A System One Approach to General Protein Evolution

Yibo Wen[1], Han Liu[1]

[1]*Northwestern University*

Abstract. Protein engineering requires deciding which mutation to test next and when to stop. Existing approaches often require target-family measurements, encode candidates separately, or return scores that are not calibrated to the assay. We introduce **Pev**, a general mutation policy that explores a System One approach to protein evolution: learning from experiments across families to make decisions in a single parallel query. Given a parent sequence and an assay context, Pev encodes the parent once and returns calibrated probabilities for choosing an edit or stopping, together with each edit's probability of improving the assay outcome. One policy covers peptides, small folded domains, and antibodies across binding and stability objectives, without fitting model weights to the target family. In ten-well screens on held-out families, Pev finds 3.3× as many beneficial mutations as random selection and 1.8× as many as zero-shot ESM-2. It also outperforms a ridge surrogate refitted on target measurements in closed-loop search on the held-out GB1 landscape. Post-hoc calibration improves probability reliability without changing decisions, while shared encoding provides over 800× the throughput of per-mutant encoding at 1,000 candidates per request. These results suggest that a System One approach could support efficient experimental decisions across protein families, with Pev providing an initial demonstration.

Correspondence: hanliu@northwestern.edu

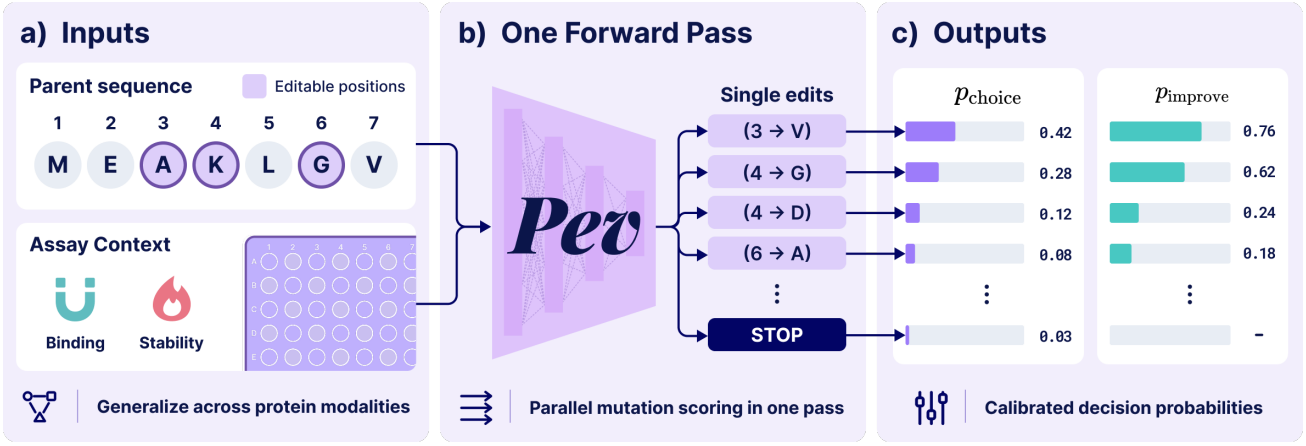


Figure 1 ***Overview of Pev.*** *Given a parent sequence and an assay context, Pev reuses one parent encoding to score all offered substitutions and STOP in parallel. Calibrated choice and improvement probabilities guide which edit to test and when to stop, supporting a System One approach to general protein evolution.*

1 Introduction

Protein engineering requires a sequence of experimental decisions under a limited assay budget. Directed evolution proceeds through rounds of mutation and selection (Romero and Arnold, 2009; Packer and Liu, 2015), with each round requiring a choice of which variants to test and whether further experiments are worthwhile. These decisions are difficult because the effect of a mutation depends on the surrounding sequence (Starr and Thornton, 2016; Wu et al., 2016). Deep mutational scanning measures thousands of substitutions at once (Fowler and Fields, 2014), but each scan serves one target and objective, so a new campaign still spends its budget a few variants at a time. Effective guidance can nevertheless make small experimental budgets productive: language-model guidance enabled affinity maturation of seven antibodies using at most twenty variants per antibody over two rounds (Hie et al., 2024). A policy that transfers knowledge across protein families and assay objectives could help new campaigns make informed choices from their first round.

Existing approaches provide useful predictions but leave a gap between scoring variants and making assay-specific decisions. Family-specific surrogates learn from measurements of the target family (Yang et al., 2019; Wu et al., 2019; Biswas et al., 2021; Jiang et al., 2025; Yang et al., 2025), using methods such as Gaussian processes and adaptive optimization to guide subsequent experiments (Romero et al., 2013; Hie and Yang, 2022). Evolutionary information and pretrained representations improve learning when measurements are scarce (Wittmann et al., 2021; Hsu et al., 2022), but these surrogates still require target-family data. Zero-shot protein language models can score variants without such measurements (Meier et al., 2021; Notin et al., 2022, 2023a), although their scores are not directly calibrated to assay outcomes and their accuracy can depend on their preference for the wild-type sequence (Gordon et al., 2025). Preference alignment adapts generative models to experimental or energy-based objectives (Rafailov et al., 2023; Widatalla et al., 2026; Wen et al., 2025), but does not by itself provide calibrated probabilities for choosing among offered edits or stopping.

Together, these limitations motivate a decision model that transfers across protein families and expresses its predictions as assay-relevant probabilities. We approach protein evolution from the System One perspective: learning to return a structured decision in a single parallel query. Jev (TypeSafe AI, 2026) exemplifies this approach by mapping structured state to typed outputs with calibrated probabilities. Protein engineering offers a natural decision structure: a parent sequence, an assay context, and a finite set of candidate edits. A zero-shot language model also reads the parent once, so a single pass is not what sets a decision model apart; what sets it apart is learning experimental decisions from measured outcomes across families, weighing every offered edit against STOP in one query, and reporting probabilities calibrated to assay events. Scoring every candidate from the shared parent representation also spares the backbone pass that encoding each mutant separately needs for every candidate.

We introduce **Pev**, a general mutation policy that selects an edit or STOP in one such query. Given a parent sequence and an assay context, it encodes the parent once and returns a choice distribution p_{choice} over the offered edits and STOP, and an improvement probability p_{improve} for each substitution. Pev trains approximately 1.6 million parameters with a frozen ESM-2 backbone (Lin et al., 2023), learning from measured outcomes across families rather than fitting weights to the family being screened. Post-hoc calibration uses separate held-out families and preserves action choices and within-panel rankings.

We summarize our **main contributions** as follows:

1. **A System One decision model.** Pev formulates protein evolution as choosing among offered edits and STOP, scoring all actions in one parallel query from a shared parent encoding.
2. **Calibrated decision probabilities.** Separate choice and improvement outputs distinguish which action is best among those offered from whether an edit improves the parent.
3. **Transfer across protein families.** A single policy learns across peptides, small folded domains, and antibodies under binding and stability objectives, without any family-specific fine-tuning.

We evaluate Pev on held-out protein families and a held-out fitness landscape, demonstrating improved experimental selection over the evaluated baselines and substantially higher scoring throughput than per-mutant encoding. Further discussion of related work is provided in App. A.

2 Problem Formulation

This section describes the decision that Pev makes in each round of protein engineering. Section 2.1 introduces the parent sequence, the assay context and the edits on offer. Section 2.2 defines how an edit is judged against the parent once its outcome is measured. Section 2.3 specifies the two probabilities that Pev returns for every decision and how they are used to select edits and to stop.

2.1 Decision Setting

Parent and context. Each decision concerns a parent sequence in an assay context. The parent $x \in \mathcal{A}^L$ is a protein sequence of length L over the twenty standard amino acids $\mathcal{A}$, and it belongs to a protein family. The context c states the objective, binding or stability, and the readout, such as $\Delta\Delta G$, melting temperature T_m, affinity, or a binding signal. For binding it may also include the sequences of a target and a partner. Within a context, a measured variant x' has an outcome $y_c(x')$, oriented so that higher is better, and a replicate standard deviation $s_c(x')$ when replicates exist. Outcomes are compared only within the same context.

Offered edits. The policy chooses among single substitutions. An edit $a = (i, r)$ replaces the residue at position i with a different amino acid r, and $x \oplus a$ denotes the edited sequence. The offered panel $\mathcal{O}(x, c)$ is the set of edits available to the policy. In a retrospective screen it contains the substitutions that were measured for the parent, so unmeasured mutations are left out rather than counted as misses. In a campaign it contains the substitutions that have not yet been assayed. The policy may also choose STOP, which declines the offered edits and retains the best measured sequence. Together, the offered edits and STOP form the action set $\mathcal{O}(x, c) \cup \{\mathsf{STOP}\}$.

