FIXED-CARDINALITY JOINT DIFFUSION FOR COMBINATORIAL PROTEIN DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Fixed-cardinality design seeks a batch of sequences with exactly K mutations relative to a reference, a constraint libraries impose at cloning that post-hoc filtering cannot restore. We introduce FC-JD (Fixed-Cardinality Joint Diffusion), a two-stream diffusion model. Its position stream diffuses mutation masks on the Johnson graph $J(N, K)$, making exact-K a construction invariant. Its simulation-free objective replaces forward rollouts with an analytic posterior over reverse transitions. We answer three questions. (Q1) Can FC-JD guarantee exact-K generation without post-hoc correction? (Q2) How does FC-JD compare with search-based methods (simulated annealing, Bayesian optimization) under the same batch budget? (Q3) When is FC-JD useful, and what are its limitations? In a pre-registered, oracle-scored benchmark (three complete-oracle landscapes, one partial-oracle case study, one regression-only dataset, four arms), FC-JD produced exact-K designs in every round, achieved substantially higher valid rates on partial-oracle landscapes, delivered batches in under two seconds, and descriptively matched or exceeded GP-BO on the extreme tail in every oracle-evaluable cell at $K \in \{5, 7\}$. Surrogate-guided simulated annealing retained the best mean fitness on all four oracle-evaluable landscapes, and pre-registered criteria confirm this division of labor: FC-JD meets the generation criteria (valid, time) while SA96 keeps the best mean. We position FC-JD as a fast, diversity-preserving generator for exact-K design. Additionally, ESM-2-conditioned variants and cross-target transfer are reported as supporting analyses.

1 INTRODUCTION

Protein engineering increasingly proceeds through *combinatorial mutagenesis*, in which multi-site libraries of a chosen order are built and assayed Phillips et al. (2021); Sarkisyan et al. (2016); Olson et al. (2014). Each variant in such a library carries exactly K amino-acid changes relative to a reference. Deep mutational scanning (DMS) now supplies near-complete fitness measurements of low- and higher-order mutants across antibodies, enzymes, and structural proteins Tsuboyama et al. (2023); Notin et al. (2023a); Dallago et al. (2021); Papkou et al. (2023); Starr et al. (2020). This turns landscape design into an explicit optimization problem: given a budget B of variants to synthesize, propose B sequences with *exactly* K mutations that maximize measured fitness. The constraint is practical: libraries are cloned and assayed at a chosen order, and it cannot be restored by post-hoc filtering. An unconstrained masked-diffusion baseline reaches 0% exact-K at $K \geq 5$ (Section 3).

Sequence design is dominated by two families of learned models: fitness predictors that must be paired with a search procedure, and generative models that propose sequences directly. Existing approaches generally require a separate search procedure or a rejection step when the design must contain exactly K mutations. Surrogate-guided search often achieves strong mean fitness but can be costly and can concentrate proposals in a small region of the design space. We do not seek to replace optimization when mean fitness is the sole objective; we target the complementary goal of diverse, valid, exact-K designs within constrained budgets and time.

We introduce **FC-JD** (Fixed-Cardinality Joint Diffusion), a two-stream diffusion model for exact-K design (Figure 1). The position stream evolves a mutation mask on the Johnson graph $J(N, K)$, whose vertices are the $\binom{N}{K}$ position sets; its closed-form transition kernel makes training simulation-free. The residue stream diffuses the amino-acid content at mutated positions. Sampling is one

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.

forward pass over T steps and guarantees exactly K mutations by construction; a proposal-and-rank variant (P+R) delivers a surrogate-ranked batch in one shot.

Our contributions are twofold. Methodologically, FC-JD combines a closed-form Johnson-graph position kernel with per-site residue diffusion, trained without simulation and sampling batches in under two seconds. Empirically, we present a pre-registered, oracle-evaluated benchmark on three complete-oracle landscapes plus a partial-oracle case study (CB6) and a regression-only dataset (GFP), across four arms (Section 4). This paper addresses three questions. **(Q1)** Can FC-JD guarantee exact-K generation without post-hoc correction? **(Q2)** How does FC-JD compare with search-based methods, SA and BO, under the same batch budget? **(Q3)** When is FC-JD useful, and what are its limitations?

2 Related Work

Discrete diffusion and flow matching. Denoising diffusion Ho et al. (2020); Song et al. (2021) has been extended to discrete state spaces through structured transition matrices Austin et al. (2021), score-entropy estimation Lou et al. (2024), and masked absorbing processes Sahoo et al. (2024). Flow matching Lipman et al. (2023) and stochastic interpolants Albergo et al. (2025) generalize this to continuous time, with protein co-design applications Campbell et al. (2024) and unified discrete frameworks Gat et al. (2024). All of these operate on the full sequence (or a masked relaxation) and train against the uniform state; none restricts the state space to a fixed cardinality K, so exact-K requires rejection or constrained sampling on top. FC-JD instead restricts the position stream to the Johnson graph $J(N, K)$, making exact-K a *construction guarantee* (Section 3). Its closed-form kernel makes training simulation-free (the analytic posterior replaces Monte-Carlo rollouts), which unconstrained diffusion cannot match; sampling still runs the T-step reverse chain, and the residue stream keeps a standard absorbing diffusion over the mutated positions only.

Generative protein design. A first line uses structure or evolutionary priors to propose sequences: structure-conditioned design Dauparas et al. (2022), backbone diffusion with in-painting sequence design Watson et al. (2023); Bose et al. (2023), and language-model samplers Ferruz et al. (2022). Antibody design has followed the same arc: antigen-conditioned diffusion Luo et al. (2022), RFdiffusion for single-domain antibodies Bennett et al. (2024), and generative antibody/nanobody models Wang et al. (2025), typically optimizing for structure or developability rather than measured binding fitness. Reward-weighted flows Bengio et al. (2021); Jain et al. (2022) sample diverse high-reward sequences. Fitness predictors: ESM-1v Meier et al. (2021), ESM-2 Lin et al. (2023), non-parametric transformers Notin et al. (2023b), autoregressive models with retrieval Notin et al. (2022), and deep generative models of variation Riesselman et al. (2018); Frazer et al. (2021); Rao et al. (2021) score variants but do not by themselves produce a diverse batch under a cardinality constraint. FC-JD targets the complementary case: a generative sampler with exact-K support, conditioned on fitness quantiles, evaluated against measured assay oracles rather than structure or likelihood proxies.

Model-based optimization on fitness landscapes. Gaussian-process Bayesian optimization Jones et al. (1998); Shahriari et al. (2016); Rasmussen & Williams (2006) and batched acquisition Wilson et al. (2018) remain the standard sample-efficient search procedures, with offline model-based optimization formalizing the design benchmark Trabucco et al. (2022; 2021). Our BO baseline implements GP-EI with fixed $\mathrm{RBF}(\ell{=}2.0){+}\mathrm{WhiteKernel}(0.01)$ hyperparameters and 2000-candidate random exact-K proposal pools per step (Section 4). We fix them because the pre-registered plan freezes every arm definition, keeping BO96 a reproducible, non-adaptive reference; adaptive tuning would add protocol complexity beyond this benchmark's scope. In the protein regime, surrogate-guided annealing with local exploration Ren et al. (2022) and data-efficient deep models for low-N protein engineering Biswas et al. (2021) show that search can outperform generation on mean fitness when evaluations are cheap. Greedy surrogate search converges to near-duplicate designs; our benchmark scores this diversity–quality tension explicitly under one frozen protocol, with search baselines and the generative family included.

