C2S: CONDITION-TO-SIMPLEX PROTEIN DESIGN WITH RESPONSE-AWARE SEQUENCE SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein design seeks sequences that realize desired behaviors, yet structure-conditioned methods largely optimize compatibility with prescribed conformations rather than behavior under a specified context. We introduce Condition-to-Simplex (C2S), which recasts condition-specific design as control over a simplex of structural states. C2S translates a functional specification into continuous state preferences and introduces π-FiLM to act on state contrasts, steering sequence probabilities according to the desired regime rather than a fixed conformation. A response model then resolves the remaining choice in functional space, selecting candidates by their predicted behavior under the specified context. Across structural-control and measured-response tasks, C2S improves both request following and functional target matching. This formulation provides a path from designing for structural states to designing for context-dependent protein behavior.Our code is available at `https://anonymous.4open.science/r/papercode`.

1 INTRODUCTION

Protein design aims to create sequences that achieve a desired function (Chu et al., 2024; Notin et al., 2024). That function depends on the conditions in which a protein is used. A receptor may need to reach a chosen response at a given ligand concentration, while a binding protein may need to interact more strongly with a particular partner. In both cases, structure describes only part of the design problem: the functional request also specifies the context and the desired outcome (Lisanza et al., 2025; Listov et al., 2024). A design model should therefore use this request to guide both the search for candidates and the choice of the final sequence.

Current methods address individual parts of this problem, but do not directly connect the functional request to the full design process. Inverse-folding models such as ProteinMPNN (Dauparas et al., 2022) and ESM-IF1 (Hsu et al., 2022) condition sequence generation on a given structure, but the structure itself does not specify the desired functional outcome. Multistate methods incorporate several conformations or structural states, but typically require their contributions to the design objective to be specified separately from the requested response. Conversely, response predictors such as Boltz-2 (Passaro et al., 2025) can estimate how a candidate behaves in an interaction context, for example through its binding affinity, but they operate on candidates that have already been generated. Thus, these methods provide the components of a generate-score-select workflow without specifying how a functional request should control both candidate generation and final selection (Figure 1(a)).

A functional request must influence not only how candidates are evaluated, but also which regions of sequence space are explored. This distinction between predicting sequence function and using predictions to guide sequence-space search is central to model-guided protein engineering (Sinai & Kelsic, 2020). For condition-specific design, it gives two requirements: the request must determine the relative preference among structural states during generation, and it must define the target response under the specified condition for selection.

We introduce Condition-to-Simplex (C2S) to implement these two requirements (Figure 1(b)). The model maps a functional request to a simplex over structural states, whose coordinates specify the requested preference between unbound and bound conformations. Rather than using these weights only to combine state-level scores, π-FiLM couples them to state-specific structural differences within a DynamicMPNN (Abrudan et al., 2026) generator, making the generated sequence distribution

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

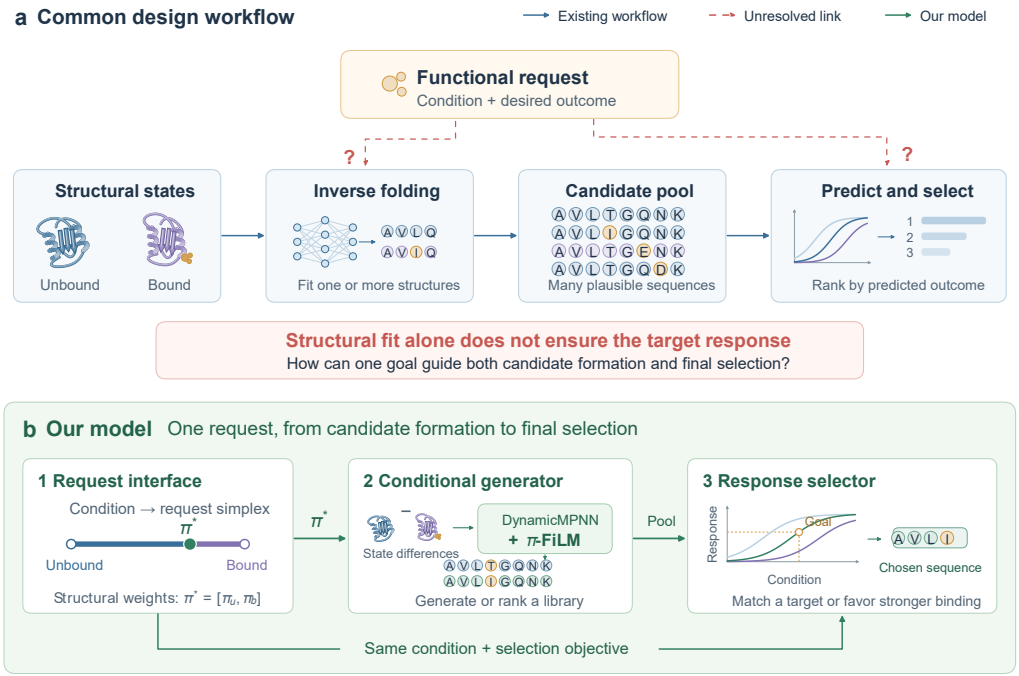

Figure 1: One functional request guides the full design model. (a) Common workflows form candidates from structures and rank their predicted responses. (b) Condition-to-Simplex uses a request interface, a conditional generator, and a response selector. The request sets both the structural weights used to form candidates and the objective used to choose a sequence. Structures and curves are schematic.

conditional on the request. The same request specifies the response context and target; a response predictor evaluates each candidate under that context, and the selector chooses the sequence closest to the target response. The framework supports both de novo generation and ranking within an existing sequence library.

Experiments validate both stages of the model. At request coordinates not seen during training, state-contrast conditioning outperforms a parameter-matched scalar-FiLM control in all nine protein–evaluator comparisons. Under a fixed eight-candidate budget, request-guided formation reduces best-in-pool target error by 0.031 relative to random candidate pools. Within fixed pools, response-aware selection reduces PYL1 target-response error by 19.1% relative to structural ranking; on PDZ, the reduction in selected binding $\Delta\Delta G$ reaches 36.9% when transferring to held-out domains.

Our contributions are:

- **Condition-to-Simplex formulation.** We introduce a unified formulation that translates condition-specific functional goals into continuous structural control and response-level design objectives.
- **State-contrast conditioning.** We develop π-FiLM, which couples simplex coordinates directly to state-feature differences, providing directional and continuous control of multistate sequence preferences.
- **Function-aligned design.** We close the loop between structural generation and functional choice, enabling sequences to be designed for condition-specific responses rather than structural compatibility alone.

2 Related Work

Structural control in multistate sequence design. Inverse-folding models such as ProteinMPNN (Dauparas et al., 2022), ESM-IF1 (Hsu et al., 2022), and PiFold (Gao et al., 2022) condition

sequence design on a target structure. Multistate methods extend this formulation to alternative conformations or structural ensembles, including DynamicMPNN (Abrudan et al., 2026), Caliby (Shuai et al., 2025), ABACUS-T MSD (Liu et al., 2025), and ProteinGenerator (Lisanza et al., 2025). ConformationalBiasing (Cavanagh et al., 2026) and CSDesign (Stern et al., 2025) further exploit differences between states to favor a target conformation. C2S extends structural control from a fixed or externally specified state preference to a continuous preference derived from the functional request.

Condition-aware functional prediction. Functional predictors estimate sequence behavior beyond structural compatibility. ProteinNPT (Notin et al., 2023) predicts protein properties from labeled support sequences, while Boltz-2 (Passaro et al., 2025) predicts interaction properties including binding affinity. Such models provide the response estimates needed to evaluate sequences in a functional context. In C2S, this capability is used to evaluate whether a candidate reaches the requested response under the specified condition, rather than to assess structural compatibility.

Functional guidance of sequence design. Recent methods move beyond post hoc evaluation by incorporating functional objectives into sequence search. MosPro (Luo et al., 2025) uses learned property predictors to bias sequence distributions toward desired properties, SAGE-Prot (Lim et al., 2025) combines generation and property scoring in closed-loop optimization, and ProteinGuide (Xiong et al., 2026) guides pretrained sequence generators toward user-specified properties. These methods establish property-guided generation as an effective design paradigm. C2S targets a different form of control: it decomposes a condition-specific functional request into a structural-state preference for candidate formation and a response target for final selection, thereby aligning multistate generation and functional choice through the same request.

3 METHOD

C2S maps a functional request and structural states to a selected sequence. The model has three components: a request interface, a conditional sequence generator, and a response selector. The request controls both candidate formation and final choice (Overview in Figure 2).

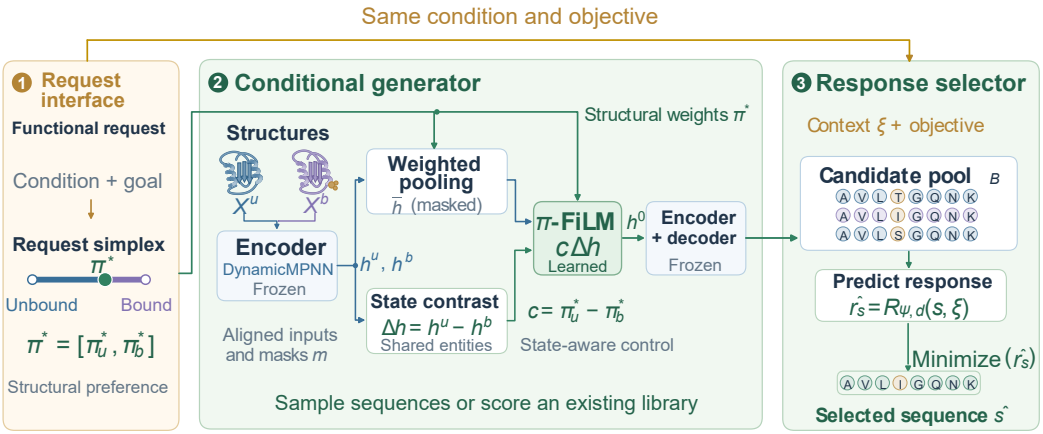

Figure 2: Overview of Condition-to-Simplex with the learned two-state generator. The request sets structural weights, response context, and the selection objective. The generator pools state features and adds a request-dependent state-contrast correction through π-FiLM. The response selector evaluates the candidates and chooses a sequence for the same goal. Structures are schematic.

3.1 MODEL DEFINITION AND REQUEST INTERFACE

Let X^u and X^b be aligned unbound and bound structures, m their node and edge availability masks, and d the protein-family and partner context. Given a request $\mathcal{R}$, the full model returns $\hat{s} = F_{\theta,\phi,\psi}(X^u, X^b, m, d, \mathcal{R})$. The request interface supplies structural weights $\pi^* = [\pi_u^*, \pi_b^*]$,

response context ξ, and objective $\mathcal{L}_{\mathcal{R}}$. The weights are nonnegative and sum to one. For a candidate budget B, inference is

$$\begin{aligned} \mathcal{S}_B &= G_{\theta,\phi}(X^u, X^b, m; \pi^*, B), \\ \widehat{s} &= \arg\min_{s \in \mathcal{S}_B} \mathcal{L}_{\mathcal{R}}(R_{\psi,d}(s, \xi)) \,. \end{aligned} \tag{1}$$

The generator $G_{\theta,\phi}$ forms at most B candidates. The selector predicts their responses with $R_{\psi,d}$ and applies the request objective. Structural weights guide the search, while ξ preserves the condition or partner identity for the final decision.