2.2 Judging an Edit

Utility. An edit is judged by how much it changes the measured outcome. Its utility is the gain over the parent, and its noise scale combines the replicate standard deviations of the two sequences:

$$u_a = y_c(x \oplus a) - y_c(x), \qquad \sigma_a = \sqrt{s_c(x \oplus a)^2 + s_c(x)^2}. \tag{1}$$

STOP leaves the parent unchanged, so $u_{\mathsf{STOP}} = 0$ and $\sigma_{\mathsf{STOP}} = 0$.

Improving and beneficial edits. Two events are used throughout the paper. An edit *improves* the parent when $u_a > 0$, which is the event forecast by the improvement probability below. It is *beneficial* when its gain also exceeds the noise scale and a readout-specific effect floor δ_ρ:

$$u_a > \max\left(\sigma_a,\ \delta_\rho\right). \tag{2}$$

The floor is fixed for each readout before any model is scored (App. C). For $\Delta\Delta G$, T_m, and affinity it is a conventional effect size, and for binding signals it is half the robust spread of the gains measured in the panel. Beneficial edits define a hit in the discovery screen (Section 4.2).

Best action. Measurement noise also means that the best action is not always unique. We take the best action $a^\star = \arg\max_j u_j$ over $\mathcal{O}(x, c) \cup \{\mathsf{STOP}\}$ and collect in the target set $\mathcal{T}$ every action whose utility is within noise of the best utility $u_{a^\star}$:

$$\mathcal{T} = \left\{ j :\ u_{a^\star} - u_j \le \sqrt{\sigma_j^2 + \sigma_{a^\star}^2} \right\}. \tag{3}$$

Since $u_{\mathsf{STOP}} = \sigma_{\mathsf{STOP}} = 0$, STOP belongs to $\mathcal{T}$ exactly when the highest-gain edit does not exceed the parent beyond its own noise scale, so STOP is the right choice when no offered edit is reliably better than the parent.

2.3 Policy Outputs

Two probabilities. Given the parent, the context and the offered panel, the policy returns two kinds of probability:

$$p_{\text{choice}} \in \Delta\big(\mathcal{O} \cup \{\mathsf{STOP}\}\big), \qquad p_{\text{improve}} \in [0, 1]^{\mathcal{O}}. \tag{4}$$

The choice distribution p_{choice} gives each action the probability of being the best on offer, with the probability shared among the actions tied within noise. It includes STOP and sums to one. The improvement probability p_{improve} gives each edit the probability of improving the parent, $u_a > 0$. Several edits can improve the same parent, so these probabilities need not sum to one. The two outputs answer different questions: p_{choice} sets the edits and STOP against each other, while p_{improve} asks of each edit alone whether it is worth trying.

Use in practice. The outputs serve both selection and ranking. The policy selects $\hat{a} = \arg\max p_{\text{choice}}$, which is an edit or STOP. A ten-well screen assays the ten edits with the highest choice probabilities, and a campaign orders each batch by improvement probability. STOP carries one meaning in each setting. In a one-step evaluation it declines to spend the assay, since no offered edit is reliably better than the parent, and in a campaign it retires the current parent once the batch in hand has been assayed, so the search continues from another parent until the budget is spent. Section 3 describes how Pev computes both outputs from one encoding of the parent.

3 Pev for General Protein Evolution

Pev turns the decision of Section 2 into a single forward pass over the parent. Three design choices define it. Parent reuse scores every candidate from one encoding of the parent, the decision objective trains the choice distribution on which action is best in place of a regression onto each edit's gain, and the improvement loss gives every edit its own probability of improving the parent. Sections 3.1 and 3.2 describe how one encoding yields both outputs, Section 3.3 explains how a single policy is trained on measured scans from many families, and Section 3.4 adds post-hoc calibration. Section 4.7 removes each design choice in turn.

3.1 One Encoding for Every Edit

Frozen backbone. Pev encodes the parent once and scores every offered edit from that encoding. The backbone is ESM-2 650M (Lin et al., 2023), kept frozen, and its encoding $h = \mathrm{Enc}(x)$ holds one state per residue of the parent. Pev trains only the context and residue embeddings and two small networks on top of the backbone, a candidate multilayer perceptron (MLP) and a STOP head, which together hold about 1.6 million parameters stored in 6 MB.

Candidate features. Each candidate edit $a = (i, r)$ is described by six concatenated parts:

$$\phi_a = \big[\, h_i;\ \bar{h};\ e_r;\ q_c;\ e_{x_i};\ \ell_a \,\big]\ , \tag{5}$$

$$\ell_a = \big[\log p_{\mathrm{LM}}(r \mid x, i) - \log p_{\mathrm{LM}}(x_i \mid x, i),\ \log p_{\mathrm{LM}}(x_i \mid x, i)\big]. \tag{6}$$

Here h_i is the parent's state at the edited position, $\bar{h}$ the pooled parent state, and e_r and e_{x_i} embed the new and the original residue. The context embedding q_c covers the objective, the readout and the projected target and partner states, which come from the same frozen backbone, once per sequence. The vector ℓ_a holds the backbone's own language-model log-odds for the edit, read from the unmasked parent at no extra cost, and its first entry is the score that the zero-shot ESM-2 baseline ranks by.

Parent reuse. All candidate features are assembled from a shared parent encoding, the assay context and edit-specific parts that need no encoder of their own, so a decision over N candidates costs one backbone pass plus a light head per candidate (Proposition C.2). Encoding every mutant separately would cost one pass per candidate. The per-mutant baseline of Section 4 differs from Pev in exactly this one place: h_i and $\bar{h}$ come from encoding $x \oplus a$ separately, and its heads, conditioning and training are the same as in Pev.

3.2 Two Outputs for Two Questions

Two heads. The candidate MLP has two hidden layers and maps each candidate to a choice logit and an improvement logit, $(z_a, v_a) = \mathrm{MLP}(\phi_a)$. A separate head with one hidden layer maps the pooled state and the context to the STOP logit z_{STOP}. With the choice logits $z = \big([z_a]_{a\in\mathcal{O}},\, z_{\mathsf{STOP}}\big)$, the two outputs are

$$p_{\mathrm{choice}} = \mathrm{softmax}(z), \qquad p_{\mathrm{improve}}[a] = \sigma(v_a), \tag{7}$$

where σ is the logistic sigmoid. The choice head is a softmax over the edits and STOP, so it sums to one, and the improvement head is an independent sigmoid for each edit, so its outputs do not sum to one. Neither substitutes for the other: p_{choice} sets the edits and STOP against each other, while p_{improve} asks of each edit alone whether it improves the parent. Because the output space is the closed set of offered edits and STOP (Figure 1), nothing is generated, and every output is a valid decision that a laboratory can act on.

3.3 Training on Wild-type Scans

Episodes. Pev trains on wild-type scans, the kind of panel that a deep mutational scan produces. An episode is a measured reference plus up to 64 of its measured single substitutions, resampled each time the reference is drawn. It has the same form as a discovery-screen panel, so one encoder pass covers every candidate in it. Because the options have the same shape in every family, a set of single substitutions plus STOP, scans from many families train one policy.

Targets and losses. Each head is trained on a target that respects measurement noise. The choice target p^* is uniform on the target set $\mathcal{T}$ of equation (3), which holds the best action and every action within noise of it. The improvement target y_a^* is a noise-aware soft label for the event $u_a > 0$ (App. C). The decision objective is the cross-entropy of the choice distribution to p^*, and the improvement loss is the binary cross-entropy (BCE) of each improvement probability to its label, averaged over the offered edits:

$$\mathcal{L}_{\mathrm{choice}} = -\sum_j p_j^* \log \mathrm{softmax}(z)_j, \qquad \mathcal{L}_{\mathrm{improve}} = \tfrac{1}{|\mathcal{O}|}\sum_{a\in\mathcal{O}} \mathrm{BCE}\big(\sigma(v_a), y_a^*\big). \tag{8}$$

Objective. The two terms are combined as $\mathcal{L}(\theta) = \mathbb{E}_\omega[w_\omega(\mathcal{L}_{\mathrm{choice}} + \lambda\,\mathcal{L}_{\mathrm{improve}})]$ with $\lambda = 1$, where θ holds the trained parameters, ω is an episode, and the weight $w_\omega \le 1$ lowers the contribution of an episode whose best action lies within noise (App. C). Both terms are strictly proper scoring rules, so each head is minimized only at

the weighted conditional mean of its own training target (Proposition C.1), the noise-aware soft label rather than the event itself. Properness therefore fixes what each head estimates, and the reported numbers are matched to the frequencies of a new assay by the calibration of Section 3.4. Censored measurements never enter either loss, and losing candidates stay in the denominator of the choice term. Training details are given in App. D.

