}} = |\mathcal{M}|^{-1} \sum_{i \in \mathcal{M}} R_i$. Independent online runs are

Table 2: Policy-only endpoint results on CAMEO22. "Same-layout global" matches the MGPO algorithmic layout but uses a global mutant baseline. All methods are evaluated from $K = 8$ policy-only samples with the same frozen structural decoder and endpoint protocol. "Coll." denotes the collapse rate defined in Section 5.1.

Method	Training signal	lDDT avg@8 ↑	TM avg@8 ↑	LFQ $\bar{d}$ ↑	3D div. ↑	Coll. ↓
SFT initialization	no RL	0.7028	0.7061	0.0400	0.1280	–
GRPO-8	8 on-policy	0.7246	0.7370	0.0388	0.1292	21.7%
GRPO-64	64 on-policy	0.7282	0.7410	0.0393	0.1310	20.4%
Mutant-GRPO	global, 8+56 pool	0.7138	0.7264	0.0345	0.1180	27.8%
Same-layout global	global, MGPO layout	0.7145	0.7270	0.0352	0.1205	25.6%
MGPO-local-only	local, no aux. weights	0.7391	0.7496	0.0458	0.1452	10.9%
MGPO (full)	local + aux. terms	0.7427	0.7532	0.0486	0.1518	5.8%
Mutant-GRPO+DivReward	explicit diversity reward	0.7286	0.7417	0.0436	0.1380	17.2%
ProteinZero-style	diversity regularizer	0.7321	0.7446	0.0444	0.1412	14.8%
SGRPO-style	set-diversity reward	0.7360	0.7479	0.0461	0.1460	11.7%
Mutant-AWR	reward-weighted imitation	0.7334	0.7442	0.0418	0.1350	16.9%

initialized from the same SFT checkpoint and use matched training schedules and seed sets where possible, but the realized pools are allowed to diverge after the first different update. The exact-sample causal comparison is therefore performed separately by the fixed-batch intervention in Section 5.3.

5.2 Main quality–diversity results

Table 2 shows that structure-local credit assignment is the main source of MGPO's gain. Under the matched Same-layout global control, replacing the global mutant baseline with structure-local credit improves both quality metrics and both diversity metrics, while substantially reducing collapse. The local-only variant already recovers most of the improvement, indicating that the local baseline is the dominant factor, with group weighting and distance attenuation providing smaller secondary gains. MGPO also outperforms the on-policy, diversity-aware, and reward-weighted imitation baselines, supporting the claim that its benefit comes from preserving useful learning signals within heterogeneous structural neighborhoods rather than from additional sampling or explicit diversity shaping.

5.3 Mechanism audit: does the global baseline reverse useful updates?

The endpoint comparison in Table 2 is complemented by a mechanism audit on saved mutant pools. At every logged training checkpoint we compute the directional FlipRate and FlipMass from Eqs. equation 2–equation 3, the opposite-direction flip rate, and the fraction of selected representatives affected by each event. We report early/middle/late training thirds separately so that an endpoint average cannot hide a transient failure mode.

Paired fixed-batch intervention. For a stratified set of checkpoints, we snapshot the same model parameters and optimizer state and construct one shared batch containing the same anchors, fully scored mutants, reward-blind groups, and representatives. Two model copies receive exactly one optimizer step: one with b_g and one with b_{global}; every other loss term and random seed is fixed. Let g_{local} and g_{global} denote the mutant-branch gradients before the optimizer transformation. We report their cosine similarity, norm ratio, and the mean post-step change in log probability of the $+\to-$ sign-flipped mutants. Because both copies use identical model parameters, optimizer state, samples, grouping, representatives, loss terms, and random-number state, the intervention isolates the effect of replacing b_g with b_{global} on the immediate update direction.

Table 3 shows that sign-flipped mutants are not a negligible fraction of the pool, that their local-advantage mass grows over training, and that the local and global mutant-branch gradients become increasingly misaligned while the global step actively decreases the log probability of the flipped mutants.

Table 3: Mechanism audit from saved pools and fixed-batch interventions. Values are averaged by target first and then summarized over early, middle, and late training thirds.

Diagnostic	Early	Middle	Late
$+\to-$ FlipRate (%)	18.6	22.4	25.7
$+\to-$ FlipMass (%)	13.2	17.9	21.6
$-\to+$ opposite FlipRate (%)	7.4	5.8	4.1
Selected representatives flipped (%)	20.1	24.8	28.3
$\cos(g_{\text{local}}, g_{\text{global}})$	0.78	0.66	0.54
$\|g_{\text{global}}\|/\|g_{\text{local}}\|$	0.94	0.89	0.84
$\Delta \log \pi$ on $+\to-$ mutants: local step	+0.021	+0.028	+0.031
$\Delta \log \pi$ on $+\to-$ mutants: global step	-0.014	-0.022	-0.027

Table 4: Pre-specified locality sensitivity. The selected mutant count is held fixed so the policy-update batch size is constant.

n_G	n_R	lDDT avg@8 ↑	TM avg@8 ↑	LFQ $\bar{d}$ ↑	3D div. ↑	FlipRate
1	24	0.7145	0.7270	0.0352	0.1205	0.0
2	12	0.7348	0.7457	0.0397	0.1332	14.6
3	8	0.7427	0.7532	0.0486	0.1518	22.4
4	6	0.7419	0.7523	0.0405	0.1376	24.8
6	4	0.7394	0.7502	0.0402	0.1366	27.1

5.4 Grouping granularity and mutation-budget sensitivity

The central hypothesis requires structure-local neighborhoods, so the default $n_G = 3$ should not be treated as an unexamined constant. We vary $n_G \in \{1, 2, 3, 4, 6\}$ while holding the number of selected mutant training sequences fixed by setting $n_R = 24/n_G$, so every setting still uses 32 training sequences including the anchors. The $n_G = 1$ condition is the natural no-locality endpoint within this granularity sweep: it has one mutant group and therefore one mutant baseline. Its endpoint metrics numerically coincide with the Same-layout global control in Table 2, as expected when every mutant shares a single reward reference; the two settings differ in how that reference is reached (single-group k-medoids here versus a global pooled mean under $n_G = 3$ grouping there) rather than in the resulting baseline.

Table 4 shows that both quality and diversity peak near the default $n_G = 3$, while FlipRate increases monotonically with n_G. The gain therefore tracks the amount of baseline heterogeneity rather than the mere presence of clustering.

5.5 Search-based imitation controls

Because mutation produces samples without a tractable behavior density, we include a reward-weighted self-training control to distinguish search-output imitation from signed local credit assignment. This control uses exactly the same anchors, mutant pool, grouping, and selected mutant representatives but removes signed local advantages. For Mutant-AWR, selected mutants receive normalized positive weights

$$\alpha_i = \frac{\exp((R_i - \bar{R}_{\mathcal{M}})/\tau)}{\sum_{j \in S} \exp((R_j - \bar{R}_{\mathcal{M}})/\tau)}, \qquad \mathcal{L}_{\text{AWR}} = -\sum_{i \in S} \alpha_i \bar{\ell}_\theta(i \mid x), \tag{10}$$

with temperature τ selected once on validation and then frozen. The anchor branch, KL term, structural evaluation budget, and training layout are unchanged. Table 2 shows that MGPO retains a clear advantage over this control, so its benefit is harder to explain as merely distilling high-reward search outputs.