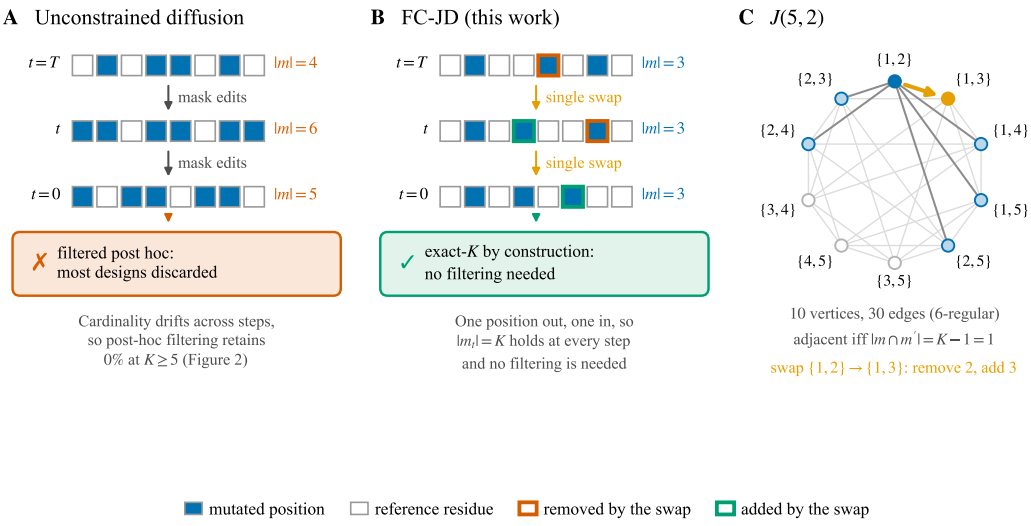


Figure 1: **Unconstrained diffusion versus the FC-JD framework** (schematic, N=8, K=3). **(A)** Unconstrained masked diffusion edits the mask freely, so the cardinality drifts across reverse steps ($|m|$=4 → 6 → 5) and the target K is met only by chance; the batch must then be filtered post hoc, which discards most designs and retains 0% at $K \geq 5$ (Figure 2). **(B)** FC-JD replaces the free edit with a single swap on the Johnson graph $J(N, K)$: one mutated position is removed and one is added at every reverse step, so $|m_t|$=K holds at t=T, at every intermediate t, and at t=0. **(C)** The Johnson graph $J(5, 2)$: its $\binom{5}{2}$=10 vertices are the 2-subsets of $\{1, \ldots, 5\}$ and two vertices are adjacent iff they share $K-1$=1 element, giving a 6-regular graph with 30 edges. The highlighted transition $\{1, 2\} \rightarrow \{1, 3\}$ (blue vertex to orange vertex, grey edges = the focal vertex's neighbourhood) removes position 2 and adds position 3 while preserving $|m|$=2.

3 METHOD

Central methodological contribution. The core of FC-JD is the position stream: diffusion on the Johnson graph $J(N, K)$ with a simulation-free objective (Section 3.2); the residue stream (Section 3.3) is auxiliary. The simulation-free training objective is derived in this section with the full pseudocode in Appendix E; the sampling procedure is given as pseudocode below (Algorithm 1); the search baselines SA96 and BO96 are defined in Section 4.2.

3.1 PROBLEM SETUP

Let $x \in \mathcal{V}^N$ be a sequence over alphabet $\mathcal{V}$ ($|\mathcal{V}| = 20$ for proteins, 4 for DNA) and r a reference sequence. A variant is characterized by its mutation mask $m \in \{0, 1\}^N$ with $m_i = 1$ iff position i is mutated relative to r, and cardinality $K = |m| = \sum_i m_i$. We are given an assayed fitness landscape $\mathcal{D}$ (deep mutational scanning), from which an oracle f is available for assayed sequences. The training set consists of rows with $1 \leq K \leq k_{\text{train}}$ (we use $k_{\text{train}} = 3$); fitness values are transformed to fractional ranks and discretized into 5 bins with edges $(0.5, 0.8, 0.9, 0.95)$, giving a conditioning label $y \in \{0, \ldots, 4\}$ (a 3/5/10-bin ablation in the appendix confirms five bins). At generation time the task is: *given a budget B, produce B sequences with exactly K mutations ($K \in \{5, 7\}$) that maximize f*. On position-set landscapes (the antibody landscapes below), variants are position sets and FC-JD uses the position stream alone; on token-valued landscapes (DHFR) it uses both streams.

3.2 POSITION STREAM: DIFFUSION ON THE JOHNSON GRAPH $J(N, K)$

Intuitively, the position stream is a random walk over sets of exactly K positions; the equations below make its transition probabilities explicit. The set of $\binom{N}{K}$ mutation masks is the vertex set of

the Johnson graph $J(N, K)$, where two masks are adjacent iff they share $K-1$ positions (Figure 1C shows $J(5, 2)$). We define a single-swap Markov chain on $J(N, K)$: at each step, with probability β_t, remove one element uniformly from the current set and add one uniformly from its complement. The chain is reversible with respect to the uniform distribution, and the overlap $j_t = |m_t \cap m_0|$ evolves on $\{j_{\min}, \dots, K\}$, $j_{\min} = \max(0, 2K - N)$, as a birth–death chain with generator

$$L_{j,j+1} = \tfrac{(K-j)^2}{d}, \qquad L_{j,j-1} = \tfrac{j\,(N-2K+j)}{d}, \qquad L_{j,j} = 1 - \tfrac{(K-j)^2+j\,(N-2K+j)}{d}, \tag{1}$$

where $d = K(N-K)$. The per-step kernels $P_t = \prod_{s=1}^{t}(I - \beta_s \mu)$, $\mu = I - \mathrm{diag}(\pi)^{-1/2} L\, \mathrm{diag}(\pi)^{1/2}$, $\pi_j = M_j / \binom{N}{K}$ with $M_j = \binom{K}{j}\binom{N-K}{K-j}$, are computed analytically (closed-form eigendecomposition; Appendix E). We use the linear retention schedule $\rho_1(t) = 1 - t/T$ with relaxation rate $\mu_1 = N/d$.

Simulation-free objective. For a training mask m_0 (Algorithm 2), sample $t \sim \mathcal{U}\{1, \dots, T\}$ and $m_t \sim P_t(\cdot \mid m_0)$; let $j = |m_t \cap m_0|$. The analytic posterior over the five possible reverse transitions has weights

$$w_c \;\propto\; \mathrm{count}_c \cdot \frac{P_{t-1}(j'_c)}{M_{j'_c}} \cdot \Big[(1-\beta_t)\,\mathbb{1}[c=0] + \tfrac{\beta_t}{d}\,\mathbb{1}[c>0]\Big], \tag{2}$$

over the five reverse-action classes defined in Appendix E. The position model is a transformer ($d = 128$, 4 layers, 4 heads) over binary mask tokens with time, cardinality K, and bin y conditioning, outputting remove logits R_i over m_t, add logits A_i over the complement, and a stay logit s. Its predictive mixture mirrors the posterior's geometry:

$$\begin{gathered} q_0 = \sigma(s), \qquad q_1 = (1-\sigma(s))\, S_1 S_3, \qquad q_2 = (1-\sigma(s))\, S_2 S_4, \\ q_3 = (1-\sigma(s))\, S_1 S_4, \qquad q_4 = (1-\sigma(s))\, S_2 S_3, \end{gathered} \tag{3}$$

where $S_1 = \sum_{i \in m_t \cap m_0} \mathrm{softmax}(R)_i$, $S_2 = \sum_{i \in m_t \setminus m_0} \mathrm{softmax}(R)_i$, $S_3 = \sum_{i \in m_0 \setminus m_t} \mathrm{softmax}(A)_i$, $S_4 = \sum_{i \notin m_t \cup m_0} \mathrm{softmax}(A)_i$. The loss is the analytic $\mathrm{KL}(w \| q)$ for $t > 1$ and a direct realization loss at $t = 1$ (Algorithm 2). Because both w and q are available in closed form, *no simulation of the forward process is required during training* (the analytic posterior replaces Monte-Carlo rollouts); sampling instead runs the T-step reverse chain as a standard Markov process.