3.4 Post-hoc Calibration

Calibration maps. Training fits the logits to noise-aware targets, so the raw probabilities need not match the frequencies observed in a new assay. Post-hoc calibration fits three parameters for each objective on held-out calibration families, against hard labels for the two events rather than the soft targets used in training (App. C): a temperature $T > 0$ that scales all choice logits within a decision, STOP included, and an affine map with positive slope α and intercept β on the improvement logits. The calibrated outputs are

$$p_{\text{choice}} = \text{softmax}(z/T), \qquad p_{\text{improve}}[a] = \sigma(\alpha v_a + \beta), \tag{9}$$

and training itself uses $T = \alpha = 1$ and $\beta = 0$.

Decisions are preserved. Calibration changes how confident the probabilities are and not which action is chosen. Both maps are monotone, so they cannot reorder the logits, which is what the following proposition states.

Proposition 3.1 (Calibration preserves choices and rankings)**.** Let $T > 0$, $\alpha > 0$, and β be the calibration parameters for a decision's objective. Then

$$\arg\max_j \text{softmax}(z/T)_j = \arg\max_j z_j,$$

and $\sigma(\alpha v_a + \beta)$ orders the offered edits exactly as v_a does. Thus, with the same tie-breaking rule, calibration preserves the selected action $\hat{a}$, the within-panel rankings under both outputs, and every metric that depends only on these choices and rankings (App. C). Metrics that use the probability values themselves, including expected calibration error (ECE), may change.

Proof. Dividing by $T > 0$ preserves the order of the choice logits, and softmax preserves their pairwise ordering. The map $v \mapsto \sigma(\alpha v + \beta)$ is strictly increasing because $\alpha > 0$ and the logistic sigmoid is strictly increasing. Both transformations therefore preserve rankings, including ties. □

4 Experiments

We evaluate whether Pev's decisions are useful, whether its probabilities can be trusted, and whether one encoding of the parent is efficient. Section 4.1 describes the evaluation settings, data and baselines. Sections 4.2 and 4.3 test one-round decisions on held-out protein families, and Section 4.4 tests sequential selection on a held-out fitness landscape. Section 4.5 checks that the predicted probabilities are reliable, Section 4.6 compares selection quality with throughput, and Section 4.7 examines the contribution of each design choice.

Table 1 ***Evaluation settings.*** *One-round screens assess candidate ranking and edit-or-stop decisions, and closed-loop campaigns assess sequential batch selection under an assay budget.*

	One-round screen (Sections 4.2 and 4.3)	Closed-loop campaign (Section 4.4)
Candidates	Measured single substitutions of a reference parent, from full wild-type scans or sparse neighborhoods	All unassayed single edits of the parent, at most 76
Budget	Ten wells (discovery) or one action (edit-or-stop)	64 assays, ordered in batches of 8
Decision	Which edits to prioritize, or whether to select STOP	Which batch to assay next and which parent to use
Metrics	Hit rate, enrichment, AUROC, wells to hit, Success@1	Best observed fitness F_B, normalized to wild type
Held out	Protein family	Fitness landscape
Population	45 panels from 39 of the 165 held-out families	One landscape, over 50 paired initializations
Eligibility	At least 100 measured substitutions of the reference, and one beneficial edit	Single edits of the parent, not yet assayed

4.1 Experimental Setup

Settings. The experiments use two evaluation settings. In a one-round screen, Pev ranks the measured single substitutions of a parent and is scored against their measured outcomes, either on a ten-well budget (Section 4.2) or one decision at a time (Section 4.3). In a closed-loop campaign (Section 4.4), Pev orders batches of unassayed edits under an assay budget, and the campaign is scored by the best sequence it reaches. In neither setting are Pev's weights fitted to the target: a one-round screen gives it no measurement of the family at all, and in a campaign the measurements in hand only decide which parent it works from. The two settings also answer different questions, one about the quality of a single ranking and one about where a sequence of rankings arrives.

Data and splits. All evaluations use protein families or a fitness landscape that were held out before training (Table 1). The corpus draws on nine public sources: eight supply the training and evaluation data, and GB1 is reserved for the closed-loop experiment (App. B). Splits are made by family, and a family held out for one property remains held out for every property. The evaluation split contains 165 families and 214,378 variants, and both one-round screens draw on it: the discovery benchmark takes 45 panels from the 39 of those families that its rule admits, and the transfer test takes the sparse neighborhoods of the same split.

Baselines. We compare Pev with methods that need no target-family measurements, a surrogate that learns from campaign measurements, and variants that isolate its design choices. Random selection and zero-shot ESM-2 are the baselines in every screen, where zero-shot ESM-2 ranks edits by the log-odds of the backbone and never selects STOP. The discovery screen includes ESM-1v (Meier et al., 2021), a language model trained for variant effect prediction, scored by masked marginals over its five published checkpoints. The closed-loop experiment adds two comparators refitted on the target's observed assays: a ridge surrogate on one-hot site features, and Bayesian optimization with a Gaussian process over the same features, using a squared-exponential kernel and ranked by expected improvement. Three variants each remove one design choice of Section 3: the per-mutant variant encodes every mutant separately instead of sharing the parent encoding, the regression variant replaces the decision objective with a point prediction trained by mean squared error, and the choice-only variant sets $\lambda = 0$ to remove the improvement loss.

Metrics and statistics. Each subsection defines its metrics where they are first used, and full definitions are given in App. C. Screen comparisons are paired by family, with uncertainty estimated by a family-clustered bootstrap, and closed-loop comparisons are paired by initialization.

4.2 Finding Beneficial Mutations in Held-out Families

Setup. The discovery screen asks how well Pev finds beneficial edits when it can test only ten. A panel is one measured reference protein with its measured single substitutions in one assay context, and Pev orders each panel with its choice head. An edit counts as a hit when it is beneficial in the sense of equation (2). Hit rate@10 is the share of the ten selected edits that are hits, and enrichment@10 divides it by the panel's share of beneficial edits, so that random selection scores one in expectation. We also report the AUROC of the full ranking and the rank of the first beneficial edit, which we call wells to first hit. The panel list is frozen before anything is scored on it.

Table 2 ***Discovery on held-out families.*** *Means over 45 panels from 39 families (27 domain, 7 peptide and 5 antibody) with 33,463 measured candidates and ten wells per panel. Hit@10 and Enr.@10 are hit rate and enrichment at ten wells, and wells to hit is the rank of the first beneficial edit. Perfect ranking is the ceiling that each panel's prevalence sets, and the best of the four methods in each block is in bold.*

	All, 45 panels				Domain, 27 panels			Peptide, 13 panels			Antibody, 5 panels		
Method	Hit @10 ↑	Enr. @10 ↑	AUROC ↑	Wells to hit ↓	Hit @10 ↑	Enr. @10 ↑	AUROC ↑	Hit @10 ↑	Enr. @10 ↑	AUROC ↑	Hit @10 ↑	Enr. @10 ↑	AUROC ↑
Perfect ranking	*0.949*	*18.94×*	*1.000*	*1.0*	*0.941*	*25.47×*	*1.000*	*0.946*	*9.27×*	*1.000*	*1.000*	*8.79×*	*1.000*
Pev	**0.424**	**7.19×**	**0.729**	**6.1**	**0.522**	**10.96×**	**0.828**	**0.246**	**1.32×**	**0.572**	**0.360**	2.16×	0.604
ESM-2	0.231	2.80×	0.545	48.3	0.219	3.73×	0.549	0.215	0.87×	0.506	0.340	**2.81×**	**0.623**
ESM-1v	0.167	2.16×	0.540	53.9	0.141	2.65×	0.531	0.177	1.24×	0.535	0.280	1.90×	0.604
Random	0.128	1.02×	0.497	27.8	0.080	1.05×	0.498	0.199	1.00×	0.495	0.202	0.93×	0.495

Overall results. Pev finds beneficial edits far more often than every baseline (Table 2). Its hit rate at ten wells is 0.424, against 0.231 for zero-shot ESM-2, 0.167 for ESM-1v and 0.128 for random selection, so the same ten wells hold 1.8×, 2.5× and 3.3× as many beneficial edits. ESM-1v is built for variant effects and still trails

zero-shot ESM-2 here: both score how likely a substitution is, not whether it improves the parent. One family can supply several panels, so averaging within each family first, the paired margins in hit rate are 0.218 over zero-shot ESM-2 ([0.099, 0.342], 95%) and 0.285 over ESM-1v ([0.162, 0.408]).

