PROPEL: GLOBAL EVOLUTIONARY LATENTS FOR AUTOREGRESSIVE PROTEIN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive protein language models capture rich sequence regularities, but represent protein family information implicitly through shared model parameters and token-level hidden states. Other family-specific models about one family are spread across weights shared with every other family, so pointing it at a new target means changing the model. We introduce PROPEL (Protein Retargeting by Optimized Posterior over Evolutionary Latents), an autoregressive protein language model with latent space that complements local autoregressive modeling with a compact global representation of evolutionary context. A target-specific latent state is shared across the sequence and injected into every decoder layer through cross-attention, separating persistent family-level information from token-level sequence computation.The latent is learned without an encoder and can be further optimized at inference time while the backbone remains frozen. This representation provides a flexible interface for incorporating both unlabelled homologs and limited fitness supervision, while preserving a single autoregressive model for both variant scoring and sequence generation. Across protein fitness prediction tasks, PROPEL consistently improves its autoregressive backbone and exhibits strong data and parameter efficiency: a 60M-parameter model can match substantially larger autoregressive protein language models under zero-shot fitness prediction and comparable alignment settings. These results show that introducing an explicit global representation can substantially enhance autoregressive protein models without sacrificing their simplicity, scalability, or generative nature.

1 INTRODUCTION

A deep mutational scan of protein G's B1 domain measured $149{,}360$ four-site variants (Wu et al., 2016). Conditioned on that domain's homologs alone, PROPEL—the 60M-parameter model in this paper—ranks its $147{,}193$ triple and quadruple mutants at Spearman $\rho = -.049$, significantly the wrong way. Writing the $2{,}167$ measured singles and doubles into PROPEL's latent state, without touching any of its 60.1 million weights, moves the same ranking to $\rho = .459$ (Table 7). The model did not need new parameters. It needed a place to put the target.

Nearly every method for pointing a protein model at a target puts it somewhere else. Supervised predictors fit a head on frozen embeddings (Notin et al., 2023b; Groth et al., 2024); preference-aligned generators fine-tune every weight and keep one model per assay (Bhatnagar et al., 2025); parameter-efficient methods shrink the update but keep its destination, a delta on weights shared with every family the model has seen (Hu et al., 2022); and retrieval methods do not adapt at all, re-reading the target's homologs in the context window and paying for that context on every sequence scored (Truong Jr. & Bepler, 2023; Rao et al., 2021). These differ in cost but share a structural property: the object holding target knowledge is not the object that generates. A head predicts a number beside the model and cannot decode; a weight delta is a copy of the model, with no second channel by which label-free evolutionary information could reach it; a prompt is neither compact nor optimizable. Each is strong at what it does—homolog-conditioned predictors remain the most accurate zero-shot variant-effect models available—but none exposes one small object that an alignment and an assay can both write into and a decoder can read.

This paper. We introduce PROPEL, a 60.1M-parameter latent-variable autoregressive protein language model that augments sequence modeling with a compact, target-specific representation of

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.

evolutionary information. A global latent conditions every decoder layer through cross-attention (Figure 1), and family-packed UniRef50 pretraining encourages it to capture information shared across homologs rather than memorize individual sequences. At test time, variational inference optimizes this latent against the frozen decoder, producing a reusable family representation z_{fam} without an encoder or backbone updates. This architecture offers three capabilities: (i) homologous sequences are compressed into a latent state that can be reused for efficient variant scoring; (ii) limited fitness measurements refine the same representation through a pairwise preference objective, $z_{\text{adapt}} = z_{\text{fam}} + \gamma \Delta z$; and (iii) the conditioned autoregressive decoder supports both variant scoring and sequence generation. Without fitness labels, PROPEL achieves .360 grouped Spearman across ProteinGym's 217 substitution assays, outperforming the evaluated ProGen checkpoints with at most 339M parameters despite using an order of magnitude fewer pretraining tokens (Notin et al., 2023a). These results demonstrate the potential of explicit evolutionary latents to enhance the adaptability and parameter efficiency of autoregressive protein language models.

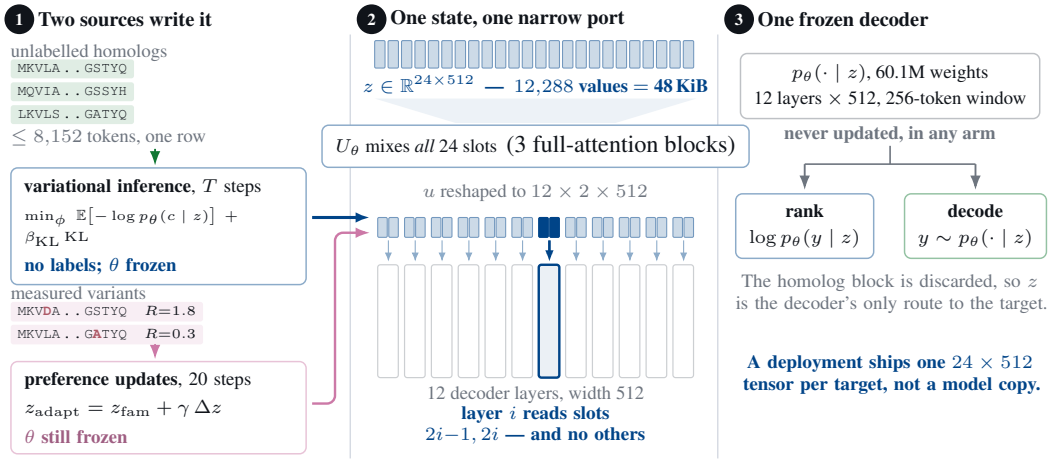


Figure 1: **One writable state.** An unlabelled alignment (green) and a laboratory assay (pink) write the *same* 24×512 tensor; every decoder layer reads it through its own two slots; one frozen decoder both ranks measured variants and decodes new ones. Details in §3.

Contributions.

1. **An autoregressive protein language model with an explicit global evolutionary representation.** Family-packed pretraining encourages a shared latent to capture family-level information. In contrast, per-sequence training improves perplexity with increasing latent capacity while degrading fitness ranking, highlighting the importance of how the representation is learned (§3.1).
2. **Inference-time refinement of evolutionary representations.** Increasing posterior optimization steps eightfold improves grouped Spearman from .2089 to .3595 across 217 ProteinGym substitution assays, with improvements on 189, without labels or backbone updates (§4.1).
3. **Data- and parameter-efficient target adaptation.** At an identical 12,288-value budget, latent adaptation outperforms weight updates on 7/8 assays. Homologs and fitness measurements provide complementary information, while near-wild-type supervision transfers to 822,049 distant variants across all eight evaluated assays (§4.2, §4.3).
4. **A unified representation for prediction and generation.** The same latent conditions both variant scoring and autoregressive generation, enabling directional changes in generated sequences, validated against wrong-family and shuffled-label controls (§4.4).

2 Related Work

Protein language models and evolutionary context. Large-scale autoregressive protein language models, including ProGen, ProGen2, and ProGen3, learn sequence distributions from diverse protein corpora and support both variant scoring and sequence generation (Madani et al., 2023; Nijkamp et al., 2023; Bhatnagar et al., 2025). Masked language models such as ESM-1v and ESM-2 learn

transferable representations that are widely used for protein property prediction (Meier et al., 2021; Lin et al., 2023). A complementary line of work explicitly incorporates evolutionary information from homologous sequences. MSA Transformer learns representations from multiple sequence alignments (Rao et al., 2021), while PoET and PoET2 use autoregressive modeling of homologous sequence sets (Truong Jr. & Bepler, 2023; 2025). Tranception combines autoregressive predictions with retrieval from evolutionary data (Notin et al., 2022a), and ProFam further explores homolog-conditioned modeling (Wells et al., 2025). For supervised fitness prediction, approaches such as Kermut, ProteinNPT, and other representation-based methods leverage pretrained models together with experimental measurements (Groth et al., 2024; Notin et al., 2023b; Hsu et al., 2022; Zhao et al., 2024).

Latent-variable models for proteins. Latent-variable models provide another approach to modeling protein sequence variation. DeepSequence introduces a family-specific variational autoencoder for modeling evolutionary constraints (Riesselman et al., 2018), while EVE extends this approach to evolutionary variant effect prediction (Frazer et al., 2021). More recent work explores latent representations in conjunction with pretrained protein models. ProT-VAE incorporates a compact variational latent space into pretrained protein representations (Sevgen et al., 2025), while PLAID uses latent generative modeling for conditional protein structure generation (Lu et al., 2024). These approaches illustrate different uses of latent variables, ranging from family-specific evolutionary modeling to conditional generation, with differences in their training objectives, inference procedures, and use of pretrained models.

Parameter-efficient and test-time adaptation. Parameter-efficient fine-tuning reduces the cost of adapting large pretrained models through low-rank updates, adapters, activation rescaling, or bias tuning (Hu et al., 2022; Houlsby et al., 2019; Liu et al., 2022a; Ben Zaken et al., 2022). Prompt and prefix tuning instead introduce trainable continuous conditioning variables (Li & Liang, 2021; Lester et al., 2021), with P-tuning v2 extending deep prompt tuning across transformer layers (Liu et al., 2022b). Test-time training and adaptation provide related approaches for improving models after pretraining (Sun et al., 2020; Hardt & Sun, 2024). In protein modeling, PRIMO combines a frozen ESM-2 backbone with preference-based adaptation using limited experimental fitness measurements (Teufel et al., 2025). Activation-steering methods offer another form of intervention, modifying model behavior through directions in representation space (Turner et al., 2023; Huang et al., 2025).

Inference-time latent optimization and preference learning. Variational autoencoders typically use amortized inference to approximate posterior distributions (Kingma & Welling, 2014). Iterative inference methods instead refine approximate posteriors through additional optimization (Marino et al., 2018; Cremer et al., 2018), while latent energy-based models have explored short-run MCMC for inference and generation (Nijkamp et al., 2020; Pang et al., 2020). More recently, the Latent Thought Model introduces optimized latent variables for conditioning autoregressive generation (Kong et al., 2025), and related work investigates latent-space optimization for molecular generation, planning, and other design objectives (Kong et al., 2023; 2024a;b; Noh et al., 2025). Separately, preference optimization has emerged as a framework for aligning generative models with observed preferences. Direct Preference Optimization (DPO) formulates alignment through relative likelihoods (Rafailov et al., 2023), while IRPO incorporates an additional likelihood objective (Pang et al., 2024). Preference-based objectives have also been applied to protein sequence modeling and fitness-guided adaptation (Hawkins-Hooker et al., 2024; Widatalla et al., 2024; Gordon et al., 2025; Bhatnagar et al., 2025).