Concentration-based requests. For ligand concentration $C > 0$ and target dissociation constant K_d^*, a one-site binding model under ligand excess gives

$$\pi_b^* = \frac{C}{C + K_d^*}, \qquad \pi_u^* = 1 - \pi_b^*. \tag{2}$$

A target bound fraction $q^* \in (0, 1)$ gives the same interface through $K_d^* = C(1 - q^*)/q^*$, so that $\pi_b^* = q^*$. Requests with the same C/K_d^* share a generator coordinate and are evaluated at their own concentrations. For a measured response system, a fixed wild-type curve provides the mapping. In PYL1, let $H_{\mathrm{WT}}(C)$ be a four-parameter Hill curve with asymptotes B_0, B_∞, midpoint EC_{50}, and coefficient n. We set

$$a_{\mathrm{WT}}(C) = \frac{H_{\mathrm{WT}}(C) - B_0}{B_\infty - B_0} = \frac{1}{1 + (\mathrm{EC}_{50}/C)^n}, \qquad \pi^*(C) = [1 - a_{\mathrm{WT}}(C), a_{\mathrm{WT}}(C)], \tag{3}$$

with $a_{\mathrm{WT}}(0) = 0$. This is a calibrated coordinate for structural control. The functional target y^* is the published normalized wild-type assay response at C, and selection minimizes $\mathcal{L}_{\mathcal{R}}(r) = |r - y^*|$.

Partner-defined requests. In PDZ, partner identity selects an ordered apo/partner-bound structural pair and the endpoint $\pi^* = [0, 1]$. The response context identifies that interaction, and the objective is $\mathcal{L}_{\mathcal{R}}(r) = r$, where r is predicted binding $\Delta\Delta G$. The request therefore favors stronger binding to the specified partner. The model supports both a quantitative response target and a partner-defined ordering of candidates.

3.2 Conditional sequence generation

The conditional generator combines a frozen DynamicMPNN backbone (Abrudan et al., 2026), with parameters θ, and a learned π-FiLM module, with parameters ϕ. The backbone encodes each structure into scalar and vector features $h_{e,k}^{\chi}$ for node or edge e, state $k \in \{0, 1\}$, and feature type $\chi \in \{s, V\}$. States 0 and 1 denote unbound and bound structures. The mask $m_{e,k}$ marks whether an entity is observed. Availability-aware pooling applies the request weights:

$$\overline{h}_e^{\chi} = \frac{\sum_{k=0}^{1} \pi_k^* m_{e,k} h_{e,k}^{\chi}}{\sum_{k=0}^{1} \pi_k^* m_{e,k}}. \tag{4}$$

This makes the relative influence of the structures depend on the request. When an entity is observed in only one state and that state has positive weight, pooling returns its feature. The contrast correction below acts only where both states are observed. This keeps the shared structural information available for request control while respecting missing observations.

State-contrast modulation. Within the generator, π-FiLM couples the request to differences between state features. We define

$$c = \pi_0^* - \pi_1^*, \qquad z = \mathrm{SiLU}(W_m |c| + b_m), \qquad \Delta h_e^{\chi} = \mathbf{1}[m_{e,0} m_{e,1} = 1](h_{e,0}^{\chi} - h_{e,1}^{\chi}). \tag{5}$$

The module updates scalar and vector channels as

$$\begin{aligned} s_e' &= \overline{s}_e + [\tanh(\alpha_e^s(z)) + \beta_e^s(z)] \odot c \Delta s_e, \\ V_e' &= \overline{V}_e + \tanh(\alpha_e^V(z)) \odot c \Delta V_e. \end{aligned} \tag{6}$$

Nodes and edges use separate learned affine projections α, β; vector scales are shared across Cartesian components. The resulting features pass to the frozen post-pooling encoder and autoregressive decoder. The generator can sample sequences or score an existing library.

The signed product $c\Delta h_e$ ties the direction of control to the difference between states. At $\pi^* = [0.5, 0.5]$, the correction is exactly zero. Swapping the structures and their weights leaves the output unchanged, since both c and Δh_e change sign. Zero-initialized output projections also preserve the parent model at initialization. These properties keep the control neutral at the midpoint and independent of state naming.

Learning the generator's request control. We train the 11,284 π-FiLM parameters while keeping the backbone fixed in evaluation mode. Each record uses state-0 weights $a \in \mathcal{A} = \{0.1, 0.5, 0.9\}$. The outer anchors use the corresponding pure-state parent predictions as teachers; the midpoint uses the two-state parent with π-FiLM disabled. Teachers and students share a masked-logit interface with sequence embeddings set to zero. Let $\tau_{a,t}$ and $p^{\phi}_{a,t}$ be their amino-acid distributions at shared site $t \in \mathcal{M}$. We minimize

$$\mathcal{L}_\phi = \operatorname*{mean}_{a\in\mathcal{A},\, t\in\mathcal{M}} \mathrm{KL}(\tau_{a,t} \| p^{\phi}_{a,t}) + 0.25 \operatorname*{mean}_{t\in\mathcal{M}} \mathrm{SmoothL1}(D^{\phi}_t - D^{\tau}_t), \tag{7}$$

where $D^{\bullet}_t = \widetilde{\ell}^{\bullet}_{0.9,t} - \widetilde{\ell}^{\bullet}_{0.1,t}$ and $\widetilde{\ell}_{a,t}(b) = \log p_{a,t}(b) - \frac{1}{20}\sum_{b'} \log p_{a,t}(b')$. The first term anchors the requested distributions. The second transfers the teacher's direction of change across requests. All anchors use the same paired record, so this term compares changes in amino-acid preference at the same sites. The request enters through a continuous coordinate. The same learned parameters can be evaluated at weights between the training anchors.

Each update uses one paired structure record and all three anchors. We optimize with AdamW for three epochs, using learning rate 10^{-3}, zero weight decay, and gradient-norm clipping at one. The final checkpoint defines the learned control module; Appendix B.1 gives the data split and parameter accounting.

An analytical generator variant. The request interface also supports an analytical generator for $K = 2$–5 states. It requires no training. Each state predicts $p_{t,k}(b)$ for the same decoded prefix. With sharpened weights $\rho_k = (\pi^*_k)^{1.5} / \sum_j (\pi^*_j)^{1.5}$, the next-token distribution is

$$w_{t,k} = \frac{m_{t,k}\rho_k}{\sum_j m_{t,j}\rho_j}, \qquad \overline{p}_t(b) = \sum_k w_{t,k} p_{t,k}(b). \tag{8}$$

For two states, an optional contrast term provides directional guidance at jointly observed sites:

$$P_{t,\gamma}(b) \propto \exp\left(\frac{\log \overline{p}_t(b) + \gamma(2\rho_0 - 1)[\log p_{t,0}(b) - \log p_{t,1}(b)]}{\tau_{\mathrm{dec}}} \right), \tag{9}$$

where γ sets guidance strength and τ_{dec} is the sampling temperature. This term favors amino acids whose relative state scores agree with the request. At a fixed prefix, increasing γ cannot decrease the expected signed local score contrast (Appendix A). This variant supplies control across multiple states. We use the learned two-state generator for the linked PYL1 and PDZ tasks.

3.3 Response-aware selection

The response selector combines a task-specific predictor with the request objective. It evaluates each candidate on the relevant assay scale and selects a sequence through Equation (1). The prediction inputs and supervision match each task.

Concentration-dependent response. For PYL1, the generator ranks the substitution library at $\pi^*(C)$ and retains the Top-B candidates. A dual-view predictor combines frozen absolute and mutation-delta representations with $a_{\mathrm{WT}}(C)$ to predict $\widehat{q}(s, C)$. The absolute view describes the variant, while the delta view captures changes from wild type. Selection minimizes $|\widehat{q}(s, C) - q_{\mathrm{WT}}(C)|$. Thus the same concentration sets the structural coordinate and the response target.

Partner-specific binding. For PDZ, the generator prioritizes substitutions by their bound-endpoint score relative to the midpoint. Writing $\ell_G(s; \pi)$ for its sequence score, this contrast is $\ell_G(s; [0, 1]) - \ell_G(s; [0.5, 0.5])$. We retain at most B candidates. A predictor uses sequence, mutation, interaction, endpoint, and aggregate generator features to estimate binding $\widehat{\Delta\Delta G}_\psi(s, d)$. The selector chooses its minimum. Binding measurements supervise the response model, and partner identity links structural preference to the final choice.

Dual-view affinity prediction. For Tite-Seq, we use a fixed candidate library to evaluate response selection. An ESM-2 encoder (Lin et al., 2023) supplies contextual embeddings $E_i(s)$. For variant s and wild type s_{WT}, the two views are

$$v_{\mathrm{abs}}(s) = \frac{1}{L}\sum_{i=1}^{L} E_i(s), \qquad v_{\Delta}(s) = \frac{1}{|\mathcal{M}_s|}\sum_{i\in\mathcal{M}_s}[E_i(s) - E_i(s_{\mathrm{WT}})], \tag{10}$$

where $\mathcal{M}_s$ contains mutated sites and L counts non-padding residues. The absolute view summarizes the full sequence context. The delta view focuses on changes at mutated positions relative to wild type. The mutation fraction records the proportion of changed sites. A 128-unit GELU–LayerNorm head combines these views with the mutation fraction $|\mathcal{M}_s|/L$. It uses dropout 0.10 and produces a binder logit $\widehat{z}_{\mathrm{bind}}(s)$ and log-affinity $\widehat{\ell}_K(s) = \widehat{\log_{10} K}_d(s)$, bounded to $[-10.5, -4.5]$. The expected assay response is

$$R_{\psi,d}(s, C) = \sigma(\widehat{z}_{\mathrm{bind}}(s))\frac{C}{C + 10^{\widehat{\ell}_K(s)}}. \tag{11}$$

Here σ is the sigmoid function. The response combines the probability of binding with the conditional bound fraction. The two predicted quantities depend on the sequence, while concentration enters through the binding curve. The same binding probability and affinity define responses across concentrations. The selector compares each response with the requested occupancy. We fine-tune the encoder and heads using class-balanced binder loss, censored SmoothL1 affinity loss, and occupancy-interval loss. The censoring term uses the measured affinity bounds, and the response term trains predictions on a fixed concentration grid. Inner-validation occupancy-interval error determines checkpoint selection. This aligns checkpoint choice with the response scale used for selection. The measured intervals preserve the information in censored assays without treating an assay limit as an exact affinity value.

We combine the 8M, 35M, and 150M ESM-2 scales using validation-based weights. If E_j^{val} is the validation error at scale j, its weight is

$$\omega_j = \frac{\exp(-E_j^{\mathrm{val}}/T_{\mathrm{ens}})}{\sum_r \exp(-E_r^{\mathrm{val}}/T_{\mathrm{ens}})}, \qquad T_{\mathrm{ens}} = 0.01. \tag{12}$$

Appendix B gives the training details. The ensemble weights favor scales with lower validation error and are fixed before test selection.

Candidate budgets and fixed libraries. For PDZ, a context with n observed substitutions contributes $\min(B, n)$ candidates. When the budget covers the inventory, all variants enter response selection. In Tite-Seq, response models use the same fixed library for each concentration–occupancy request.

Request-guided inference. We train generator control and response predictors separately. At inference, the generator samples sequences or ranks a library under π^*. The selector evaluates the candidate pool in the requested context and minimizes target-response error or predicted binding $\Delta\Delta G$ (Equation (1)).

4 Experiments

Our experiments are organized around four questions. First, can the generator follow structural preferences at request coordinates not seen during training? Second, does response-aware selection improve the functional outcome of the selected sequence? Third, under a limited candidate budget, does request-guided generation make better candidates available for selection? Finally, how does the choice of response model affect functional selection?