Sequence classes. Pev's discovery gains vary by sequence class (Figure 2). They are largest on domains, with a hit rate of 0.522 across 27 held-out families, versus 0.219 for zero-shot ESM-2 and 0.080 for random selection. The paired margin over random is 0.442 (95% bootstrap interval [0.312, 0.578]). On peptides, Pev leads zero-shot ESM-2 in hit rate, enrichment, and AUROC (0.246 versus 0.215, 1.32× versus 0.87×, and 0.572 versus 0.506). On the 5 antibody panels, Pev has higher hit rate (0.360 versus 0.340), while zero-shot ESM-2 has higher enrichment (2.81× versus 2.16×) and AUROC (0.623 versus 0.604).

Ranking and first hit. Pev achieves enrichment 7.19× and AUROC 0.729 (Table 2), versus 2.80× and 0.545 for zero-shot ESM-2. Its first beneficial edit ranks 6.1 on average, versus 48.3 for ESM-2, 53.9 for ESM-1v and 27.8 for random selection. Full-panel ranks can exceed the ten-well budget of the screen.

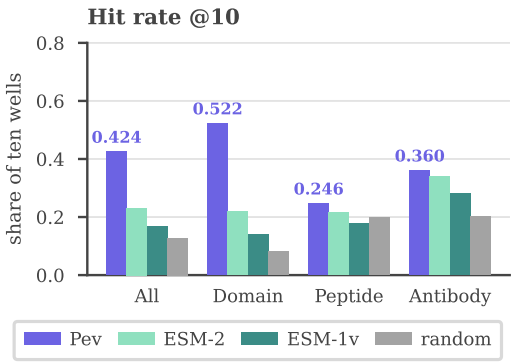


Figure 2 ***Pev leads in hit rate overall and across sequence classes.*** *Hit rate@10 is the fraction of ten selected edits that are beneficial.*

4.3 Transfer to Sparse Measured Neighborhoods

Setup. We evaluate Pev on sparse neighborhoods with a median of sixteen measured single-edit neighbors per parent, roughly half with positive gains. Most panels contain fewer hits than the ten-well budget, so screening metrics saturate; we instead report two budget-independent metrics. An edit clears its noise when its gain exceeds the replicate noise, $u_a > \sigma_a$. Success@1 credits a panel when the selected edit clears its noise or the model chooses STOP when no offered edit does. AUROC evaluates the ranking of all offered edits using the same label. Panels are weighted by class, objective, and family (App. C).

Table 3 ***Transfer to sparse neighborhoods.*** *Pev scores highest on both Success@1 and AUROC.*

Method	Success@1 ↑	AUROC ↑
Perfect decisions	*1.000*	*1.000*
Pev	**0.502**	**0.575**
Zero-shot ESM-2	0.425	0.549
Per-mutant	0.475	0.549
Regression	0.357	0.563
Choice-only	0.437	0.570
Random	0.386	0.517

Results. Pev achieves the highest Success@1 and AUROC among the six evaluated methods (Table 3). Its Success@1 is 0.502, exceeding zero-shot ESM-2 by 18%. Both methods require no measurements from the target family. However, Success@1 credits correct stopping, which Pev and its variants support but zero-shot ESM-2 and random selection do not. AUROC evaluates edit rankings independently of STOP, with Pev scoring 0.575 versus 0.549 for zero-shot ESM-2.

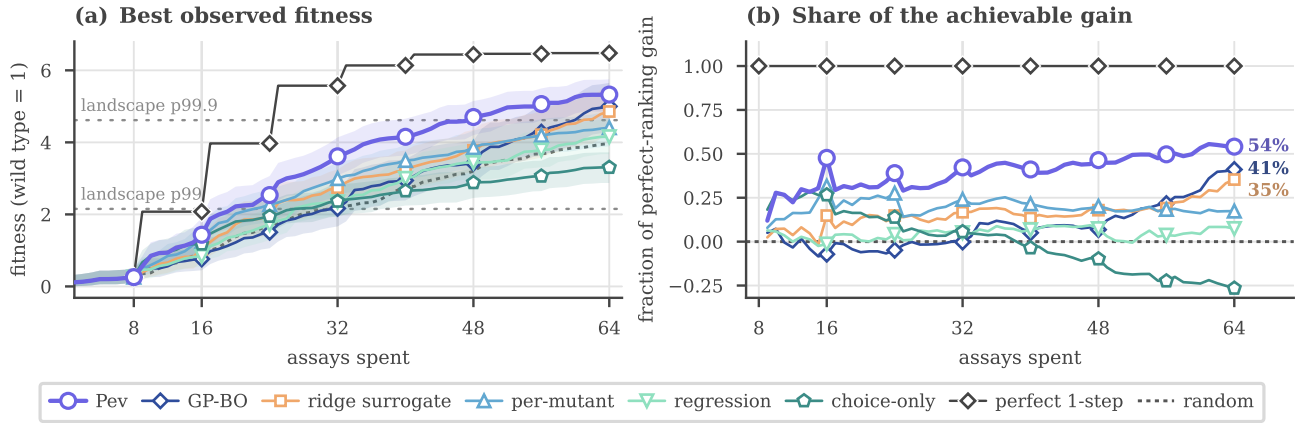


Figure 3 ***Closed-loop campaign on the GB1 four-site landscape.*** *(a) Mean best observed fitness over 50 paired initializations, with bootstrap bands, markers every 8 assays, and grey lines at the landscape's 99th and 99.9th percentiles. (b) Each method's fraction of the gap from random selection to perfect one-step ranking across assay budgets.*

4.4 Closed-loop Protein Optimization

Setup. The closed-loop experiment tests whether Pev's decisions yield better sequences under limited assay budgets. We run campaigns on the held-out GB1 four-site landscape. Each round, the policy selects a batch of 8 from at most 76 unassayed single edits of the current parent. All arms use the full 64-assay budget, including initialization. Pev ranks edits by calibrated improvement probability; STOP retires the parent after its batch, and the campaign continues from another.

Table 4 ***Closed-loop GB1 campaign.*** *Mean best observed fitness after B assays.*

Method	B=16 ↑	B=32 ↑	B=64 ↑
Perfect ranking	*2.08*	*5.57*	*6.48*
Pev	**1.43**	**3.61**	**5.33**
GP-BO	0.76	2.17	5.00
Ridge surrogate	1.03	2.74	4.86
Per-mutant	1.24	2.99	4.41
Regression	0.83	2.35	4.16
Choice-only	1.18	2.36	3.30
Random	0.85	2.17	3.97

Results. The ridge surrogate and the Gaussian process are both fitted to the target's own assays and refit after each round, and all arms share initial sets, batch rules, and budgets. Across 50 paired initializations, Pev achieves higher best observed fitness $F_B = \max_{t \le B} y(x_t)$ than both at all three budgets ($B = 16, 32, 64$), with wild-type fitness set to one (Table 4). Its advantage emerges in the earliest batches, when a model fitted to the target has the least training data. At 64 assays, Pev reaches 5.33 against 4.86 for the surrogate and 5.00 for the Gaussian process.

Share of the achievable gap. Figure 3b rescales the campaigns of Figure 3a by the gap between random selection and perfect one-step ranking, which assays the 8 best unassayed neighbors in every round. Pev closes 54% of that gap by 64 assays, the Gaussian process 41% and the ridge surrogate 35%.

4.5 Calibrated Probabilities

Metric. A decision policy is most useful when its probabilities can be read as probabilities. We measure this with the expected calibration error (ECE), the weighted mean absolute gap between predicted probability and observed frequency across ten equal-width bins, on test families that are disjoint from the families used to fit the calibration (App. C). The improvement probability is scored against whether each edit improves the parent, and the choice probability of the top-ranked edit against whether that edit is the best improving edit of its panel.