Preference learning for protein models. Preference optimization provides a complementary framework for adapting generative models to observed preferences or task-specific objectives. Direct Preference Optimization (DPO) formulates alignment through relative likelihoods of preferred and dispreferred outputs (Rafailov et al., 2023). IRPO incorporates an additional likelihood objective on preferred outputs, combining relative preference learning with likelihood-based regularization (Pang et al., 2024). Preference-based objectives have also been explored for protein sequence modeling and fitness-guided adaptation (Hawkins-Hooker et al., 2024; Widatalla et al., 2024; Gordon et al., 2025). ProGen3 applies IRPO to protein fitness alignment, illustrating how experimental measurements can guide the adaptation of autoregressive protein generators (Bhatnagar et al., 2025).

These methods establish preference optimization as a practical approach to incorporating experimental fitness information into protein language models.

3 METHOD

3.1 PROPEL: A PROTEIN LANGUAGE MODEL WHOSE STATE IS INFERRED BY OPTIMIZATION

Let $x = (x_1, \ldots, x_L)$ be an amino-acid sequence over a 32-token vocabulary. PROPEL is a latent-variable autoregressive language model,

$$p_\theta(x) = \int p(z)\, p_\theta(x \mid z)\, \mathrm{d}z, \qquad p(z) = \mathcal{N}(0, I_{N_z d}) \text{ over } \mathrm{vec}(z),$$
$$p_\theta(x \mid z) = \prod_{t=1}^{L} p_\theta(x_t \mid x_{<t}, z)\,, \tag{1}$$

with $z \in \mathbb{R}^{N_z \times d}$, $N_z = 24$ slots of width $d = 512$: 12,288 values. The decoder is 12 pre-norm layers of width 512 with RMSNorm, a SwiGLU feed-forward of hidden width 1,376, rotary position embeddings, tied input and output embeddings. Its attention layer is a special causal self-attention restricted to a 256-token sliding window, so a residue attends to 256 tokens directly and roughly 3,000 through composition across depth.

How the latent reaches the decoder. The routing is what makes z an interface rather than a prompt. A shared prior network U_θ, three full-attention blocks over the 24 slots, maps z to $u = U_\theta(z)$, which is reshaped to $12 \times 2 \times 512$ and given learned layer and slot embeddings; decoder layer i cross attends to its own two slots of u and to no others. Every layer therefore reads the same object through a fixed, narrow port. The property that later lets one tensor be written by two sources and read by one unchanged decoder. Because the decoder consumes u rather than z, it sees the pushforward $U_{\theta\#}\mathcal{N}(0, I)$, the structured prior Kong et al. (2025) identify as missing, while the variable we optimize keeps an isotropic prior and a closed-form diagonal Gaussian KL. Of the 60.1M parameters, 38.0M are decoder self-attention and feed-forward, 12.6M are the twelve cross-attention blocks that read z, and 9.5M are U_θ: the machinery carrying the latent is a third of the model, not an adapter bolted onto a language model (Appendix B).

Inference is optimization: fast latents, slow weights. PROPEL has no encoder, which makes it fall back to the family of autoregressive language model. The posterior over z is obtained by *optimizing* a Gaussian variational distribution against the frozen decoder: for a block c we hold $\phi = (\mu, \log \sigma^2)$, set $q_\phi(z) = \mathcal{N}(\mu, \mathrm{diag}(\sigma^2))$, and minimize the negative ELBO

$$\mathcal{F}(\phi; c, \theta) = \mathop{\mathbb{E}}_{\epsilon \sim \mathcal{N}(0,I)} [-\log p_\theta(c \mid \mu + \sigma \odot \epsilon)] \; + \; \beta_{\mathrm{KL}}\, \mathrm{KL}(q_\phi(z) \,\|\, p(z))\,, \tag{2}$$

whose first term we estimate with a single reparameterized sample and whose second is available in closed form for diagonal Gaussians, by T AdamW steps on ϕ alone, θ fixed; write $\mathrm{Infer}_T(c; \theta)$ for the result. Pretraining alternates this inner loop over ϕ with a slow outer update to θ on the same objective, so the decoder is trained from initialization to be conditioned by a state that is itself the solution of an optimization problem rather than the output of a network. A new target's state is thus defined by a problem instead of a forward pass. It exists for any target with homologs, and no decoder weight moves to obtain it. We train at $T = 16$ and infer at $T = 128$ (§4.1).

Pretraining on family-packed homolog blocks. What a latent learns to be depends on what it is asked to explain. Asked to explain a single sequence, a protein latent memorizes residue identity, because the reconstruction saved by encoding a residue exceeds the KL that encoding it costs and perplexity rewards exactly that, which makes it a misleading model-selection signal here: in preliminary per-sequence runs validation perplexity improved monotonically with capacity while fitness ranking degraded. PROPEL is therefore never trained on a single sequence. Each example is a *family-packed block*: unaligned homologs concatenated into one 2,048-token row, built PoET-style (Truong Jr. & Bepler, 2023) from a 10M-query DIAMOND search over UniRef50. Since 12,288 values cannot memorize six or seven sequences at once, what survives compression is what they share. The backbone is frozen everywhere below except in the arms labelled *LoRA* or *full weights* (Appendix B).

Why a target should be cheap to write into this state. At scoring time that port is the decoder's *only* route to target identity, the homolog block is consumed once by Eq. 2 and discarded, so writing a target into z changes the *argument* of a function the weights already decode rather than the function itself. That bounds the hypothesis class an assay selects from before any label arrives, and is why 532–2,167 measured variants are plausible data for it though they are very little for 60.1M weights. A weight delta of the same size must instead build its own route into the computation, and edits parameters every family shares.

3.2 THREE OPERATIONS ON ONE STATE

Protocol detail for all three operations is in Appendix C.

The label-free write. Given a target with an MSA we build a homolog block and infer $z_{\mathrm{fam}} = \mathrm{Infer}_T(c_{\mathrm{hom}}; \theta)$ at $T = 128$ with the backbone frozen, from ProteinGym's supplied alignments; five prompts at different maximum-identity thresholds give five states, and a sequence is scored by the mean of its conditional log-likelihood over them. No fitness label enters. The block is paid for once and then discarded, which means that scoring a variant afterwards is one pass over its own residues (§4.3), and the family is now a tensor, which is what makes the next operation possible.

The supervised write. Let $\{(y_i, R_i)\}$ be measured variants. We introduce one zero-initialized residual $\Delta z \in \mathbb{R}^{N_z \times d}$, shared across the five prompts, and set $z_{\mathrm{adapt}} = z_{\mathrm{fam}} + \gamma\, \Delta z$ with $p_{\mathrm{ref}}(y) = p_\theta(y \mid z_{\mathrm{fam}})$ and $p_{\mathrm{adapt}}(y) = p_\theta(y \mid z_{\mathrm{adapt}})$. Because the reference is the same frozen decoder at $\Delta z = 0$, the log-ratio $s(y) = \log p_{\mathrm{adapt}}(y) - \log p_{\mathrm{ref}}(y)$ is exactly the effect of the write. On blocks of n variants we minimize IRPO (Pang et al., 2024; Bhatnagar et al., 2025) with Δz as the only free variable:

$$\mathcal{L}(\Delta z) = \sum_{(i,j)\,:\,R_i > R_j} \log\Big(1 + e^{-\beta[s(y_i) - s(y_j)]}\Big) \;+\; \frac{\alpha}{|\mathcal{B}^\star|} \sum_{i \in \mathcal{B}^\star} \left(-\frac{\log p_{\mathrm{adapt}}(y_i)}{|y_i|}\right), \tag{3}$$

where (i, j) ranges over ordered pairs within the block, β is the preference temperature, and $\mathcal{B}^\star = \arg\max_i R_i$ is its set of maximal-reward variants. The first term ranks every strictly ordered pair; the second anchors the likelihood of the best variants and trades fitness alignment against retained naturalness. Two operating points, *raw* and *constrained*, differing in α, the residual scale γ, and whether the pair term is reduced by sum or by mean (Appendix C), are declared in advance and selected on near-wild-type training labels, never re-tuned.

The read by the generator. Nothing further is required: the same $p_\theta(\cdot \mid z)$ that ranks variants is the distribution we decode from, so one write serves both jobs, and a deployment keeps the adapted state itself, one 24×512 tensor per prompt.

4 EXPERIMENTS

We evaluate PROPEL along four questions. First, does the global latent capture family-level evolutionary information that is useful to an autoregressive protein model? Second, can that representation be refined at inference time without modifying the shared decoder? Third, can the same representation incorporate limited fitness supervision efficiently and generalize beyond the labelled sequence neighborhood? Finally, does the latent remain part of the generative model strongly enough to condition the sequences it produces?

4.1 FAMILY-LEVEL EVOLUTIONARY REPRESENTATIONS FOR AUTOREGRESSIVE PREDICTION

We first ask whether the global latent provides a useful representation of family-level evolutionary information. Starting from the same frozen 60.1M-parameter autoregressive decoder, we compare an unconditioned model ($z = 0$) with a 24×512 latent inferred from homologous sequences. Neither condition uses fitness labels or updates model parameters, so this comparison isolates the contribution of the inferred family representation.

With $z = 0$, the frozen backbone ranks distant variants at $\rho = .0790$. Inferring the same 12,288 latent values from homologs alone raises performance to $.3545$ on the eight controlled splits (Table 2).

Table 1: **Zero-shot ProteinGym against the autoregressive generative models PROPEL is directly comparable to**, official v1.3 grouped Spearman over 217 substitution assays. PROPEL reads a homolog block while the ProGen checkpoints read only the query, so this compares pretraining budgets. The full 13-model table is Table 5.

Model	Parameters	Pretrain tokens	ρ
ProGen3-112M	112M	—	.282
ProGen2-small	151M	175B	.336
ProGen3-339M	339M	200B	.347
PROPEL (ours)	**60M**	**13.5B**	**.360**
ProGen2-base	764M	200B	.378
ProGen2-xlarge	6.4B	350B	.391
ProGen3-3B	3.0B	500B	.392

Table 2: **Complementarity of homolog context and fitness supervision.** Equal-assay mean ρ on the eight distant-variant test sets when homolog information and fitness supervision are independently enabled. Each source improves performance in the absence of the other, and their combination gives the strongest result.

	Fitness labels	
Homolog context	absent	present
absent	.0790	.3184
present	.3545	**.5182**

Across all 217 ProteinGym v1.3 substitution assays, the same label-free procedure reaches **.360** grouped Spearman.

