# CodeFP2: ONTOLOGY-GUIDED *De Novo* FUNCTIONAL PROTEIN DESIGN

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

*De novo* functional protein design aims to generate proteins from scratch with desired molecular activities, enabling applications such as therapeutics and biocatalysis. Protein functions are hierarchically organized, with related functions sharing higher-level functional structure. However, existing generators either condition directly on target functions or enrich them with auxiliary functional context, leaving the hierarchical structure among functions largely unexploited. To address this challenge, we propose **CodeFP2**, an ontology-guided conditioning framework for joint sequence–structure diffusion that explicitly models functional hierarchies through broad-to-specific conditioning during denoising and pairwise relation encoding based on shared ancestry. Specifically, Ontology-Annealed Function Conditioning (OAFC) anneals each target's representation over its Gene Ontology ancestors during denoising, while Common-Ancestor Relation Encoding (CARE) encodes function pairs by symmetrically composing the paths connecting them to a shared ancestor. Experiments show consistent gains over the strongest baseline, including 2.7% in F1-Macro and 8.5% in pLDDT success rate on the full-distribution benchmark, and 5.7% in per-target exact recall on OOD tests.

## 1 INTRODUCTION

*De novo* functional protein design aims to create proteins with desired molecular activities, with applications spanning catalysis (Yeh et al., 2023), therapeutics (Chevalier et al., 2017; Cao et al., 2020), molecular recognition (Cao et al., 2022), and synthetic biology (Langan et al., 2019; Chen et al., 2020). By designing proteins from scratch, *de novo* approaches expand protein design beyond natural starting points, creating opportunities to explore a broader functional space and to control multiple molecular functions within a single protein (Kortemme, 2024).

A key property of protein functions is their hierarchical organization, in which specific biological activities refine broader functional capabilities. Consequently, functional targets are not independent categories: related functions may share higher-level properties while differing in more specific constraints. From a learning perspective, explicitly representing this hierarchy provides a natural inductive bias for sharing information across related functions and combinations, rather than treating each specific target or combination as an isolated condition.

Recent work formulates *de novo* functional protein design as conditional generation, yet does not fully exploit the hierarchical organization of protein function. Most methods are target-centric, representing each requested function at the target level using discrete identifiers or embeddings (Madani et al., 2023; Yin et al., 2025) or natural-language descriptions (Ingraham et al., 2023; Liu et al., 2025), leaving the function's hierarchical context implicit. A smaller class of context-enriched methods leverages additional functional information, but only partially captures the underlying hierarchy. ProteoGAN augments target annotations with inherited higher-level functions (Kucera et al., 2022) without explicitly modeling relations across hierarchical levels within each function. While Func2Seq and CodeFP initialize target-level representations with hierarchical information (Zhu et al., 2026; Chen et al., 2026), the hierarchy is collapsed into individual function embeddings rather than explicitly modeled during training. This limitation is particularly consequential for out-of-distribution (OOD) generation, where familiar functions are requested in combinations absent from training. Although the exact target-level combination is unseen, it may still share substantial higher-level functional structure with combinations observed during training.



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.

To exploit the hierarchical organization of biological function, we introduce **CodeFP2**, an ontology-guided framework for joint sequence–structure diffusion that models hierarchical function conditioning through two complementary factors: the specificity of each requested function and the pairwise relations among requested functions. We instantiate the functional hierarchy using the Gene Ontology (GO), whose molecular-function branch forms a directed acyclic graph (DAG) from broad functional concepts to more specific activities (Ashburner et al., 2000; Aleksander et al., 2023). Motivated by timestep-dependent semantic effects and coarse-to-fine denoising in text-to-image diffusion (Görgün et al., 2026; Yi et al., 2024), we hypothesize that functional conditioning should shift from broad semantics at highly corrupted protein states to target-specific constraints as the sequence and structure become progressively resolved. This hypothesis is also conceptually consistent with evolutionary models in which promiscuous or generalist activities precede functional specialization (Soskine & Tawfik, 2010). Accordingly, **O**ntology-**A**nnealed **F**unction **C**onditioning (OAFC) anneals each function condition over the GO hierarchy, progressively shifting from broader functional concepts toward the requested function as denoising proceeds, directly incorporating functional hierarchy into the denoising process. **C**ommon-**A**ncestor **R**elation **E**ncoding (CARE) models pairwise functional relations using the hierarchical paths that connect a function pair through their shared ancestor, thereby exposing hierarchical relations that can be reused across function combinations.