4.1 Evaluation setup

Data and evaluation settings. We train the request-control module on 4,530 records from the Protein Ensemble Database (PED) (Lazar et al., 2021) and evaluate on 518 held-out records after overlap filtering. Three independently trained DynamicMPNN models (Abrudan et al., 2026) serve as frozen

generator backbones. We additionally test structural request control on the open–closed conformations of LplA, ADK, and MBP. Functional evaluation uses two experimentally measured substitution maps: the PYL1-ABI1 map, which measures mutational effects across ligand concentrations (Stammnitz & Lehner, 2026), and the PDZ map, which measures folding and binding effects across five domains and seven domain-peptide interactions (Martí-Aranda & Lehner, 2026). PYL1 requests specify a concentration-dependent target response, whereas PDZ requests specify a binding partner.

Comparison protocol. We evaluate response selectors on the same candidate pools and candidate-formation methods with the same response selector. The main library comparisons use an eight-candidate budget, denoted Top-8. PYL1 Top-19 retains all 19 non-wild-type substitutions at each position. In the original functional tests, candidates and predictions are fixed before held-out measurements are used to score the selected variants. We average repeated decisions within mutation positions or position groups. Appendix B gives the training settings, paired bootstrap procedures, and complete evaluation protocols.

4.2 Control at unseen request weights

State contrasts make the generator more responsive to the requested structural preference. We train at state-0 weights of 0.1, 0.5, and 0.9, then test at 0.2, 0.35, 0.65, and 0.8. We measure request sensitivity as the slope of the state log-likelihood difference against the requested weight. A larger slope means that changing the request produces a stronger shift toward the desired state.

Figure 3 compares state-contrast conditioning with scalar-FiLM across three proteins and three frozen inverse-folding evaluators. The gain is positive in **all nine comparisons**. Both models have 11,284 trainable parameters and share the parent checkpoints, training data, teachers, optimizer, and sampling budget. Across 11,232 generated sequences in the six-policy evaluation, the standardized slope gain is 0.491 over scalar-FiLM and 0.539 over probability arithmetic. This comparison links the stronger response to the use of state differences in conditioning.

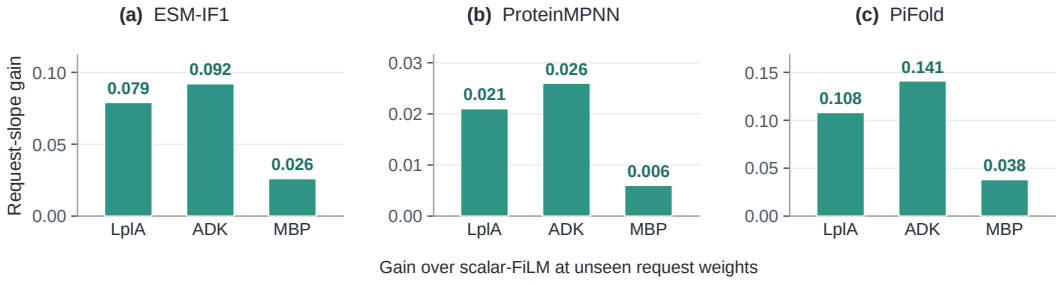

Figure 3: **Stronger request control at unseen weights.** Bars show preference–request slope gains over the parameter-matched scalar-FiLM control. Each evaluator uses its own score scale. All nine gains are positive; full paired intervals and companion quality measures are reported in Appendix C.

The effect also holds across the 518 held-out PED records and all three parent seeds. The aggregate slope gain over the frozen parent is 0.030 at unseen weights. On 169 MotifDiv structure pairs(Stern et al., 2025), the same ten motif positions are designed in both target directions. Relative to scalar-FiLM, the target-minus-alternate log-likelihood margin increases by 0.113 under both ESM-IF1 and PiFold. Thus, state contrasts improve request sensitivity in the original ensemble setting and in new motif-design contexts. Appendix C reports the seed-wise results, target-state compatibility, and complete transfer comparisons.

4.3 Functional selection from a fixed candidate pool

Matching the requested response. The PYL1–ABI1 map covers substitutions at 169 positions and 12 ABA concentrations (Stammnitz & Lehner, 2026). Concentration sets the structural weights and the wild-type response target. Among the same eight candidates, response-aware selection reduces mean absolute target-response error from 0.298 to 0.241, a **19.1% reduction** (Table 1). The selected variant improves over structural ranking at 119 positions. Since candidate formation is shared, this comparison measures the benefit of choosing for the functional target.

Table 1: **Functional outcomes of the selected sequences.** Structural ranking and response-aware selection use the same Top-8 pools in the position-held-out tests. The domain-held-out test and hit-rate evaluation retain their own protocols. Binding $\Delta\Delta G$ is in kcal mol^{-1}; pp denotes percentage points. Values are rounded for display; relative reductions use unrounded means. Full results are in Appendix D.

Task / outcome	Structural	Response-aware	Improvement
PYL1 target-response MAE ↓	0.298	**0.241**	19.1% lower
PDZ binding $\Delta\Delta G$: positions ↓	0.218	**0.147**	32.8% lower
PDZ binding $\Delta\Delta G$: domains ↓	0.277	**0.175**	36.9% lower
PDZ binding-hit rate ↑	40.1%	**44.3%**	+4.2 pp
PDZ joint-hit rate ↑	35.9%	**39.8%**	+3.9 pp

Selecting for a binding partner. The PDZ map spans five domains and seven peptide interactions (Martí-Aranda & Lehner, 2026). On held-out position groups, response-aware selection lowers mean binding $\Delta\Delta G$ from 0.218 to 0.147 kcal mol^{-1}. The selected mutations are therefore less disruptive to binding than those chosen by structural ranking. Binding labels supervise the selector; folding outcomes also improve when evaluated after selection. The hit-rate evaluation gives the same practical picture: binding hits rise from 40.1% to 44.3%, and joint binding-and-folding hits rise from 35.9% to 39.8%.

Selecting in unseen domains. When an entire PDZ domain is held out, mean selected binding $\Delta\Delta G$ falls from 0.277 to 0.175 kcal mol^{-1}, a **36.9% reduction**. All five held-out domains have favorable point estimates. The benefit therefore extends beyond mutations at unseen positions to selection in unseen domains. A second response architecture also reproduces the PYL1 benefit, reducing error relative to its structural-ranking baseline at both candidate budgets (Appendix E.2).

4.4 Candidate formation under a limited budget

Request-guided candidate formation improves the options available to the response selector. With the response model fixed and the budget held at eight candidates, our pools reduce final target error by 0.027 relative to uniform random selection of the pool (Figure 4(a)). The best-in-pool target error also falls by 0.031. This quantity is the lowest measured target error among the retained candidates, evaluated after selection. Its reduction shows that closer-to-target variants enter the pool before the response model makes its final choice.



Figure 4: **Better candidates under an eight-sequence budget.** (a) Error reductions relative to uniform random pools with the same frozen selector. (b) Reductions from using the correct q80 request at a fixed concentration. Positive bars favor the requested pool; whiskers show paired 95% intervals. Panels use separate scales. Full target-group results are in Appendix D.3.

The requested target matters even when concentration is fixed. In the q80 target group, the correct request lowers final error by 0.0046 relative to a midpoint request and by 0.0064 relative to a reversed request (Figure 4(b)). This comparison changes the request within the same candidate-formation procedure. It complements the random-pool test by linking candidate quality to the target supplied to the model.

Once all 19 substitutions are retained, these requests produce the same inventory and the common response selector chooses the same variants. The budget comparison locates the request effect in

which candidates are retained. This is the decision that matters when only a subset of the available variants can be considered. Appendix D.3 gives the full-inventory and target-group controls, and Appendix F compares external candidate sources.

4.5 Response models for functional selection

Choosing the response composition on validation data further improves target matching. We use nested validation within each outer fold in a six-seed evaluation. On the same evaluation cases, the selected composition reaches Top-8 error 0.218, compared with 0.241 for dual-view, 0.232 for one-hot, and 0.237 for full ProteinNPT (Notin et al., 2023) (Table 2). Its error is 0.217 when the complete substitution inventory is available. The selected composition improves target matching with both a restricted pool and the complete inventory.

Table 2: **PYL1 target-response error by response model.** Methods use the same evaluation cases at each budget. The validation-selected composition is a post-hoc exploratory extension. † Paired 95% intervals favor this model over all three Top-8 baselines. Full estimates and fold-wise choices are in Appendix E.3.

Response model	Top-8 error ↓	Top-19 error ↓
Original dual-view	0.241	0.246
One-hot	<u>0.232</u>	<u>0.228</u>
Full ProteinNPT	0.237	0.235
Validation-selected composition	$\mathbf{0.218}^{\dagger}$	**0.217**

Fixed-library Tite-Seq tests provide a second setting for response selection. At 20 held-out mutation positions, dual-view ESM-2 reaches occupancy-interval error 0.323, compared with 0.351 for the matched-loss ProteinNPT axial-core adaptation and 0.340 for mutation-delta ESM-2. The candidates and requests are shared, so this comparison measures which response representation makes better choices from the same library. Appendix E.1 reports all models and the separate ensemble evaluation.

5 Conclusion and Limitations

Condition-to-Simplex introduces a request-centered formulation of protein design in which a condition-specific functional goal is translated into both structural-state control and a response-level objective. Its key mechanism, π-FiLM, turns continuous simplex preferences into directional sequence control through structural-state contrasts, while response-aware selection closes the design loop at the functional level. Together, these results establish a route from designing sequences for prescribed structural states to designing them for desired responses under specified conditions.

Limitations. C2S currently targets settings in which functional requests can be related to defined structural states. Extending this formulation to functions governed by heterogeneous or latent conformational ensembles will require jointly resolving the relevant states and their functional roles. Future work will extend C2S to these settings by jointly identifying relevant conformational states and learning their functional roles from condition-resolved measurements.

Ethics Statement

This work is a purely computational study. All data used are publicly available and non-sensitive: released protein structure ensembles and previously published protein assay measurements. No human subjects, personally identifiable information, or newly generated wet-lab data are involved. We are not aware of any ethical concerns arising from this work.

Reproducibility Statement

Our code is available at the anonymized repository linked in the abstract. The main text and appendices describe the model, data splits, training settings, and evaluation protocols needed to reproduce our results.

AI Use Statement

In this work, we used generative AI tools to assist in providing feedback on experimental design and on the interpretation and presentation of experimental results. We also used generative AI tools to draft and edit portions of the manuscript, improve the clarity and organization of the writing, conduct literature search, identify relevant references, and format citations. The authors reviewed and verified all content and take full responsibility for the final manuscript.

References

Alex Abrudan, Sebastian Pujalte Ojeda, Chaitanya Joshi, Matthew Greenig, Felipe Engelberger, Alena Khmelinskaia, Jens Meiler, Michele Vendruscolo, and Tuomas Knowles. Multi-state protein sequence design with dynamicmpnn. In *International Conference on Learning Representations*, volume 2026, pp. 153620–153641, 2026.

Rhys M Adams, Thierry Mora, Aleksandra M Walczak, and Justin B Kinney. Measuring the sequence-affinity landscape of antibodies with massively parallel titration curves. *Elife*, 5:e23156, 2016.

Peter E Cavanagh, Andrew G Xue, Shizhong A Dai, Albert Qiang, Tsutomu Matsui, and Alice Y Ting. Computational design of conformation-biasing mutations to alter protein functions. *Science*, 391(6790):eadv7953, 2026.

Alexander E Chu, Tianyu Lu, and Po-Ssu Huang. Sparks of function by de novo protein design. *Nature biotechnology*, 42(2):203–215, 2024.

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

Zhangyang Gao, Cheng Tan, Pablo Chacón, and Stan Z Li. Pifold: Toward effective and efficient protein inverse folding. *arXiv preprint arXiv:2209.12643*, 2022.