For context, Table 1 compares PROPEL with autoregressive ProGen models scored by sequence likelihood. PROPEL exceeds the evaluated ProGen checkpoints up to 339M parameters while using substantially fewer pretraining tokens. This is not an equal-input comparison: PROPEL uses homologous sequences at test time to infer its family latent, whereas these ProGen checkpoints score the query sequence alone. We therefore use this comparison to characterize the effectiveness of the resulting autoregressive model relative to backbone scale and pretraining budget, rather than as evidence that PROPEL is a stronger general-purpose fitness predictor.

Specialized homolog-based models are included in the broader comparison in Appendix B. Like PROPEL, these methods exploit evolutionary information from homologous sequences, but they differ in how that information is represented and incorporated into prediction. Our focus here is not to outperform this class of methods in absolute fitness-prediction accuracy, but to evaluate whether a compact inferred evolutionary representation can effectively augment a shared autoregressive protein language model.

Posterior inference provides a test-time compute axis. Because the family representation is obtained through iterative posterior inference, we next vary only the number of inference steps while holding the checkpoint, homolog context, and scoring procedure fixed. Increasing posterior optimization from 16 to 128 steps raises grouped Spearman monotonically from .2089 to .3595 across the 217 ProteinGym substitution assays, an end-to-end gain of $+.1506$ that is positive on 189 of 217 assays (Figure 2). Each doubling yields a smaller gain than the previous one over the range studied. Thus, additional inference compute can improve the target-specific representation while leaving the shared autoregressive backbone unchanged.

The latent amortizes homolog conditioning. Once inferred, the family representation can be reused across variants rather than reconstructing the target representation from the homolog block for every query. With the decoder fixed, each scored residue attends to 258 latent key–value entries per layer rather than 8,408 entries in the homolog context, corresponding to $32.6\times$ less attention work per scored residue and $8{,}152\times$ less resident conditioning memory. Under our implementation, the one-time posterior-inference cost is recovered after 393 scored variants (Table 6).

4.2 Efficient target adaptation through the global representation

We next ask whether the same family-level representation can incorporate experimental information in addition to homologous context. We first independently enable homolog information and fitness supervision to test whether the two sources provide complementary information. We then compare adaptation through the global conditioning representation with a capacity-matched weight update, holding the backbone, labels, objective, optimization budget, and number of trainable values fixed.

The two information sources are complementary (Table 2). Starting from $z = 0$, fitness supervision alone raises mean Spearman from .0790 to .3184, while homolog inference alone raises it to .3545.

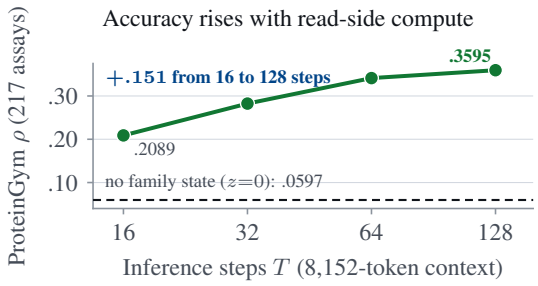


Figure 2: **Inference-time compute is a scaling axis for this model, on frozen weights.** All 217 ProteinGym substitution assays, scored label-free with PROPEL's checkpoint and the headline protocol held fixed. Accuracy is monotone in T, each doubling buying less than the last ($+.0735$, $+.0590$, $+.0182$); the end-to-end gain is positive on **189 of 217 assays** (95% CI $[+0.131, +0.171]$) and every doubling excludes zero. The curve ends at .360 of Table 1. Protocol and the companion context sweep: Appendix C.

Combining homolog inference with fitness adaptation reaches .$\mathbf{5182}$. Experimental measurements therefore provide information beyond that already captured from homologous sequences.

Before comparing adaptation mechanisms, we characterize the supervised update itself. Rescaling the family representation changes the eight-assay mean by only $+0.0028$, whereas one rank-1 direction recovers 44% of the full adaptation gain, indicating that the useful update is predominantly directional rather than a simple change in magnitude. A pre-declared γ provides a tradeoff between fitness and naturalness: our constrained operating point gives up 21% of the fitness gain while removing **74%** of its likelihood cost (Appendix E).

Parameter-efficient adaptation. To test how efficiently the global representation incorporates supervision, we compare it with a rank-12 LoRA update at the same 12,288-value budget. Both arms use the same labels, pairwise preference objective, frozen backbone, and optimization schedule, with each learning rate selected independently on training labels (Appendix C). We additionally vary the available adaptation capacity to compare how performance scales with the number of target-specific trainable values.

At the exact match, latent adaptation reaches .5182, compared with .4657 for LoRA, a mean difference of $+\mathbf{0.0526}$ on **7/8** assays (95% CI $[+0.023, +0.079]$; Figure 3, Table 3). Across the upper part of the capacity sweep, the LoRA update gains $+0.0136$ per doubling, compared with $+0.0268$ for latent adaptation. The same ordering holds at our constrained operating point (Table 3, Appendix F). Within the range studied, the global conditioning representation therefore provides an effective parameter-efficient interface for target adaptation.

Is the effect specific to the latent parameterization? The comparison with LoRA leaves open whether the gain arises from the latent-variable machinery itself—the isotropic prior, closed-form KL, and learned transport U_θ—or from the compact conditioning interface through which target-specific information enters the decoder. We therefore optimize Δu directly in activation space, $u = U_\theta(z_{\text{fam}}) + \gamma \Delta u$, using the same 12,288 values at the same twelve sites under the same objective, labels, budget, and frozen backbone, differing only in that the update does not pass through the transport and carries no prior.

Direct activation-space adaptation reaches .5243, compared with .5182 for latent adaptation: a paired difference of -0.0061 whose interval spans zero. By contrast, the latent–LoRA difference is $+0.0526$ with an interval excluding zero. This ablation suggests that the supervised adaptation benefit is associated primarily with the compact target-specific conditioning interface rather than with a particular latent coordinate system. The variational parameterization remains important for constructing that representation from homologs and for the inference-time refinement studied in §4.1.

4.3 Generalization from limited fitness supervision

We next ask whether supervision confined to a local region of sequence space can improve predictions beyond the labelled neighborhood. The ProGen3 splits provide a controlled setting for this question: adaptation uses only near-wild-type variants, while all more distant variants remain held out for evaluation. Across the eight assays, this gives 9,554 labelled adaptation examples and 822,049 test variants spanning 2–44 substitutions.

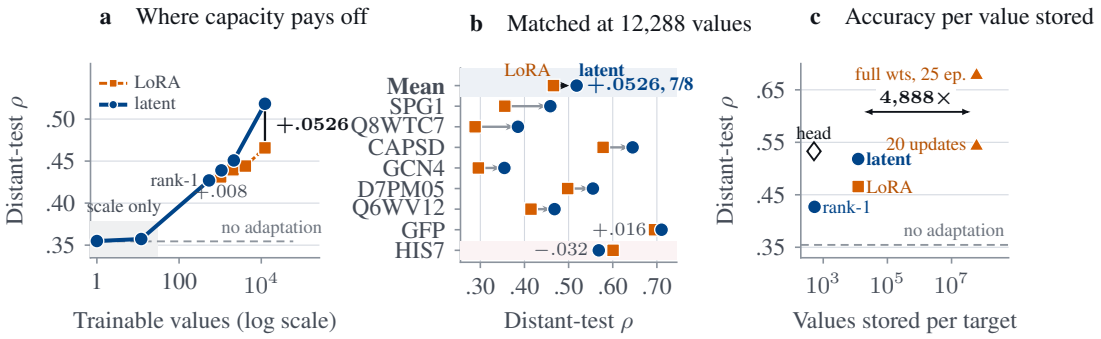


Figure 3: **Parameter-efficient adaptation through the global conditioning representation.** Both arms share the frozen backbone, labels, objective, and optimization budget, with each arm's learning rate selected on training labels alone. (a) Equal-assay mean over the eight distant-variant test sets against trainable values; the marked gap grows from $+.008$ near 1,000 values to $+.0526$ at 12,288. (b) At the exact match, latent adaptation improves mean Spearman by $+0.0526$ over LoRA (95% CI $[+0.023, +0.079]$), with gains on 7/8 assays. (c) Accuracy against what a deployment must *store* per target; marker shape denotes a state, low-rank weight delta, full model copy, or supervised head.

Table 3: **Controlled adaptation on the eight near-wild-type splits.** $\Delta\rho$ is ours minus the comparison row, so positive favours the latent. The upper block controls capacity, labels, objective, backbone, and optimization budget; the lower block provides reference points using different adaptation objects or budgets. Intervals are the locked seed-1337 paired assay bootstrap, and one excluding zero is the only verdict we assert.

Method	Trainable	ρ	$\Delta\rho$ (ours − row)	95% CI	Wins
Capacity-, label-, objective- and backbone-matched					
No adaptation (z_{fam} only)	0	.3545	+.1637	[+.084, +.274]	8/8
Rank-12 LoRA (weight delta)	12,288	.4657	**+.0526**	**[+.023, +.079]**	**7/8**
same row, constrained latent			+.0186	[−.003, +.039]	5/8
Latent state, constrained	**12,288**	**.4843**	—	—	—
Prefix Δu (activation space)	12,288	.5243	−.0061	[−.015, +.007]	2/8
Latent state, raw (ours)	**12,288**	**.5182**	—	—	—
Different update target; not capacity-matched					
Supervised head (EOS ridge / LLR residual)	515	.4995 / .5329	+.0187 / −.0147	[−.070, +.099]	5/8, 4/8
Full weights, 20 updates	60.1M	.5427	−.0245	[−.053, +.006]	2/8
Full weights, 25 epochs	60.1M	.6776	−.1594	[−.280, −.048]	1/8

Refining the homolog-derived latent with near-wild-type fitness measurements raises the equal-assay mean from .3545 to **.5182** on the distant variant, positive on **8/8 assays** (Table 3), with reseeding the optimizer worth less than .007 (Appendix C). Removing SPG1 and CAPSD, the two assays whose "distant" sets are combinatorial rather than positional, the remaining six still rise from .4136 to .5070, **+0.0934** on **6/6**. Thus, limited local supervision refines the family representation in a way that remains useful over a substantially broader sequence neighborhood.

Adaptation efficiency and backbone capacity. The controlled comparisons above characterize how efficiently the global representation can incorporate supervision. We additionally test whether the 60.1M autoregressive backbone itself limits supervised performance. On the same backbone, increasing the optimization budget raises ρ to .5427 after 20 full-weight updates and .6776 at the 25-epoch ProGen3 budget, while a 515-value log-likelihood-ratio head reaches .5329 (Table 3).