5.6 Geometry visualization and remaining audits

To inspect whether the diversity statistic corresponds to visibly different search paths, we project one complete target trace (target 13) into a shared two-dimensional PCA basis

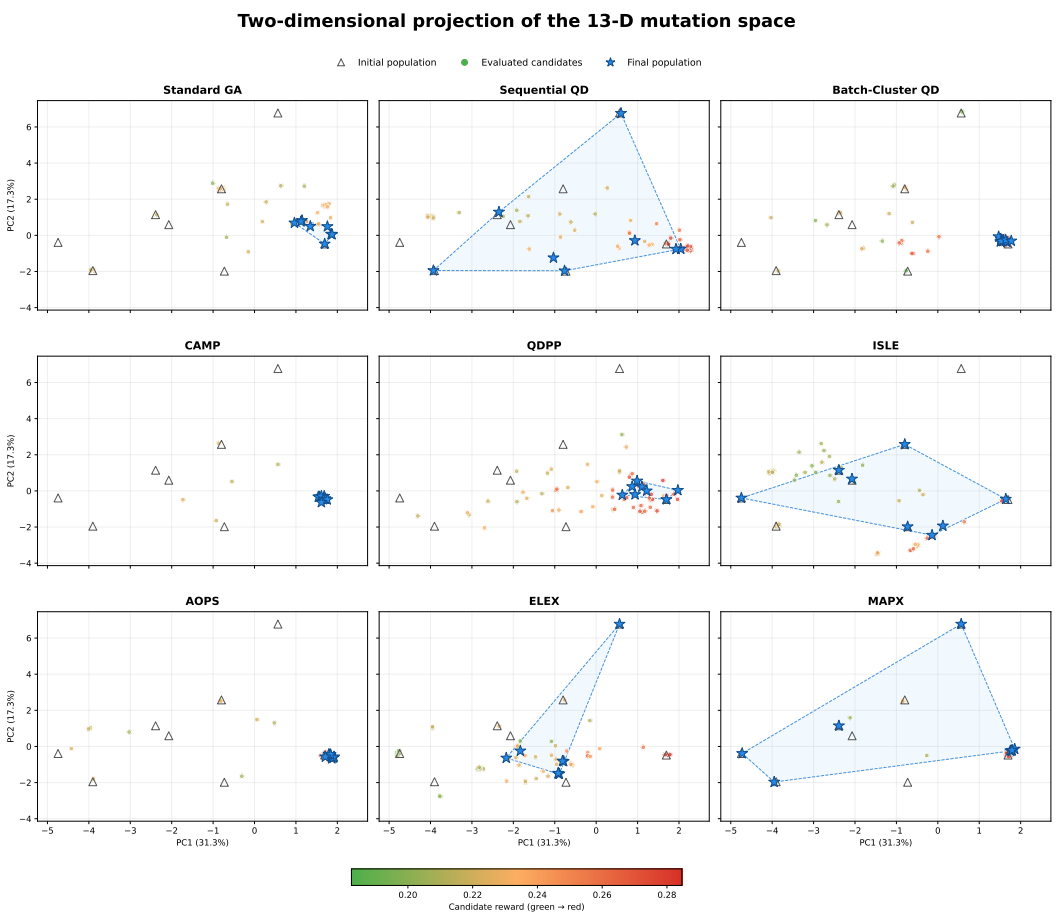

Figure 2: Shared-PCA view of a complete 64-evaluation trace. Features are per-position mean occupancies of the 13 latent token bits. Triangles denote the initial population, circles decoded candidates, and stars the final population; all nine panels use the same PCA basis and axis limits. Candidate color encodes lDDT-CA.

(Figure 2). The panel is illustrative; the quantitative conclusions rest on the metrics in Tables 2–3. Each backend contributes 8 initial anchors, 64 decoded candidate points, and 8 final survivors; each point is represented by its 13-dimensional mean bit-occupancy vector (the average over residues of the 13 LFQ bits, a different feature space from the $13L$-bit flattened matrix used for r_{eff}). The first two components explain 31.3% and 17.3% of the pooled variance, respectively; the projection is therefore used as a visualization and not as a replacement for the full Hamming metric. Standard GA's final stars occupy a compact region, consistent with elitist convergence. Sequential QD and ISLE preserve broader regions, while QDPP retains a high-reward final set without collapsing to a single cluster.

The remaining geometry, proxy, and backend audits are reported in Appendix D. Appendix D reports the within/between-group decoded distances and the LFQ–3D adjusted Rand indices; those results support interpreting reward-blind LFQ groups as algorithmic structural neighborhoods rather than recovered biological states. Appendix D.1 reports the nine mutation-backend diagnostics, including their quality–diversity trade-off. Appendix F reports the statistics and efficiency accounting, including the correctness-preserving acceleration of the exact MGPO protocol and the approximate proxy-reward engineering variant.

Statistical reporting. The statistical unit for endpoint evaluation is the held-out target. Tables 2 and 4 report aggregate means over the same CAMEO22 benchmark under matched $K = 8$ policy-only evaluation. The mechanism audit first aggregates diagnostics within

target and then summarizes them over early, middle, and late training. The primary causal evidence comes from the large same-layout contrast and the exactly matched fixed-batch intervention; smaller component-level differences are reported descriptively and are not assigned statistical-significance claims.

6 Conclusion

We introduced Mutant Group Policy Optimization (MGPO), a mutation-aware policy-optimization framework that couples structured evolutionary exploration with neighborhood-aware credit assignment for protein structure generation. MGPO generates structurally diverse mutants through reward-blind mutation and quality-diversity search, groups them using reward-independent structural geometry, and computes each mutant's advantage within its complete structural neighborhood before representative compression. Because the baseline is computed on the full cluster prior to compression, MGPO preserves learning signals from mutants that improve within their local neighborhoods but would be suppressed by global population statistics. Same-layout comparisons and local ablations show that structure-local credit assignment is the primary source of improvement, while evolutionary search supplies the diverse structural neighborhoods on which this mechanism operates, and group weighting and distance terms serve as auxiliary mechanisms for stabilization and compression. Overall, this work makes two main contributions: it systematically introduces mutation and evolutionary search into reinforcement-learning fine-tuning for protein structure generation and compares nine mutation/evolutionary strategies; and it identifies the sign-failure between local and global advantages in mutation-augmented training and proposes structure-local credit assignment as its remedy. MGPO thereby improves the exploration effectiveness, structural diversity, and prediction quality of protein structure models, offering a mutation-aware policy-optimization framework for protein structure generation.