Experiments demonstrate that CodeFP2 improves functional recovery and foldability on both full-distribution and OOD benchmarks. Against the strongest baseline, it achieves absolute gains of 2.7% in F1-Macro and 8.5% in pLDDT $\geq 70$ success rate on the full test set, and 5.7% in per-target exact recall on OOD combinations observed in natural proteins. Our contributions are summarized as follows:

- We introduce CodeFP2, an ontology-guided diffusion framework that integrates the hierarchical organization of biological function into *de novo* functional protein design.
- We propose OAFC to align functional specificity with the denoising trajectory and CARE to encode pairwise functional relations from the GO hierarchy.
- CodeFP2 achieves state-of-the-art functional recovery and foldability on both full-distribution and OOD benchmarks.

## 2 Related Work

***De novo* functional protein design.** *De novo* functional protein design aims to create proteins that realize desired biological activities from scratch. Recent generators can be grouped by how they represent requested functions. *(1) Target-centric conditioning.* ProGen uses protein-family and property control tags (Madani et al., 2023); CFP-Gen and ProDiT use categorical functional annotations (Yin et al., 2025; Jing et al., 2025); and Chroma, Pinal, and related text-guided models use natural-language descriptions (Ingraham et al., 2023; Dai et al., 2024; Liu et al., 2025). *(2) Context-enriched conditioning.* ProteoGAN augments each annotation with its inherited GO ancestors (Kucera et al., 2022), Func2Seq combines GO descriptions with graph topology (Zhu et al., 2026), and CodeFP grounds individual GO terms with hierarchy-aware motifs (Chen et al., 2026). CodeFP2 instead incorporates the GO hierarchy into conditional generation by annealing the specificity of each requested function and modeling their relations through shared ancestry.

## 3 Method

We present the ontology-guided conditioning framework of CodeFP2, beginning with the problem formulation and conditional diffusion process (Section 3.1). We then introduce OAFC to align each requested function's semantic specificity with denoising progress (Section 3.2) and CARE to explicitly encode hierarchical relations among co-requested functions (Section 3.3).

### 3.1 Ontology-Guided Diffusion for Functional Protein Design

**Problem formulation.** Let a protein $P = (\mathbf{s}, \mathbf{x})$ contain an amino-acid sequence $\mathbf{s} = [s_1, \ldots, s_L]$ and backbone coordinates $\mathbf{x} \in \mathbb{R}^{L\times 4\times 3}$. Given a set of target molecular-function GO terms $\mathcal{Y} = \{y_1, \ldots, y_m\}$, the goal is to learn the conditional distribution $p_\theta(P \mid \mathcal{Y})$. Following DPLM-2 and