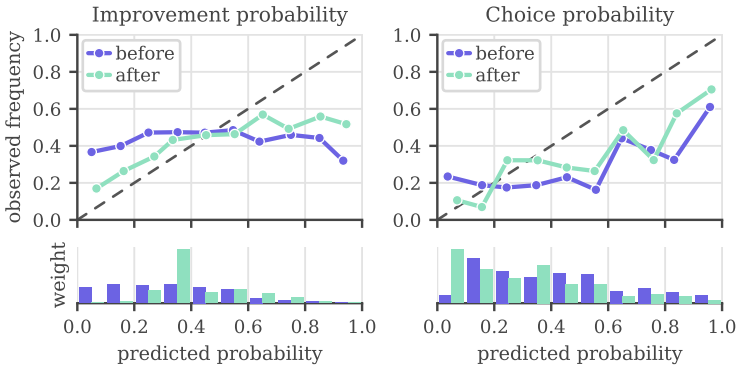


Figure 4 ***Calibration outcomes on test families.*** *Observed versus predicted probabilities across ten bins, before and after calibration. The dashed diagonal indicates perfect calibration, and bars below show the distribution of predictions across bins.*

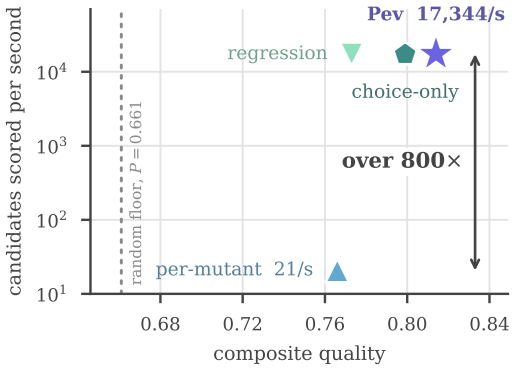


Figure 5 ***Quality against throughput.*** *Pev has the best composite quality P, and scores over 800× faster than the per-mutant encoder at N=1000 candidates in one request.*

Results. A score can rank edits well and still overstate how likely they are to help, and post-hoc calibration corrects the second without touching the first. It roughly halves the calibration error of both probabilities (Figure 4). The ECE of the improvement probability falls from 0.182 to 0.096, a fall of 47%. For the choice probability of the top-ranked edit it falls from 0.203 to 0.114, a fall of 44%. Calibration changes no decision (Proposition 3.1), so only the probabilities that Pev reports are affected.

4.6 The Quality and Throughput Frontier

Setup. We test whether sharing a parent encoding improves scoring efficiency without sacrificing selection quality. We compare Pev with three variants trained under the same protocol. The matched per-mutant variant differs only in whether candidates reuse the parent encoding, isolating the effect of this design choice. Selection quality

is measured by P, the fraction of a panel's best gain captured by the top eight edits, averaged within each sequence class and then equally across classes (App. C). Throughput is measured in candidates per second for requests of N=1000 candidates, with CUDA synchronization (App. D).

Results. Shared encoding improves throughput while retaining strong selection quality (Figure 5). Pev achieves the highest P among the four variants and over 800× the throughput of the matched per-mutant encoder. It scores 17,344 candidates per second versus 21, while achieving $P = 0.814$ on held-out parents versus 0.766. These results show a Pareto improvement over per-mutant encoding in the evaluated setting: Pev selects better edits at substantially lower computational cost. The shared backbone pass also avoids repeating the dominant computation as the candidate set grows (Proposition C.2).

4.7 Ablation Studies

Setup. We evaluate three matched variants to assess the contributions of parent reuse, the decision objective, and the improvement loss (Table 5). The per-mutant variant encodes each candidate separately, the regression variant replaces the decision objective with mean squared error, and the choice-only variant omits the improvement loss. All other settings are held fixed, with random selection providing a model-free baseline.

Table 5 ***Ablations.*** *Each variant removes one design choice; speed is candidates scored per second.*

Method	P↑	Speed↑	Success@1↑	Best fitness↑
Pev	**0.814**	**17,344**	**0.502**	**5.33**
Per-mutant	0.766	21	0.475	4.41
Regression	0.773	17,216	0.357	4.16
Choice-only	0.799	17,244	0.437	3.30
Random	0.661	n/a	0.386	3.97

Results. Each variant scores below Pev on P, Success@1, and best observed fitness at 64 assays. Shared parent encoding improves P by 0.048 and Success@1 by 0.027 over per-mutant encoding, alongside over 800× higher throughput at N=1000 candidates. The decision objective improves these quality metrics over regression by 0.041 and 0.145, respectively, and best observed fitness at 64 assays by 1.17. Adding the improvement loss raises P by 0.015 and Success@1 by 0.065 over the choice-only variant. Together, these comparisons support the contribution of each component in the evaluated settings.

5 Limitations

Our evaluation focuses on choosing which single substitution of a measured parent to test next, or whether to stop. This shared decision format supports evaluation across protein families and objectives. Multi-site edits and explicit conditioning on experimental history remain outside the scope of the current policy.

All experiments use previously measured outcomes. One-round screens assess selection among measured substitutions, while closed-loop campaigns assess sequential decisions within the held-out GB1 landscape. These settings do not establish performance on unmeasured candidates or across new landscapes. Coverage also varies by sequence class, with antibodies represented by 5 panels. Prospective assays, additional landscapes, and broader family coverage would help establish where a shared policy transfers reliably and where further adaptation may be needed.

6 Conclusion

Pev explores a System One approach to protein evolution: mapping a parent sequence, an assay context, and offered edits to an experimental decision in one parallel query. A shared parent encoding enables efficient scoring, while calibrated choice and improvement probabilities guide edit selection and stopping. Results on held-out families and the GB1 landscape suggest that experimental knowledge pooled across families can support useful decisions without fitting model weights to the target family, giving new campaigns an informed starting point.

This approach could extend from single substitutions to multi-site edits, from individual decisions to policies informed by experimental history, and from binding and stability to broader protein functions. A shared decision interface offers a way to learn jointly from these settings as more measurements become available. Combining these fast decisions with feedback from real experiments offers a path toward scalable, closed-loop protein engineering.

Code Availability. We release the Pev model code, weights and frozen benchmark. The code of the main experiments is at https://github.com/Yibo-Wen/Pev, the weights are at https://huggingface.co/Yibooooo/Pev, and the project page is at https://yibow.me/pev.

References

Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. 2020. Model-based reinforcement learning for biological sequence design. In *International Conference on Learning Representations*.

Surojit Biswas, Grigory Khimulya, Ethan C. Alley, Kevin M. Esvelt, and George M. Church. 2021. Low-N protein engineering with data-efficient deep learning. *Nature Methods*, 18:389–396.

Christian Dallago, Jody Mou, Kadina E. Johnston, Bruce J. Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K. Yang. 2021. FLIP: Benchmark tasks in fitness landscape inference for proteins. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Payam Dibaeinia, Sudarshan Babu, Mei Knudson, Ali ElSheikh, Yibo Wen, Han Liu, Jason Perera, and Aly A. Khan. 2026. Virtual cells need context, not just scale. *bioRxiv*.

Ali ElSheikh, Rui-Xi Wang, Weimin Wu, Yibo Wen, Payam Dibaeinia, Jennifer Yuntong Zhang, Jerry Yao-Chieh Hu, Mei Knudson, Sudarshan Babu, Shao-Hua Sun, Aly A. Khan, and Han Liu. 2026. Cell-JEPA: Latent representation learning for single-cell transcriptomics. Preprint arXiv:2602.02093.

Douglas M. Fowler and Stanley Fields. 2014. Deep mutational scanning: a new style of protein science. *Nature Methods*, 11(8):801–807.

Tilmann Gneiting and Adrian E. Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Cade Gordon, Amy X. Lu, and Pieter Abbeel. 2025. Protein language model fitness is a matter of preference. In *International Conference on Learning Representations (ICLR)*. BioRxiv preprint, doi:10.1101/2024.10.03.616542.

Kevin P. Greenman, Ava P. Amini, and Kevin K. Yang. 2025. Benchmarking uncertainty quantification for protein engineering. *PLOS Computational Biology*, 21(1):e1012639.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330.

Brian L. Hie, Varun R. Shanker, Duo Xu, Theodora U. J. Bruun, Payton A. Weidenbacher, Shaogeng Tang, Wesley Wu, John E. Pak, and Peter S. Kim. 2024. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2):275–283.

Brian L. Hie and Kevin K. Yang. 2022. Adaptive machine learning for protein engineering. *Current Opinion in Structural Biology*, 72:145–152.

Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. 2022. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 40(7):1114–1122.

Kaiyi Jiang, Zhaoqing Yan, Matteo Di Bernardo, et al. 2025. Rapid in silico directed evolution by a protein language model with EVOLVEpro. *Science*, 387:eadr6006.

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

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. 2021. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, volume 34, pages 29287–29303.

Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan N. Gomez, Debora Marks, and Yarin Gal. 2022. Tranception: Protein fitness prediction with autoregressive transformers and inference-time retrieval. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16990–17017. PMLR.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, et al. 2023a. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*.

Pascal Notin, Ruben Weitzman, Debora Marks, and Yarin Gal. 2023b. ProteinNPT: Improving protein property prediction and design with non-parametric transformers. In *Advances in Neural Information Processing Systems*, volume 36, pages 33529–33563.