3.3 Residue stream (token-valued landscapes)

For token-valued landscapes, a separate masked-residue stream denoises mutated positions while unmutated positions stay fixed at the reference; position-set landscapes need no residue stream (details in Appendix E).

3.4 Sampling and the P+R pipeline

Algorithm 1 details the sampling procedure: the reverse chain runs over $t = T, \dots, 1$ from a uniform K-subset; each position-stream step keeps the mask with probability $\sigma(s)$ or performs a single swap, so $|m_t| = K$ holds at every step and *exact-K is guaranteed by construction.* A batch of 96 designs costs one forward pass (0.5–1.7 s on one RTX 3070 Ti). **Proposal+Ranker (P+R)**: draw a pool of $4B$ FC-JD samples, score them with the surrogate (below), and return the top B; one forward pass plus $O(4B)$ CPU evaluations, no iteration.

3.5 Baselines

SA96, surrogate-guided simulated annealing; **BO96**, Gaussian-process expected improvement with fixed hyperparameters, n_{cand}=2000 random exact-K proposals per step, refit every 10 steps, B=96 (full specification in Section 4.2). Both search baselines evaluate designs only through the surrogate (zero oracle calls during search). **B5**, frequency baseline (position marginals + residue frequencies), reported in the appendix. **D0 (masked diffusion)**, a standard absorbing-state diffusion baseline without cardinality enforcement, trained on the same $K \leq 3$ rows with masked cross-entropy loss and T-step ancestral sampling; the exact-K rate is measured post-hoc on the generated batch and is the construction-guarantee contrast in the introduction and results (23–26% at K=4, 0% at K=5–9 on DHFR; range over 3 training seeds).

Algorithm 1 FC-JD sampling: exact-K generation in one pass

Require: position model θ_{pos}; reference r; cardinality K; top bin $y{=}4$; steps T; batch size B
Ensure: batch of B designs with exactly K mutations

```
B ← ∅
for i = 1, ..., B do
    m_T ~ Uniform(K-subsets of {1, ..., N})
    for t = T, ..., 1 do
        keep m_t with probability σ(s); otherwise m_{t-1} ← (m_t \ {i}) ∪ {j} with i ~ softmax(R|_{m_t}), j ~ softmax(A|_{m̄_t})   ▷ |m_{t-1}|=K by construction
    end for
    B ← B ∪ {m_0}
end forreturn B
  (token-valued landscapes: the residue stream denoises mutated positions
   in the same loop; Appendix E)
```

Table 1: Landscape statistics. Sources: CR9114-h1 Phillips et al. (2021), GFP-Sarkisyan Sarkisyan et al. (2016), DHFR Papkou et al. (2023). The benchmark consists of three complete-oracle landscapes (CR9114, CH65, DHFR) plus two special cases: CB6, a partial-oracle case study, and GFP, retained for regression only. n_{train} = rows with $1 \le K \le 3$; n_K = assayed rows at cardinality K used as the oracle evaluation set. Oracle coverage = fraction of the exact-K space present in the assay ("complete" = all exact-K variants assayed; "partial" = the fraction shown). †Partial-oracle case study: SA96's designs land predominantly outside the assayed subset; FC-JD achieves 46–61 valid designs per round (details in Appendix D). ‡Regression-only: all arms land outside GFP's exact-$K{=}5$ assay (valid$= 0$); details in Appendix D.

Landscape	Type	N	n_{train}	$n_{K=5}$	$n_{K=7}$	Oracle coverage
Complete-oracle benchmark landscapes						
CR9114-h1	position set	16	697	4,368	11,440	complete (2^{16})
CH65-MA90 (CR6261)	position set	16	697	4,368	11,440	complete
DHFR	tokens 4^9	9	2,619	30,539	78,473	complete
Special cases						
CB6†	position set	12	299	784	726	30.5% (1250/4096)
GFP-Sarkisyan‡	positions + fill	233	26,197	6,825	2,526	$\sim 10^{-12}$ (support ratio)

3.6 Protocol, metrics, and pre-registration

The evaluation protocol (landscapes, budget, seeds, metrics, verdict criteria) is defined in Section 4; all arms make zero oracle calls during search, and full per-round data are released with the code.

4 Experimental Setup

4.1 Landscapes

Table 1 summarizes the three complete-oracle benchmark landscapes and the two auxiliary datasets; CB6 is not pooled with the complete-oracle results, and GFP contributes regression results only.

All models train on rows with $1 \le K \le k_{\text{train}}{=}3$ and generate at $K \in \{5, 7\}$; GFP masks are completed by the pre-registered fill rule.

4.2 Implementation details

Seeds and rounds. 10 training seeds (model or surrogate weights) $\times$ 10 sampling seeds give 100 rounds per (landscape, K, arm) cell (40,000 evaluated rounds); K-ladder seed counts are noted in the appendix.

Surrogate (shared by SA96 and P+R). Additive single-mutant table plus an ensemble of three attention MLPs (d=64) fit to the $K \leq 3$ rows; the MLP residual is z-scored against the additive base. DHFR uses the same family (150 epochs).

SA96. 96 chains, 200 swap-move steps, geometric temperature $0.3 \rightarrow 0.01$; each move swaps one mutated position or one residue while preserving cardinality; proposals are scored by the surrogate and accepted with probability $\min(1, \exp(\Delta \hat{f}/T_s))$ (zero oracle calls).

BO96. GP-EI with $\mathrm{RBF}(\ell{=}2.0) + \mathrm{WhiteKernel}(0.01)$, fixed hyperparameters, n_{cand}=2000 random exact-K proposals per step, refit every 10 steps, budget B=96; on DHFR features are the 36-dim one-hot of the 4^9 sequence.

FC-JD. Two transformer streams trained with AdamW; architecture, schedules, and optimization settings are listed in Appendix E.

4.3 Metrics and statistics

Per round we report: *mean* fitness of the unique valid designs; *hits@1%* and *hits@0.1%* (fractions of designs above the per-K 0.99/0.999 quantiles of assayed values); *unique* designs; *valid* designs (present in the assay); *wall-clock* seconds per B=96 round (per-landscape medians in Table 5, Appendix C); and *compute cost* (training plus per-round time, with zero oracle calls for every arm). Arm comparisons use the paired Wilcoxon test over the 100 rounds (per landscape, K, arm) and seed-level percentile bootstrap 95% CIs (B=10,000) for headline differences. Cells where fewer than 48% of designs are evaluable by the oracle are flagged as partial-oracle in all tables.

4.4 Pre-registration and reproducibility

Protocols, arm definitions, seed schedules, and verdict criteria were frozen before execution and are released with the code. Hardware: one RTX 3070 Ti laptop GPU; training is 120 epochs of a $\sim$1M-param transformer per landscape, and generation is one forward pass per round.

5 Results

We answer the three questions from Section 1 in turn. All intervals are seed-level percentile bootstrap 95% CIs over 100 rounds per cell. "Oracle-evaluable" below means scorable against the $K \in \{5, 7\}$ assay (CR9114, CH65, CB6, DHFR); GFP is regression-only (Section 6; Table 1, footnote ‡). Table 1 defines the landscapes; Figure 3 in Appendix A summarizes the quality–diversity–cost picture.

5.1 Q1: can FC-JD guarantee exact-K generation without post-hoc correction?

The answer is yes, and it is structural rather than empirical. The position stream samples on the Johnson graph $J(N, K)$, so every design FC-JD outputs has exactly K mutated positions: the cardinality invariant holds at each of the T reverse steps, and no post-hoc filtering is applied anywhere in the pipeline (Section 3). The unconstrained baseline D0 reached 23–26% exact-K at K=4 and 0% at K=5–9 on DHFR, discarding most of its budget before oracle evaluation (Figure 2). A second guarantee matters where the oracle is partial: FC-JD keeps its designs inside the assayed region (the CB6 case study, Section 5.2). Exact-K validity held in every round; the D0 contrast shows the cost of dropping the constraint.