Table 4 places these results alongside published methods on the same eight splits. Under the matched 25-epoch full-weight IRPO protocol, PROPEL's 60.1M backbone reaches **.6776**, compared with approximately .574 for ProGen3-339M and .673 for ProGen3-46B (Bhatnagar et al., 2025). PROPEL additionally uses homologous context, so we interpret this comparison as evidence that the small family-packed autoregressive backbone remains a strong substrate when the adaptation budget is increased, rather than as the primary efficiency claim of the paper.

The compact representation amortizes homolog conditioning. The family latent also reduces the cost of repeatedly evaluating variants from the same target. Once the homolog block has been compressed into the latent, subsequent queries reuse this representation rather than repeatedly

Table 4: **Supervised results on the same eight near-wild-type splits, sorted by ρ.** Rows differ in target inputs, adaptation mechanism, and backbone scale. The three full-weight IRPO rows share the objective, splits, and 25-epoch optimization budget (Bhatnagar et al., 2025, their Table 6). Additional comparison details are given in Appendix D.

Method	Target inputs	Adapts by	Per-target state	ρ
LoRA-IRPO, rank 12 (ours)	homolog block	weight delta	48 KiB	.4657
Latent-IRPO (ours)	**homolog block**	**state**	**48 KiB**	**.5182**
Full-weight IRPO, 20 updates (ours)	homolog block	all weights	229 MiB	.5427
ProGen3-339M IRPO	query only	all weights	1.26 GiB	≈.574
KERMUT (Groth et al., 2024)	query + structure	GP on features	—	.628
ProGen3-46B IRPO	query only	all weights	171 GiB	.673
Full-weight IRPO, 25 epochs (ours)	**homolog block**	**all weights**	**229 MiB**	**.6776**
ConFit (Zhao et al., 2024)	query only	all weights	model copy	.679

processing the full homolog context. Granting an in-context homolog model a fully cached block, what each *scored variant* costs afterwards still differs: with the decoder held fixed, a residue reaching its family through the state performs 258 key–value lookups per layer against $8{,}408$ in context ($\mathbf{32.6}\times$ less attention work per scored residue, $\mathbf{8{,}152}\times$ less resident memory), and inferring the family state is a one-time cost repaid after **393 scored variants**, fewer than every assay in our panel contains (Table 6).

4.4 The evolutionary representation conditions sequence generation

Fitness prediction alone does not establish that the inferred family representation is integrated into the autoregressive generative model. A representation could improve variant ranking while exerting little influence on the model's output distribution. We therefore test whether conditioning changes both the probability assigned to measured variants and sequences sampled directly from the frozen decoder. Shuffled-label and wrong-family states provide controls for changes unrelated to the target information encoded by the latent.

We first read the conditioned decoder as a distribution over each assay's measured library, $q_z(v) \propto p(v \mid z)$. Fitness adaptation places **.4597** of the probability mass in the top fitness decile, $4.6\times$ a random ranking, or .3825 excluding SPG1 (Figure 5a). Applying the identical pipeline with shuffled fitness labels reaches only .1812 (Table 10, Appendix F), showing that the probability shift tracks the information contained in the measurements.

Conditioning also changes the sequences themselves, not only the ranking of a fixed library. Scoring 500 decoded proposals per assay per condition with a five-model ESM-1v ensemble (Meier et al., 2021), family-conditioned proposals outperform unconditioned ones at every mutation budget for which we have intervals, with the advantage growing from $+0.64$ at $m = 2$ to $+1.62$ at $m = 5$ (Figure 5b).

The effect is also visible directly in sequence composition. Composition KL to the natural family is .1416 under $z = 0$; the target family representation reduces it to **.0360**, whereas a *wrong*-family representation moves it in the opposite direction to .1748. The same substitution reverses the sign of the decoder's own family preference (Table 11). Together, these results show that the family-level representation is not merely an auxiliary feature for variant ranking: it participates directly in the autoregressive model's output distribution and provides target-specific conditioning for generation (Appendix F).

5 Conclusion

A supervised update has to go somewhere. For a protein language model it belongs in the small conditional state the weights already consume: because that state is produced by optimization rather than by an encoder, an alignment and an assay can both write it and one frozen decoder reads both, turning target knowledge into an object a laboratory can keep, send and rewrite.

AI USE STATEMENT

We used generative AI tools to help implement methods, write code, interpret results, draft parts of this paper, and improve its readability. We did not use them to develop theoretical models or conceptual frameworks, propose or refine hypotheses, design experiments, advise on methodology, translate text, or clean or reformat datasets. Synthetic data generation, mathematical claims and proofs, and qualitative analysis were not applicable to this work. The authors tested AI-assisted code, checked all reported numbers against runs we conducted, reviewed AI-assisted text and interpretations against our results, and verified every reference against its original source. We take responsibility for the final paper, including content produced with the help of generative AI.

REFERENCES

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022. doi: 10.18653/v1/2022.acl-short.1.

Aadyot Bhatnagar, Sarthak Jain, Joel Beazer, Samuel C. Curran, Alexander M. Hoffnagle, Kyle S. Ching, Michael Martyn, Stephen Nayfach, Jeffrey A. Ruffolo, and Ali Madani. Scaling unlocks broader generation and deeper functional understanding of proteins. In *Advances in Neural Information Processing Systems*, 2025.

Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1078–1086. PMLR, 2018. URL https://proceedings.mlr.press/v80/cremer18a.html.

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, others, and D. Baker. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7112–7127, 2022. doi: 10.1109/TPAMI.2021.3095381.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13:4348, 2022. doi: 10.1038/s41467-022-32007-7.

Quentin Fournier, Robert M. Vernon, Almer van der Sloot, Benjamin Schulz, Sarath Chandar, and Christopher James Langmead. Protein language models: Is scaling necessary? *bioRxiv*, 2024. doi: 10.1101/2024.09.23.614603. Introduces AMPLIFY.

Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K. Min, Kelly Brock, Yarin Gal, and Debora S. Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021. doi: 10.1038/s41586-021-04043-8.

Cade W. Gordon, Amy X. Lu, and Pieter Abbeel. Protein language model fitness is a matter of preference. In *International Conference on Learning Representations (ICLR)*, 2025.

Peter Mørch Groth, Mads Herbert Kerrn, Lars Olsen, Jesper Salomon, and Wouter Boomsma. Kermut: Composite kernel regression for protein variant effects. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2305.18466.

Alex Hawkins-Hooker, Shikha Surana, Jack Simons, Jakub Kmec, Oliver Bent, and Paul Duckworth. Likelihood-based fine-tuning of protein language models for few-shot fitness prediction and design. *bioRxiv*, 2024. doi: 10.1101/2024.05.28.596156.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018.

Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. RITA: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*, 2022.

Thomas A. Hopf, John B. Ingraham, Frank J. Poelwijk, Charlotta P. I. Schärfe, Michael Springer, Chris Sander, and Debora S. Marks. Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35(2):128–135, 2017. doi: 10.1038/nbt.3769.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *PMLR*, pp. 2790–2799, 2019.

Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 40(7):1114–1122, 2022. doi: 10.1038/s41587-021-01146-5.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.09685.

Long-Kai Huang, Rongyi Zhu, Bing He, and Jianhua Yao. Steering protein language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. arXiv:2509.07983.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. arXiv:1312.6114.

Deqian Kong, Bo Pang, Tian Han, and Ying Nian Wu. Molecule design by latent space energy-based modeling and gradual distribution shifting. In Robin J. Evans and Ilya Shpitser (eds.), *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 1109–1120. PMLR, 2023. URL `https://proceedings.mlr.press/v216/kong23a.html`. arXiv:2306.14902.

Deqian Kong, Yuhao Huang, Jianwen Xie, Edouardo Honig, Ming Xu, Shuanghong Xue, Pei Lin, Sanping Zhou, Sheng Zhong, Nanning Zheng, and Ying Nian Wu. Molecule design by latent prompt transformer. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 89069–89097. Curran Associates, Inc., 2024a. doi: 10.52202/079017-2826. arXiv:2402.17179.

Deqian Kong, Dehong Xu, Minglu Zhao, Bo Pang, Jianwen Xie, Andrew Lizarraga, Yuhao Huang, Sirui Xie, and Ying Nian Wu. Latent plan transformer for trajectory abstraction: Planning as latent space inference. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 123379–123401. Curran Associates, Inc., 2024b. doi: 10.52202/079017-3922. arXiv:2402.04647.

Deqian Kong, Minglu Zhao, Dehong Xu, Bo Pang, Shu Wang, Edouardo Honig, Zhangzhang Si, Chuan Li, Jianwen Xie, Sirui Xie, and Ying Nian Wu. Latent thought models with variational Bayes inference-time computation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31419–31436. PMLR, 2025. URL `https://proceedings.mlr.press/v267/kong25c.html`. arXiv:2502.01567.

Elodie Laine, Yasaman Karami, and Alessandra Carbone. GEMME: A simple and fast global epistatic model predicting mutational effects. *Molecular Biology and Evolution*, 36(11):2604–2619, 2019. doi: 10.1093/molbev/msz179.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021. doi: 10.18653/v1/2021.acl-long.353.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level

protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022a. Introduces $(IA)^3$ and T-Few.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, 2022b. doi: 10.18653/v1/2022.acl-short.8.

Amy X. Lu, Wilson Yan, Sarah A. Robinson, Kevin K. Yang, Vladimir Gligorijevic, Kyunghyun Cho, Richard Bonneau, Pieter Abbeel, and Nathan Frey. Generating all-atom protein structure from sequence-only training data. *bioRxiv*, 2024. doi: 10.1101/2024.12.02.626353.

Yunan Luo, Guangde Jiang, Tianhao Yu, Yang Liu, Lam Vo, Hantian Ding, Yufeng Su, Wesley Wei Qian, Huimin Zhao, and Jian Peng. ECNet is an evolutionary context-integrated deep learning framework for protein engineering. *Nature Communications*, 12:5743, 2021. doi: 10.1038/s41467-021-25976-8.

Ali Madani, Ben Krause, Eric R. Greene, Subu Subramanian, Benjamin P. Mohr, James M. Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z. Sun, Richard Socher, James S. Fraser, and Nikhil Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41:1099–1106, 2023. doi: 10.1038/s41587-022-01618-2.

Joe Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3403–3412. PMLR, 2018. URL https://proceedings.mlr.press/v80/marino18a.html.