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pp. 8946–8970. PMLR, 2022.

Tamas Lazar, Elizabeth Martínez-Pérez, Federica Quaglia, András Hatos, Lucía B Chemes, Javier A Iserte, Nicolás A Méndez, Nicolás A Garrone, Tadeo E Saldaño, Julia Marchetti, et al. Ped in 2021: a major update of the protein ensemble database for intrinsically disordered proteins. *Nucleic acids research*, 49(D1):D404–D411, 2021.

Hocheol Lim, Geon-Ho Lee, Hyein Cho, and Kyoung Tai No. Sage-prot: scoring-assisted generative exploration for multi-objective protein design. *Briefings in Bioinformatics*, 26(6):bbaf585, 2025.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Sidney Lyayuga Lisanza, Jacob Merle Gershon, Samuel WK Tipps, Jeremiah Nelson Sims, Lucas Arnoldt, Samuel J Hendel, Miriam K Simma, Ge Liu, Muna Yase, Hongwei Wu, et al. Multistate and functional protein design using rosettafold sequence space diffusion. *Nature biotechnology*, 43 (8):1288–1298, 2025.

Dina Listov, Casper A Goverde, Bruno E Correia, and Sarel Jacob Fleishman. Opportunities and challenges in design and optimization of protein function. *Nature Reviews Molecular Cell Biology*, 25(8):639–653, 2024.

Yufeng Liu, Rui Wu, Xinyu Wang, Sheng Wang, Linghui Chen, Fudong Li, Quan Chen, and Haiyan Liu. Enhancing functional proteins through multimodal inverse folding with abacus-t. *Nature Communications*, 16(1):10177, 2025.

Jiaqi Luo, Kerr Ding, and Yunan Luo. Pareto-optimal sampling for multi-objective protein sequence design. *Iscience*, 28(3), 2025.

Aina Martí-Aranda and Ben Lehner. Seven complete comparative maps of allosteric mutations in a protein family. *Nature Communications*, 2026.

Pascal Notin, Ruben Weitzman, Debora Marks, and Yarin Gal. Proteinnpt: Improving protein property prediction and design with non-parametric transformers. *Advances in neural information processing systems*, 36:33529–33563, 2023.

Pascal Notin, Nathan Rollins, Yarin Gal, Chris Sander, and Debora Marks. Machine learning for functional protein design. *Nature biotechnology*, 42(2):216–228, 2024.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Richard W Shuai, Tianyu Lu, Subhang Bhatti, Petr Kouba, and Po-Ssu Huang. Ensemble-conditioned protein sequence design with caliby. *bioRxiv*, 2025.

Sam Sinai and Eric D Kelsic. A primer on model-guided exploration of fitness landscapes for biological sequence design. *arXiv preprint arXiv:2010.10614*, 2020.

Maximilian R Stammnitz and Ben Lehner. The genetic architecture of an allosteric hormone receptor. *Nature Communications*, 2026.

Jacob A Stern, Siba Alharbi, Anandsukeerthi Sandholu, Stefan T Arold, and Dennis Della Corte. Conformation-specific design: a new benchmark and algorithm with application to engineer a constitutively active map kinase. *bioRxiv*, pp. 2025–04, 2025.

Junhao Xiong, Ishan Gaur, Maria Lukarska, Hunter Nisonoff, Luke M Oltrogge, David F Savage, and Jennifer Listgarten. Property guidance for protein sequence generative models with proteinguide. *Nature Biotechnology*, pp. 1–13, 2026.

APPENDIX

The appendix gives the model properties, training settings, and full experimental results. We organize the additional experiments around request control, functional selection, response models, and candidate sources.

Results tables emphasize task outcomes. Appendix G collects the full-precision estimates and intervals.

A MODEL PROPERTIES

The request interface has two useful properties. A balanced request recovers the parent generator, and exchanging the two state labels leaves the output unchanged when their weights and features are exchanged together. We derive both properties and check the balanced request in the implementation. We also show how the analytical decoder controls the next-residue distribution.

A.1 Properties of the learned request interface

Consider an entity e observed in at least one state with positive request weight. Write $c = \pi_0 - \pi_1$, $z = \mathrm{SiLU}(W_m|c| + b_m)$, and

$$\Delta h_e = \mathbf{1}[m_{e,0}m_{e,1} = 1](h_{e,0} - h_{e,1}). \tag{A.1}$$

The learned correction is a function of z multiplied by $c\Delta h_e$, as in Equation (6).

Balanced requests. At $\pi = [0.5, 0.5]$, $c = 0$. Both scalar and vector corrections vanish for every entity, regardless of the learned parameters. The generator therefore uses the parent's pooled features at the balanced request. Zero-initialized output projections also make both corrections zero at initialization for every request.

Consistent state exchange. Exchange $(h_{e,0}, m_{e,0}, \pi_0)$ with $(h_{e,1}, m_{e,1}, \pi_1)$. The weighted pool is unchanged. Both c and Δh_e change sign, while $|c|$ and z remain unchanged. Thus $c\Delta h_e$ and the modulated features are unchanged. The downstream decoder receives the same inputs. With matched decoding order and random draws, this equality carries through the autoregressive sequence.

Partially observed structures. If an entity is observed in one state, the contrast mask sets $\Delta h_e = 0$. Whenever the pooling denominator is positive, availability normalization assigns the observed state all effective weight. The correction therefore uses state differences only where both states provide features.

A.2 Numerical agreement at the balanced request

The balanced request reproduces the parent's decisions in every evaluated context. All 2,025 contexts have identical complete rankings, Top-8 and Top-19 sets, and generator winners. All 12,150 frozen-response pools and winners also agree. Across 38,214 score rows, the largest absolute difference is 2.861×10^{-6}, within standard float32 tolerance. Thus, the scores agree to numerical precision and give exactly the same decisions.

A.3 Properties of the analytical decoder

Fix one decoded prefix and one next-residue position. Let $p_k(a)$ be the state-conditioned amino-acid probabilities and let $\rho_k = \pi_k^{1.5} / \sum_j \pi_j^{1.5}$. For availability mask m_k, define

$$Z = \sum_k m_k \rho_k > 0, \qquad q(a) = \frac{\sum_k m_k \rho_k p_k(a)}{Z}. \tag{A.2}$$

Since Z does not depend on a, it cancels in the tempered softmax:

$$\mathrm{softmax}_a\left(\frac{\log q(a)}{\tau_{\mathrm{dec}}}\right) = \mathrm{softmax}_a\left(\frac{\log \sum_k m_k \rho_k p_k(a)}{\tau_{\mathrm{dec}}}\right). \tag{A.3}$$

Availability normalization makes the effective state weights explicit while preserving the resulting distribution.

When both states are observed, write $u = \rho_0$ and $q(a) = up_0(a) + (1-u)p_1(a)$. The guided distribution is

$$P_\gamma(a) = \mathrm{softmax}_a\left[\frac{\log q(a) + \gamma(2u-1)(\log p_0(a) - \log p_1(a))}{\tau_{\mathrm{dec}}}\right]. \tag{A.4}$$

Setting $\gamma = 0$ recovers tempered arithmetic pooling. At $u = 1/2$, guidance vanishes at every amino acid. Exchanging the states and replacing u by $1 - u$ leaves the mixture and the signed contrast unchanged. At singly observed positions, the contrast is disabled and the decoder uses the available-state distribution.

Local monotonicity. For two amino acids a and b, the softmax normalizer cancels in their log odds:

$$\log \frac{P_\gamma(a)}{P_\gamma(b)} = \frac{1}{\tau_{\mathrm{dec}}} \left[\log \frac{q(a)}{q(b)} + \gamma(2u-1) \log \frac{p_0(a)/p_1(a)}{p_0(b)/p_1(b)} \right]. \tag{A.5}$$

Guidance shifts residue odds according to their relative scores under the two states.

Hold the prefix, state probabilities, request, and temperature fixed. Let $c = 2u - 1$ and $r(a) = \log p_0(a) - \log p_1(a)$. Differentiating the log partition function gives

$$\frac{\mathrm{d}}{\mathrm{d}\gamma} \mathbb{E}_{a \sim P_\gamma}[c\, r(a)] = \frac{1}{\tau_{\mathrm{dec}}} \operatorname{Var}_{a \sim P_\gamma}[c\, r(a)] = \frac{c^2}{\tau_{\mathrm{dec}}} \operatorname{Var}_{a \sim P_\gamma}[r(a)] \geq 0. \tag{A.6}$$

Increasing guidance therefore increases or preserves the expected signed local score contrast. Equality holds at the balanced request or when the state log-ratio is constant on the distribution's support. This result describes the analytical next-token distribution at a fixed prefix; sequence-level request following is evaluated empirically.

B Implementation and Evaluation Details

We specify the data splits, candidate budgets, and selection rules for each evaluation. Comparisons of response selectors use the same candidate pools. Comparisons of candidate sources use the same frozen response selector and candidate budget.

B.1 Generator data and training

We train the request-control module on Protein Ensemble Database records (Lazar et al., 2021). The starting split contains 4,691 development and 521 held-out records. Filtering identities shared with external inputs leaves 4,676 and 518 records. Removing development records that overlap the retained held-out set by identity, cluster, PDB membership, or native sequence gives the final 4,530/518 split.

We keep three independently trained DynamicMPNN parents frozen in evaluation mode. Each update uses one paired structure record at all three training anchors. Teacher and student predictions share the masked-logit interface, with sequence embeddings set to zero. The outer anchors use pure-state parent teachers; the midpoint uses the parent with the control module disabled. We retain the final checkpoint from each three-epoch run.

Table 3: Training configuration shared by state-contrast and scalar-FiLM conditioning.

Setting	Value
Parent seeds	17, 42, 73
Trainable parameters	11,284
State-0 training weights	$0.1, 0.5, 0.9$
Optimizer / learning rate	AdamW / 10^{-3}
Weight decay / gradient clipping	0 / norm 1
Training epochs	3
Contrast-loss coefficient	0.25
Node scalar / vector dimensions	128 / 16
Edge scalar / vector dimensions	32 / 4
Request hidden dimension	32

Node and edge modulation use separate affine maps. Vector scales are shared across Cartesian components. Including the request embedding, the parameter count is

$$2 \times 32 + (32 + 1)\,[2(128) + 16 + 2(32) + 4] = 11{,}284. \tag{B.1}$$

The scalar-FiLM control matches this count, the data, teachers, optimizer, training duration, parent checkpoints, and sampling budgets. It conditions on the request without multiplying by signed state-feature differences.

B.2 External generation and request-control metrics

The open–closed inputs are LplA (`3A7R-A`/`1X2G-A`), ADK (`4AKE-A`/`1AKE-A`), and MBP (`1OMP-A`/`1ANF-A`). For each of the four unseen requests, LplA uses five decoding orders and 12 samples; ADK and MBP each use six orders and eight samples. This gives 624 sequences per policy and checkpoint. The six-policy, three-checkpoint comparison contains $624 \times 6 \times 3 = 11{,}232$ sequences, including 1,872 from scalar-FiLM.

For evaluator e, preference is the difference between per-residue log likelihoods under the two structures, $\ell_{e,0}(s) - \ell_{e,1}(s)$. We regress preference on the requested state-0 weight. A positive slope indicates that sequences follow the requested direction. For standardized external gains, each system–evaluator pair uses one score standard deviation pooled over the learned generator and its seed-matched parent across checkpoints, requests, decoding orders, and samples. Raw scorer-scale comparisons keep each evaluator's original units.

The external slope intervals use 30,000 crossed resamples over systems, decoding orders, and samples, retaining matched evaluator views. System-specific endpoint comparisons use crossed checkpoint-seed and shared-decoding-order resampling, with complete sampled trajectories retained within an order. We report slopes and endpoint differences separately because they measure different aspects of request following.