5.2 Q2: how does FC-JD compare with search-based methods under the same batch budget?

Mean. SA96 achieved the best mean on all four oracle-evaluable landscapes (CR9114 $\Delta{-}0.21$, CH65 $\Delta{-}0.42$, CB6, DHFR $\Delta{-}0.24$ to -0.32, FC-JD minus SA96). Against BO96 the picture was mixed: FC-JD was significantly better at CR9114 K=5 ($+0.15$ [0.12, 0.18]) and CH65 K=5 ($+0.03$ [0.005, 0.059]), and significantly worse at CH65 K=7, CB6 K=5, and DHFR K=7 (all Wilcoxon $p < 0.05$).

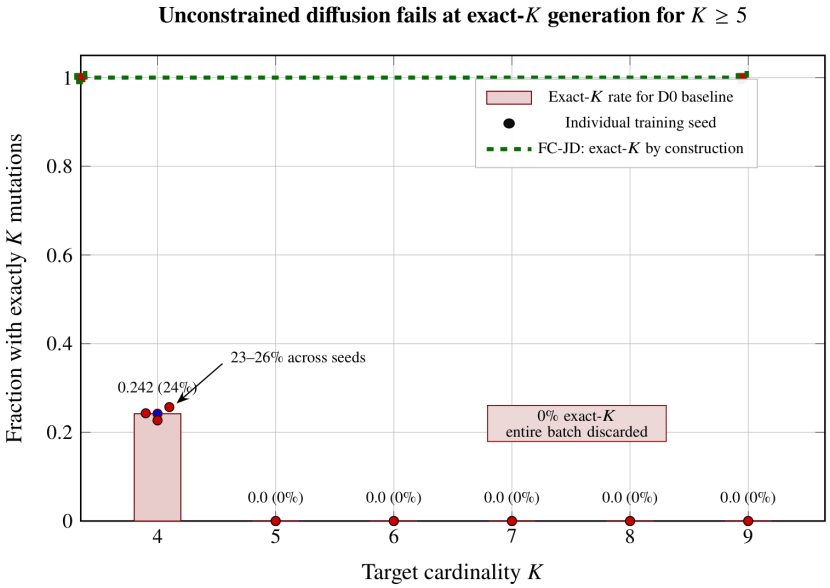


Figure 2: **Unconstrained diffusion fails at exact-K generation for $K \geq 5$.** Bars show the mean fraction of generated designs with exactly K mutations for the unconstrained masked-diffusion baseline D0 on DHFR; black dots are the individual seed values (3 training seeds). D0 reaches 23–26% exact-K at $K{=}4$ but collapses to 0% at $K{=}5$–9, discarding the entire $B{=}96$ batch before oracle evaluation. The dashed green line at 1.0 is FC-JD's construction guarantee: the Johnson-graph formulation enforces $|m| = K$ at every reverse step, so exact-K holds by design with no post-hoc filtering.

Diversity. FC-JD returned 80–96 unique designs per round on the three antibody landscapes (CR9114 93.0/95.4, CH65 93.5/95.4, CB6 80.8/85.7 at $K{=}5/7$) and 70–82 on DHFR (69.7/82.1), versus 3–11 for SA96 bare chains on the antibody landscapes and 30–45 on DHFR. The pre-registered ≥ 90 bar is met on CR9114 and CH65, and missed on CB6 and DHFR. BO96 matched FC-JD's diversity (90–96) but at 5–60$\times$ the wall-clock. Enlarging SA to 960 chains plus deduplication restored diversity on DHFR (59.6–85.0) but not on CR9114 (8.4–13.1): the collapse is structural on the antibody landscapes, not a budget artifact.

Cost. FC-JD delivered a batch in 0.49–1.68 s on one laptop GPU, versus 2.3–4.4 s for SA96, 5.3–29.2 s for BO96, 15–32 s for SA960, and 27–128 s for BO384 (Table 5). The pre-registered cost criterion held on $4/5$ landscapes, with DHFR $K{=}7$ missing the SA960 bar by 0.08 s (14.92 s, reported).

Extreme tail. At $K \in \{5, 7\}$ the FC-JD family (FC-JD or P+R, whichever is higher) matched or exceeded BO96 on hits@0.1% in all eight oracle-evaluable cells (means; per-cell significance is not claimed). The generative family was never tail-dominated by GP-BO at these cardinalities (full table in Appendix B). Against SA96 the tail was contested: SA96 was non-zero at CR9114 $K{=}5$ (0.0101), CH65 $K{=}7$ (0.0083), and DHFR $K{=}7$ (0.0385, the highest tail hit-rate in the study). The generative family alone was non-zero with SA empty at CR9114 $K{=}7$ (0.0019–0.0023), CH65 $K{=}5$ (0.0069–0.0071), and both CB6 cells. Performance across additional cardinalities is reported in Appendix B (Figure 4); the trend is landscape-dependent.

Partial-oracle landscape (CB6). CB6's assay covers 30.5% of the design space. FC-JD evaluated 2.2$\times$ more of its proposed designs than BO96 (valid 60.7 [59.8, 61.5] vs. 26.8 [26.0, 27.7] at $K{=}5$; 45.8 [45.0, 46.6] vs. 28.2 [27.4, 29.0] at $K{=}7$), while SA96 proposed designs almost entirely outside the assayed subset (valid 0.4 [0.2, 0.6] at $K{=}5$) and its mean and hits are undefined in most rounds; the generative family alone reached the 0.1% tail in both CB6 cells, and CB6 misses the ≥ 90-unique bar (80.8/85.7).

The pre-registered criteria confirm the division of labor between exact-K generation and fitness optimization (Figure 3A;Table 2). FC-JD meets the 2 generation criteria it was designed for (*valid*, *time*); *mean* and *hits@1%* measure what optimization was designed for, where SA96 achieves the best mean on all four oracle-evaluable landscapes and the only positive hits@1% cell is DHFR $K{=}5$ (P+R 0.731 vs. SA96 0.384, Wilcoxon $p < 0.05$). The remaining three, *hits@0.1%*, *unique*, and *P+R*, are FC-JD's own bars that it missed. The frozen verdict is 2/7. Accordingly, we do not position FC-JD as a near-optimizer and treat P+R as an optional rather than recommended strategy; both pre-registered contingencies are documented in Appendix A.

5.3 Q3: WHEN IS FC-JD USEFUL, AND WHAT ARE ITS LIMITATIONS?

FC-JD is the method of choice when a campaign needs a diverse batch of exact-K candidates in one pass: antibody landscapes at $K \leq 7$, where it matches or exceeds GP-BO on the extreme tail and returns 80–96 unique designs per round, and partial-oracle landscapes such as CB6, where its designs stay inside the assayed region. It is the wrong tool when mean fitness is the only objective: SA96 wins mean on all four landscapes, and FC-JD's diversity has a blind spot (the CB6 ≥ 90 bar). P+R lifts FC-JD's mean on CR9114 ($+0.17/+0.22$) and CH65 ($+0.23/+0.25$) while keeping 84–93 unique designs, but stays below SA96's mean on both and collapses diversity on CB6/DHFR (35.6–67.3 unique). Its pre-registered criterion, mean improvement *over raw FC-JD* with diversity ≥ 70 on $\geq 3/5$ landscapes, passed $2/5$ (40%), so we offer it as an optional strategy, not a recommended pipeline. Additional ESM-2-conditioned and transfer experiments are reported in Appendix B and are not used for the main claims; limitations bounding these answers are in Section 6.