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 *Advances in Neural Information Processing Systems*, volume 34, 2021.

Erik Nijkamp, Bo Pang, Tian Han, Linqi Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning multi-layer latent variable model via variational optimization of short run MCMC for approximate inference. In *Computer Vision – ECCV 2020*, pp. 361–378. Springer, 2020. doi: 10.1007/978-3-030-58539-6_22. arXiv:1912.01909.

Erik Nijkamp, Jeffrey Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978.e3, 2023. doi: 10.1016/j.cels.2023.10.002.

Donghun Noh, Deqian Kong, Minglu Zhao, Andrew Lizarraga, Jianwen Xie, Ying Nian Wu, and Dennis Hong. Latent adaptive planner for dynamic manipulation. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2430–2448. PMLR, 2025. URL https://proceedings.mlr.press/v305/noh25a.html. arXiv:2505.03077.

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

Pascal Notin, Lood Van Niekerk, Aaron W. Kollasch, Daniel Ritter, Yarin Gal, and Debora S. Marks. TranceptEVE: Combining family-specific and family-agnostic models of protein sequences for improved fitness prediction. *bioRxiv*, 2022b. doi: 10.1101/2022.12.07.519495. Also presented at the NeurIPS 2022 LMRL workshop.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Hansen Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora S. Marks. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64331–64379, 2023a.

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

Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21994–22008. Curran Associates, Inc., 2020. arXiv:2006.08205.

Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, volume 37, pp. 116617–116637, 2024. arXiv:2404.19733.

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

Roshan M. Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA transformer. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *PMLR*, pp. 8844–8856, 2021.

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.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118.

Emre Sevgen, Joshua Moller, Adrian Lange, John Parker, Sean Quigley, Jeff Mayer, Poonam Srivastava, Sitaram Gayatri, David Hosfield, Clayton Dilks, others, and Andrew L. Ferguson. ProT-VAE: Protein transformer variational autoencoder for functional protein design. *Proceedings of the National Academy of Sciences*, 122(41):e2408737122, 2025. doi: 10.1073/pnas.2408737122.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *PMLR*, pp. 9229–9248, 2020.

Felix Teufel, Aaron W. Kollasch, Yining Huang, Ole Winther, Kevin K. Yang, Pascal Notin, and Debora S. Marks. Few-shot protein fitness prediction via in-context learning and test-time training. *arXiv preprint arXiv:2512.02315*, 2025. AI for Science Workshop, NeurIPS 2025. Introduces PRIMO.

Timothy Truong Jr. and Tristan Bepler. PoET: A generative model of protein families as sequences-of-sequences. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3384.

Timothy F. Truong Jr. and Tristan Bepler. Understanding protein function with a multimodal retrieval-augmented foundation model. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025. Introduces PoET-2; arXiv:2508.04724.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023. Introduces Activation Addition (ActAdd); earlier title "Activation Addition: Steering Language Models Without Optimization".

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, others, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620 (7976):1089–1100, 2023. doi: 10.1038/s41586-023-06415-8.

Ruben Weitzman, Peter Mørch Groth, Lood van Niekerk, Aoi Otani, Yarin Gal, Debora Marks, and Pascal Notin. Protriever: End-to-end differentiable protein homology search for fitness prediction. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *PMLR*, 2025. arXiv:2506.08954.

Jude Wells, Alex Hawkins-Hooker, Micha Livne, Weining Lin, David Miller, Christian Dallago, Nicola Bordin, Brooks Paige, Burkhard Rost, Christine Orengo, and Michael Heinzinger. ProFam: Open-source protein family language modelling for fitness prediction and design. *bioRxiv*, December 2025. doi: 10.64898/2025.12.19.695431. Introduces ProFam-1 (251M params).

Talal Widatalla, Rafael Rafailov, and Brian Hie. Aligning protein generative models with experimental fitness via direct preference optimization. *bioRxiv*, 2024. doi: 10.1101/2024.05.20.595026. ProteinDPO.

Nicholas C. Wu, Lei Dai, C. Anders Olson, James O. Lloyd-Smith, and Ren Sun. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife*, 5:e16965, 2016. doi: 10.7554/eLife.16965.

Junming Zhao, Chao Zhang, and Yunan Luo. Contrastive fitness learning: Reprogramming protein language models for low-N learning of protein fitness landscape. *bioRxiv*, 2024. doi: 10.1101/2024.02.11.579859.

A Extended related work

Protein language models. Autoregressive models learn a distribution over sequences and score variants by likelihood (Madani et al., 2023; Nijkamp et al., 2023; Bhatnagar et al., 2025; Hesslow et al., 2022; Ferruz et al., 2022); masked models do the same with pseudo-likelihood (Rives et al., 2021; Lin et al., 2023; Meier et al., 2021; Elnaggar et al., 2022; Fournier et al., 2024). In both cases a family's information is distributed across parameters shared with every other family.

Homolog conditioning. MSA Transformer attends over aligned sets (Rao et al., 2021); PoET and PoET-2 treat a family as a sequence of sequences (Truong Jr. & Bepler, 2023; 2025); ProFam trains on packed homolog sets, the closest architectural relative to our pretraining (Wells et al., 2025); Tranception and Protriever add retrieval (Notin et al., 2022a;b; Weitzman et al., 2025); EVE, DeepSequence and EVmutation fit per-family models (Frazer et al., 2021; Riesselman et al., 2018; Hopf et al., 2017); GEMME uses evolutionary conservation directly (Laine et al., 2019). Their cost is paid per scored sequence and their context is not differentiable. ProT-VAE places a small family latent between frozen transformer blocks (Sevgen et al., 2025) and is wet-lab validated, but that latent comes from a per-family encoder rather than test-time optimization against a shared decoder.

Supervised fitness prediction. Gaussian-process kernels over embeddings (Groth et al., 2024), non-parametric transformers over labelled neighbours (Notin et al., 2023b), ridge on augmented density features (Hsu et al., 2022), supervised residue networks (Luo et al., 2021) and contrastive low-N objectives (Zhao et al., 2024) all fit a predictor beside a frozen representation.

Parameter-efficient adaptation and steering. LoRA (Hu et al., 2022), adapters (Houlsby et al., 2019), $(\mathrm{IA})^3$ (Liu et al., 2022a) and BitFit (Ben Zaken et al., 2022) keep the update in weight space; see Lialin et al. (2023) for a survey. Prefix and prompt tuning (Li & Liang, 2021; Lester et al., 2021; Liu et al., 2022b) optimize activations. Test-time training adapts on the instance at hand (Sun et al., 2020; Hardt & Sun, 2024), and PRIMO brings it to this benchmark (Teufel et al., 2025). Activation addition edits hidden states without optimization (Turner et al., 2023), including for protein models (Huang et al., 2025).

Generation with latent or structural conditioning. PLAID generates all-atom structure from sequence-only training (Lu et al., 2024); ProteinMPNN and RFdiffusion condition on structure (Dauparas et al., 2022; Watson et al., 2023); ESM3 generates across tracks (Hayes et al., 2025). None exposes a small per-target state written by both an alignment and an assay.

B Architecture, data, and the full label-free protocol

Backbone. 12 pre-norm layers, width 512, 8 heads, SwiGLU hidden dimension 1376, RMSNorm ($\epsilon = 10^{-5}$), rotary position embeddings, tied embedding/unembedding, GPT-2 scaled residual initialization ($\sigma = 0.02/\sqrt{2L}$), dropout 0, 32-token vocabulary (20 canonical amino acids plus `BOS`/`EOS`/`PAD`/`UNK` and rare-residue codes). Sliding-window causal self-attention with window 256. $N_z = 24$ latent slots, two per layer, with learned layer and slot embeddings and rotary positions strided over `max_seq_len`. The prior network U_θ is three full-attention transformer blocks over the 24 slots with no rotary embedding. Parameter counts: 37,965,312 in decoder self-attention and feed-forward blocks, 12,589,056 in the twelve cross-attention modules, 9,489,920 in U_θ, 32,768 tied embedding values; 60,077,056 total.

Corpus. PoET-style blocks from a DIAMOND search of 10M UniRef50 queries against the full database. Greedy best-fit packing to a 2,048-token budget; homolog h sampled with weight $\mathrm{bits}(h)^\beta/(c+\mathrm{appearances}(h))^\alpha$ at $\alpha = \beta = c = 1$; per-protein appearance cap $K = 10$; singletons dropped; seed 1337. Result: 6,120,136 blocks over 18,054,482 distinct accessions, 11,993,195,285

Table 5: **Where the label-free read sits on ProteinGym v1.3.** Official grouped Spearman with the benchmark's own bootstrap standard error, from the v1.3 summary file (Notin et al., 2023a). Target information differs across blocks and this is *not* an information-matched ranking: our model and the alignment blocks read homologs, the generative and encoder blocks read only the query. The row that matters most for reading this paper is Site-Independent—a first-order position-specific frequency profile built from the same alignments we sample from—which our score does not separate from. Our per-category scores are Expression .399, Stability .375, Activity .366, OrganismalFitness .337, Binding .323; the flat 217-assay mean is .36047, which is not the official statistic.

Model	Parameters	Pretrain tokens	Target information	ρ
Alignment profile and per-family models				
Site-Independent	—	—	homologs	.359
EVmutation	per family	—	homologs	.395
EVE (ensemble)	per family	—	homologs	.439
GEMME	—	—	homologs	.455
Autoregressive generative models				
ProGen3-112M	112M	—	query sequence	.282
ProGen2-small	151M	175B	query sequence	.336
ProGen3-339M	339M	200B	query sequence	.347
PROPEL (ours)	**60M**	**13.5B**	**homolog block**	**.360**
ProGen2-base	764M	200B	query sequence	.378
ProGen2-xlarge	6.4B	350B	query sequence	.391
ProGen3-3B	3.0B	500B	query sequence	.392
Masked / encoder models				
ESM-2	35M	—	query sequence	.321
ESM-2	650M	—	query sequence	.414
Homolog-conditioned and retrieval models				
MSA Transformer (ens.)	100M	—	homologs	.432
TranceptEVE-L	—	—	query + homologs	.456
PoET	200M	—	homologs	.470
Protriever	—	—	query + retrieved	.479
Current v1.3 leaders, for scale				
ESCOTT	—	—	structure + homologs	.476
ProSST (K=2048)	—	—	query + structure	.507
VenusREM	—	—	structure + homologs	.518
AIDO Protein-RAG	16B	—	structure + homologs	.518

non-padding tokens at a 95.685% fill ratio, 200 shards, 5,949,807 training and 170,329 validation blocks. 3,850,193 candidate rows were dropped so that training and validation shards share no sequence identifier.