B.3 Functional tasks and held-out splits

PYL1 concentration-dependent requests. The public PYL1–ABI1 map contains 19 substitutions at each of 169 positions and measurements at 12 ABA concentrations (Stammnitz & Lehner, 2026). A fixed wild-type Hill curve maps concentration to the generator's activation coordinate. The published normalized wild-type response at that concentration supplies the selection target. This coordinate is calibrated from the functional assay.

Three generator checkpoints form candidate pools. Three response seeds use ten mutation-position outer folds. The response predictor combines frozen absolute and mutation-delta sequence representations with the concentration-derived coordinate. It selects the candidate closest to the target response. Pools, predictions, and winners are fixed before held-out measurements are joined. Error is absolute deviation from the requested wild-type response. Paired intervals use 50,000 shared resamples of mutation positions.

PDZ partner-defined requests. The PDZ compendium spans five domains and seven domain–peptide interactions (Martí-Aranda & Lehner, 2026). Partner identity selects an author-released apo/partner-complex structure pair and the endpoint $\pi^* = [0, 1]$. The generator encodes the PDZ chain backbone. The response model receives interaction identity and predicts binding $\Delta\Delta G$, which is minimized for selection. The request is a partner-defined structural preference.

The inventory contains 675 interaction–position contexts. Of these, 628 have 19 substitutions, 37 have 18, six have 17, and one each has 15, 14, 6, and 3. Budget B is an upper cap: a context with n measured substitutions contributes $\min(B, n)$ candidates. Every context is retained with its observed inventory.

Three generator checkpoints and three response seeds produce 6,075 decisions. Ten outer folds hold out complete aligned or domain-local mutation-position groups, giving 129 evaluation units. Binding labels from the training and validation folds supervise the response model. Outer-test labels are absent from candidate formation, fitting, and checkpoint selection. Binding and folding outcomes are joined after winners are fixed. Paired intervals use 50,000 shared position-group resamples.

Tite-Seq libraries and held-out measurements. We use the public 4-4-20 anti-fluorescein Tite-Seq data (Adams et al., 2016). For each outer position $p \in \{0, \ldots, 19\}$, every sequence mutated at p is absent from training and validation. Position $(p + 10) \bmod 20$ supplies inner validation and is also absent from training. One assay replicate supplies target measurements, while the median of the other two supplies source labels. We rotate all three replicates as targets and train three response-model seeds.

Multi-mutant sequences are assigned without labels to one mutated outer position. Seventeen previously examined low-throughput clones are excluded from all splits and candidate pools. Every eligible outer-position sequence receives a prediction before target-replicate affinity availability or value is loaded.

For each of 15 concentration–occupancy requests, the response model selects the candidate whose predicted occupancy is closest to the target. Let q^* be that target and $[q_{\mathrm{lo}}, q_{\mathrm{hi}}]$ the interval implied by the target-replicate measurement. The selection error is

$$d(q^*, [q_{\mathrm{lo}}, q_{\mathrm{hi}}]) = \max\{q_{\mathrm{lo}} - q^*,\, 0,\, q^* - q_{\mathrm{hi}}\}. \tag{B.2}$$

This endpoint respects affinity censoring. Requests, replicates, and seeds are averaged within each position before 50,000 shared paired resamples of the 20 positions.

B.4 Tite-Seq response models and optimization

For Tite-Seq, the dual-view family fine-tunes ESM-2 at 8M, 35M, and 150M scales, with 6, 12, and 30 layers (Lin et al., 2023). The absolute view averages contextual variant embeddings over non-padding residues. The mutation-delta view averages variant-minus-wild-type embeddings over mutated residues. A 128-unit GELU–LayerNorm head combines both views and the mutation fraction. It produces a binder logit and a log-affinity estimate bounded to $[-10.5, -4.5]$.

The loss combines class-balanced binder cross-entropy, censored SmoothL1 affinity loss, and occupancy-interval loss on a fixed concentration grid. Inner-validation occupancy error selects the checkpoint and the weights of the three-scale ensemble. The absolute-only and mutation-delta controls retain their named input view and match the backbone scales, optimization, losses, checkpoint selection, and ensembling.

Table 4: Tite-Seq optimization settings for the learned ESM-2 response family.

Setting	Value
Optimizer	AdamW
Backbone / head learning rate	10^{-5} / 3×10^{-4}
Weight decay / gradient clipping	10^{-4} / norm 1
Head dropout	0.10
SmoothL1 width	0.25
Minimum / maximum epochs	20 / 120
Early-stopping patience	15 epochs
Minimum validation improvement	10^{-4}
Ensemble softmax temperature	0.01

The ProteinNPT comparison uses its official axial-attention core on frozen ESM-2-650M features (Notin et al., 2023). It uses the same censored-response loss, outer positions, prediction freeze, candidate libraries, requests, and evaluation rule. We identify it as an axial-core adaptation: it uses neither ProteinNPT's MSA context nor its MLM training recipe. This comparison evaluates the adapted response-encoder system under a common selection protocol.

Mutation-count and training-mean predictors use training data only. Exact random selection reports the expected error over the common candidate library. The within-pool oracle uses target labels to identify the best available candidate and provides a retrospective reference.

B.5 Selection outcomes and paired uncertainty

For a request $\mathcal{R}$, let $e_{\mathcal{R}}(s)$ be the measured target error of candidate s. The final target error and best-in-pool target error are

$$e_{\mathrm{selected}} = e_{\mathcal{R}}(\widehat{s}), \qquad e_{\mathrm{pool}} = \min_{s \in \mathcal{S}_8} e_{\mathcal{R}}(s). \tag{B.3}$$

The best-in-pool error is also known as pool-oracle error. Its difference from the final error is the selector gap, or selection regret:

$$e_{\mathrm{regret}} = e_{\mathrm{selected}} - e_{\mathrm{pool}}. \tag{B.4}$$

We compute the best-in-pool error from held-out measurements after selection. It describes the closest match available among the candidates, while the selector gap measures how far the final choice is from that match.

In the original functional evaluations, candidate construction, model fitting, validation choices, and selected sequences are fixed before held-out outcomes are joined. Repeated requests, model seeds, and assay replicates are combined within the stated evaluation unit before paired resampling. In the PDZ position test, 6,075 seed–context decisions are nested in 129 position groups.

Table 5: Evaluation units and interval construction. The separate five-domain PDZ result is given in Appendix D.2.

Evaluation	Paired unit	Interval construction
PED unseen requests	518 records	Record / hierarchical aggregate
MotifDiv learned transfer	169 structure pairs	30,000 crossed seed–pair resamples
PYL1 functional selection	169 mutation positions	50,000 position resamples
PDZ position-held-out selection	129 position groups	50,000 group resamples
Tite-Seq response selection	20 mutation positions	50,000 position resamples

Results tables show task means or paired gains. Bold marks the best mean within a comparison; † marks a paired difference whose reported 95% interval excludes zero. Unmarked entries may have uncertain differences or no reported interval. Appendix G gives the full estimates. Paired differences can differ from subtraction of displayed means: some means use all available cases, while paired comparisons use shared cases. Captions define the comparison and its direction. PDZ binding $\Delta\Delta G$ measures the selected variant's effect on binding; PYL1 and Tite-Seq errors measure deviation from a request.

B.6 Internal validation and additional selection tests

An additional matched tuning protocol splits the 4,530 development records into 3,624 training and 906 validation records. It evaluates 18 candidate recipes with the same tuning budget for learned and shared-gain controls. The selected recipes produce six full-data retraining units. Outer-test and external labels do not enter recipe selection. This protocol supplies the tuned comparisons in Appendix C.4.

The fixed-concentration tests vary the target request and compare its pool with midpoint and reversed requests. The pool-quality test instead compares the correct-request pool with an exact uniform-random pool, keeping the response selector frozen and the budget at Top-8. These tests separate the effect of the requested target from the value of candidate formation under a fixed budget.

The response-composition extension uses nested validation in a six-seed evaluation. It is a post-hoc exploratory extension of the original PYL1 evaluation. All methods in this extension are compared on the same cases; its means are reported separately from the original PYL1 results.

C Request Control and Transfer

We examine how the generator responds to a change in the requested structural preference. Matched controls test the conditioning mechanism, while new request coordinates and motif-design contexts test its transfer. We also assess sequence diversity, likelihood, and predicted structure.

C.1 Matched controls across systems and evaluators

State contrasts make the generator more responsive to the request than all four controls in Table 6. The standardized slope gain is 0.49056 over scalar-FiLM at equal capacity and 0.48444 over the frozen parent. Gains over probability arithmetic and local-product pooling are 0.53864 and 0.54423. These comparisons show that using state differences improves request following beyond conditioning on the request alone or directly combining state predictions.

Table 6: Standardized request-slope gains on LplA, ADK, and MBP. Positive values favor state-contrast conditioning over the named control.

Control	Slope gain
Same-parameter scalar-FiLM	$+0.491^{\dagger}$
Seed-matched parent	$+0.484^{\dagger}$
Endpoint probability arithmetic	$+0.539^{\dagger}$
Endpoint local product	$+0.544^{\dagger}$

The scalar-FiLM comparison gives positive standardized gains for all three parent seeds: 0.5280, 0.4685, and 0.4752 for seeds 17, 42, and 73. Figure 5 shows that the gain extends across all nine system–evaluator pairs. Table 7 gives the scorer-scale values and the separate endpoint-direction comparison.

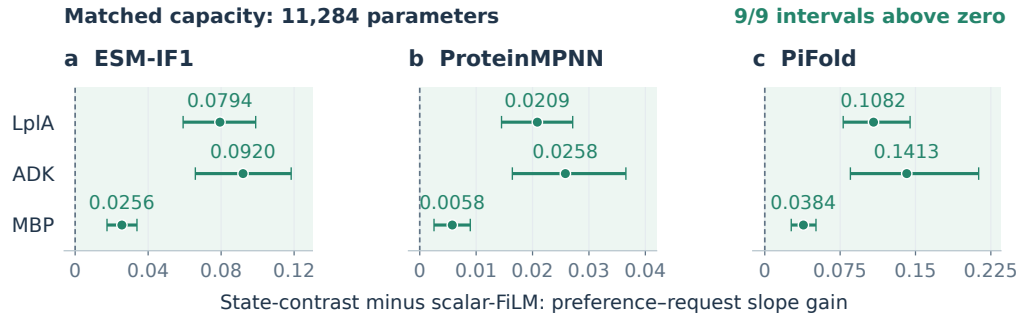

Figure 5: Request control across three systems and three evaluators. Points show slope gains over parameter-matched scalar-FiLM; bars show 95% intervals. State contrasts improve control in all nine comparisons at request coordinates absent from training. Each evaluator uses its own score scale.

Table 7: Request control in each system and evaluator. Entries are raw score gains. (a) Preference–request slope over scalar-FiLM. (b) Endpoint direction over the parent. Each evaluator uses its own score scale.

System	ESM-IF1	ProteinMPNN	PiFold
(a) Slope gain over scalar-FiLM			
LplA	$+0.079^{\dagger}$	$+0.021^{\dagger}$	$+0.108^{\dagger}$
ADK	$+0.092^{\dagger}$	$+0.026^{\dagger}$	$+0.141^{\dagger}$
MBP	$+0.026^{\dagger}$	$+0.006^{\dagger}$	$+0.038^{\dagger}$
(b) Endpoint-direction gain over the parent			
LplA	$+0.020^{\dagger}$	$+0.008^{\dagger}$	$+0.082^{\dagger}$
ADK	$+0.021^{\dagger}$	$+0.006^{\dagger}$	$+0.114^{\dagger}$
MBP	$+0.009^{\dagger}$	$+0.002^{\dagger}$	$+0.028^{\dagger}$

The generated sequences also tend to follow successive changes in the request. Across 468 matched trajectories, the fractions of adjacent requests with increasing state-0 preference are 0.8611, 0.8148, and 0.8682 under ESM-IF1, ProteinMPNN, and PiFold. The corresponding arithmetic-control fractions are 0.8511, 0.7828, and 0.8618. These fractions describe the sampled trajectories; Figure 5 gives the paired slope comparisons.