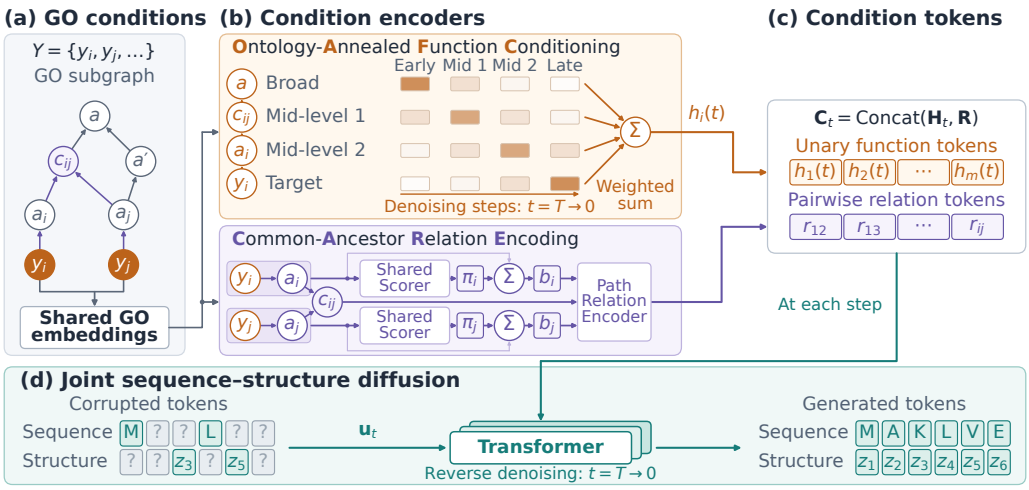



Figure 1: Overview of CodeFP2. OAFC shifts function semantics from broad GO ancestors to requested terms during denoising; CARE encodes function pairs via their closest common ancestor.

CodeFP (Wang et al., 2025; Chen et al., 2026), a pretrained structure tokenizer discretizes $\mathbf{x}$ into residue-level tokens $\mathbf{z} = [z_1, \ldots, z_L]$, yielding the clean state $\mathbf{u}_0 = (\mathbf{s}, \mathbf{z})$. We therefore model the objective as $p_\theta(\mathbf{u}_0 \mid \mathcal{Y})$.

**Forward absorbing diffusion.** For modality $v \in \{\mathrm{aa}, \mathrm{str}\}$, let $\mathbf{u}_0^v$ denote the clean amino-acid or structure-token sequence. The forward process independently replaces tokens with a modality-specific mask token according to an absorbing Markov chain (Austin et al., 2021). At diffusion timestep $t$, the marginal for modality $v$ is

$$q_v(u_{t,\ell}^v \mid u_{0,\ell}^v) = \mathrm{Cat}\left(u_{t,\ell}^v; u_{0,\ell}^v \overline{Q}_t^v\right), \qquad v \in \{\mathrm{aa}, \mathrm{str}\}, \tag{1}$$

where $u_{0,\ell}^v$ is represented as a one-hot row vector and $\overline{Q}_t^v$ is the cumulative transition matrix from the clean state to timestep $t$. We write the joint corrupted state as $\mathbf{u}_t = (\mathbf{s}_t, \mathbf{z}_t)$.

**Reverse denoising with time-dependent functional conditioning.** Starting from a corrupted joint state $\mathbf{u}_t$, the reverse process iteratively predicts the clean amino-acid and structure tokens and samples $\mathbf{u}_{t-1}$. Unlike prior approaches that use a fixed functional representation throughout generation, CodeFP2 constructs a time-dependent ontology condition that evolves from coarse functional semantics to the specific target functions as reverse denoising proceeds:

$$\mathcal{C}_t(\mathcal{Y}) = \left[\underbrace{\{\mathbf{h}_{y_i}(t)\}_{i=1}^m}_{\text{OAFC}} ; \underbrace{\{\mathbf{r}_{ij}\}_{(i,j)\in\mathcal{I}(\mathcal{Y})}}_{\text{CARE}}\right]. \tag{2}$$

The unary token $\mathbf{h}_{y_i}(t)$ varies with $t$ and implements the coarse-to-specific shift that aligns functional granularity with reverse denoising. The CARE component $\mathbf{r}_{ij}$ provides complementary pairwise ontology context. Here $\mathcal{I}(\mathcal{Y}) = \{(i,j) \mid 1 \le i < j \le m\}$ denotes unordered target pairs.

The resulting condition guides the clean-state predictor in the reverse transition:

$$p_\theta(\mathbf{u}_{t-1} \mid \mathbf{u}_t, \mathcal{Y}) = \sum_{\widehat{\mathbf{u}}_0} q(\mathbf{u}_{t-1} \mid \mathbf{u}_t, \widehat{\mathbf{u}}_0)\, p_\theta(\widehat{\mathbf{u}}_0 \mid \mathbf{u}_t, \mathcal{C}_t(\mathcal{Y})), \tag{3}$$

where $q(\cdot \mid \mathbf{u}_t, \widehat{\mathbf{u}}_0)$ is the posterior induced by the known absorbing process.