Pretraining. Outer loop: AdamW, learning rate 10^{-4}, $\beta = (0.9, 0.95)$, weight decay 0.1, gradient clipping 1.0, 1,000 warmup iterations, cosine decay to 10^{-5}, bfloat16 autocast, 6-way data parallel at a global batch of $6 \times 64 \times 3 = 1152$ blocks. Inner loop: AdamW on $(\mu, \log \sigma^2)$, $T = 16$ steps, constant inner learning rate 0.3, $\mu \leftarrow z_{\text{prev}}$ (persistent per example), $\log \sigma^2 \leftarrow -5 + 0.1 \cdot \mathcal{N}(0, 1)$, one noise draw reused across the T steps, $\beta_{\text{KL}} = 1$. 6,000 iterations process 1.354×10^{10} non-padding tokens, 1.16 epochs. (The checkpoint's `total_trained_tokens` field records 2,256,096,581, the count on one of the six data-parallel ranks; the global figure is six times that.) Checkpoint `ckpt_6000.pt`, SHA1 `6bc0cc93...a95614`, `per_layer` latent architecture, held-out reconstruction cross-entropy 1.99 nats per token under a 32-step posterior; as in Kong et al. (2025) this is a conditional quantity and is not comparable to the marginal perplexity of a model without a latent.

Test-time label-free write. $z_0 = 0.01 \cdot \mathcal{N}(0, I)$ at generator seed 1337; $T = 128$ AdamW steps at inner learning rate 0.34 with cosine decay to one tenth; β_{KL} at its pretraining value. ProteinGym v1.3 is aggregated by the official procedure: the per-assay Spearman is averaged within each (UniProt ID, functional category) group, then within each of the five functional categories, then unweighted across the five. This is not the arithmetic mean over the 217 assays, which is .36047.

Table 6: **Cost of reaching the family, within one architecture.** The decoder is held fixed; only the route from the homolog block to a prediction varies. The in-context column assumes a fully cached block, which favours it. Lookups are per residue per layer; footprint is what must be resident per target while variants are scored.

	Through the state	Block in context	Ratio
Attention lookups / residue / layer	258	8,408	32.6×
Resident per-target footprint	48 KiB	382 MiB	8,152×
Prepaid before scoring	T=128, 5 prompts	none	—
Variants to break even	393		—

Where the label-free read sits. Table 5 places our .360 against the official v1.3 summary. The row that matters most is Site-Independent at $.359 \pm .012$: a first-order position-specific frequency profile built from the same alignments we sample from, which our score does not separate from. Two homolog-conditioned models published after our comparison set was fixed are also above us: ProFam-1 (251M, packed homolog sets) at .47 and PoET-2 (182M) (Wells et al., 2025; Truong Jr. & Bepler, 2025).

C Experimental details

Backbone and label-free write. The decoder has 12 layers of width 512 (60.1M parameters) with 256-token sliding-window attention and $N_z = 24$ latent slots, two per layer, injected by cross-attention. Pretraining processes 1.354×10^{10} non-padding tokens in family-packed UniRef50 blocks of 2,048 tokens—6,000 optimizer iterations at a global batch of $6 \times 64 \times 3$ blocks and a 95.69% fill ratio, or 1.16 epochs over the 5.95M-block training split. (The checkpoint's `total_trained_tokens` field records 2,256,096,581, which is the count on one of the six data-parallel ranks; the global figure is six times that.) All results use checkpoint `ckpt_6000.pt` (SHA1 `6bc0cc93...a95614`) with the `per_layer` latent architecture. At test time the family state z_{fam} is inferred with 128 posterior steps from a label-free homolog block of up to 8,152 tokens while the backbone stays frozen. Variants are ranked by length-normalized conditional log-likelihood. ProteinGym v1.3 is aggregated by the official procedure: the per-assay Spearman is averaged within each (UniProt ID, functional category) group, then within each of the five functional categories, then unweighted across the five. This is not the arithmetic mean over the 217 assays, which is .36047.

Full-weight arms. Both full-weight arms update all 60.1M parameters under the objective of Eq. 3 with the same labels, splits and five-prompt scoring as the latent arms. The 20-update arm exists to match the latent arm's optimizer budget; the 25-epoch arm exists to match Bhatnagar et al. (2025), whose Table 6 specifies 25 ProteinGym epochs at every model size. Both epoch counts were therefore fixed by those two external constraints before any distant-test variant was scored, and no distant-test label entered the choice. **We have not run a learning-rate sweep for either full-weight arm, and we report neither a selection rule nor a seed replication for them**; the latent and LoRA arms carry both. That asymmetry in reporting discipline is a limitation of the full-weight rows and a reason to read them as context rather than as a controlled result.

Variant scoring protocols. Sequences up to the 2,048-token scoring row are scored by full-sequence conditional log-likelihood. Longer proteins use one of three windowed likelihood-ratio estimators, selected per assay from the assay's own metadata and never from any score: a mutation-centred crop, a first-mutation-to-last-mutation span extended by 256 residues, and a leftmost-suffix variant, each scoring $\log p(\text{variant window} \mid z) - \log p(\text{WT window} \mid z)$ and summing over windows when a variant's mutations do not fit one. Windows are placed from mutated positions and the wild-type crop only. 105 of the 217 substitution assays exceed 256 residues and 55 exceed 512, so a substantial share of the label-free number is computed under a windowed estimator; we did not ablate the estimator against full-sequence scoring on the assays where both apply.

Homolog block occupancy. Token budget, not alignment depth, is what limits the family state. On SPG1, whose ProteinGym MSA holds 3,109 records, an 8,152-token block admits 21 sequences at 96–99% fill. The context-length panel of Figure 2b is therefore also a statement about how many homologs the state sees, and it is flat above roughly 2,048 tokens.

Supervised write and operating points. The residual Δz receives 20 pairwise-preference updates; the adapted state is $z_{\text{adapt}} = z_{\text{fam}} + \gamma\,\Delta z$. Two operating points are fixed before any distant-test evaluation: *raw* ($\alpha = 0.2$, $\beta = 0.05$, $\gamma = 0.75$, lr $= 0.03$, sum pair reduction) and *constrained* ($\alpha = 0.4$, $\beta = 0.05$, $\gamma = 1.0$, lr $= 0.03$, mean pair reduction), where α and β are the terms of Eq. 3 and γ scales the learned residual as in §3.2. Blocks hold $n = 64$ variants ($n = 32$ when the write set is smaller than 1,000, following Bhatnagar et al. (2025)); every strictly ordered pair within a block contributes, giving roughly 2,000 pairs per update, and $\log p_{\text{adapt}}$ is the mean over the five prompt states. Reference log-likelihoods are cached once at $\Delta z = 0$. The 20 updates are 20 optimizer steps, with 10% linear warmup and cosine decay to half the peak learning rate. The budget itself came from the same near-wild-type selection as the rest of the objective: the staged sweep scored inner-validation Spearman at checkpoint steps $\{1, 2, 5, 10, 20\}$, and the selected configuration's median best step across assays was 20, the maximum of that grid, so the budget is a boundary value rather than a demonstrated saturation point; we did not extend the grid. Every adapted arm—latent, weight delta and activation-space write—receives the same 20 updates, so the matched comparisons hold optimization budget fixed alongside capacity. What the budget does not establish is any arm's ceiling, and §4.3 shows a longer full-weight schedule is worth considerably more on this backbone.

Rank-12 LoRA is applied to `layers.11.attention.wo`, which gives exactly 12,288 trainable values; it shares the labels, objective, frozen backbone and optimizer schedule with the latent arms. Its learning rate was selected independently: a $\{0.003, 0.01, 0.03\}$ sweep scored on near-wild-type training labels with inner validation only, under the pre-declared rule "maximize mean per-assay validation Spearman, break ties toward the lower rate", giving mean validation Spearman $.4425$, $.4315$ and $.4079$ respectively and selecting 0.003. No distant-test label entered that selection. Confidence intervals are paired assay bootstraps with 100k resamples at seed 1337.

Generation. Measured-library steering uses $q_z(v) \propto p(v \mid z)^{1/\tau}$ at $\tau = 1$ and reports top-decile fitness mass. Decoding allows substitutions only at assay-mutable positions, with budgets $m \in \{1, 2, 3, 5\}$ and 500 proposals per cell; the independent score is a five-model ESM-1v ensemble with a hierarchical assay-by-proposal bootstrap (10,000 resamples, seed 20260824). Generative-competence diagnostics use 100 trajectories per condition on GFP, SPG1 and CAPSD.

Replication and prompt ensembling. Optimizer-order reseeding (seeds 1337/2027/2029) gives adapted eight-assay means of .5182/.5120/.5123, i.e. 0.5142 ± 0.0035 (sample SD across seeds); the seed-averaged lift is $+0.1597$ with paired assay-bootstrap 95% interval $[+0.082, +0.263]$. Before the supervised write, a five-prompt family ensemble beats the best single prompt by $+0.0138$ (.3545 versus .3407); after the write the gap is $+0.0040$ (.5182 versus .5142) and all five single prompts land in $[.4863, .5142]$.

Test-time grid for the label-free write. Figure 2 uses a 32-assay length-stratified ProteinGym panel with exactly eight assays per length bucket (128–256, 256–512, 512–1024 and >1024 residues), the same checkpoint as everywhere else, and the headline scoring protocol held fixed (`poet_neighbors` homolog sampling over max-identity thresholds $\{1.0, 0.95, 0.9, 0.7, 0.5\}$, fast learning rate 0.34 with cosine decay, mutation-span scoring).

Posterior steps (Figure 2). Run over all 217 substitution assays at 8,152-token context: .2089 (T=16), .2824 (T=32), .3413 (T=64), .3595 (T=128), against .0597 with no family state. Mutants per assay are capped at 5,000 by a subsample seeded on the assay id, so every cell scores the same variants; on six published baselines the cap shifts grouped Spearman by $-.0004$ to $-.0007$, and on our own checkpoint T=128 capped gives .3595 against .3605 uncapped. Paired assay bootstrap over the 217 assays, 20,000 resamples: each doubling is positive with an interval excluding zero ($+.0735$ $[+.063, +.085]$, $+.0590$ $[+.047, +.071]$, $+.0182$ $[+.010, +.027]$), and end to end $+.1506$ $[+.131, +.171]$ on 189 of 217 assays.