AI Use Statement

Generative AI tools were used to assist with manuscript editing, scientific figure preparation, code editing, and to provide feedback on experimental design and interpretation of experimental results. All AI-assisted methodological suggestions, code modifications, and result interpretations were reviewed and verified by the authors. All reported experimental results were produced by the authors' experimental pipeline, and the authors take full responsibility for the content, claims, code, and experimental results in this work.

Ethics Statement

This work studies conditional protein-structure generation for sequences provided as input and does not introduce a de novo amino-acid sequence-design pipeline. Generated structures are computational hypotheses rather than experimental validation. The method inherits the dual-use profile of protein modeling, and any downstream practical use should apply the usual structural, functional, and experimental verification procedures.

Reproducibility Statement

The main text and appendices specify the MGPO objective, reward components, evolutionary backends, sampling and structural-evaluation budgets, key training hyperparameters, causal controls, and systems implementation used in the reported experiments.

References

Josh Abramson, Jonas Adler, Jack Dunger, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. Nature, 630:493–500, 2024. doi: 10.1038/s41586-024-07487-w.

Álvaro Fialho, Marc Schoenauer, and Michèle Sebag. Analysis of adaptive operator selection techniques on the royal road and long k-path problems. In Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 779–786, 2009. doi: 10.1145/1569901.1570009.

Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. IEEE Transactions on Evolutionary Computation, 22(2):245–259, 2018. doi: 10.1109/TEVC.2017.2704781.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

Jürgen Haas, Alessandro Barbato, Dario Behringer, Gabriel Studer, Steven Roth, Martino Bertoni, Khaled Mostaguir, Rafal Gumienny, and Torsten Schwede. Continuous automated model evaluation (cameo) complementing the critical assessment of structure prediction in casp12. Proteins: Structure, Function, and Bioinformatics, 86(S1):387–398, 2018. doi: https://doi.org/10.1002/prot.25431. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25431.

Thomas Hayes, Roshan Rao, Halil Akin, et al. Simulating 500 million years of evolution with a language model. bioRxiv, 2024. doi: 10.1101/2024.07.01.600583.

John H. Holland. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI, 1975.

John Jumper, Richard Evans, Alexander Pritzel, et al. Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Leonard Kaufman and Peter J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 1990. doi: 10.1002/9780470316801.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. Foundations and Trends in Machine Learning, 5(2–3):123–286, 2012. doi: 10.1561/2200000044.

William La Cava, Lee Spector, and Kourosh Danai. Epsilon-lexicase selection for regression. In Proceedings of the 2016 Annual Conference on Genetic and Evolutionary Computation, pp. 741–748, 2016. doi: 10.1145/2908812.2908898.

Nicolas Le Roux, Marc G. Bellemare, Jonathan Lebensold, et al. Tapered off-policy REINFORCE: Stable and efficient reinforcement learning for LLMs. arXiv preprint arXiv:2503.14286, 2025.

Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. Evolutionary Computation, 19(2):189–223, 2011. doi: 10.1162/EVCO_a_00025.

Zeming Lin, Halil Akin, Roshan Rao, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. Science, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. arXiv preprint arXiv:1504.04909, 2015.

Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality diversity: A new frontier for evolutionary computation. Frontiers in Robotics and AI, 3:40, 2016. doi: 10.3389/frobt.2016.00040.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

Guobin Shen, Chenxiao Zhao, Xiang Cheng, Lei Huang, and Xing Yu. VESPO: Variational sequence-level soft policy optimization for stable off-policy LLM training. arXiv preprint arXiv:2602.10693, 2026.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, et al. AlphaFold protein structure database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Research, 50(D1):D439–D444, 2022. doi: 10.1093/nar/gkab1061.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. DPLM-2: A multimodal diffusion protein language model. In International Conference on Learning Representations, 2025a.

Yuyang Wang, Jiarui Lu, Navdeep Jaitly, Josh Susskind, and Miguel Angel Bautista. Simplefold: Folding proteins is simpler than you think, 2025b. URL https://arxiv.org/abs/2509.18480.

Ziwen Wang, Jiajun Fan, Ruihan Guo, Thao Nguyen, Heng Ji, and Ge Liu. ProteinZero: Self-improving protein generation via online reinforcement learning. arXiv preprint arXiv:2506.07459, 2025c.

Darrell Whitley, Soraya Rana, and Robert B. Heckendorn. The island model genetic algorithm: On separability, population size and convergence. Journal of Computing and Information Technology, 7(1):33–47, 1999.

Xinwu Ye, He Cao, Hao Li, Bin Feng, Zijing Liu, Xiangru Tang, Yu Li, and Shenghua Gao. Pushing biomolecular utility-diversity frontiers with supergroup relative policy optimization. arXiv preprint arXiv:2605.08659, 2026.

Lijun Yu, José Lezama, Nitesh B. Gundavarapu, et al. Language model beats diffusion—tokenizer is key to visual generation. arXiv preprint arXiv:2310.05737, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yue Yu, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In Advances in Neural Information Processing Systems, 2025. arXiv:2503.14476.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. arXiv preprint arXiv:2507.18071, 2025.

A Notation summary

Table 5: Notation used throughout the paper.

Symbol	Meaning
x, y	Prompt and generated completion
L, t_i, $\mathbf{t}(y)$	Structure length, token i, full structure path
$d(\cdot,\cdot)$	Normalized LFQ bit-Hamming distance
z_i, b_{ik}, m_i	Token ID, its k-th bit, XOR mutation mask
$\mathcal{A}$, n_A	Anchor set and its size
$\mathcal{M}$, n_M	Mutant pool and its size
G_g, n_G	Structural group g and number of groups
S_g, n_R	Selected representatives in group g and their count
$R(x,y)$, R_r, ω_r	Total reward, reward component r, its weight
b_g, s_M	Group baseline and mutant-pool scale
A_i^M, A_a	Mutant and anchor advantages
μ_g, s_μ, p_g	Group mean, between-group std, softmax weight
W_g, $w_{\min}$	Group weight and its floor
δ_i, κ, q_i	Distance proxy, calibration, pseudo-ratio
ω_i, ω_i^-	Mutant weight and negative-advantage attenuation
K, E, Q, P_g	Population size, evaluated children, cheap pool, population
π_θ, π_{ref}, $J(\theta)$	Policy, reference policy, KL-regularized objective
$\mathcal{D}$	Prompt distribution over single-chain sequence targets
ρ_a, ω_a^{TOPR}	Rollout ratio and TOPR weight
$\bar{\ell}_\theta$, $\mathcal{L}_{\text{MGPO}}$, β	Mean log-prob, MGPO loss, KL coefficient

B Baseline methods

All endpoint comparisons use the same SFT initialization and the same 183 held-out targets. Unless stated otherwise, evaluation is policy-only with $K = 8$ samples per target and uses the same frozen DPLM-2 structural decoder. The baselines are organized to separate ordinary on-policy RL, mutation with a global reference, diversity-aware alternatives, and search-output imitation.

SFT initialization. This is the RL cold-start model before reinforcement learning and serves as the zero-step reference for all reported RL deltas.