6 LIMITATIONS

First, GFP is not oracle-evaluable for exact-K generation: the support ratio at $K{=}5$ is approximately 10^{-12}, so we use it for regression only. Second, CB6 has only 30.5% oracle coverage and is therefore reported as a partial-oracle case study rather than pooled with the complete landscapes. Third, FC-JD is not a general-purpose optimizer: SA96 achieves the best mean fitness on all four oracle-evaluable landscapes, and the diversity advantage of FC-JD is landscape-dependent. Additional analyses of oracle coverage, pipeline dependence, and the prospective framework are provided in Appendix D.

7 CONCLUSION

In a pre-registered, oracle-scored benchmark over five landscapes and four arms, exact-K validity held in every round, partial-oracle validity on CB6 was $2.2\times$ BO96's, batches took 0.49–1.68 s, and the generative family matched or exceeded BO96 on the extreme tail in all eight cells at $K \in \{5, 7\}$. Pre-registered criteria confirm the division of labor: FC-JD meets the 2 generation criteria (*valid*, *time*), SA96 keeps the best mean on all four oracle-evaluable landscapes, and the frozen verdict is 2/7.

AI USE STATEMENT

We used generative AI tools for drafting, editing, and language polishing of parts of the manuscript, for summarizing the related literature, for formatting references and structure, and for generating the plotting code and figure drafts. We did not use them for the required-disclosure tasks (synthetic data, theoretical or mathematical development, hypotheses, methodology or experiment design, method implementation, data cleaning, qualitative analysis, or interpretation of results). We reviewed all AI-assisted work and take full responsibility for the final content, including text, claims, or artifacts produced with the aid of generative AI; the pre-registered protocol and all result files predate and are independent of any AI assistance with the prose.

REPRODUCIBILITY STATEMENT

Code. All code (model implementations, training scripts, and evaluation pipelines) is released with the paper; an anonymous repository is provided as supplementary material.

Data. All datasets are public DMS assays (CR9114-h1, CH65-MA90, CB6, GFP-Sarkisyan, DHFR), summarized with sources in Table 1.

Compute. All runs used one RTX 3070 Ti laptop GPU (8 GB VRAM); training is 120 epochs of a $\sim$1M-param transformer per landscape, and generation is one forward pass per round (0.5–1.7 s per B=96 batch).

Hyperparameters and protocol. All hyperparameters are stated in Section 4 and Algorithm 2; the frozen pre-registered protocol (arms, seeds, verdict criteria) and all per-round arrays are released as JSON with the code.

REFERENCES

Albergo, Michael S., et al. Stochastic interpolants: A unifying framework for flows and diffusions. *JMLR 2025*, 2025.

Austin, Jacob, et al. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS 2021*, 2021.

Yoshua Bengio, T. Deleu, J. Hu, S. Lahlou, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 2021. doi: 10.48550/arXiv.2111.09266.

N. Bennett, Joseph L. Watson, R. Ragotte, A. J. Borst, DéJenaé L. See, Connor Weidle, Riti Biswas, Yutong Yu, et al. Atomically accurate de novo design of antibodies with rfdiffusion. *bioRxiv (preprint)*, 2024. doi: 10.1101/2024.03.14.585103.

Surojit Biswas, Grigory Khimulya, Ethan C. Alley, Kevin M. Esvelt, and George M. Church. Low-n protein engineering with data-efficient deep learning. *Nature Methods, 18: 389-396*, 2021. doi: 10.1038/s41592-021-01100-y.

A. Bose, Tara Akhound-Sadegh, Kilian Fatras, Guillaume Huguet, Jarrid Rector-Brooks, Cheng-Hao Liu, A. Nica, Maksym Korablyov, et al. Se(3)-stochastic flow matching for protein backbone generation. In *ICLR 2024*, 2023. doi: 10.48550/arXiv.2310.02391.

Campbell, Andrew, et al. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *ICML 2024*, 2024.

Christian Dallago, Jody Mou, Kadina E. Johnston, Bruce J. Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. In *NeurIPS 2021 Datasets and Benchmarks (proceedings); preprint: bioRxiv*, 2021. doi: 10.1101/2021.11.09.467890.

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. Ragotte, L. Milles, B. Wicky, A. Courbet, et al. Robust deep learning-based protein sequence design using proteinmpnn. *Science*, 2022. doi: 10.1126/science.add2187.

Noelia Ferruz, Steffen Schmidt, and B. Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature Communications*, 2022. doi: 10.1038/s41467-022-32007-7.

J. Frazer, Pascal Notin, M. Dias, Aidan N. Gomez, Joseph K. Min, Kelly P. Brock, Y. Gal, and D. Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature 599 (7883): 91-95*, 2021. doi: 10.1038/s41586-021-04043-8.

Gat, Itai, et al. Discrete flow matching. In *NeurIPS 2024*, 2024.

Ho, Jonathan, et al. Denoising diffusion probabilistic models. In *NeurIPS 2020*, 2020.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, C. Ekbote, Jie Fu, Tianyu Zhang, et al. Biological sequence design with gflownets. In *ICML 2022 (Spotlight)*, 2022. doi: 10.48550/arXiv.2203.04115.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization (Springer), 13(4): 455-492*, 1998. doi: 10.1023/A:1008306431147.

Zeming Lin, Halil Akin, Roshan Rao, Brian L. Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science 379 (6637)*, 2023. doi: 10.1126/science.ade2574.

Lipman, Yaron, et al. Flow matching for generative modeling. In *ICLR 2023*, 2023.

Lou, Aaron, et al. Discrete diffusion modeling by estimating the ratios of the data distribution. In *ICML 2024 (Best Paper)*, 2024.

Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In *NeurIPS 2022*, 2022. doi: 10.1101/2022.07.10.499510.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In *NeurIPS 2021*, 2021. doi: 10.1101/2021.07.09.450648.

Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan Gomez, Debora S. Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *ICML 2022*, 2022. doi: 10.48550/arXiv.2205.13760.

Pascal Notin, Aaron W. Kollasch, Daniel P. Ritter, Lood van Niekerk, Steffan B. Paul, Hansen Spinner, Nathan Rollins, Ada Y. Shaw, et al. Proteingym: Large-scale benchmarks for protein design and fitness prediction. In *NeurIPS 2023 Datasets and Benchmarks (proceedings); preprint: bioRxiv*, 2023a. doi: 10.1101/2023.12.07.570727.

Pascal Notin, Ruben Weitzman, D. Marks, and Yarin Gal. Proteinnpt: Improving protein property prediction and design with non-parametric transformers. In *NeurIPS 2023*, 2023b. doi: 10.1101/2023.12.06.570473.

C. Anders Olson, Nicholas C. Wu, and Ren Sun. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Current Biology*, 2014. doi: 10.1016/j.cub.2014.09.072.

Andrei Papkou, Lucía García-Pastor, José Antonio Escudero, and Andreas Wagner. A rugged yet easily navigable fitness landscape. *Science*, 2023. doi: 10.1126/science.adh3860.

Angela M. Phillips, Katherine R. Lawrence, Alief Moulana, Thomas Dupic, Jeffrey Chang, Milo S. Johnson, Ivana Cvijović, Thierry Mora, et al. Binding affinity landscapes constrain the evolution of broadly neutralizing anti-influenza antibodies. *eLife*, 2021. doi: 10.7554/eLife.71393.

Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, J. Canny, P. Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *ICML 2021*, 2021. doi: 10.1101/2021.02.12.430858.

Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning. *MIT Press (Adaptive Computation and Machine Learning series), ISBN 0-262-18253-X*, 2006. doi: 10.7551/mitpress/3206.001.0001.

Zhizhou Ren, Jiahan Li, Fan Ding, Yuanshuo Zhou, Jianzhu Ma, and Jian Peng. Proximal exploration for model-guided protein sequence design. In *ICML 2022 (PMLR v162, ren22a)*, 2022. doi: 10.1101/2022.04.12.487986.