Homolog context (Figure 4). Run on the 32-assay length-stratified panel (eight assays per length bucket) at T=128, every cell an equal-assay mean over all 32: .3214 (2,048 tokens), .3469 (5,096), .3415 (8,152), .3484 (16,304). No pair of settings is separable—all six paired 95% intervals span zero, including 5,096 versus 8,152, whose median favours 8,152—so the panel is read as a saturation point and not as a ranking.

Table 7: **Near-wild-type write, distant-variant test.** k is the largest substitution count seen during adaptation; the test set holds every variant beyond it. Both predeclared latent arms improve all eight assays.

Assay	k	Write labels	Distant test	z_{fam}	Raw	Constr.	Raw $- z_{\mathrm{fam}}$
D7PM05_CLYGR	1	1,169	23,346	.4456	.5551	.5130	$+.1095$
GFP_AEQVI	1	1,084	50,630	.6643	.7106	.6897	$+.0463$
Q6WV12_9MAXI	1	1,141	30,260	.3401	.4681	.4118	$+.1280$
Q8WTC7_9CNID	1	1,201	32,309	.2393	.3852	.3253	$+.1459$
GCN4_YEAST	4	617	2,021	.2881	.3545	.3543	$+.0664$
SPG1_STRSG	2	2,167	147,193	$-.0492$	.4586	.3906	$+.5078$
CAPSD_AAV2S	1	532	41,796	.4032	.6450	.6273	$+.2418$
HIS7_YEAST	2	1,643	494,494	.5044	.5684	.5622	$+.0640$
Mean	—	**9,554**	**822,049**	**.3545**	**.5182**	**.4843**	$\mathbf{+.1637}$

D SAME-SPLIT COMPARISON: PROVENANCE AND CAVEATS

What a bigger backbone buys, and what it costs. Four published methods report on these exact splits (Table 4), and our latent arm is below all of them on ranking accuracy. Fully fine-tuning our own 60.1M backbone for 25 epochs reaches **.6776**, which is within .005 of ProGen3-46B IRPO's .673 and below ConFit's .679. We report that as a method-class-matched comparison rather than an interface result, with two asymmetries stated together: our model reads a homolog block and the ProGen models read only the query sequence, which favours us; and their backbones and token budgets are far larger, which favours them. The optimization budget *is* matched—Bhatnagar et al. (2025) report 25 ProteinGym epochs at every model size in their Table 6, and our converged arm uses 25 epochs; our separate 20-update arm (.5427) is a different, deliberately short-budget arm and is not the comparison point. The .005 difference carries no interval on either side. Every weight-space row in that table costs a model copy per assay. What separates .4657 from .5182 is the destination at a fixed budget; what separates .5182 from .5427 and .6776 is optimization budget on one backbone; and what separates those from the ProGen rows is a different backbone entirely. We make no claim on the last of those.

KERMUT and ConFit are supervised methods of a different class again and are included only to locate the frontier.

For storage context, one 48 KiB adapted state is $4{,}888\times$ smaller than an fp32 copy of our own 60.1M backbone (≈229 MiB), roughly $27{,}600\times$ smaller than ProGen3-339M, and about 3.7×10^6 times smaller than ProGen3-46B. The headline .5182 uses a five-prompt ensemble and therefore five states, 240 KiB, dividing each of those ratios by five ($977\times$, $5{,}500\times$, 7.4×10^5); a single-prompt deployment costs 48 KiB and scores .4863–.5142 depending on which prompt is fixed in advance (§4.3).

E ADAPTATION ANATOMY

What is in the space. Three measurements say what kind of object the state is. *Supervision writes directions, not scale*: rescaling the family state does almost nothing—one global gain moves the eight-assay mean by $+0.0004$ and twelve per-layer gains by $+0.0028$—while a single rank-1 *direction*, 536 values, recovers 44% of the full write's lift, distributed across all twelve layers with a mild bias toward the last third. *One pre-declared parameter trades fitness against naturalness on the same object*: the write moves the model off the natural-sequence distribution and γ sets how far, so our constrained operating point gives up 21% of the fitness lift to remove **74%** of the write's likelihood cost on an identity-disjoint natural-family holdout and still lands above matched LoRA, with the same movement visible in generation (§4.4); both endpoints were fixed before any distant-test variant was scored, so a deployment picks a point on that trade-off without re-tuning. *Homologs and labels write complementary information into the same values*: labels alone lift the $z = 0$ cell to .3184, homologs alone reach .3545 with no label, and the joint write is worth 85% of the sum of the two gains (Table 2).

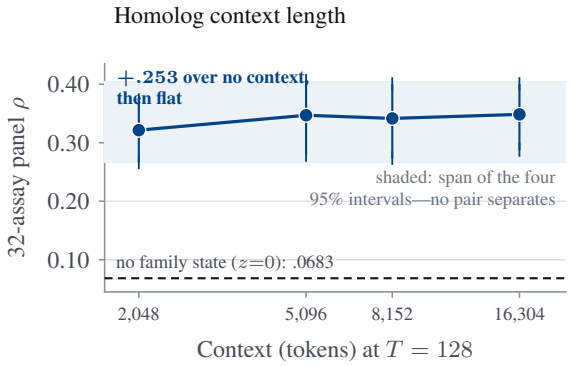


Figure 4: **Homolog context saturates.** Context at the matched step budget $T = 128$ on the same 32-assay panel. The curve rises out of 2,048 tokens and is flat thereafter: *no* pair of context settings is separable—all six paired 95% intervals span zero, including 5,096 versus 8,152, whose median favours 8,152. We therefore read this as a saturation point, not a ranking, and attach no significance claim to the ordering of its three largest settings. Token budget rather than alignment depth is the binding constraint: on SPG1, whose MSA holds 3,109 records, an 8,152-token block admits 21 sequences.

Table 8: **Capacity ladders.** Equal-assay mean ρ on the eight distant-variant test sets. Both families of writes use the same frozen backbone, labels, and objective. The first two latent rows can only rescale the family state.

Trainable values	Latent ρ	LoRA ρ
1 (global scale)	.3549	—
12 (per-layer scale)	.3573	—
536 (rank-1 direction)	.4270	—
1,024 (LoRA r=1)	—	.4312
1,072 (rank-2)	.4390	—
2,048 (LoRA r=2)	—	.4397
2,144 (rank-4)	.4508	—
4,096 (LoRA r=4)	—	.4441
12,288 (full / r=12)	**.5182**	.4657

Table 9: **Layerwise write energy.** Equal-assay mean of each layer's share of $\|\Delta z_l\|_F^2$, in percent; a uniform write would put 8.33% in every layer. The two operating points share the same depth-biased routing profile.

Layer	1	2	3	4	5	6	7	8	9	10	11	12
Raw	7.185	7.009	7.547	7.360	7.811	6.730	8.282	8.819	9.677	7.797	10.037	11.748
Constrained	5.886	7.607	7.069	7.261	7.726	6.635	9.084	8.854	9.910	7.213	10.376	12.379

Naturalness is a dial, and the write sets it. The write moves the model away from the natural-sequence distribution, and the amount is tunable. The constrained operating point removes **74%** of the write's likelihood cost on an identity-disjoint natural-family holdout while retaining **79%** of its fitness lift (Appendix E), and still lands above matched LoRA. The same movement shows up in generation: amino-acid composition KL to the natural family rises from .0360 under z_{fam} to .1201 under the raw write (§4.4). Both endpoints were fixed before any test evaluation, so a deployment can pick a point on the accuracy–naturalness dial without re-tuning. The write also removes the model's dependence on prompt ensembling: the five-prompt ensemble beats the *best* single prompt by $+0.0138$ before the write and $+0.0040$ after it. A deployment must fix its prompt in advance, and the five adapted single-prompt scores span $[.4863, .5142]$, so dropping four of five scoring passes costs between $.004$ and $.032$ depending on that choice—the upper end is 60% of the latent's advantage over matched LoRA, so we do not recommend a single prompt without per-target validation (Appendix C).

Depth profile of the write. The write is distributed across depth rather than concentrated at one adapter site, with a mild bias toward the last third of the network, and the two operating points share that profile (Table 9).

Likelihood cost of the write. On an identity-disjoint natural-family holdout, mean perplexity rises from 11.86 under z_{fam} to 23.71 under the raw write (worse on 8/8 assays) and to 14.89 under the constrained write (worse on 7/8; one assay improves). The constrained arm is not likelihood-preserving; it costs 74% less likelihood than the raw arm and keeps 79% of its fitness lift.

F EXTENDED LIMITATIONS

Release. We release the pretrained checkpoint, the family-packed block construction, the posterior inference and write code, and the per-assay states behind every arm in this paper, so that the latent space itself—not only the numbers read off it—is the artifact.

Where the numbers sit, and how far each dial goes. PROPEL is not the most accurate predictor on either task. Read label-free it sits among alignment-reading methods rather than above them: a site-independent frequency profile built from the same alignments reaches $.359$ against our $.360$, and homolog-conditioned and structure-aware methods lead the benchmark (Table 5). On the eight supervised splits a 515-value likelihood-ratio head reaches $.5329$ against the latent arm's $.5182$, and the full-weight arm at matched updates reaches $.5427$; eight paired assays cannot separate that from the latent ($\Delta = -0.0245$, 95% CI $[-0.053, +0.006]$) and the point estimate favours it, so that comparison is a failure to separate and not an equivalence result. The read-side compute dial is bounded and visibly so: the three doublings return $+.0735$, $+.0590$ and $+.0182$, and extrapolating that decay puts the ceiling between .37 and .38, short of the v1.3 leaders; we have not run $T{=}256$ to test the extrapolation. Homolog context length is the weaker and flatter dial, saturating by roughly 5,000 tokens with no pair of settings separable (Figure 4), so $8{,}152$ is an operating point nothing measurably beats rather than a peak. The matched-budget result is established at the raw operating point and not at both: the constrained arm is $+0.0186$ over matched LoRA with a 95% interval of $[-0.003, +0.039]$ and 5/8 wins—the same sign, unresolved at this sample size.

What the evaluation panel does and does not support. Four of the eight supervised assays are green-fluorescent-protein homologs scored by fluorescence, so an assay-level bootstrap that treats them as exchangeable is optimistic and the effective number of families is five. "Distant" also means different things across the panel: HIS7 and the four fluorescent proteins are genuine extrapolation to unseen positions, while SPG1 and CAPSD are saturated combinatorial windows whose held-out sets are higher-order combinations of positions their write sets already cover, which is why §4.3 reports the panel with both removed. The label budget is fixed by the benchmark, so what we measure is near-wild-type-to-distant transfer rather than general sample efficiency. The $z = 0$ cell is a point the decoder never saw in training, so it upper-bounds the state's contribution rather than isolating it, and the 85% decomposition in Table 2 is descriptive only. Our two operating points are two declared settings of γ, not a dose–response.