GRPO-8. GRPO-8 uses eight on-policy samples per prompt and applies the standard group-relative on-policy update without evolutionary mutants. It measures the gain obtainable from ordinary small-group RL.

GRPO-64. GRPO-64 uses 64 on-policy samples per prompt. Its 64 structurally evaluated policy samples match the total number of expensive structural evaluations in the default MGPO data-generation step (8 anchors plus 56 mutants), providing a larger-budget on-policy control.

Mutant-GRPO. Mutant-GRPO trains on the 8-anchor plus 56-mutant pool but computes mutant advantages relative to one pooled global mutant baseline. It therefore tests whether evolutionary mutation alone is sufficient when heterogeneous mutant families share a single reward reference.

Same-layout global. Same-layout global is the closest causal control for MGPO. It preserves the anchor count, 56-mutant pool, reward-blind grouping, representative-selection pipeline, group weighting, distance attenuation, and 32-sequence update layout, but replaces every group baseline b_g with one global mutant baseline b_{global}.

MGPO-local-only. MGPO-local-only retains reward-blind structural grouping and the same representative-selection pipeline, but removes the auxiliary group and distance weights by setting $W_g = 1$ and $\omega_i = 1$. Its mutant learning signal is therefore determined by the structure-local baseline itself.

Mutant-GRPO+DivReward. This diversity-aware control augments the mutant-GRPO scalar reward with a per-sample diversity contribution; the diversity coefficient is selected on validation. It tests whether explicit diversity reward shaping can replace the change in credit-assignment reference used by MGPO.

ProteinZero-style. This is our adaptation of the pairwise diversity-regularization strategy used in ProteinZero (Wang et al., 2025c). It keeps the same initialization, prompt stream, training horizon, and policy-only endpoint protocol while adding an explicit pairwise diversity regularizer rather than changing the mutant baseline.

SGRPO-style. This is our adaptation of the set-diversity objective in SGRPO (Ye et al., 2026). The baseline explicitly rewards diversity at the sample-set level and is evaluated under the same endpoint protocol as MGPO.

Mutant-AWR. Mutant-AWR uses the same anchors, 56-mutant pool, structural groups, and selected 24 mutant representatives as MGPO, but replaces signed local advantages with positive normalized reward weights. Its mutant objective is given in Eq. equation 10, with τ selected from $\{0.1, 0.2, 0.5, 1.0\}$ on validation. This control tests whether MGPO can be explained simply as reward-weighted imitation of search outputs.

Full MGPO. The full method combines structure-local mutant advantages with group weighting and negative-advantage distance attenuation as defined in Eqs. equation 4–equation 6. It is included here only to make the design differences to each baseline explicit.

C Experimental protocol and audit details

C.1 Data split

Held-out targets are separated from the training data at the sequence-cluster level: near-duplicate members of an existing sequence cluster are kept within a single split.

C.2 Component ablations

Table 6 reports the one-at-a-time component ablations. The distance heuristic and representative compression have the smallest point-estimate effects among the ablated components.

Table 6: One-at-a-time MGPO component ablations.

Variant	lDDT avg@8 ↑	TM avg@8 ↑	LFQ $\bar{d}$ ↑	3D div. ↑
Full MGPO	0.7427	0.7532	0.0486	0.1518
− group-local advantage	0.7145	0.7270	0.0352	0.1205
− reward-blind grouping	0.7312	0.7410	0.0396	0.1330
− nonzero-mass floor	0.7350	0.7440	0.0412	0.1372
− stratified selection	0.7385	0.7480	0.0448	0.1420
− distance heuristic	0.7400	0.7510	0.0472	0.1478
− 56-to-24 compression	0.7410	0.7520	0.0479	0.1495

C.3 Fixed-batch intervention: implementation details

The protocol is specified in Section 5.3; this appendix records the implementation-level details. Checkpoints are chosen before inspecting the sign-flip outcome and sampled across early, middle, and late training. For every checkpoint and target batch we save the complete

$8 + 56$ scored pool and the selected 24 representatives. Two copies of the same model and optimizer state receive one update with identical random-number states; the only changed tensor is the mutant baseline. Before the optimizer step we save the flattened mutant-branch gradients to compute cosine similarity and norm ratio. After the step we re-score log probabilities of the same samples.

D Geometry, proxy, and backend audits

Within- versus between-group geometry. Reward-blind LFQ groups should be interpreted as algorithmic neighborhoods, not automatically as biological states. On saved mutant pools, the mean within/between-group decoded distances are 0.112/0.284 on validation and 0.118/0.271 on CAMEO22, with LFQ–3D adjusted Rand indices of 0.610 and 0.580. The substantially larger between-group than within-group decoded distances, together with the moderate LFQ–3D adjusted Rand indices, indicate that the reward-blind LFQ partition captures meaningful 3D geometry while not being equivalent to a decoded-structure clustering. We therefore interpret the groups as algorithmic structural neighborhoods rather than as recovered biological conformational states.

Proxy validity. The LFQ shift proxy has only moderate association with old-policy rarity: Spearman $\rho(\delta, \Delta\text{NLL}) = 0.46$ on validation and 0.43 on CAMEO22 pools. Removing the proxy changes lDDT avg@8 from 0.7427 to 0.7400 in the ablation. This is why the paper treats distance attenuation as a stabilization heuristic rather than as a theoretical correction.

D.1 Backend diagnostics

Before end-to-end RL, we characterize the nine candidate backends of Table 1 at the search level. These are backend diagnostics, not comparisons of nine independent training objectives: the downstream MGPO grouping and loss are held fixed. All aggregates use single-chain targets, population size $K = 8$, and the same lDDT-CA evaluator. Figure 3 plots each backend's final mean and final best lDDT-CA against mean pairwise token diversity.

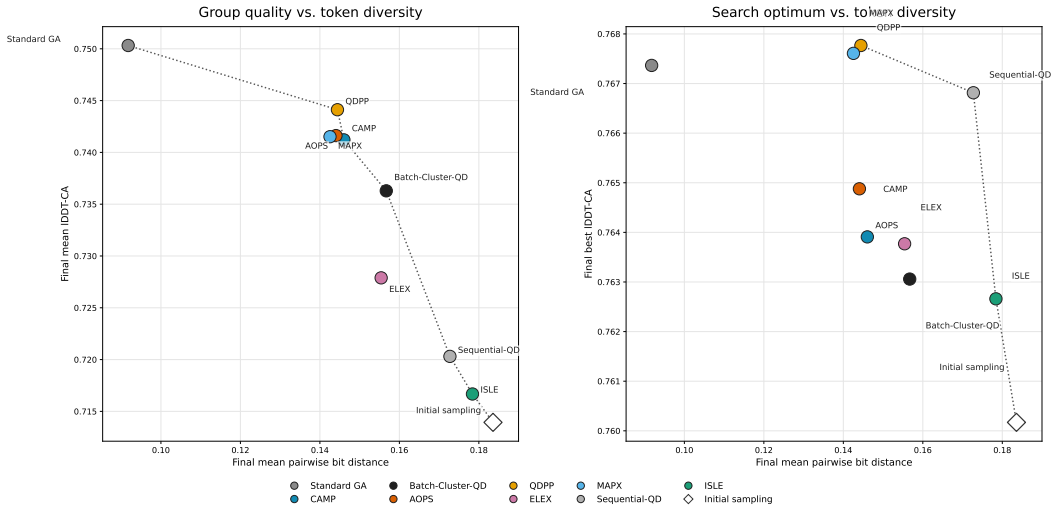

Figure 3: Low-budget mutation-backend ablation over the CAMEO22 benchmark. Each point is a backend-level aggregate after eight decoded child evaluations per target; the initial anchor is the diamond. Left: final population mean lDDT-CA; right: final population best lDDT-CA. Dotted lines connect the non-dominated frontier.