Adam J. Riesselman, John B. Ingraham, and Debora S. Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods 15 (10): 816-822*, 2018. doi: 10.1038/s41592-018-0138-4.

Sahoo, Subham Sekhar, et al. Simple and effective masked diffusion language models. In *NeurIPS 2024*, 2024.

Karen S. Sarkisyan, Dmitry A. Bolotin, Margarita V. Meer, Dinara R. Usmanova, Alexander S. Mishin, George V. Sharonov, Dmitry N. Ivankov, Nina G. Bozhanova, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 2016. doi: 10.1038/nature17995.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE, 104(1): 148-175*, 2016. doi: 10.1109/JPROC.2015.2494218.

Song, Yang, et al. Score-based generative modeling through stochastic differential equations. In *ICLR 2021*, 2021.

Tyler N. Starr, Allison J. Greaney, Sarah K. Hilton, Daniel Ellis, Katharine H. D. Crawford, Adam S. Dingens, Mary Jane Navarro, John E. Bowen, et al. Deep mutational scanning of sars-cov-2 receptor binding domain reveals constraints on folding and ace2 binding. *Cell*, 2020. doi: 10.1016/j.cell.2020.08.012.

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *ICML 2021 (PMLR v139)*, 2021.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. In *ICML 2022 (PMLR)*, 2022.

Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J. Weinstein, Niall M. Mangan, Sergey Ovchinnikov, et al. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 2023. doi: 10.1038/s41586-023-06328-6.

Rubo Wang, Fandi Wu, Xingyu Gao, Jiaxiang Wu, Peilin Zhao, and Jianhua Yao. Iggm: A generative model for functional antibody and nanobody design. In *ICLR 2025*, 2025. doi: 10.1101/2024.09.19.613838.

Joseph L. Watson, David Juergens, N. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, A. J. Borst, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 2023. doi: 10.1038/s41586-023-06415-8.

James T. Wilson, Frank Hutter, and Marc Peter Deisenroth. Maximizing acquisition functions for bayesian optimization. In *NeurIPS 2018*, 2018.

Table 2: The frozen seven-dimension verdict matrix (frozen protocol): pre-registered criteria and the division of labor. FC-JD meets the 2 generation criteria (*valid*, *time*); *mean* and *hits@1%* measure what optimization was designed for, while *hits@0.1%*, *unique*, and *P+R* are FC-JD's own bars that it missed. The frozen verdict is 2/7. CIs are seed-level percentile bootstrap 95%, 100 rounds per cell.

Criterion	Verdict	Evidence
valid	PASS	FC-JD valid 60.7 [59.8, 61.5] vs. BO96 26.8 [26.0, 27.7] (CB6 $K{=}5$); 45.8 [45.0, 46.6] vs. 28.2 [27.4, 29.0] ($K{=}7$)
time	PASS	FC-JD $\leq$ 5 s and enlarged baselines $\geq$ 15 s on $4/5$ landscapes (DHFR $K{=}7$ misses SA960 bar by 0.08 s; reported)
mean	FAIL	SA96 best mean on $4/4$ oracle-evaluable landscapes ($\Delta{-}0.21$ to -0.42, FC-JD minus SA96)
hits@1%	FAIL	single positive cell: DHFR $K{=}5$, P+R 0.731 vs. SA96 0.384 (Wilcoxon $p < 0.05$)
hits@0.1%	FAIL	pre-registered criterion (generative family non-zero, SA zero, all K) fails; SA96 non-zero at CR9114 $K{=}5$, CH65 $K{=}7$, DHFR $K{=}7$
unique	FAIL	CB6 (80.8/85.7) and DHFR (69.7/82.1) miss the $\geq$ 90 bar; CR9114/CH65 pass (93.0–95.4)
P+R	FAIL	mean gains on CR9114/CH65 but diversity collapse on CB6/DHFR; criterion passes $2/5$

Table 3: Mean hits@0.1% over 100 rounds ($B{=}96$) on the four oracle-evaluable landscapes, reported in units of 10^{-3}. Each entry is the round-averaged mean with its seed-level percentile bootstrap 95% CI in brackets. A dash marks cells in which every round was zero, so no nontrivial interval is defined. CB6 is a partial-oracle case study; GFP is omitted because its exact-K generation is not oracle-evaluable.

Landscape	K	SA96	BO96	FC-JD	P+R
CR9114-h1	5	10.1 [2.0, 20.0]	4.7 [3.0, 6.0]	5.6 [4.0, 7.0]	7.1 [6.0, 9.0]
	7	0.0 [–]	1.5 [1.0, 2.0]	1.9 [1.0, 3.0]	2.3 [1.0, 3.0]
CH65-MA90	5	0.0 [–]	0.8 [0.0, 2.0]	7.1 [6.0, 9.0]	6.9 [5.0, 9.0]
	7	8.3 [3.0, 15.0]	4.5 [3.0, 6.0]	3.0 [2.0, 4.0]	6.1 [5.0, 8.0]
CB6[a]	5	0.0*[–]	1.1 [0.0, 3.0]	7.9 [6.0, 10.0]	6.4 [4.0, 9.0]
	7	0.0*[–]	3.4 [2.0, 6.0]	6.6 [5.0, 9.0]	8.2 [6.0, 11.0]
DHFR	5	0.9 [0.0, 2.0]	1.7 [1.0, 3.0]	2.7 [1.0, 4.0]	0.7 [0.0, 2.0]
	7	38.5 [29.0, 49.0]	11.6 [7.0, 17.0]	5.5 [4.0, 7.0]	15.3 [12.0, 19.0]

[a] CB6 has partial oracle coverage: SA96 has 20 evaluable rounds at $K{=}5$ and 71 at $K{=}7$, so its values are computed over the evaluable rounds only and are not directly comparable to the complete-oracle landscapes.
* Partial-oracle round; see note (a).

A PRE-REGISTERED VERDICT MATRIX AND THE FULL TAIL TABLE

Table 2 reproduces the seven-dimension verdict matrix frozen in the pre-registered plan (released with the code) and evaluated under the frozen protocol: 5 landscapes $\times$ 4 arms $\times$ $K \in \{5, 7\}$ $\times$ $10{\times}10$ seeds $\times$ 7 metrics (420 cells, 40,000 evaluated rounds). Verdict: **2/7**.

Both pre-registered contingencies were executed (the "near-optimizer mean" claim is withdrawn; P+R is demoted to optional strategy), and excluding the oracle-infeasible GFP landscape leaves the verdict at 2/7, reported as a robustness check. Table 3 reports the complete hits@0.1% results for the four oracle-evaluable landscapes, one row per (landscape, K) pair and values scaled by 10^3; CB6 is flagged as partial-oracle, and GFP is omitted because its exact-K generation is not oracle-evaluable.