Michael S. Packer and David R. Liu. 2015. Methods for the directed evolution of proteins. *Nature Reviews Genetics*, 16(7):379–394.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36.

Philip A. Romero and Frances H. Arnold. 2009. Exploring protein fitness landscapes by directed evolution. *Nature Reviews Molecular Cell Biology*, 10(12):866–876.

Philip A. Romero, Andreas Krause, and Frances H. Arnold. 2013. Navigating the protein fitness landscape with Gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3):E193–E201.

Robert Schmirler, Michael Heinzinger, and Burkhard Rost. 2024. Fine-tuning protein language models boosts predictions across diverse tasks. *Nature Communications*, 15(1):7407.

Tyler N. Starr and Joseph W. Thornton. 2016. Epistasis in protein evolution. *Protein Science*, 25(7):1204–1218.

TypeSafe AI. 2026. Introducing System One models & Jev. Blog post.

Ziqing Wang, Yibo Wen, William Pattie, Xiao Luo, Weimin Wu, Jerry Yao-Chieh Hu, Abhishek Pandey, Han Liu, and Kaize Ding. 2026. POLO: Preference-guided multi-turn reinforcement learning for lead optimization. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD 2026, oral; arXiv:2509.21737.

Yibo Wen, Chenwei Xu, Jerry Yao-Chieh Hu, Kaize Ding, and Han Liu. 2025. Pareto-optimal energy alignment for designing nature-like antibodies. In *Advances in Neural Information Processing Systems*, volume 38, pages 46294–46321. Curran Associates, Inc. NeurIPS 2025 Main Conference Track; arXiv:2412.20984.

Talal Widatalla, Ashir A. Borah, Samuel H. King, Claudia L. Driscoll, Rafael Rafailov, and Brian L. Hie. 2026. Aligning protein-generative models to experimental fitness with ProteinDPO. *Nature Methods*, 23(9):1805–1813.

Bruce J. Wittmann, Yisong Yue, and Frances H. Arnold. 2021. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Systems*, 12(11):1026–1045.e7.

Nicholas C. Wu, Lei Dai, C. Anders Olson, James O. Lloyd-Smith, and Ren Sun. 2016. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife*, 5:e16965.

Weimin Wu, Xuefeng Song, Yibo Wen, Qinjie Lin, Zhihan Zhou, Jerry Yao-Chieh Hu, Zhong Wang, and Han Liu. 2025. Genome-Factory: A library for tuning, deploying, and interpreting genomic foundation models. Preprint arXiv:2509.12266.

Zachary Wu, S. B. Jennifer Kan, Russell D. Lewis, Bruce J. Wittmann, and Frances H. Arnold. 2019. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, 116(18):8852–8858.

Jason Yang, Ravi G. Lal, James C. Bowden, Raul Astudillo, Mikhail A. Hameedi, Sukhvinder Kaur, Matthew Hill, Yisong Yue, and Frances H. Arnold. 2025. Active learning-assisted directed evolution. *Nature Communications*, 16(1):714.

Kevin K. Yang, Zachary Wu, and Frances H. Arnold. 2019. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694.

Appendix

A Related Work 13

B Data, Splits and Licenses 14

C Formal Definitions 14

D Training and Evaluation Details 16

E Algorithms 17

F LLM Usage 17

A Related Work

Zero-shot mutation-effect prediction. Zero-shot predictors rank the effect of mutations without any measurement of the target protein. Protein language models trained on sequence alone do this: ESM-1v scores a substitution by the masked marginal at its position (Meier et al., 2021), and Tranception adds retrieval of homologous sequences at inference (Notin et al., 2022). ProteinGym collects deep mutational scans to benchmark such models (Notin et al., 2023a), and their accuracy depends on how strongly pretraining prefers the wild-type sequence (Gordon et al., 2025). ESM-2 (Lin et al., 2023) is the backbone of Pev, kept frozen. Its pretrained language-model head gives the wild-type-marginal log-odds of every edit from one pass over the parent (Meier et al., 2021), which is one of our two zero-shot baselines and also a feature of each candidate in Pev. The other is ESM-1v. Pev keeps that single pass over the parent and adds heads, trained on measured outcomes, that return calibrated probabilities for each edit on offer.

Machine-learning-guided directed evolution. Directed evolution improves a protein through repeated rounds of mutation and selection, and machine learning now guides which variants to assay next. The classic scheme uses random mutation and artificial selection (Romero and Arnold, 2009), and surrogate models fitted to a family's own measurements guide the next round (Yang et al., 2019). Gaussian-process models use their uncertainty to choose the sequences to test next (Romero et al., 2013), and adaptive machine learning selects sequences for measurement across rounds of training and optimization (Hie and Yang, 2022). Machine-learning-assisted evolution works with combinatorial libraries (Wu et al., 2019), and zero-shot predictors can design its training sets (Wittmann et al., 2021). Low-N engineering starts from a handful of characterized variants (Biswas et al., 2021), and few-shot active learning runs on language-model embeddings with a regression head (Jiang et al., 2025). Biswas et al. (2021) and Jiang et al. (2025) encode each candidate's whole sequence, a design that the per-mutant encoder of Section 4.6 also uses. Active learning with uncertainty quantification pairs with batch Bayesian optimization (Yang et al., 2025). Combining evolutionary density features with assay-labeled data helps when labeled data are limited (Hsu et al., 2022), and ProteinNPT targets label-scarce, multi-property prediction (Notin et al., 2023b). General protein language models have also guided antibody affinity maturation over two rounds (Hie et al., 2024). Most of these methods fit a model to measurements of the target family. Pev is trained once across families and proposes a first round with no measurement of the target. In the closed loop of Section 4.4 it stays ahead of a ridge surrogate and a Gaussian process, both refit on the target's own assays, at every budget.

Fine-tuning and preference alignment. Fine-tuning, preference optimization and reinforcement learning adapt models to experimental or energy-based objectives. Supervised fine-tuning of protein language models improves predictions across tasks and is especially useful for small fitness datasets (Schmirler et al., 2024), and GenomeFactory packages fine-tuning for genomic foundation models (Wu et al., 2025). Direct preference optimization trains a language model from preference data in a single stage, without fitting a separate reward model (Rafailov et al., 2023). ProteinDPO uses it to align a structure-conditioned protein language model with experimental stability measurements (Widatalla et al., 2026). AlignAb extends antibody direct preference optimization to Pareto optimality over several energy-based objectives for antibody sequence–structure co-design (Wen et al., 2025). Reinforcement learning has also been applied to biological sequence design, with a policy trained on a model fitted to earlier experimental rounds (Angermueller et al., 2020). POLO optimizes a lead compound over several turns under a fixed oracle budget, a setting like our closed loop but on molecules (Wang et al., 2026). Pev instead leaves the backbone frozen and trains about 1.6 million parameters of embeddings and heads directly on measured outcomes of candidate edits. It acts in one pass over a closed set of edits and returns two calibrated probabilities for each edit, one that it is the best action on offer and one that it improves the parent.

Context and coverage in biological foundation models. Work on biological foundation models asks how well they carry over to contexts absent from training. For virtual-cell models, generalization to held-out contexts depends on the diversity of contexts in the training data and not on scale alone (Dibaeinia et al., 2026). Cell-JEPA predicts cell embeddings from partial observations instead of reconstructing sparse, noisy counts (ElSheikh et al., 2026). Pev applies that view to proteins: one policy trained on many families across peptides, domains and antibodies, and scored on families it never saw. It predicts a distribution over an action set where a regression head would predict a noisy outcome, and Section 4.7 compares the two formulations.

Landscapes for closed-loop evaluation. Our closed loop runs on a held-out measured landscape. It is the four-site GB1 landscape (Wu et al., 2016), which also provides low-resource and extrapolative tasks in the FLIP benchmark (Dallago et al., 2021). GB1 is reserved from training, so every proposal in the campaign is made on a target outside the training data.

Calibration and proper scoring. Calibration asks whether a predicted probability matches how often the event happens. Strictly proper scoring rules reward honest probabilities (Gneiting and Raftery, 2007), and temperature scaling, the single-parameter variant of Platt scaling, was studied for modern networks by Guo et al. (2017). Uncertainty estimates for protein engineering have been benchmarked for accuracy and calibration (Greenman et al., 2025). Pev trains both heads with strictly proper scoring rules. It then calibrates them post hoc, with a temperature on the choice logits and an affine map on the improvement logit, both fitted on separate calibration families. On held-out families, Section 4.5 shows that this roughly halves the expected calibration error and changes no decision.