Standard GA has the highest final mean in this scan, but its LFQ diversity is only marginally above the initial anchor value and its decoded 3D diversity is below the SFT reference in Table 2. QDPP reaches the highest final best score while retaining LFQ diversity on the

corrected scale, illustrating why best-of-group and population-average quality should not be reported without a diversity axis. Sequential QD and ISLE retain still more diversity, but at a larger mean-quality cost.

Interpretation boundary. The native-reference reward and the diversity metrics above establish only that the generated population is structurally broader under the stated quality objective. They do not establish recovery of biologically valid alternative apo/holo or metastable states. Accordingly, we restrict our claim to structural population diversity under the stated quality objective and do not interpret this evidence as recovery of biologically valid alternative conformational states.

E Evolutionary backend strategies

This section provides full specifications for the nine mutation and survivor strategies summarized in Table 1. Eight of them adapt established algorithms from evolutionary computation (Holland, 1975; Lehman & Stanley, 2011; Pugh et al., 2016; Cully & Demiris, 2018; Kaufman & Rousseeuw, 1990; Kulesza & Taskar, 2012; Whitley et al., 1999; Álvaro Fialho et al., 2009; La Cava et al., 2016; Mouret & Clune, 2015), while CAMP is a method-specific combination introduced in this work.

E.1 Standard GA

The control follows the canonical genetic-algorithm pattern of variation and fitness-based survival (Holland, 1975). Each generation mixes three reward-blind proposal families: a CRF path assembled from group-level transition statistics, an LFQ consensus/Fréchet proposal, and unique one-bit jumps from distinct parents. After scoring, parents and children are sorted by R, with deterministic generation order breaking ties, and the top K survive. Thus Standard GA applies no explicit pressure for novelty; it measures the quality gain and diversity collapse produced by pure elitism.

E.2 Sequential QD

Sequential QD combines novelty search (Lehman & Stanley, 2011) with the quality-diversity principle of optimizing within behavioral niches (Pugh et al., 2016; Cully & Demiris, 2018). For archive $\mathcal{H}$, we define the novelty of z as the mean distance to its k_n nearest compatible archive members,

$$n(z;\mathcal{H}) = \frac{1}{k'} \sum_{h \in \mathrm{kNN}_{k_n}(z,\mathcal{H})} d(z,h), \qquad k' = \min(k_n, |\mathcal{H}|), \tag{11}$$

with $k_n = 3$. Children are processed in deterministic proposal order. A child c competes against its nearest current resident $r(c)$; the sole population best is protected by redirecting competition to the second-nearest resident. The child replaces $r(c)$ if it is unique and either

$$R(c) > R(r(c)), \quad \text{or} \quad n(c) \geq n(r(c)) + \delta_n \ \wedge \ R(c) \geq \max(R^0_{\min}, R^g_{\max} - \delta_q). \tag{12}$$

Every sufficiently novel probe may enter a bounded historical archive even when it loses local competition. When the archive exceeds its capacity, the most crowded entry is removed.

E.3 Batch-Cluster QD

Batch-Cluster QD asks whether broad mutation can be made evaluation-efficient. It first generates $Q = 64$ unique paths using crossover, LFQ-guided mutations, consensus mutations, and occasional wider jumps. Without observing child reward, it reduces this pool to $E = 8$ actual paths by farthest-first k-medoids with two refinement passes (Kaufman & Rousseeuw, 1990). Because a medoid is an actual valid path, no synthetic cluster center must be decoded. Only these medoids receive structural reward evaluations. In the default MGPO configuration, this generation is repeated seven times per prompt, so the seven $E = 8$ evaluated medoid sets accumulate to the $n_M = 56$ mutant pool used in Section 4.1.

The $Q = 64$ cheap pool is regenerated per generation and does not consume structural evaluations.

Survivor selection uses q quality slots (six when $K = 8$), filled by the highest-reward unique paths. Each remaining slot greedily selects

$$z^* = \arg \max_{z:R(z)\geq\phi_g} \min_{s\in S} d(z, s), \tag{13}$$

where S is the partial survivor set. The floor $\phi_g = \max(\bar{R}^0 - \delta_0, R^g_{\max} - \delta_g)$ prevents diversity slots from accumulating arbitrarily poor structures; δ_g is annealed during search. This quality/crowding separation is related to diversity-preserving multi-objective selection (Deb et al., 2002), but uses explicit slots rather than Pareto ranks.

E.4 CAMP

CAMP changes where and how mutations are made. At residue position j, let

$$u_j = 1 - \max_v \frac{1}{K} \sum_{x\in P_g} \mathbb{I}[t_j(x) = v] \tag{14}$$

be token disagreement. We greedily select four positions with largest u_j subject to a separation of at least $\max(1, \lfloor L/10 \rfloor)$ residues. Each center produces one local and one broad child. Local mutation chooses a parent and flips one or two entropy-weighted bits inside radius $\max(1, \lfloor L/32 \rfloor)$; broad mutation first performs uniform bitwise crossover, then flips three to five bits inside radius $\max(1, \lfloor L/10 \rfloor)$. For position j and bit b, the sampling weight is $p_{jb}(1 - p_{jb}) + \epsilon$, so variation targets uncertain group coordinates without reading reward. Survivors use the quality/max–min rule in equation 13. CAMP is a method-specific combination of multi-center disagreement sampling and QD selection.

E.5 QDPP

Determinantal point processes (DPPs) assign high mass to subsets that are jointly high-quality and mutually dissimilar (Kulesza & Taskar, 2012). QDPP generates the same $Q = 64$ cheap pool as Batch-Cluster QD, but assigns proposal i the proxy

$$\widetilde{R}_i = R(p_i) + \lambda_n d(\mathbf{t}_i, \mathbf{t}_{p_i}), \tag{15}$$

where p_i is its nearest parent and $\lambda_n = 0.03$. The proxy uses only already-known parent fitness; child structural reward is not queried. Scores are mapped to positive qualities q_i, and the similarity and DPP kernels are

$$S_{ij} = \exp\left(-\frac{d(\mathbf{t}_i, \mathbf{t}_j)^2}{2\sigma^2}\right), \qquad L_{ij} = q_i S_{ij} q_j, \tag{16}$$

where σ is the median nonzero pairwise distance. Greedy MAP inference selects the E proposals maximizing successive $\log\det L_S$, with the highest-proxy item forced into the set. After true evaluation, the same greedy DPP rule is applied to parents and children using actual R, restricted to an annealed quality band.