**Training objective.** We optimize the amino-acid and structure modalities jointly with the standard multimodal denoising objective:

$$\mathcal{L}_{\mathrm{denoise}} = \mathbb{E}_{\mathbf{u}_0, \mathcal{Y}, t}\left[\lambda(t)\left(\mathcal{L}_{\mathrm{aa}} + \mathcal{L}_{\mathrm{str}}\right)\right], \tag{4}$$

where $\mathcal{L}_{\mathrm{aa}}$ and $\mathcal{L}_{\mathrm{str}}$ are masked-token reconstruction losses and $\lambda(\cdot)$ weights their contributions at different timesteps.

### 3.2 ONTOLOGY-ANNEALED FUNCTION CONDITIONING

Modeling the hierarchy of protein function allows broader functional properties to inform the learning of more specific targets, coupling related functions across levels of specificity. Existing context-enriched methods expose ancestral information but flatten its hierarchical structure (Kucera et al., 2022), while others encode hierarchical context but collapse it into a static target-level representation (Chen et al., 2026; Zhu et al., 2026), leaving the reuse of hierarchical structure during training implicit. To address these limitations, OAFC models the hierarchy from the perspective of each individual requested GO term: it represents the target together with its ancestors and varies their contributions from broad to target-specific semantics as reverse denoising progresses. For each target $y$, OAFC produces the time-dependent condition $\mathbf{h}_y(t)$ in three stages: target-relative specificity encoding, timestep-dependent ancestor weighting, and target-preserving condition composition.

**Target-relative specificity encoding.** The molecular-function branch of GO forms a rooted directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, whose root $r_{\mathrm{MF}}$ represents the broadest molecular-function concept (Ashburner et al., 2000; Aleksander et al., 2023). Each node represents a functional concept, and each directed *is-a* edge connects a specific function to a broader parent. For a target term $y \in \mathcal{V}$, let $\mathcal{S}(y)$ contain $y$ and its ancestors in $\mathcal{G}$, and let $e_a \in \mathbb{R}^H$ denote the shared embedding of node $a \in \mathcal{S}(y)$. Because targets in different GO branches occur at different depths, OAFC assigns each represented node a target-normalized ontology-depth coordinate:

$$\delta_y(a) = \mathrm{clip}\left(\frac{d(a)}{\max(d(y), 1)}, 0, 1\right), \qquad a \in \mathcal{S}(y), \tag{5}$$

where $d(\cdot)$ denotes shortest-path depth from $r_{\mathrm{MF}}$. This normalization places the target at $\delta_y(y) = 1$ and assigns smaller values to ancestors that are shallower than the target, providing a common depth-based proxy for specificity across GO branches.

**Timestep-dependent ancestor weighting.** For timestep $t \in [0, T]$, let $\xi(t) = t/T$ denote normalized corruption and $\eta(t) = 1 - \xi(t)$ denote denoising progress. We posit that functional conditioning should evolve from broad to specific during denoising, while allowing the precise contribution of each ancestor at individual timesteps to adapt to the generation objective. OAFC therefore combines an explicit depth prior that encodes the global coarse-to-specific trend with a learned residual that refines these weights for protein generation:

$$\begin{aligned} \ell^{\mathrm{prior}}_{y,a}(t) &= -\frac{|\delta_y(a) - \eta(t)|}{\tau}, \\ \ell^{\mathrm{res}}_{y,a}(t) &= f_\phi([\delta_y(a), 1 - \delta_y(a), \xi(t), \eta(t), \mathbf{1}[a = y]]), \\ w_{y,a}(t) &= \mathrm{softmax}_{a \in \mathcal{S}(y)} \left(\lambda_p \ell^{\mathrm{prior}}_{y,a}(t) + \lambda_{\mathrm{res}} \ell^{\mathrm{res}}_{y,a}(t)\right). \end{aligned} \tag{6}$$

Here $f_\phi$ is a learned MLP, $\tau$ controls the selectivity of the depth prior, and $\lambda_p$ and $\lambda_{\mathrm{res}}$ balance the two logits. The prior is centered at the timestep-dependent specificity level $\eta(t)$, assigning larger logits to ancestors whose normalized depth $\delta_y(a)$ is closer to the current denoising progress. The shared MLP $f_\phi$ learns task-specific deviations from this global trajectory. The resulting weights produce the hierarchy summary $\bar{\mathbf{e}}_y(t) = \sum_{a \in \mathcal{S}(y)} w_{y,a}(t) e_a$.