C.2 Seed stability and unseen requests

The control effect holds across all three parent seeds on 518 held-out PED records. Directional gains are positive at the training anchors, and slope gains remain positive at four intermediate requests that

were not used for training (Table 8). Midpoint agreement is 1.000 for every seed, consistent with the balanced-request property in Appendix A.2.

Table 8: Request control on 518 held-out PED records. Entries are learned-generator minus parent changes. Higher direction and slope indicate stronger control; higher NLL indicates lower likelihood. Unseen weights are $\{0.2, 0.35, 0.65, 0.8\}$.

Parent seed	Anchor direction	Anchor ΔNLL	Unseen slope
17	$+0.019^{\dagger}$	$+0.012^{\dagger}$	$+0.029^{\dagger}$
42	$+0.021^{\dagger}$	$+0.019^{\dagger}$	$+0.027^{\dagger}$
73	$+0.017^{\dagger}$	$+0.014^{\dagger}$	$+0.034^{\dagger}$
Hierarchical aggregate	—	—	$+0.030^{\dagger}$

The accompanying unseen-request target-NLL change is $+0.00616$.

The unseen-request comparison covers 49,728 sequences and gives an aggregate slope gain of 0.03014. The effect therefore extends between the training anchors across the held-out record set. Table 8 also gives the target-NLL changes, which measure sequence likelihood separately from request sensitivity.

C.3 Transfer to motif design

We test the learned generator on MotifDiv (Stern et al., 2025). A rule fixed before scoring retains 169 of 200 structure pairs. Both target directions must have canonical non-motif context under the target-residue-then-alternate-residue rule, and the official CSDesign non-motif residues must match that context. The same exclusions apply to every method. Each checkpoint-linked method designs the same ten motif positions in both directions with three parent checkpoints. The four learned/control methods produce $169 \times 2 \times 3 \times 4 = 4{,}056$ greedy candidates. CSDesign contributes 338 fixed official candidates, without duplication across parent checkpoints.

ESM-IF1 and PiFold score the same ten positions. Target log likelihood measures compatibility with the requested structure. Direction is target-minus-alternate log likelihood. We average the two directions within each pair and checkpoint and use 30,000 crossed checkpoint-seed and pair resamples for the paired comparisons.

Table 9: MotifDiv transfer on 169 structure pairs. Values are learned generator minus comparator; higher values are favorable for both metrics. Matched controls isolate conditioning choices; CSDesign is an external comparison.

Comparator	Δ target log likelihood	Δ direction margin
ESM-IF1 evaluation		
Scalar-FiLM	$-0.047^{\dagger}$	$+0.113^{\dagger}$
Seed-matched parent	$-0.046^{\dagger}$	$+0.114^{\dagger}$
Probability arithmetic	-0.018	$+0.048^{\dagger}$
CSDesign	-0.088	$-1.303^{\dagger}$
PiFold evaluation		
Scalar-FiLM	$-0.087^{\dagger}$	$+0.113^{\dagger}$
Seed-matched parent	$-0.087^{\dagger}$	$+0.113^{\dagger}$
Probability arithmetic	$-0.040^{\dagger}$	$+0.063^{\dagger}$
CSDesign	$+0.609^{\dagger}$	$-1.261^{\dagger}$

State contrasts increase the target-minus-alternate margin over all three matched controls under both evaluators. Against scalar-FiLM, the gains are 0.11285 under ESM-IF1 and 0.11342 under PiFold. Target likelihood and direction margin need to be read together: stronger separation from the alternate state need not mean a higher likelihood under the target state. CSDesign gives larger direction margins, while our generator has higher target likelihood under PiFold. The transfer gain concerns request direction relative to matched conditioning controls.

C.4 Contrast strength and complementary sequence assessments

We compare learned modulation with a fixed gain $g = 1/8$ and a trained shared gain. These controls separate the use of state differences from the choice of gain parameterization. Table 10 reports structural control and functional selection on their respective scales.

Table 10: Learned modulation versus fixed and shared gains. Positive gains favor learned modulation. Structural gains are changes in standardized request slope; functional gains are reductions in selected error.

Test	Control	Gain
Structural slope	Fixed $g = 1/8$	$+0.021$
Structural slope	Trained shared gain	$+0.009$
q50 selected error	Fixed $g = 1/8$	$+0.00099^{\dagger}$
All-target selected error	Fixed $g = 1/8$	$+0.00034$
All-target selected error	Trained shared gain	$+0.00038$
Tuned Top-8 selection	Fixed $g = 1/8$	$\approx +0.00018$

Learning the gain gives a small improvement in q50 selection over both fixed and shared gains. The broader all-target, structural-slope, and tuned comparisons do not establish a consistent advantage for learned modulation. The clearest effect is therefore the use of state contrasts; the benefit of learning their strength depends on the task.

PDZ selection changes little across the three gain choices. Mean selected binding $\Delta\Delta G$ is 0.14663 for learned modulation, 0.14559 for fixed gain, and 0.14546 for shared gain. Paired comparisons with learned modulation do not resolve differences in the selected outcome, best-in-pool outcome, or selector gap. At Top-19, all three give identical results. The tuned Top-19 comparison also agrees across all five methods.

Candidate diversity and likelihood. Learned, fixed-gain, and shared-gain generation each attain a unique-sequence rate of 100%, with Hamming diversity approximately 0.741. All three preserve broad variation in the sampled pools, and the paired target-NLL comparisons show no clear separation. In the external scalar-FiLM comparison, the companion balanced-state ProteinMPNN NLL change is $+0.00515$, and native recovery changes by -0.00077. These measures describe sequence quality alongside the request-control result.

Independent structural assessment. We use frozen Boltz-2 structure prediction (Passaro et al., 2025) to assess structural direction, motif self-consistency RMSD, and scaffold RMSD. Controls are endpoint probability arithmetic, same-parameter scalar-FiLM, and the parent with control disabled. None of the nine paired comparisons establishes an improvement on these structure-prediction measures. The demonstrated control gain concerns inverse-folding preferences; it does not by itself imply better predicted structures. Appendix D evaluates functional selection against assay measurements.

D Candidate Formation and Functional Selection

Functional design depends on both the available candidates and the final choice. We first evaluate response selection within fixed pools. We then hold the response selector fixed and vary the request used to form the pool. Appendix B gives the task protocols and evaluation units.

D.1 Functional selection on PYL1

Response-aware selection finds variants closer to the requested PYL1 response at both candidate budgets (Table 11). With eight candidates, mean error falls from 0.29832 under structural ranking to 0.24138, a 19.1% reduction. The selected outcome improves at 119 mutation positions. At Top-19, error is 0.24646 compared with 0.29832 for structural ranking. Thus, choosing for the functional target improves on structural ranking in both the restricted pool and the full substitution inventory.

Table 11: PYL1 target-response error. Structural ranking and response controls use the same candidate pool unless a different pool is named. Markers compare each row with our pool plus dual-view response.

Selection rule	Top-8 error ↓	Top-19 error ↓
Our pool + dual-view response	0.241	0.246
Structural ranking	$0.298^{\dagger}$	$0.298^{\dagger}$
Midpoint pool + same response	0.242	0.246
Same pool + one-hot condition	$\mathbf{0.233}$	$\mathbf{0.228}^{\dagger}$
Same pool + frozen-ESM condition	0.237	$0.238^{\dagger}$

The remaining controls separate candidate formation from response prediction. At Top-8, comparisons with the midpoint pool and direct-condition predictors do not establish a clear ordering. At Top-19, the one-hot and frozen-ESM predictors achieve lower errors than dual-view. The advantage over structural ranking is therefore distinct from the choice of response architecture; Appendix E examines that choice further.

Stability across active-like structures. Replacing the active-like PYL1 input `3KDJ` with `3JRS` gives mean Top-8 pool Jaccard overlap 0.90687 and median overlap 1.0. The top-ranked substitution is unchanged in 86.5% of position–condition decisions. Candidate prioritization is therefore stable across these two structural inputs.

D.2 Partner-specific selection and domain transfer on PDZ

On held-out PDZ position groups, response-aware selection lowers mean binding $\Delta\Delta G$ from 0.21826 to 0.14663 kcal mol^{-1}. The selected mutations are less disruptive to binding than those chosen by structural ranking. Binding labels supervise the selector; folding outcomes also improve when evaluated after selection (Table 12(a)).

Table 12: PDZ outcomes of the selected variants. Positive gains favor response selection. Energy gains are structural minus response-selected energy; hit-rate gains are response minus structural rate, expressed in percentage points. Panel (b) is a separate aggregate evaluation.

(a) Paired Top-8 improvements		
Outcome	Unit	Gain
Binding $\Delta\Delta G$	kcal mol^{-1}	$+0.072^{\dagger}$
Folding $\Delta\Delta G$	kcal mol^{-1}	$+0.102^{\dagger}$
Binding-hit rate	percentage points	$+4.181^{\dagger}$
Folding-hit rate	percentage points	$+4.801^{\dagger}$
Joint-hit rate	percentage points	$+3.861^{\dagger}$
(b) Absolute hit-rate evaluation		
Endpoint	Structural ranking	Response selection
Binding hit	40.1%	**44.3%**
Joint binding-and-folding hit	35.9%	**39.8%**

The hit-rate evaluation gives a complementary view. Binding hits increase from 40.1% to 44.3%, and joint binding-and-folding hits increase from 35.9% to 39.8%. These are gains of about 4.2 and 3.9 percentage points. A joint hit satisfies both endpoint criteria, so this measure rewards selected variants that meet both requirements.

Response-aware selection also improves binding and folding over the one-hot condition model (Table 13). Comparisons with frozen-ESM and midpoint or parent pools do not resolve a further binding gain. The strongest PDZ result concerns the final choice for the requested partner within a fixed candidate pool.

Table 13: Matched PDZ Top-8 comparisons. Gains are comparator minus our selected $\Delta\Delta G$, in kcal mol^{-1}. Direct-condition controls use the same pool; midpoint and parent controls change the pool. Dashes indicate unavailable estimates.

Comparator	Binding gain	Folding gain
Structural ranking	$+0.072^{\dagger}$	$+0.102^{\dagger}$
One-hot condition	$+0.054^{\dagger}$	$+0.111^{\dagger}$
Frozen-ESM condition	-0.001	—
Midpoint or parent pool	-0.004	—

Generalization across domains. A separate leave-one-domain-out test evaluates all five PDZ domains. Mean selected binding $\Delta\Delta G$ decreases from 0.27735 to 0.17490 kcal mol^{-1}, a 36.94% reduction. Each held-out domain has a favorable point estimate (Table 14). The benefit therefore extends from mutations at unseen positions to selection in unseen domains.

Table 14: PDZ selection with each domain held out in turn. Binding energies are in kcal mol^{-1}. Response selection reduces the mean by 36.94%, with favorable point estimates in all five domains.