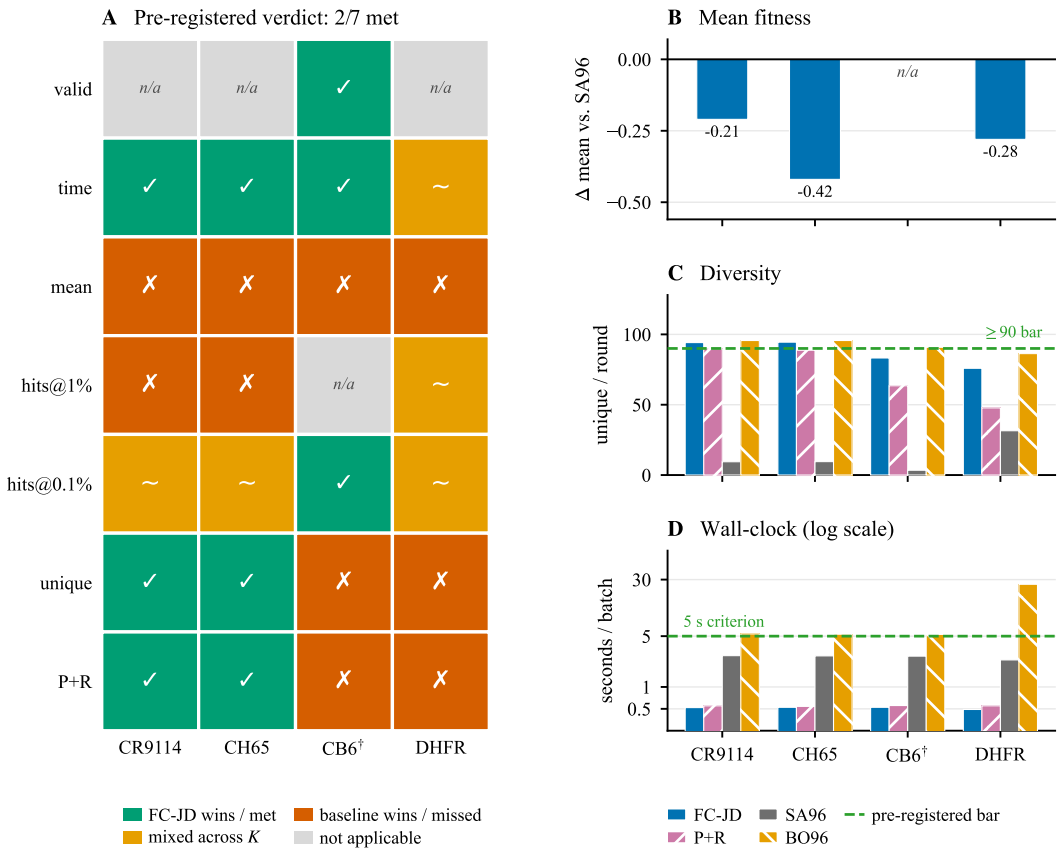


Figure 3: **Benchmark results separate generation quality from fitness optimization.** **(A)** Verdict matrix on the 7 pre-registered criteria (rows) across the 4 oracle-evaluable landscapes (columns): green ✓ = FC-JD wins or the criterion is met, amber ∼ = mixed across K, vermillion × = baseline wins or the criterion is missed, grey = not applicable; the frozen verdict is 2/7 (Table 2). Columns: CR9114, CH65, and DHFR are complete oracles; CB6† is a partial-oracle case study included to measure validity under partial coverage. **(B)** Mean-fitness gap, FC-JD minus SA96: SA96 leads on all four landscapes. **(C)** Unique designs per round, pooled over $K{=}5/7$: FC-JD returns 7–33× SA96's count on the antibody landscapes and meets the pre-registered $\geq$ 90 bar on CR9114 and CH65. **(D)** Wall-clock seconds per $B{=}96$ round on a log axis (medians from Table 5): FC-JD delivers sub-2-second batches. Panels **(B)**–**(D)** share the x axis.

Table 4: K-ladder hits@0.1% (means; per-K arrays and CIs released with the code). "overlap" = overlapping bootstrap CIs at $K7$.

Landscape	$K5$	$K6$	$K7$	$K8$	$K9$
CH65 FC-JD	0.0061	0.0043	overlap	0.0021	0.0020
CH65 BO384	0.0003	0.0005	overlap	0.0095	0.0045
CR9114 FC-JD vs. BO384		FC-JD leads on $4/5$ of $K5$–9			

B K-LADDERS AND SUPPORTING REGIMES

K-ladders. Table 4 extends the $B{=}96$ tail comparison along $K{=}5$–9 with the enlarged BO384 budget (seed counts noted where they differ from the $10{\times}10$ protocol). CH65 splits by cardinality: FC-JD wins $K5$–6, BO384 wins $K8$–9, and the two overlap at $K7$; on CR9114 FC-JD leads BO384 on $4/5$ of $K5$–9, with no crossover. Full per-K arrays are released with the code.

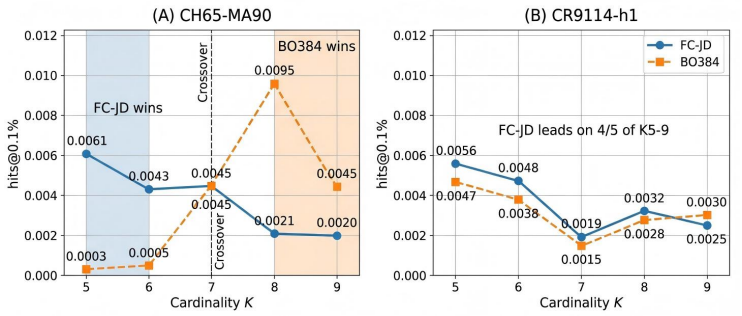


Figure 4: **K-ladder reveals cardinality-dependent tail performance.** Mean hits@0.1% versus cardinality K for FC-JD (blue circles, solid line) and BO384 (orange squares, dashed line) on two antibody landscapes. **(A)** CH65-MA90: FC-JD wins the extreme tail at $K5$–6 ($0.0061/0.0043$ vs. $0.0003/0.0005$), overlaps at $K7$ (means 0.0033 vs. 0.0026, overlapping bootstrap CIs), and BO384 wins $K8$–9 ($0.0095/0.0045$ vs. $0.0021/0.0020$). **(B)** CR9114-h1: FC-JD leads BO384 on $4/5$ of $K5$–9, with no crossover; the pattern is landscape-dependent. K is discrete, so connected lines are visual aids. Full per-seed arrays and bootstrap CIs are released with the code (Table 4).

Table 5: Wall-clock seconds per $B{=}96$ round (median of 100 rounds), listed as $K{=}5/K{=}7$. SA960/BO384 are three repeats for the cost criterion. All arms make zero oracle calls during search; BO384 on DHFR exceeds 2 minutes. SA960 plus deduplication (seconds-level overhead) restores diversity on DHFR (59.6–85.0 unique) but remains collapsed on CR9114/CH65 (8.4–13.1 unique; Section 5).

Landscape	FC-JD	P+R	SA96	BO96	SA960	BO384
CR9114-h1	0.51/0.53	0.55/0.55	2.70/2.69	5.38/5.36	21.91/21.07	28.25/27.98
CH65-MA90	0.53/0.52	0.55/0.53	2.73/2.62	5.31/5.31	21.43/21.70	27.88/28.03
CB6	0.53/0.52	0.55/0.56	2.69/2.62	5.31/5.25	20.25/20.43	27.22/27.41
GFP	1.68/1.68	6.63/6.69	4.36/4.22	8.59/8.48	30.42/32.19	50.16/50.78
DHFR	0.49/0.49	0.54/0.56	2.37/2.34	22.41/29.24	16.01/14.92	96.68/125.96

D0 (unconstrained masked diffusion). Trained on the same $K \leq 3$ rows with masked cross-entropy and T-step ancestral sampling, D0 reaches 23–26% exact-K at $K{=}4$ and 0% at $K{=}5$–9 on DHFR (range over 3 training seeds); the exact-K rate is measured post-hoc on the generated batch (Section 1 and Section 5.1).

ESM-2 conditioning (small-sample regime). Conditioning the position stream on ESM-2 embeddings of the input sequence amplifies the small-sample regime: the generation Spearman gap over the vanilla arm grows monotonically as training shrinks, from $+0.21$ at 2% to $+0.38$ at 20% training on UBE4B, with consistent significant gains on BBC1 and YNZC (post-hoc regularities; Section 6).

Cross-target transfer. Pre-training the position stream on CR9114 and fine-tuning on a 58-sequence CR6261 target beats from-scratch training by $+0.19$ position-marginal correlation ($+0.18$–$+0.29$ across $K{=}4$–7).