**Target-preserving condition composition.** Broad ancestors provide useful coarse functional guidance early in denoising, but they are shared by many more specific functions and therefore do not uniquely determine the requested endpoint. We therefore maintain an explicit target signal whose strength increases over time, allowing broad semantics to dominate early generation while increasingly anchoring the trajectory to the requested function as denoising proceeds:

$$\mathbf{h}_y(t) = W_h \left(\bar{\mathbf{e}}_y(t) + \eta(t) e_y\right), \tag{7}$$

where $W_h$ is a learned projection. The hierarchy summary supplies reusable ancestral semantics, while the residual progressively reinforces the requested target function as denoising proceeds.

**Inference schedule.** The construction above defines OAFC for any semantic progress $\eta$. During generation, we use a delayed cubic schedule that preserves broad semantic guidance early and concentrates the transition to target-specific conditioning at later, more resolved stages:

$$\rho(t) = 1 - \frac{t}{T}, \qquad \eta_{\mathrm{inf}}(t) = \rho(t)^3, \qquad \xi_{\mathrm{inf}}(t) = 1 - \eta_{\mathrm{inf}}(t), \tag{8}$$

where the weighting and composition equations use $\eta_{\text{inf}}(t)$ and $\xi_{\text{inf}}(t)$ at reverse timestep $t$.

### 3.3 COMMON-ANCESTOR RELATION ENCODING

OAFC models each requested function together with its ancestors in a shared GO embedding space, establishing a hierarchy-aware unary basis for functional conditioning. However, unary representations alone do not explicitly characterize how co-requested functions are related within the function hierarchy. To address this gap, CARE builds on the same node representations to encode each target pair through its closest common ancestor and connecting ontology branches. Its construction comprises three stages: selecting the closest common ancestor, encoding the two branches with shared parameters, and composing their representations symmetrically into a relation token.

**Closest common ancestor.** CARE represents the shared ontology context of a target pair using the common ancestor that is closest to both targets, measured by the minimum total upward distance:

$$c_{ij} = \underset{a \in \text{Anc}_{\mathcal{G}}(y_i) \cap \text{Anc}_{\mathcal{G}}(y_j)}{\arg\min} \left[ D_{\mathcal{G}}(y_i, a) + D_{\mathcal{G}}(y_j, a) \right], \tag{9}$$

where $\text{Anc}_{\mathcal{G}}(y)$ includes $y$ itself, and $D_{\mathcal{G}}(y, a)$ is the shortest upward distance from $y$ to $a$. The shortest upward branches $B_i : y_i \rightsquigarrow c_{ij}$ and $B_j : y_j \rightsquigarrow c_{ij}$ then capture the ontology refinements between each endpoint and the shared ancestor.

**Shared branch encoding.** CARE treats $c_{ij}$ as the shared *center* of the pair and assigns every other node on $B_i$ and $B_j$ a shared *branch* role. Each path node $v$ combines its GO embedding $e_v$ shared with OAFC, a learned role embedding, and a learned projection of an ontology-and-path attribute vector $\mathbf{f}_v$ that summarizes node specificity and relative path position (Appendix C):

$$\mathbf{z}_v = e_v + e_{\text{role}(v)} + W_f \mathbf{f}_v. \tag{10}$$

Let $B_k^\circ = B_k \setminus \{c_{ij}\}$ denote the non-center nodes on branch $k$. CARE summarizes each branch with a learned scalar scorer $s_\psi$ shared by both sides:

$$\begin{aligned} \pi_{k,v} &= \frac{\exp(s_\psi(\mathbf{z}_v))}{\sum_{u \in B_k^\circ} \exp(s_\psi(\mathbf{z}_u))}, \quad v \in B_k^\circ, \\ \mathbf{b}_k &= \sum_{v \in B_k^\circ} \pi_{k,v} \mathbf{z}_v, \quad k \in \{i, j\}. \end{aligned} \tag{11}$$

The resulting $\mathbf{b}_i$ and $\mathbf{b}_j$ provide fixed-size summaries of the two branches for pairwise composition.