Selection rule	Binding $\Delta\Delta G \downarrow$
Structural ranking	0.277
Response selection	$\mathbf{0.175}^{\dagger}$

D.3 Correct requests at a fixed concentration

Changing the request can improve the final choice even when concentration stays fixed. We compare correct, midpoint, and reversed requests in the q20, q50, and q80 target groups. At q80, the correct request reduces selected error by 0.00456 relative to the midpoint request and by 0.00643 relative to the reversed request (Table 15(a)). Since the selector is shared, this improvement comes from changing which candidates enter the pool. The q20 and q50 comparisons do not resolve a consistent request effect.

D.4 Candidate availability under a fixed budget

Request-guided pools also provide better candidates than exact uniform-random pools under the same eight-candidate budget. With the response selector frozen, they reduce final target error by 0.0273589 and best-in-pool target error by 0.0306078 across targets (Table 15(b)). The latter is the lowest measured error among the eight retained variants. Its reduction shows that candidate formation makes closer-to-target variants available before the selector makes its choice.

Table 15: Request and candidate-pool controls at Top-8. Reductions are control error minus correct-request error. Panel (a) changes the request at fixed concentration. Panel (b) changes the pool under the same frozen selector. Best-in-pool error is measured after selection.

Target / outcome	Control	Error reduction
(a) Correct versus altered requests		
q80 / selected error	Midpoint request	$+0.0046^{\dagger}$
q80 / selected error	Reversed request	$+0.0064^{\dagger}$
(b) Request-guided versus random pools		
All / selected error	Exact uniform random	$+0.0274^{\dagger}$
All / best-in-pool error	Exact uniform random	$+0.0306^{\dagger}$
q80 / selected error	Exact uniform random	$\approx +0.0496^{\dagger}$

The selected-error gain is present at q50 and q80, reaching approximately 0.04960 at q80; the q20 result is uncertain. The selector gap shows no clear change. The direct evidence is that better variants enter the pool and the final target error falls. Together with the fixed-concentration comparison, this links the functional request to candidate quality under a limited budget.

D.5 Full-inventory controls

At Top-19, each PYL1 position retains all non-wild-type substitutions. Correct, midpoint, and reversed requests then give the same candidate sets and winners under the common pure-response selector. Once every variant is available, request-dependent prioritization no longer changes that selector's options. This equality requires shared candidate inventories, response scores, and selection rules. Appendix F examines rules that also use generator ordering in the final choice.

E Response Models and Validation-Based Selection

The response model determines which candidate best matches the functional target. We compare sequence representations on fixed Tite-Seq libraries, repeat PYL1 selection with a second architecture, and use validation data to choose a response composition. Appendix B gives the training settings and data splits.

E.1 Response selection on fixed Tite-Seq libraries

Dual-view ESM-2 makes better choices than the matched-loss ProteinNPT axial-core adaptation from the same Tite-Seq libraries. Mean occupancy-interval error is 0.32287 versus 0.35125, with better outcomes at 15 of the 20 held-out mutation positions (Table 16). Because the candidates and requests are shared, this comparison measures the effect of the response representation on the final choice.

Table 16: Tite-Seq occupancy-interval error on fixed candidate libraries. Markers compare each model with dual-view ESM-2. Bold identifies the lowest predictive-model error. The oracle uses target labels and is a retrospective reference.

Response model	Error ↓
Dual-view ESM-2	**0.323**
ProteinNPT axial-core adaptation	0.351†
Absolute-only ESM-2	0.333
Mutation-delta ESM-2	0.340†
Mutation-count regression	0.348†
Training mean	0.404†
Exact random selection	0.403†
Within-pool oracle	0.057

Combining the two sequence views also improves on mutation-delta ESM-2, whose error is 0.33988. The absolute view therefore adds useful information beyond the change from wild type. Absolute-only ESM-2 reaches 0.33262; its smaller difference from dual-view is not resolved by the paired comparison. The table includes simpler predictors and random selection to show how these models perform on the same selection task.

Frozen response ensembles. In a separate evaluation, the prespecified validation-weighted ensemble reaches error 0.3218592, compared with 0.3389165 for mutation-delta (Table 17). The paired comparison supports this improvement. Equal-weight combinations give slightly lower errors than the primary ensemble, while the remaining comparisons do not identify a clear ordering. This experiment supports the benefit over mutation-delta without identifying a uniquely best ensemble.

Table 17: Tite-Seq frozen response ensembles, evaluated separately from Table 16. Markers compare each method with the validation-weighted primary. Four decimals distinguish the closely spaced means.

Response model	Error ↓
Validation-weighted primary	0.3219
Absolute view	0.3276
Mutation-delta	0.3389†
Dual-view	0.3225
ProteinNPT axial-core	0.3394
Dual + axial, equal weights	0.3198
Four families, equal weights	**0.3193**

E.2 Response architectures on PYL1

The PYL1 selection benefit also appears with a ProteinNPT axial-core response model. Error falls from 0.29828 under structural ranking to 0.23463 at Top-8, a 21.34% reduction. At Top-19, it falls from 0.29817 to 0.22595, a 24.22% reduction (Table 18). Response-aware selection therefore improves target matching with both the dual-view and axial-core architectures.

Table 18: PYL1 target-response error with a second response architecture. Reductions use unrounded means from the same evaluation and candidate budget.

B	Structural ranking	Axial-core response	Reduction
8	0.298	**0.235**	21.34%
19	0.298	**0.226**	24.22%

Full ProteinNPT comparison. We also evaluate full ProteinNPT on PYL1, separately from the axial-core adaptation above. Its mean errors are lower than those of the original dual-view model at both budgets, but the paired comparisons do not establish a clear difference (Table 19).

Table 19: PYL1 target-response error with original dual-view and full ProteinNPT. The paired comparisons do not establish a clear difference at either budget.

B	Dual-view error ↓	Full ProteinNPT error ↓
8	0.241	**0.237**
19	0.246	**0.235**

E.3 Validation-selected response composition

Validation-based response selection improves PYL1 target matching in a six-seed, post-hoc exploratory study. All methods use the same evaluation cases. Top-8 error is 0.218030, below dual-view, one-hot, and full ProteinNPT. Paired comparisons support all three gains (Table 20). At Top-19, error is 0.217016. Gains over dual-view and full ProteinNPT remain, while the comparison with one-hot is less conclusive.

Across ten folds, validation chooses the architecture mean seven times, one-hot twice, and ESM once; dual-view alone is never chosen. The weighting parameter is $\lambda = 1.0$ in seven folds and $\lambda = 0.75$ in three; $\lambda = 0.5$ is never chosen.

Table 20: PYL1 response models on shared evaluation cases. Markers compare each baseline with the validation-selected composition. This six-seed, post-hoc exploratory evaluation is separate from the original PYL1 comparison.

Response model	Top-8 error ↓	Top-19 error ↓
Validation-selected composition	**0.218**	**0.217**
Original dual-view	0.241†	0.246†
One-hot	0.232†	0.228
Full ProteinNPT	0.237†	0.235†

F Candidate Sources and Selection Rules

The candidate source determines which variants reach the response selector. We compare sources at the same budget, then test whether generator rankings add useful information to the final choice. These experiments distinguish control of a structural request from the functional quality of the resulting candidates.

F.1 Candidate sources under a fixed response selector

ProteinMPNN endpoint pools. The external control uses ProteinMPNN `v_48_020` (Dauparas et al., 2022) with available N, CA, C, and O atoms from both PYL1 structures. Our endpoint combination applies the same wild-type coordinate to its probabilities and ranks the same substitution inventory. ProteinMPNN uses three native-context decoding-order seeds from one checkpoint; our generator uses three trained parent checkpoints.

Table 21: PYL1 candidate sources with the same frozen dual-view selector. Markers compare ProteinMPNN with our pool. Both sources give identical results when all 19 substitutions are available.

Candidate source	Top-8 error ↓	Top-19 error ↓
Our pool	0.241	**0.246**
ProteinMPNN endpoint	**0.228**†	**0.246**

At Top-8, the ProteinMPNN endpoint pool gives error 0.22761 compared with 0.24138 for our pool (Table 21). Their Jaccard overlap is 0.50950, so the selector receives different sets of candidates. At Top-19, both sources include all substitutions and the common response selector returns the same result. The difference at the smaller budget therefore comes from which variants are retained.

Additional conformational candidate sources. We also compare two Conformational Biasing (CB) formulations and Caliby (Shuai et al., 2025). Our pool reduces Top-8 selected error by approximately 0.0321 relative to official endpoint-CB (Table 22). The linear-state-mix formulation gives lower error than our pool. Caliby also has a lower point estimate on the 169 positions with complete backbone support in both conformations. The endpoint-CB result therefore depends on how the state information is combined.

Table 22: Additional Top-8 candidate sources. Approximate reductions are comparator error minus our error. Positive values favor our source. CB formulations share one evaluation; Caliby uses common complete-backbone positions.

Candidate source	Error reduction
CB: official endpoint	$\approx +0.032$†
CB: linear state mix	≈ -0.018
Caliby: complete-backbone positions	≈ -0.012

At Top-19, these sources return identical results under the common pure-response rule. Candidate prioritization therefore matters when the budget restricts the available variants.

F.2 Combining generator and response rankings

Within the same ProteinMPNN pools, we blend normalized generator rank r_G and response-closeness rank r_R:

$$r_\lambda = (1-\lambda)r_G + \lambda r_R, \qquad \lambda \in \{0, 0.25, 0.5, 0.75, 1\}. \tag{F.1}$$

Inner-validation positions select one weight per outer fold and budget using all generator and response seeds. Pools and weights are fixed before outer-test outcomes are joined.

Table 23: Selection within frozen ProteinMPNN endpoint pools. Markers compare each rule with generator ranking. The validation-selected blend has the lowest mean error at both budgets.

Selection rule	Top-8 error ↓	Top-19 error ↓
Generator ranking	0.224	0.224
Pure response selection	0.228	$0.246^{\dagger}$
Validation-selected blend	$\mathbf{0.213}^{\dagger}$	$\mathbf{0.210}^{\dagger}$

The validation-selected blend lowers error from 0.22409 under generator ranking to 0.21293 at Top-8 and 0.21008 at Top-19 (Table 23). Pure response selection shows no clear advantage over generator ranking at Top-8 and gives higher error at Top-19. Generator ordering therefore contains useful information that can complement the response predictions within these ProteinMPNN pools.

F.3 Matched validation and seed aggregation

We next apply the same nested blend grid, inner-validation cases, seeds, and prediction-freezing procedure to both candidate sources. A second comparison averages normalized full-candidate ranks across three generator seeds, breaking ties by candidate identifier. The blend strength is selected on shared inner-validation cases. Positions without the required common support are omitted from both methods, without imputing response values.

Table 24: PYL1 candidate sources under matched validation and seed aggregation. Both sources use the same selection rule. Markers indicate paired differences between the two sources.

Protocol	B	Our error ↓	ProteinMPNN error ↓
Nested blend	8	0.240	$\mathbf{0.213}^{\dagger}$
Nested blend	19	0.233	$\mathbf{0.210}^{\dagger}$
All-seed ranks	8	0.241	$\mathbf{0.208}^{\dagger}$
All-seed ranks	19	0.231	$\mathbf{0.208}^{\dagger}$

ProteinMPNN pools give lower target error under both matched protocols (Table 24). The all-seed comparison covers 169 paired positions, with 6,021 common cases at Top-8 and 6,052 at Top-19. At Top-19, blended selection can still differ between sources because it uses generator ordering even when the candidate sets are identical. Pure response selection uses no such ordering and gives the identical results in Table 21.

F.4 Scope of the candidate-source results

The model improves request following over matched controls and functional selection over structural ranking. It does not have the lowest functional error for every candidate source. On PYL1, both candidate prioritization and the final ranking rule affect the outcome. These comparisons specify the model's performance at a given budget and functional request.