What the generation evidence covers. Per-assay steering effects are heterogeneous: relative to the family state the raw write improves top-decile mass on 5/8 assays and worsens it on three, and two

assays supply 76% of the $+.1956$ mean (Table 10). At the decoded-proposal level the fitness-aligned versus shuffled contrast is not resolved at this sample size. ESM-1v is external to our model but is not an independent instrument for family membership, since it is a likelihood model of the same natural-sequence universe our backbone was pretrained on, and the unconditioned $z = 0$ baseline it is compared against is weak. Conditioning has two costs: sampled trajectories retain a .89–.91 canonical-residue rate where $z = 0$ and natural sequences are at 1.00, so conditioning is what introduces the non-canonical tokens and masking the sampling logits would remove them; and the fitness write partly undoes the family conditioning it is layered on, with composition KL rising from .0360 under z_{fam} to .1201 under the raw write against .1416 at $z = 0$.

Controls we did not run. The matched comparison varies the destination but not the injection topology. Two arms at exactly $12{,}288$ values would separate them: rank-1 LoRA across all twelve output projections, and a per-layer prefix of two free 512-dimensional vectors at the cross-attention slots, which has the shape and site of u with the prior, the KL and U_θ removed. The prefix arm in particular would decide how much of the effect is the *latent-variable* structure rather than activation space as such. Neither is run, nor are the tiny-capacity weight baselines $(\mathrm{IA})^3$ and BitFit (Liu et al., 2022a; Ben Zaken et al., 2022), a training-free protein steering baseline (Huang et al., 2025), a one-hot site-independent ridge on the write labels (Hsu et al., 2022)—the natural baseline for a saturated combinatorial design like SPG1—or a write into a site-independent profile, which would test whether the advantage is the destination or the write. We report no head-to-head against PRIMO (Teufel et al., 2025), whose splits differ from ours. Our generation claims concern the decoder's output distribution—composition, family preference, and the fitness region of a measured library—and we make no claim that decoded sequences are members of the target family; the identity statistics in Table 11 bound that directly. Scoring decoded proposals against the assays' own measured values, rather than against a likelihood model, is the evaluation we would add next.

Cost we did not normalize. We report per-target storage rather than wall-clock. Filling the state costs $T = 128$ forward-backward passes over an 8,152-token block per prompt, paid once per target; scoring a variant afterwards is one forward pass over that variant. We have not normalized either against the per-variant cost of an in-context homolog model, which pays its context on every sequence scored.

Contamination and replication. The pretraining corpus is UniRef50, so the assay targets and their homologs are present in it, as they are for every protein language model evaluated on ProteinGym; we performed no decontamination. This matters more for us than for a single-sequence model, because our claim is that z_{fam} is inferred at test time from a target's homologs rather than recalled from pretraining, and the $z = 0$ ablation does not separate those hypotheses. A state inferred from an unrelated family, scored on the ranking task, would; we run that control for generation but not for ranking. All results come from one checkpoint, and replication on an independently trained backbone is future work.

G GENERATION DIAGNOSTICS

Decoded-proposal contrasts. The family-conditioned advantage over unconditioned decoding is $+0.6438$ $[+0.0127, +1.7358]$ at $m = 2$, $+1.0016$ $[+0.0523, +2.6333]$ at $m = 3$, $+1.6188$ $[+0.0415, +4.2797]$ at $m = 5$, and $+1.0881$ $[+0.0317, +2.8746]$ pooled; all four intervals exclude zero. Adapted conditioning is also positive against unconditioned decoding ($+0.7549$, $+1.2190$, $+1.8342$; pooled $+1.2694$ raw and $+1.3490$ constrained; means only). At $m = 1$ the corresponding means are $+0.3224$ (family), $+0.3574$ (adapted raw), $+0.3935$ (adapted constrained), and $+0.1890$ for adapted raw against the shuffled control. Two further contrasts are not resolved at this sample size and we make no claim from them: adapted raw minus shuffled pooled is $+0.6114$ $[-0.1064, +1.5892]$, and adapted raw minus family pooled is $+0.1813$ $[-0.4535, +1.0897]$.

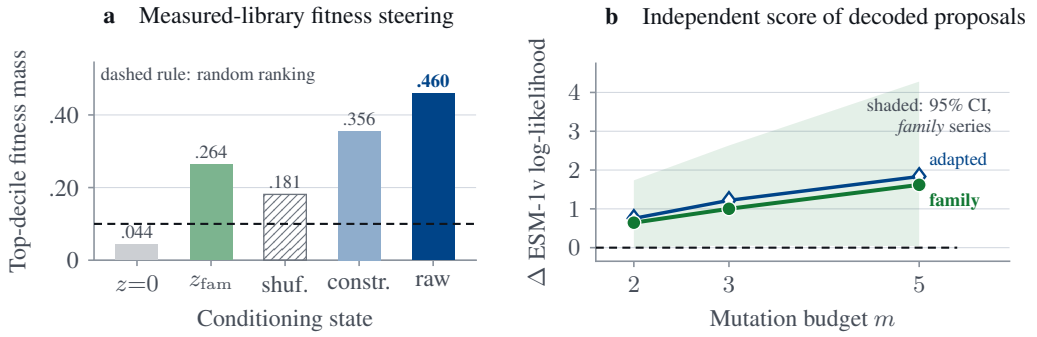


Figure 5: **One state, written two ways, moves both the fitness region and the family the frozen decoder produces.** (a) Top-decile measured-fitness mass under the induced distribution q_z, equal-assay means over eight assays; .10 is what a random ranking gives, and the shuffled-reward arm (hatched) is the same pipeline with permuted labels. Per-assay effects are heterogeneous (5/8 positive; Table 10). (b) ESM-1v ensemble score of 500 decoded proposals per assay per cell relative to unconditioned decoding. The shaded band is the asymmetric 95% hierarchical (assay × proposal) bootstrap on the *family* series; it is an absolute, unpaired interval, and the paired adapted-minus-family differences it does not describe are +.111, +.217 and +.215. Only means are stored for the adapted series. The family-swap diagnostic is Figure 6.

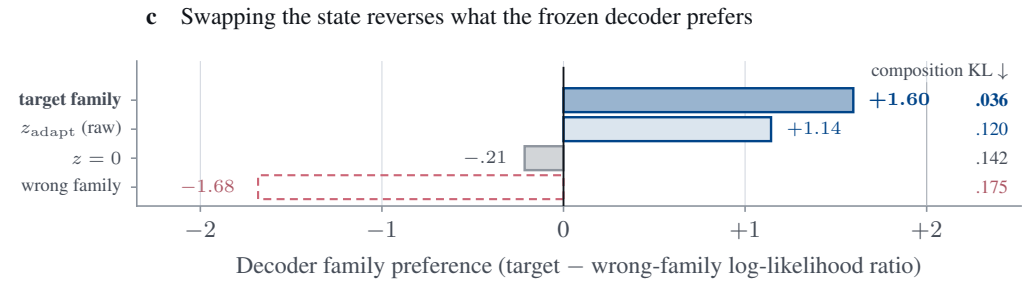


Figure 6: **Swapping the state reverses what the frozen decoder prefers.** Equal-assay means over GFP, SPG1 and CAPSD, 100 trajectories per condition. Target- and wrong-family states flip the sign of the decoder's preference (AUROC 1.0 on all three families); composition KL to the natural family is printed at right (held-out natural sequences: .0003). The preference score is a decoder self-consistency diagnostic, not an independent family classifier, and the AUROC is saturated on a contrast separable by length and composition alone.

Table 10: **Per-assay change in top-decile measured-fitness mass relative to the family state.** Equal-assay means are −0.0829 (shuffled reward), +0.0918 (constrained) and +0.1956 (raw). Per-assay effects are heterogeneous: the raw and constrained writes are positive on 5/8 assays and the shuffled control on 3/8, and SPG1 and Q6WV12 supply 76% of the raw write's mean. The raw and constrained entries for SPG1 are identical to four decimals because both saturate: each puts essentially all of the mass (0.99999 and 1.00000) in the top decile of a library whose top decile is 14,720 of 147,193 variants, so the two arms cannot be distinguished on that assay by this statistic. Absolute top-decile mass for the same conditions is .0436 ($z = 0$), .2641 (z_{fam}), .1812 (shuffled), .3559 (constrained) and .4597 (raw), against .10 for a random ranking.

Assay	Shuffled reward	Constrained	Raw
CAPSD	−.2883	−.1036	−.1359
GFP	+.0342	−.0261	−.0535
GCN4	−.0983	−.1089	−.0015
HIS7	+.0861	+.0917	+.1099
D7PM05	−.0337	+.0359	+.1476
Q8WTC7	−.0648	+.0870	+.3136
Q6WV12	+.0211	+.1197	+.5455
SPG1	−.3191	+.6389	+.6389
Equal-assay mean	**−.0829**	**+.0918**	**+.1956**

Table 11: **Generative competence over GFP, SPG1 and CAPSD**, 100 trajectories per condition. Target conditioning gives the lowest composition KL and reverses the decoder's family preference relative to a wrong-family state; that preference measures decoder self-consistency rather than independent recognition. Target-versus-wrong AUROC is exactly 1.0 on all three families, as is the adapted-versus-wrong and the held-out natural-versus-wrong comparison. Three saturated AUROCs do not validate the probe: GFP, SPG1 and the AAV2 capsid are 238, 448 and 735 residues with unrelated folds, so this contrast is separable on length and composition alone, and the probe cannot resolve anything finer. A composition-matched shuffle control and a within-family contrast (the four fluorescent proteins of our own panel are homologs at 20–30% identity) are the tests that would. Sampled trajectories retain a .89–.91 canonical-token rate (failures are kept, not resampled), and maximum identity to natural inference sequences stays near .26 against .957 for held-out natural sequences, so these are family-level rather than sequence-level competence results.

Conditioning	Canonical rate	Composition KL ↓	Max identity	Pairwise identity	Family preference
$z = 0$	1.00	.1416	.2643	.2678	$-.214$
z_{fam} (target)	.91	**.0360**	.2692	**.2500**	$\mathbf{+1.596}$
z_{fam} (wrong family)	.91	.1748	.2579	.2719	-1.682
z_{adapt} (raw)	.89	.1201	.2636	.2617	$+1.143$
Held-out natural family	1.00	*.0003*	*.9572*	*.5947*	—