System One models. System One models make fast, structured decisions that software can use directly. TypeSafe AI (2026) describe them as taking structured program state as input and returning type-safe values that carry calibrated probabilities. Every output comes from one parallel query, with no string generation. Pev adopts that interface for protein evolution, with a closed and typed action set. It trains the interface by supervised minimization of strictly proper scoring rules, with calibration fitted on separate calibration families.

B Data, Splits and Licenses

This appendix lists the public data sources and their licenses, explains how families are split between training, calibration and evaluation, and names the lineages held out from training.

Table 6 ***Public data sources.*** *Each source with what it contributes and its license or terms where the source states them; GB1 is reserved for the closed loop.*

Source	Contributes	License or terms
MegaScale	domain stability	CC BY 4.0
FireProtDB	stability	see primary source
ProteinGym	deep mutational scans	MIT
SKEMPI 2.0	binding affinity (pK_d)	see primary source
SLiM DMS	peptide binding	CC BY 4.0
AlphaSeq (Ab-14 releases)	antibody binding	CC BY-NC-SA 4.0
AbAgym	antibody–antigen scans	non-commercial
FLAb	antibody mutation series	per-study terms
GB1 four-site	closed-loop landscape only	CC BY 4.0

Sources. Pev uses nine public sources, listed with their licenses in Table 6. Of these, eight supply training and evaluation data, and the GB1 landscape is reserved for the closed loop. The repository names each source, with its license or terms where the source gives them, and points to the primary data. The evaluation split behind the benchmark is distributed as one archive whose variant table is pinned by SHA-256, and it carries the terms of the sources from which it is derived.

Splits. Splits are made by family, and a family held out for one property stays held out for every property. The evaluation split holds 165 families, 214,378 variants and 157 assay contexts, and the frozen benchmark of Section 4.2 is drawn from it (App. C). Every family outside the evaluation split supplies data for training, validation or calibration, and in that role alone: training fits the heads, validation selects the recipe, and the calibration families fit the maps of Section 3.4 and appear in no table of results. Training draws 10,000 episodes in each of three epochs and validation 2,500, and a calibration pool is accepted at a minimum of three families and 200 episodes (App. D).

Held-out lineages. Whole lineages are held out from training alongside whole families. GB1 with every Protein G homologue is held out, matched at the homologue level because MegaScale and ProteinGym carry it under other names, and it serves as the closed-loop landscape. Others include the Ab-14 lineage in both AlphaSeq releases, the PSD95 PDZ3 domain and twenty MegaScale domains. The Ab-14 lineage and the PSD95 PDZ3 domain give antibody and domain-binding panels, and the MegaScale domains give domain-stability panels.

C Formal Definitions

Beneficial edit. An edit is beneficial when its gain exceeds both the measurement noise and a predefined effect floor (equation (2)). The noise σ_a combines the replicate standard deviations of the edited sequence and the reference. The reference value is the median of its replicate measurements, and its standard deviation is the

larger of its recorded replicate standard deviation and the standard error across those measurements. Missing replicate standard deviations are imputed using the median within the assay context, or set to zero if none are recorded. The effect floor δ_ρ is fixed before model evaluation at 0.5 kcal/mol for $\Delta\Delta G$, 1 °C for T_m, and 0.3 pK_d units for affinity. For binding signals, it is half the robust spread of the panel's gains, $0.5 \cdot 1.4826 \cdot \mathrm{MAD}(u)$, where MAD denotes the median absolute deviation.

Benchmark panels. The discovery benchmark comprises a fixed list of 45 panels spanning 39 families and 33,463 measured candidates, released with the code. Each panel contains a measured unmutated reference and its measured single substitutions within one assay context and protein family. A panel must contain at least 100 candidates with edits inside the encoder's 1,022-residue window and at least one beneficial edit. Panels without beneficial edits are excluded from the discovery benchmark.

Training targets. Both heads use soft targets that account for measurement noise. The choice target p^* assigns equal probability to actions in the noise-tie set $\mathcal{T}$ (equation (3)). The improvement target is $y_a^* = \Phi(u_a/\sigma_a)$ when $\sigma_a > 0$ and $1[u_a > 0]$ otherwise, where Φ is the standard normal cumulative distribution function. Under Gaussian measurement noise and a flat prior on the gain, this target is the posterior probability that the true gain is positive. Each episode's loss is weighted by the separation between its two highest-utility actions:

$$w = 0.1 + 0.9\,\Phi\left(\frac{u_{(1)} - u_{(2)}}{\sqrt{\sigma_{(1)}^2 + \sigma_{(2)}^2}}\right),$$

where (1) and (2) index those actions, including STOP with $u = 0$ and $\sigma = 0$. When the combined noise is positive, the weight is 0.55 for tied utilities and approaches 1 as their gap grows relative to the noise. Without recorded noise, the weight is 1. Calibration instead uses targets based directly on measured outcomes. The choice target assigns equal probability to the highest-gain edits when their gain is positive, or all probability to STOP otherwise. The improvement target is $1[u_a > 0]$ for each edit.

Statements and proofs. Two propositions state what each head estimates and what parent reuse costs.

Proposition C.1 (What each head estimates)**.** Let $\omega = (x, c, \mathcal{O})$, let $p^* \in \Delta(\mathcal{O} \cup \{\mathsf{STOP}\})$ be the choice target and $y_a^* \in [0, 1]$ the improvement target, and let $w \in [0.55, 1]$ be the episode weight defined under Training targets. Write $\mathbb{E}_w[\,\cdot \mid \omega] = \mathbb{E}[w \cdot \mid \omega]/\mathbb{E}[w \mid \omega]$ for the expectation under the training distribution reweighted by w. Over all measurable heads, the weighted cross-entropy $\mathbb{E}[-w \sum_j p_j^* \log p_{\mathrm{choice}}[j](\omega)]$ is minimized uniquely by $p_{\mathrm{choice}}[j](\omega) = \mathbb{E}_w[p_j^* \mid \omega]$, and the weighted binary cross-entropy of $p_{\mathrm{improve}}[a](\omega)$ against y_a^* by $p_{\mathrm{improve}}[a](\omega) = \mathbb{E}_w[y_a^* \mid \omega]$. With $w \equiv 1$ and hard targets, $p_{\mathrm{choice}}[j](\omega)$ is the posterior probability that j is the best action on offer, ties shared equally, and $p_{\mathrm{improve}}[a](\omega)$ is the posterior probability that the gain of a is positive. With soft targets they are the weighted conditional means of those targets.

Proof. Fix ω and write $q = p_{\mathrm{choice}}(\omega)$. The weighted cross-entropy is linear in $w\,p^*$, so its conditional mean is $-\sum_j \mathbb{E}[w\,p_j^* \mid \omega] \log q_j = \mathbb{E}[w \mid \omega]\,\big(H(\bar{p}) + \mathrm{KL}(\bar{p} \,\|\, q)\big)$ with $\bar{p} = \mathbb{E}_w[p^* \mid \omega] \in \Delta(\mathcal{O} \cup \{\mathsf{STOP}\})$ and $\mathbb{E}[w \mid \omega] > 0$. The KL term is non-negative and vanishes only at $q = \bar{p}$, which is strict properness (Gneiting and Raftery, 2007). The binary term is the same computation with two outcomes. □

Proposition C.2 (Cost of parent reuse)**.** Let C_{enc} be the cost of one backbone pass over a length-L sequence and C_{head} the cost of scoring one candidate from its features. For a decision over N candidates, parent reuse costs $C_{\mathrm{reuse}} = C_{\mathrm{enc}} + N\,C_{\mathrm{head}}$. Encoding every mutant separately costs $C_{\mathrm{pm}} = (N+1)\,C_{\mathrm{enc}} + N\,C_{\mathrm{head}}$, because each mutant has the parent's length and the parent is still encoded for the STOP head and the language-model features. The ratio $S(N) = C_{\mathrm{pm}}/C_{\mathrm{reuse}}$ is increasing in N and tends to $1 + C_{\mathrm{enc}}/C_{\mathrm{head}}$.

Proof. $S(N) = \dfrac{N(C_{\mathrm{enc}} + C_{\mathrm{head}}) + C_{\mathrm{enc}}}{N\,C_{\mathrm{head}} + C_{\mathrm{enc}}} = 1 + \dfrac{C_{\mathrm{enc}}}{C_{\mathrm{head}} + C_{\mathrm{enc}}/N}$, which increases in N and tends to $1 + C_{\mathrm{enc}}/C_{\mathrm{head}}$ from below as $N \to \infty$. □

Screen metrics. The screen scores each panel on the ten edits it ranks first and on its full ranking. Hit rate@k averages the labels of the top k edits of a panel, with $k = 10$ since every panel holds at least 100 candidates. Enrichment@k divides it by the panel prevalence, so random ranking scores one in expectation. AUROC is

the Mann–Whitney statistic per panel, and wells to first hit is the one-based rank of the first beneficial edit in the full ranking. Perfect ranking places the beneficial edits first, ordered by measured effect, and the random baseline averages twenty permutations per panel. Pev ranks each panel by its choice logits, for both objectives. Zero-shot ESM-2 ranks by the wild-type-marginal log-odds of the edit from one pass over the parent, and ESM-1v by the masked marginal at the edited position, averaged over its five published checkpoints. The screen's tables are means over panels.