G Detailed Statistical Estimates

Tables in Appendices C–F emphasize task outcomes and use rounded values for readability. Table 25 collects the full-precision paired estimates and available marginal intervals. Signs and units follow the named comparison. Appendix B gives the resampling procedures.

Table 25: Full-precision estimates and 95% intervals. Table references identify the corresponding results. Hit-rate differences use rate units here; Table 12 displays them in percentage points.

Statistic / comparison	Estimate	95% interval
Standardized request-slope gains (Table 6). Learned generator minus the named control.		
Same-parameter scalar-FiLM	0.49056	[0.37729, 0.61740]
Seed-matched parent	0.48444	[0.37053, 0.61625]
Endpoint probability arithmetic	0.53864	[0.37040, 0.72015]
Endpoint local product	0.54423	[0.37013, 0.74249]
External request control (Table 7). Raw score gains: slopes over scalar-FiLM and endpoint direction over the parent.		
Slope: LplA / ESM-IF1	0.07936	[0.05917, 0.09894]
Slope: LplA / ProteinMPNN	0.02085	[0.01450, 0.02712]
Slope: LplA / PiFold	0.10823	[0.07805, 0.14462]
Slope: ADK / ESM-IF1	0.09200	[0.06587, 0.11839]
Slope: ADK / ProteinMPNN	0.02582	[0.01642, 0.03655]
Slope: ADK / PiFold	0.14129	[0.08521, 0.21288]
Slope: MBP / ESM-IF1	0.02561	[0.01746, 0.03388]
Slope: MBP / ProteinMPNN	0.00577	[0.00253, 0.00896]
Slope: MBP / PiFold	0.03844	[0.02634, 0.05096]
Endpoint: LplA / ESM-IF1	0.01960	[0.01546, 0.02306]
Endpoint: LplA / ProteinMPNN	0.00819	[0.00625, 0.01012]
Endpoint: LplA / PiFold	0.08205	[0.06972, 0.09812]
Endpoint: ADK / ESM-IF1	0.02082	[0.01571, 0.02657]
Endpoint: ADK / ProteinMPNN	0.00632	[0.00376, 0.00876]
Endpoint: ADK / PiFold	0.11447	[0.07082, 0.17512]
Endpoint: MBP / ESM-IF1	0.00913	[0.00638, 0.01228]
Endpoint: MBP / ProteinMPNN	0.00237	[0.00135, 0.00354]
Endpoint: MBP / PiFold	0.02752	[0.01937, 0.03576]
Held-out PED requests (Table 8). Learned generator minus parent; NLL and direction are separate outcomes.		
Anchor direction / seed 17	0.01930	[0.01443, 0.02364]
Anchor target NLL / seed 17	0.01167	[0.00777, 0.01553]
Anchor direction / seed 42	0.02078	[0.01660, 0.02632]
Anchor target NLL / seed 42	0.01931	[0.01414, 0.02615]
Anchor direction / seed 73	0.01651	[0.01275, 0.02079]
Anchor target NLL / seed 73	0.01400	[0.01083, 0.01794]
Unseen slope / 17	0.02917	[0.02329, 0.03654]
Unseen slope / 42	0.02686	[0.02269, 0.03250]
Unseen slope / 73	0.03438	[0.02942, 0.04258]
Unseen slope / Hierarchical aggregate	0.03014	[0.02453, 0.03713]
Unseen requests / target NLL	+0.00616	[0.00430, 0.00782]
MotifDiv transfer (Table 9). Learned generator minus comparator; higher values favor the learned generator.		
ESM-IF1 / Scalar-FiLM / target LL	−0.04747	[−0.07751, −0.01786]
ESM-IF1 / Scalar-FiLM / direction	0.11285	[0.07650, 0.15224]
ESM-IF1 / Seed-matched parent / target LL	−0.04552	[−0.07586, −0.01556]
ESM-IF1 / Seed-matched parent / direction	0.11412	[0.07658, 0.15532]
ESM-IF1 / Probability arithmetic / target LL	−0.01837	[−0.03792, 0.00120]
ESM-IF1 / Probability arithmetic / direction	0.04771	[0.00908, 0.09131]
ESM-IF1 / CSDesign / target LL	−0.08798	[−0.22651, 0.05394]
ESM-IF1 / CSDesign / direction	−1.30319	[−1.43297, −1.17631]
PiFold / Scalar-FiLM / target LL	−0.08727	[−0.12564, −0.05078]

Continued on the next page

Table 25 (continued)

Statistic / comparison	Estimate	95% interval
PiFold / Scalar-FiLM / direction	0.11342	[0.06996, 0.16586]
PiFold / Seed-matched parent / target LL	−0.08683	[−0.12516, −0.05098]
PiFold / Seed-matched parent / direction	0.11250	[0.06798, 0.16613]
PiFold / Probability arithmetic / target LL	−0.03985	[−0.06771, −0.01607]
PiFold / Probability arithmetic / direction	0.06304	[0.01713, 0.13558]
PiFold / CSDesign / target LL	0.60882	[0.33639, 0.90160]
PiFold / CSDesign / direction	−1.26096	[−1.48167, −1.03442]
Gain parameterizations (Table 10). Positive values favor learned modulation; metrics retain their original units.		
Structural slope / Fixed $g = 1/8$	0.021179	[−0.024640, 0.067416]
Structural slope / Trained shared gain	0.009053	[−0.030458, 0.047818]
q50 selected error / Fixed $g = 1/8$	0.000985	[0.000108, 0.002291]
All-target selected error / Fixed $g = 1/8$	0.000335	[−0.000066, 0.000825]
All-target selected error / Trained shared gain	0.000380	[−0.000001, 0.000861]
Tuned Top-8 selection / Fixed $g = 1/8$	$\approx$ 0.00018	[−0.00023, 0.00066]
PYL1 functional selection (Table 11). Our pool with dual-view response minus comparator error.		
Top-8 / Structural ranking	−0.05708	[−0.07463, −0.03997]
Top-8 / Midpoint pool + same response	−0.00066	[−0.00375, 0.00208]
Top-8 / Same pool + one-hot condition	0.00847	[−0.00409, 0.02129]
Top-8 / Same pool + frozen-ESM condition	0.00444	[−0.00168, 0.01056]
Top-19 / Structural ranking	−0.05201	[−0.07167, −0.03271]
Top-19 / Same pool + one-hot condition	0.01830	[0.00160, 0.03513]
Top-19 / Same pool + frozen-ESM condition	0.00850	[0.00205, 0.01503]
PDZ selected-variant outcomes (Table 12). Energy: structural minus response selection; hit rate: response minus structural.		
Binding $\Delta\Delta G$	0.07162	[0.04434, 0.10091]
Folding $\Delta\Delta G$	0.10241	[0.05859, 0.14738]
Binding-hit rate	0.04181	[0.00954, 0.07609]
Folding-hit rate	0.04801	[0.01500, 0.08095]
Joint-hit rate	0.03861	[0.00308, 0.07550]
Matched PDZ controls (Table 13). Comparator minus our selected energy, in kcal mol^{-1}.		
Structural ranking / binding	0.07162	[0.04434, 0.10091]
Structural ranking / folding	0.10241	[0.05859, 0.14738]
One-hot condition / binding	0.05368	[0.02368, 0.08753]
One-hot condition / folding	0.11131	[0.04365, 0.18488]
Frozen-ESM condition / binding	−0.00144	[−0.00703, 0.00422]
Midpoint or parent pool / binding	−0.00420	[−0.01016, 0.00160]
Held-out PDZ domains (Table 14). Structural minus response-selected binding energy, in kcal mol^{-1}.		
Paired gain [95% CI]	0.10244	[0.05891, 0.14427]
Request and pool controls (Table 15). Control error minus correct-request error at Top-8.		
q80 / selected error / Midpoint request	0.00456	[0.00120, 0.00851]
q80 / selected error / Reversed request	0.00643	[0.00274, 0.01076]
All / selected error / Exact uniform random	0.0273589	[0.0159259, 0.0385650]
All / best-in-pool error / Exact uniform random	0.0306078	[0.0199222, 0.0412863]
q80 / selected error / Exact uniform random	$\approx$ 0.04960	[0.03323, 0.06600]
Tite-Seq response selection (Table 16). Comparator minus dual-view occupancy-interval error.		
ProteinNPT axial-core adaptation	0.02838	[0.00942, 0.04773]
Absolute-only ESM-2	0.00976	[−0.00075, 0.01888]
Mutation-delta ESM-2	0.01701	[0.00544, 0.02879]
Mutation-count regression	0.02559	[0.00214, 0.04887]
Training mean	0.08073	[0.05662, 0.10483]
Exact random selection	0.07980	[0.05586, 0.10354]
Tite-Seq frozen ensembles (Table 17). Method minus validation-weighted-primary error.		
Absolute view	0.0057589	[−0.0072510, 0.0183536]
Mutation-delta	0.0170573	[0.0002068, 0.0352813]

Continued on the next page

Table 25 (continued)

Statistic / comparison	Estimate	95% interval
Dual-view	0.0006870	[−0.0117442, 0.0141659]
ProteinNPT axial-core	0.0175656	[−0.0034369, 0.0389753]
Dual + axial, equal weights	−0.0020238	[−0.0108951, 0.0069843]
Four families, equal weights	−0.0025588	[−0.0100834, 0.0047336]
Full ProteinNPT on PYL1 (Table 19). Original dual-view minus full ProteinNPT error.		
Top-8	0.004511	[−0.006408, 0.015506]
Top-19	0.012025	[−0.002608, 0.026732]
Validation-selected response composition (Table 20). Baseline minus selected-composition error on shared evaluation cases.		
Top-8 / Original dual-view	0.022980	[0.013228, 0.033054]
Top-8 / One-hot	0.014390	[0.003350, 0.026002]
Top-8 / Full ProteinNPT	0.018655	[0.007225, 0.029978]
Top-19 / Original dual-view	0.029365	[0.015740, 0.042933]
Top-19 / One-hot	0.011096	[−0.001438, 0.023952]
Top-19 / Full ProteinNPT	0.017518	[0.001682, 0.032694]
ProteinMPNN candidate pools (Table 21). Our pool minus ProteinMPNN pool error under the same response selector.		
Top-8	0.01354	[0.00528, 0.02193]
Additional candidate sources (Table 22). Comparator minus our selected error; endpoint-CB values are approximate.		
CB: official endpoint	≈ +0.0321	[0.0186, 0.0461]
Selection within ProteinMPNN pools (Table 23). Selection-rule error minus generator-ranking error.		
Top-8 / Pure response selection	0.003285	[−0.011938, 0.018690]
Top-8 / Validation-selected blend	−0.011516	[−0.022148, −0.001380]
Top-19 / Pure response selection	0.022267	[0.005534, 0.038972]
Top-19 / Validation-selected blend	−0.014349	[−0.024817, −0.004413]
Matched candidate-source comparisons (Table 24). Our pool minus ProteinMPNN pool error.		
Nested blend / Top-8	0.026484	[0.016428, 0.037223]
Nested blend / Top-19	0.022866	[0.013179, 0.032916]
All-seed ranks / Top-8	0.03243	[0.02017, 0.04527]
All-seed ranks / Top-19	0.02298	[0.01236, 0.03386]
Marginal outcome intervals		
PYL1 / dual-view Top-8 error	0.24138	[0.21785, 0.26592]
PDZ / structural-ranking binding energy	0.21826	[0.15558, 0.28468]
PDZ / response-selected binding energy	0.14663	[0.09428, 0.20327]
Tite-Seq / dual-view error	0.32287	[0.29969, 0.34624]
Tite-Seq / axial-core adaptation error	0.35125	[0.33253, 0.36936]