E.6 ISLE

Island models preserve multiple evolutionary trajectories by restricting interaction between subpopulations (Whitley et al., 1999). ISLE assigns the $K = 8$ lineages to four two-member islands according to their root-parent index. A child for island r is formed by uniform crossover between its resident lineage and the next island's lineage, followed by a population-guided bit mutation. This neighbor mating is the only information exchange; ISLE does not use unrestricted global mating. Within each island, survivor selection first retains the highest-reward member and then the member farthest from that island best, breaking distance ties by reward. Any unused global slots are filled by reward.

E.7 AOPS

AOPS adapts operator selection online, following bandit-based adaptive operator selection (Álvaro Fialho et al., 2009). Its arms are crossover, LFQ, CRF-guided mutation, wide mutation, and targeted mutation. Untried arms are scheduled first; subsequently arm a is selected by

$$\mathrm{UCB}_a = \bar{U}_a + c_{\mathrm{ucb}}\sqrt{\frac{\log N}{N_a}}, \qquad c_{\mathrm{ucb}} = 0.20, \tag{17}$$

where N_a is its past trial count. After child evaluation, its utility is

$$U(c) = R(c) - R(p_{\mathrm{near}}) + \eta \min_{p \in P_g} d(c, p), \qquad \eta = 0.10. \tag{18}$$

Thus prior reward outcomes choose the next operator, but the chosen primitive does not consult reward when selecting mutation bits. Survivors follow the same quality/max–min rule as CAMP.

E.8 ELEX

Lexicase selection considers cases separately rather than collapsing them into one objective, and ϵ-lexicase admits near-best candidates in continuous spaces (La Cava et al., 2016). ELEX proposes population-guided wide mutations. For survivor selection, the cases are the reward components $\{R_1, \ldots, R_J\}$ available to the trainer plus archive novelty from equation 11; it never uses withheld answers or test labels. For each survivor draw, cases are randomly permuted. Starting with all remaining candidates, case c retains

$$\mathcal{S} \leftarrow \{i \in \mathcal{S} : f_{ic} \geq \max_{j \in \mathcal{S}} f_{jc} - \epsilon_c\}, \qquad \epsilon_c = \max(\epsilon_{\mathrm{lex}}, \mathrm{MAD}_{j \in \mathcal{S}}\, f_{jc}). \tag{19}$$

Filtering stops when one candidate remains or all cases are exhausted; one survivor is sampled and removed, and the procedure repeats until K are selected. The value of ϵ_{lex} is reported in the implementation details rather than fixed by the method definition. ELEX consequently favors specialists that are strong on at least one ordering of reward/diversity cases, rather than only high scalar reward.

E.9 MAPX

MAPX instantiates MAP-Elites (Mouret & Clune, 2015), a canonical quality-diversity archive that stores the best solution in each behavioral cell. Its two descriptors are

$$d_{\mathrm{init}}(\mathbf{t}) = d(\mathbf{t}, \mathbf{t}^0_{\mathrm{best}}), \qquad \Delta\rho(\mathbf{t}) = \rho(\mathbf{t}) - \rho_0, \quad \rho(\mathbf{t}) = \frac{1}{L}\sum_i \frac{\mathrm{popcount}(t_i)}{13}, \tag{20}$$

where $\mathbf{t}^0_{\mathrm{best}}$ is the best anchor and ρ_0 is the initial mean bit density. Distance edges $(0.005, 0.02, 0.06)$ and density-shift edges $(-0.015, 0, 0.015)$ define a 4×4 archive; malformed or length-mismatched paths are quarantined in an overflow cell. Each cell keeps only its highest-reward member. The next population takes cell elites in descending reward order and, if fewer than K cells are occupied, fills the remainder from the best unselected candidates. Mutation radius increases periodically across proposals, encouraging transitions between neighboring and more distant cells.

F Parallel Evolution

The evolutionary backends of Appendix E are specified at the level of proposal and survivor mechanisms, but running them inside the MGPO update adds two per-prompt costs on top of the anchor rollout: (i) mutation construction, which builds the cheap candidate pool, computes the bit-Hamming distance matrix of equation 1, clusters with balanced k-medoids, and draws mutants; and (ii) structural reward evaluation, which decodes each candidate and runs the structure scorers. This appendix records how we parallelize the construction stage, the measured 10.4× full-scale speedup this yields, and the profiling result that then locates the dominant remaining cost in the evaluation stage rather than in construction. It supplies the efficiency accounting summarized in Section 5.

Algorithm 1 Parallel per-prompt mutation construction for one generation.

1: Input: live groups $\{(g, P_g)\}$, generation t, worker count W, base seed.
2: Compose tasks $\{(g, P_g, \xi_{g,t})\}$ with a deterministic per-group seed $\xi_{g,t}$.
3: Map $f(\cdot)$ over all tasks with a thread pool of $\min(W, |\text{tasks}|)$ workers, collecting $\mathcal{C}_g = f(P_g; \xi_{g,t})$.
4: Mark a group failed if $|\mathcal{C}_g| < e$; otherwise append its candidates to the pending proposal batch.
5: Score the surviving proposals in one batched reward call (the evaluation stage, not construction).

F.1 Embarrassingly parallel construction

Candidate construction is independent across prompts: the mutant pool of one prompt is determined entirely by its anchor population and its per-prompt seed, and no prompt reads another prompt's state during construction. The construction cost is concentrated in the LFQ bit-Hamming distance matrix and the balanced k-medoids refinement of Appendix G.2; both are implemented as C-level NumPy kernels that release Python's global interpreter lock, so a thread pool can overlap them across prompts even though the surrounding orchestration remains single-threaded Python. This is the data-parallel analogue of the island model's independent subpopulations (Whitley et al., 1999), with the parallelism placed over prompts (each prompt is a separate search problem) rather than over lineages within one search.

We therefore replace the sequential per-prompt loop with a fixed worker pool that applies the unchanged candidate builder $f(\cdot)$ to every prompt in parallel, returning the per-group candidate set $\mathcal{C}_g$. Each task is seeded by a deterministic function of the global step, the group index, and the generation index, combined with a base seed; refill attempts add a fixed prime offset to obtain a fresh stream. Because this seed is deterministic and f is deterministic, the parallel build reproduces the sequential build up to floating-point reduction order. The per-prompt mutation output is therefore unchanged, so the proposal and survivor mechanisms of Appendix E are untouched. The worker count is the configuration key mgpo_mutation_workers; a value of 0 selects $\min(16, \max(1, \text{os.cpu_count}()))$ automatically, and the effective pool size is capped at the number of live groups to avoid oversubscribing the Ray controller. Algorithm 1 summarizes one parallel generation.