Transfer metrics. The transfer diagnostic labels an edit by the noise margin alone and scores one decision per panel. An edit clears its noise there when $u_a > \sigma_a$, with σ_a the replicate sd of the difference ($u_a > 0$ where none is recorded) and no effect floor. Success@1 is one if the chosen edit clears its noise, or if STOP is chosen when no edit does. Pev and its variants can choose STOP, while zero-shot ESM-2 and random always choose an edit. Panel AUROC uses the same label and does not depend on STOP. Perfect decisions choose an edit that clears its noise when one exists and STOP otherwise, and rank the edits that clear their noise first, so both metrics equal one. Rows are weighted by class, then objective, then (family, context), then panel.

Closed-loop and frontier metrics. The closed loop scores the best sequence a campaign reaches, and the frontier scores quality against speed. The best observed fitness after B assays is $F_B = \max_{t \le B} y(x_t)$, where x_t is the t-th assayed sequence and fitness is relative to the wild type, $y(\mathrm{WT}) = 1$. The perfect one-step ranking reference follows the same protocol but assays, each round, the 8 unassayed neighbors of the current parent with the highest measured fitness. The share of the achievable gap in subfigure (b) of Figure 3 is $(F_B - F_B^{\mathrm{rnd}})/(F_B^{\mathrm{perfect}} - F_B^{\mathrm{rnd}})$, with F_B^{rnd} the value for random selection. For a panel and the set R of the top eight edits returned, $P = \max(0, \max_{a \in R} u_a)/\max_{a \in \mathcal{O}} u_a$, with $P = 0$ when no edit improves; it is averaged within each class, family and context and then over the three classes with equal weight. Throughput is the reciprocal of the median seconds per candidate over repeated timed requests of 1,000 candidates.

Calibration error. Expected calibration error is the weighted gap between predicted probability and observed frequency. It is computed on the held-out test families, on panels that each hold a measured parent with its measured single-edit neighbors, and never on the calibration families on which the maps are fitted. It uses ten equal-width bins m, the last including 1, and $\mathrm{ECE} = \sum_m \frac{W_m}{W} |\mathrm{pred}_m - \mathrm{freq}_m|$, where W_m is the total weight of the forecasts in bin m, $W = \sum_m W_m$, and pred_m and freq_m are their weighted mean predicted probability and weighted observed frequency. Weights give equal mass to each objective, then to each (family, context) within it, then to each panel, with candidates equal within a panel. The improvement forecast is scored against $1[u_a > 0]$ for every candidate. The choice forecast is the probability of the top-ranked edit, scored against the hard choice target of Training targets, so it counts as correct when that edit is a best positive-gain edit of its panel.

Uncertainty. Intervals come from a paired bootstrap over families on the screen and over initializations in the closed loop. One family can supply several panels, so the screen resamples the per-family means of the two methods compared 2,000 times, and the intervals are 95% percentile intervals.

D Training and Evaluation Details

Model. The frozen ESM-2 650M backbone has hidden size $d = 1280$, and the trained parts hold 1,560,643 parameters. The context embedding q_c has $32 + 32 + 2 \times 129 = 322$ dimensions: a 32-dimensional objective embedding, a 32-dimensional readout embedding, and the mean-pooled target and partner states, each projected from 1280 to 128 dimensions with a presence flag. With a 32-dimensional residue embedding shared by e_r and e_{x_i}, the candidate feature $\phi_a = [h_i;\ \bar{h};\ e_r;\ q_c;\ e_{x_i};\ \ell_a]$ has $1280 + 1280 + 32 + 322 + 32 + 2 = 2{,}948$ dimensions. The candidate head maps it through two hidden layers of width 256 (GELU, dropout 0.1) to the choice and improvement logits. The STOP head maps $[\bar{h}; q_c]$, of $1280 + 322 = 1{,}602$ dimensions, through one hidden layer of width 256 to the STOP logit.

Optimization. Pev is trained with AdamW for three epochs of 10,000 wild-type-scan episodes. The learning rate is 3×10^{-4}, weight decay 0.01, gradient clipping 1.0, and training uses bf16 autocast and gradient accumulation over 8 episodes. Episodes are drawn evenly across the sequence-class and objective cells, and validation loss is computed on 2,500 validation episodes.

Calibration fit. Calibration is fitted separately for each objective on the calibration families. L-BFGS starts from the identity and fits T on the weighted choice log-loss and (α, β) on the weighted binary cross-entropy, with equal mass per objective, then (family, context), then episode. The bounds are $\log T, \log \alpha \in [-5, 5]$ and $\beta \in [-10, 10]$. Each map is fitted at the first of four pools that gives an accepted fit: the objective's own calibration episodes

from its primary sources, all calibration episodes of that objective, both objectives pooled, and the identity. A fit is accepted when its pool has at least three families and 200 episodes and, for the affine map, both outcome classes, and when it does not run to a flattening bound. The pool reached is published with the parameters.

Closed-loop rollout. Every arm runs the rollout of Algorithm 2 and spends its whole budget of 64 assays in batches of 8, counting the initial set. Each of the 50 initializations is a measured sequence and seven of its neighbors, chosen by hash without reading any outcome. Pev scores by calibrated p_{improve}, the ridge surrogate by its predicted fitness, the Gaussian process by expected improvement, and variants without a trained improvement head by their own score. The ridge surrogate uses one-hot features over the four sites (4×20) with penalty 1 and is refit in closed form every round on all measured data. The Gaussian process uses the same features with a squared-exponential kernel, refit the same way. Neither comparator chooses STOP.

Throughput. Throughput is measured on one GPU. Each request holds single substitutions of one held-out parent, and timing is CUDA-synchronized with the frozen context cache warmed first.

Reproducibility. Two commands reproduce the discovery tables from the released checkpoint and the evaluation archive. `make check` confirms the variant table, the evaluation split and the 45 panels against their pins, and `make discovery` writes every discovery table.

Broader impact. Pev edits existing sequences toward measured assay outcomes. It chooses among single substitutions of a given sequence rather than generating new ones, and it is trained on public data. We release a checkpoint, a frozen evaluation split and the code of the discovery screen.

E Algorithms

Algorithm 1 gives one Pev decision, and Algorithm 2 wraps it in a closed-loop campaign.

Algorithm 1: one Pev decision

Input: parent x, context c, offered edits $\mathcal{O}$ (the measured substitutions in a screen, the unassayed ones in a campaign); calibration (T, α, β) for the objective of c.
Encode the parent once with the frozen backbone, $h \leftarrow \text{Enc}(x)$, and embed the context as q_c.
For every edit $a = (i, r) \in \mathcal{O}$ build ϕ_a from h_i, $\bar{h}$, e_r, q_c, e_{x_i} and ℓ_a, and compute (z_a, v_a) in one batched call.
Compute the STOP logit z_{STOP} from $[\bar{h}; q_c]$ with the STOP head.
$p_{\text{choice}} \leftarrow \text{softmax}\left(([z_a]_a,\ z_{\text{STOP}})/T\right)$ and $p_{\text{improve}}[a] \leftarrow \sigma(\alpha v_a + \beta)$.
Return $\hat{a} = \arg\max p_{\text{choice}}$ (an edit or STOP), with p_{choice} and p_{improve}.

Algorithm 2: closed-loop campaign with Pev

Input: assayed starting set S, budget B, batch size b, the decision rule of Algorithm 1.
While $|S| < B$:
    parent $\leftarrow$ the best assayed sequence in S that is not retired and has unassayed neighbors.
    Score every unassayed candidate edit of the parent with Algorithm 1.
    Rank by p_{improve}, take the best edit at each site, then fill the batch of b by rank.
    Assay the batch and add it to S; if Algorithm 1 returned $\hat{a} = \text{STOP}$, retire the parent.
Return the best observed fitness in S.

F LLM Usage

Coding agents helped implement the model, the training pipeline and the evaluation benchmark, and helped edit this paper. The authors proposed the decision formulation, designed the experiments, made the calibration decisions, and checked every claim and every number reported here against the benchmark outputs.