Bin ablation. The conditioning label uses 5 bins with edges $(0.5, 0.8, 0.9, 0.95)$; a $3/5/10$-bin ablation confirms five bins.

C PER-CELL DATA

All per-round arrays (mean/hits@1%/hits@0.1%/unique/valid/wall-clock for every one of the 420 cells, including the enlarged-budget runs) are released as JSON with the code; the headline cells are cited in Section 5 and reproduced in Tables 2, 3, and 5.

Algorithm 2 Simulation-free training of the position stream

Require: training masks m_0 with cardinalities K and bin labels y; pre-computed forward kernels $P_1, \dots, P_T$ for $K{=}1, \dots, 8$; $T{=}100$
Ensure: position model θ (remove logits R, add logits A, stay logit s)

for epoch $= 1, \dots, 120$ **do**
  sample a mini-batch of (m_0, K, y) from the training set
  for each (m_0, K, y) in the batch **do**
    $t \sim \mathcal{U}\{1, \dots, T\}$; $m_t \sim P_t(\cdot \mid m_0)$ ▷ direct kernel draw; no forward rollout
    $j \leftarrow |m_t \cap m_0|$; compute the analytic posterior w_c (Eq. 2) and the model mixture q_c (Eq. 3) from $\sigma(s)$ and $S_1, \dots, S_4$
    if $t > 1$ **then**
      $\mathcal{L} \leftarrow \mathrm{KL}(w \| q)$
    else
      if $m_t \neq m_0$ **then**
        $\mathcal{L} \leftarrow -\log(1{-}\sigma(s)) - \log p_R(i_{\mathrm{rem}}) - \log p_A(i_{\mathrm{add}})$
      else
        $\mathcal{L} \leftarrow -\log \sigma(s)$
      end if
    end if
    $\mathcal{L} \leftarrow \big(1 + 3\mathbb{1}[y{=}4]\big)\,\mathcal{L}$ ▷ focal weight on the top bin
  end for
  update θ by gradient descent (AdamW) on the batch loss
end for

D LIMITATIONS: SUPPORTING EVIDENCE

GFP structural diagnosis. Random sampling confirms the structural cause of GFP oracle-infeasibility: $0/5{,}000$ uniformly sampled $K{=}5$ masks followed by the fill rule hit the assay, and only $3/2{,}000$ assayed $K{=}5$ rows are fill-rule-compatible. GFP is retained for regression evaluation and protocol consistency only.

CB6 partial oracle. SA96's optima lie predominantly outside the assayed subset: only 20/71 evaluable rounds (out of 100) at $K{=}5/7$; mean and hits are undefined in the remaining rounds. FC-JD achieves 46–61 valid designs per round (48–63% valid rate) on the same landscape.

Prospective framework. The epi-proxy framework was evaluated in three pre-registered rounds but did not yield reliable predictions ($3/15$ positive confirmations; in the third round all six predicted-WIN datasets lost to the additive baseline, $0/6$, while the ESM-2-conditioned arm reached $8/9$ on the same datasets, disclosed as arm-selection-influenced). All positive results in this paper are therefore validated post-hoc regularities rather than predicted outcomes.

Pipeline concentration. The antibody evidence is concentrated in the Desai-lab Tite-Seq measurement pipeline (4 of 7 antibody/landscape data sources), with only one cross-pipeline dataset each (Wagner, Arnold, Toprak); pipeline-independent generalization remains open. The in-sample record ($20/20$ across internal, external, and prospective rounds) reflects arm selection within the project; the paper's claims rely on the frozen out-of-sample verdicts (Appendix A).

E METHOD DETAILS

The full pseudocode for the simulation-free training objective (Section 3.2) and the complete two-stream sampling procedure (including the DHFR residue stream) are given below.

Kernel computation and mixing. The per-step kernels $P_t = \prod_{s=1}^{t}(I - \beta_s \mu)$, $\mu = I - \mathrm{diag}(\pi)^{-1/2} L\, \mathrm{diag}(\pi)^{1/2}$, $\pi_j = M_j / \binom{N}{K}$ with $M_j = \binom{K}{j}\binom{N-K}{K-j}$ (Eq. 1), are computed in closed form by eigendecomposition of the symmetrized operator, verified against analytic eigenvalues (er-

Algorithm 3 FC-JD sampling: complete two-stream procedure

Require: position model θ_{pos}; residue model θ_{res} (token-valued landscapes only); reference r; cardinality K; top bin $y{=}4$; steps T; batch size B
Ensure: batch of B designs with exactly K mutations
$\mathcal{B} \leftarrow \emptyset$
for $i = 1, \ldots, B$ **do**
  $m_T \sim \text{Uniform}(K\text{-subsets of } \{1, \ldots, N\})$; token-valued landscapes: $x_T \leftarrow r$ with the positions in m_T replaced by MASK
  for $t = T, \ldots, 1$ **do**
    Position stream: keep m_t with probability $\sigma(s)$; otherwise $m_{t-1} \leftarrow (m_t \backslash \{i\}) \cup \{j\}$ with $i \sim \text{softmax}(R|_{m_t})$, $j \sim \text{softmax}(A|_{\bar{m}_t})$ ▷ $|m_{t-1}|{=}K$ by construction
    (DHFR: swap with per-step probability p_t from the schedule, Gumbel top-1)
    if token-valued landscape **then**
      Residue stream: mutated, still-masked positions keep MASK with probability $\gamma_t{=}(1{-}\bar{\alpha}^{\text{res}}_{t-1})/(1{-}\bar{\alpha}^{\text{res}}_{t})$; otherwise the residue is sampled at temperature 0.3; at the final step all mutated positions are decoded (reference residue excluded); positions outside the mask carry r
    end if
  end for
  $\mathcal{B} \leftarrow \mathcal{B} \cup \{x_0\}$
end forreturn $\mathcal{B}$

ror $< 10^{-9}$); the linear retention schedule $\rho_1(t) = 1 - t/T$ with relaxation rate $\mu_1 = N/d$ mixes to the hypergeometric distribution over $J(N, K)$ (TV $< 10^{-6}$ at $t = T$).

Reverse-action classes. The five reverse-action classes entering the analytic posterior (Eq. 2), with (reverse action; new overlap j'_c; count): stay $(j; 1)$, remove-shared/add-in-m_0 $(j;\ j(K-j))$, remove-m_t-only/add-neither $(j;\ (K-j)u)$, remove-shared/add-neither $(j-1;\ ju)$, remove-m_t-only/add-in-m_0 $(j+1;\ (K-j)^2)$, with $u = N - 2K + j$.

Residue-stream details. At mutated positions, the content is diffused by an absorbing-state (masked) process with cosine schedule: position i is masked at step t with probability $1 - \bar{\alpha}^{\text{res}}_t$, and a second transformer (vocabulary $|\mathcal{V}| + 1$ including a mask token) is trained with cross-entropy on masked positions only. The reference is the per-position marginal argmax of single-mutant fitness (reproducible from $\mathcal{D}$); on GFP the pre-registered fill rule (per-position argmax single-mutant DMS) completes generated masks.

FC-JD implementation settings. Both streams use a transformer with $d{=}128$, 4 layers, 4 heads, trained 120 epochs with AdamW (lr$=2 \times 10^{-4}$), batch 512 (GFP: batch 64, 8,192 rows/epoch). $T{=}100$; kernels are pre-computed (error $< 10^{-9}$; Algorithm 2). Conditioning: t (128-dim sinusoidal), K, and the 5-bin fitness label y; sampling conditions on the top bin ($y{=}4$) with focal weight $1{+}3\mathbb{1}[y{=}4]$ during training. The residue stream (DHFR) uses vocabulary $|\mathcal{V}|{+}1{=}5$ (4 nucleotides + mask), a cosine absorbing schedule, and cross-entropy on masked positions only; unmutated positions are pinned to the reference, and the reference is the per-position marginal-argmax single mutant.