F.2 Measured speedup at full scale

Table 7 compares the pre-parallelization run with the parallelized run at the full 4B configuration (8 GPUs, TP=1/DP=8, training batch 256, the same MGPO 56-mutant layout). The intentionally optimized stage is mutation construction, which drops from 427 s to 41 s, a 10.4× speedup; the GPU-side stage times are reported as observed wall-clock rather than as paired utilization measurements. Structural reward evaluation is optimized independently and is summarized in Table 8. After construction parallelization alone, the end-to-end step shortens from 1452 s to 1092 s (−24.8%). Applying the exact structural-reward acceleration in Table 8 reduces the exact-MGPO step to approximately 845 s (the exact reward-evaluation stage is about 175 s instead of the 160 s proxy value in Table 9); the approximate proxy-reward variant reaches approximately 830 s and is reported only as a systems upper bound. In a separate profiling run, training throughput rises from 868 to 1102 tok/s (+27%) with parallelization; this throughput measurement should not be interpreted as the exact reciprocal of the step-time reduction. The remaining logged stages (advantage computation 0.19 s, weight update 5.0 s, and rollout dump 12.0 s) are individually sub-1% and are absorbed into the residual.

The 10.4× construction speedup is the direct effect of the thread pool: construction is the only stage whose wall-clock scales with the number of prompts without GPU involvement, so overlapping its NumPy kernels across prompts removes the largest single CPU serialization in the step. The unchanged reward-evaluation stage confirms that the speedup did not come

Table 7: Full-scale before/after of the per-prompt mutation parallelization at the 4B configuration (8 GPUs, training batch 256; seven pre-parallelization steps versus five parallelized steps, means over steps). "Before" is the sequential construction run; "After" uses the thread pool of Section F.1. The GPU-side stages are reported for completeness and are "unchanged within variation" rather than exact paired measurements. Training throughput is measured on a separate profiling run and is reported as observed wall-clock rather than as a paired ratio to the step time.

Stage	Before (s)	After (s)	Change
Mutation construction	427	41	10.4× faster
Structural reward evaluation	469	469	optimized separately (Table 8)
Rollout generation	149	156	+5%
Old log-probability	154	163	+6%
Actor update	253	263	+4%
Total step	1452	1092	−24.8%
Training throughput (tok/s)	868	1102	+27%

from a budget or evaluator change: the same 56 mutants per prompt are still decoded and scored in both runs.

Structural reward acceleration. Structural reward evaluation was then optimized independently of construction. Table 8 separates three correctness-preserving changes from one approximate variant. Shared decode caching lets lDDT-CA and TM-align reuse one decode; TM-align subprocess parallelization increases the worker pool from 8 to 16; and torch.compile accelerates the frozen structure decoder after warm-up. These changes preserve the full reward for all 56 mutants required by Eq. equation 4 and reduce the structural scoring stage from 469 s to approximately 175 s. The final row, proxy-reward evolution, is different: the 56 mutants are first ranked with a cheap lDDT-CA + token-accuracy proxy and full structural reward is computed only for the selected mutant representatives (the 8 anchors are already fully scored). This mode reaches about 160 s but changes reward accounting, so it is reported as an approximate engineering ablation rather than as the implementation of the headline MGPO objective.

Table 8: Structural reward evaluation acceleration at the full 4B configuration (step-1 measurement; each row is cumulative). The first four rows preserve exact full-reward accounting for all 56 mutants; the final row is an approximate proxy variant and is not used as evidence for the exact MGPO objective unless separately validated.

Optimization	Reward eval (s)	Saving
Baseline (decode cache 32, TM-align 8 workers)	469	–
+ shared decode cache	305	−164
+ TM-align 16 workers	235	−70
+ torch.compile (warm; exact reward)	∼175	∼−60
+ proxy-reward evolution (approx.)	∼160	∼−15

F.3 Resulting breakdown and next bottleneck

After parallelization and structural reward acceleration, mutation construction and decode/TM-align scoring are no longer leading costs. Table 9 uses the approximate proxy path because that was the fastest profiled configuration; it should therefore be read as a systems upper-bound on throughput rather than as the timing of the exact headline objective. The exact compiled-decoder path differs by only about 15 s in the reward-evaluation stage in this profile (approximately 175 s instead of 160 s), the proxy path does not compute full R_i for every mutant and is therefore reported separately from the exact objective. Individual stage times are approximate and may not sum exactly to the measured total, which varies with batch size, sequence length, and TM-align worker count.

Table 9: Per-step wall-clock breakdown for the fastest profiled 4B configuration, which includes the approximate proxy-reward path (8 GPUs, TP=1 / DP=8, training batch 256, maximum response length 9216, with parameter/gradient/optimizer offload). Exact-MGPO scoring is approximately 15 s slower in the reward-stage profile.

Stage	Time (s)	Share	Device
Structural reward evaluation (decode + TM-align)	∼160	19%	CPU
Actor update	262	32%	GPU
Old log-probability	163	20%	GPU
Rollout generation	156	19%	GPU
Mutation construction	41	5%	CPU
Other	∼48	5%	–
Total	∼830	100%	–

Two consequences follow. First, after parallelization and the reward acceleration, both mutation construction and structural reward evaluation are no longer dominant, so further engineering should be judged by correctness rather than by large throughput gains. Second, the dominant costs are now the standard GPU-side rollout and optimization stages, so the next efficiency target is the actor/rollout pipeline rather than the CPU decode-and-score path; the residual CPU decode/TM-align cost could still be reduced by an in-process TM-score implementation.

F.4 Interpretation and limitations

The parallel construction is a correctness-preserving engineering change whose value is scheduling: it overlaps CPU-side mutation work across prompts while the controller would otherwise idle, and it removes a sequential bottleneck that grows with the number of prompts. At a much smaller configuration (batch size 4, maximum response length 6784), the same change moves throughput by only +2.3%, because mutation construction is already a negligible fraction of the step. The 10.4× construction speedup becomes visible only once the number of prompts is large enough that construction dominates the CPU. The fractions in Table 9 are configuration-dependent (they shift with batch size, sequence length, and whether TM-align is batched), and the before/after numbers in Table 7 compare two runs on the same model and batch size but with slightly different rollout scheduling settings, so the GPU-side stages should be read as "unchanged within variation" rather than as exact paired measurements.

G MGPO implementation details

This appendix provides the derivations and algorithmic details of the MGPO components summarized in Section 4.1.

G.1 Problem setup details

The scalar reward used by training and survivor selection is the weighted sum of available reward functions,

$$R(x, y) = \sum_{r=1}^{J} \omega_r R_r(x, y), \qquad \omega_r \geq 0, \tag{21}$$

with invalid or undecodable completions assigned zero mutation weight and excluded from population statistics. The four components are (i) token accuracy, the exact positional match rate between the predicted structure tokens $\mathbf{t}(y)$ and the reference tokens obtained by encoding the native structure with the frozen DPLM-2 encoder ($\frac{1}{L}\sum_i \mathbb{I}[t_i = t_i^{\text{ref}}]$, with mismatched lengths scored as zero); (ii) lDDT-CA; (iii) TM-align (TM-score via the TM-align program), both computed by decoding $\mathbf{t}(y)$ back to coordinates with the frozen decoder and comparing against the native backbone; and (iv) a format term that checks the structure block is present and length-consistent. Because the reference is the native structure

of the prompt itself, all methods share the same supervised signal, and the reported diversity/quality differences cannot be attributed to a different reward. For the exact MGPO objective, this full scalar $R(x, y)$ is computed for all $n_M = 56$ mutants before grouping and baseline computation. A cheap proxy may be used for proposal construction, but replacing R by the proxy for members of $\mathcal{M}$ defines an approximate variant and must be labeled separately. The mutation interface operates directly on the LFQ binary code; the structure-token representation is defined in Section 3.1. The fixed per-prompt data flow of the default MGPO update is summarized in Table 10.

Table 10: Fixed per-prompt data flow of the default MGPO update.

Data block	Evaluated	Trained	Role
Policy anchors	8	8	On-policy TOPR group
Evolutionary mutants	56	0	Grouping and representative pool
Selected mutant representatives	–	24	Three groups, eight per group
Final training batch	–	32	Joint MGPO sequence loss

G.2 Reward-blind balanced structural grouping

We construct the full $n_M \times n_M$ distance matrix with equation 1 and apply deterministic balanced k-medoids with $k = n_G$ (Kaufman & Rousseeuw, 1990):

$$\mathcal{M} = G_1 \cup \cdots \cup G_{n_G}, \qquad G_g \cap G_h = \varnothing \ (g \neq h), \qquad |G_g| \geq n_R. \tag{22}$$

The first medoid minimizes total distance to the pool and the remaining $n_G - 1$ medoids are initialized farthest-first. Samples are assigned to the nearest medoid. If a cluster has fewer than n_R members, a non-medoid point is moved from a cluster above capacity by minimizing the induced assignment-cost increase. Each medoid is then updated to the real member with minimum within-cluster distance, for at most R_{ref} refinement rounds. The clustering stage never reads reward.

G.3 Reward-stratified coverage selection

Each G_g is compressed to n_R distinct training representatives S_g. We rank its members by reward and create n_S quantile strata, with a quota of n_q representatives from each stratum. The medoid and highest-reward member are included first when their stratum quotas permit. Remaining slots are filled greedily by

$$\text{Score}(i) = \lambda_{\text{cov}}\, \Delta\,\text{Cov}(i) + \lambda_{\text{div}}\, \text{Div}(i) + \lambda_{\text{qual}}\, \text{Qual}(i), \tag{23}$$

with $\lambda_{\text{cov}} + \lambda_{\text{div}} + \lambda_{\text{qual}} = 1$, subject to the stratum quotas, where

$$\text{Div}(i) = \min_{j \in S} d(i, j), \qquad \text{Qual}(i) = \frac{R_i - R_{\min,g}}{\max(R_{\max,g} - R_{\min,g}, \varepsilon)},$$

and the marginal coverage improvement is

$$\Delta\,\text{Cov}(i) = \frac{1}{|G_g|} \sum_{u \in G_g} \left[d(u, S) - \min\{d(u, S), d(u, i)\} \right]. \tag{24}$$

This selection deliberately retains negative, near-baseline, and positive local contrasts; MGPO is consequently not an elite-only behavior-cloning method. Because the full-cluster baseline is computed before this compression, the selected subset need not itself have zero-mean advantage.

G.4 Group baseline and anchor alignment

The baseline of group g uses every member of the full cluster, whereas the common scale uses the complete mutant pool:

$$b_g = \frac{1}{|G_g|} \sum_{i \in G_g} R_i, \qquad s_M = \text{Std}_{i \in \mathcal{M}}(R_i),$$

$$A_i^M = \frac{R_i - b_g}{s_M}, \qquad i \in S_g. \tag{25}$$

Computing b_g before the $n_M \to n_G n_R$ compression prevents the reward-stratified selector from shifting the advantage baseline. Anchors keep the native prompt-group GRPO advantage (Shao et al., 2024), denoted A_a, and form group $G_0 = \mathcal{A}$.

Let μ_0 be the anchor mean and $\mu_g = b_g$ for mutant groups. With $\bar{\mu} = (n_G + 1)^{-1} \sum_{g=0}^{n_G} \mu_g$ and $s_\mu = [(n_G+1)^{-1} \sum_g (\mu_g - \bar{\mu})^2]^{1/2}$, we compute the softmax stably after subtracting the maximum group mean:

$$z_g = \frac{\mu_g - \max_h \mu_h}{s_\mu}, \qquad p_g = \frac{\exp(z_g)}{\sum_h \exp(z_h)}.$$

If $s_\mu = 0$, we set $p_g = 1/(n_G+1)$ and hence $W_g = 1$ for every group. Otherwise, W_g follows equation 5. Subtracting the maximum prevents overflow, and the $s_\mu = 0$ branch removes the undefined near-zero-temperature case.

G.5 Distance proxy calibration

Let $\widetilde{\delta}$ be the median positive distance in the mutant pool. We calibrate

$$\kappa = -\frac{\log q_{\text{med}}}{\widetilde{\delta}}, \qquad q_{\text{med}} \in (0,1), \qquad q_i = \exp(-\kappa \delta_i), \tag{26}$$

with $q_i = 1$ when all distances vanish. Negative-advantage attenuation follows equation 6. We restrict the attenuation parameters to $0 < b \le a$; then for every $q \in (0,1]$, $\log(q^a e^{b(1-q)}) \le (b-a)(1-q) \le 0$, so the multiplicative weight lies in $(0,1]$. The default is $a = b = 1$. This bound establishes only that the heuristic attenuates negative updates; it does not make the mutant branch an unbiased off-policy estimator.

G.6 Training procedure details

The optimization step is described in Section 4.1. Candidate paths must be distinct from anchors, earlier mutants, and their current generation. A deterministic refill with new seeds is attempted at most R_{refill} times; failure to obtain e unique valid mutants aborts the update rather than padding with duplicates. Evolutionary early stopping is disabled so that the complete n_M-mutant pool is available. Cheap token-space proposals do not consume structural evaluations. Metadata are gathered across data-parallel workers before grouping, and the fixed $n_A + n_G n_R$ sequence layout is re-sharded uniformly.

H Scope and interpretation

MGPO's core claim is that structure-defined local baselines can preserve learning signals from multiple mutant families while retaining a fixed, well-audited training layout. Its anchor branch is rollout-based TOPR, but its mutant branch is not an unbiased off-policy policy-gradient estimator. LFQ distance may not perfectly track 3D structural distance or policy likelihood, and q_i in equation 26 is not an importance ratio. A strict correction would require a tractable recorded mutation density. We therefore present distance mixtures, group count, group-weight floor, representative-selection terms, mutant-pool size, and the nine evolutionary backends as explicit design choices rather than as separate headline algorithms.