**Symmetric relation composition.** The relation between two requested functions depends on both the functional context they share and how their refinement paths diverge from that context. CARE therefore combines shared and contrasting information from the two branch representations with their common ancestor and global path geometry:

$$\mathbf{p}_{ij} = \text{LN}\big(f_{\text{path}}\left([\mathbf{b}_i + \mathbf{b}_j, |\mathbf{b}_i - \mathbf{b}_j|, \mathbf{z}_{c_{ij}}, f_g(\mathbf{g}_{ij})]\right)\big). \tag{12}$$

Here $\mathbf{g}_{ij} = \mathbf{g}_{ji}$ is an attribute vector summarizing the global geometry of the two branches and their shared center in GO (Appendix C). The resulting $\mathbf{p}_{ij}$ provides a symmetric representation of the pairwise functional relation induced by their shared ancestry and refinement paths.

To ground the ontology-derived relation in the specific target pair, CARE additionally incorporates a symmetric endpoint-pair representation:

$$\mathbf{q}_{ij} = \text{LN}\big(f_{\text{end}}([e_{y_i} + e_{y_j}, |e_{y_i} - e_{y_j}|])\big). \tag{13}$$

CARE then augments the path-based relation representation with this endpoint-pair signal:

$$\mathbf{r}_{ij} = \lambda_{\text{pair}} W_r \, \text{LN}(\alpha \mathbf{p}_{ij} + \mathbf{q}_{ij} + e_{\text{pair}}), \tag{14}$$

where $f_{\text{path}}$, $f_g$, $f_{\text{end}}$, and $W_r$ are learned projections, and $\alpha$ and $\lambda_{\text{pair}}$ are fixed path and output scales. The learned embedding $e_{\text{pair}}$, shared across target pairs, distinguishes CARE relation tokens from unary OAFC tokens in the conditioning sequence. Together, the path-based relation representation $\mathbf{p}_{ij}$ and auxiliary endpoint-pair signal $\mathbf{q}_{ij}$ yield an order-invariant relation token $\mathbf{r}_{ij}$ that encodes how the two requested functions are semantically related in the GO hierarchy. The resulting relation tokens are concatenated with the OAFC tokens in $\mathcal{C}_t(\mathcal{Y})$.

# 4 EXPERIMENTS

## 4.1 EXPERIMENTAL SETUP

**Datasets.** Following CodeFP (Chen et al., 2026), we adopt the dataset collected by Yin et al. (2025), comprising 95.6K training, 831 validation, and 8.3K test proteins annotated with 375 molecular-function GO terms derived from UniProt (Bateman et al., 2025) and InterPro (Blum et al., 2025). To assess whether models capture transferable functional structure beyond the training set, we additionally construct an OOD test set with two complementary subsets. The *natural* subset contains 200 GO-term combinations extracted from reviewed UniProt proteins, each including at least one term pair unseen during training. The *hypothetical* subset tests whether models can generalize from observed function combinations to closely related but unobserved combinations motivated by functional promiscuity and shifts in substrate or cofactor specificity (Tawfik & S, 2010; Eom & Song, 2019). For example, a natural condition combines urease activity (GO:0009039) with nickel cation binding (GO:0016151), while its hypothetical counterpart replaces nickel binding with the sibling manganese ion binding term (GO:0030145). Based on this construction, we construct 150 hypothetical pairs by replacing one function in a training-observed pair with a closely related GO sibling, retaining only combinations absent from both training and reviewed UniProt. To reduce evaluator-induced false negatives, we retain only functions reliably recognized on natural proteins. We generate five candidates per condition, yielding 1,750 proteins in total. Full construction and filtering details are provided in Appendix B.

**Implementation details.** CodeFP2 follows the training framework of CodeFP (Chen et al., 2026) and is initialized from the pretrained DPLM-2 (Wang et al., 2025) backbone. We retain its joint sequence–structure diffusion architecture and training objectives, replace the one-hot GO conditioning with the OAFC and CARE tokens defined in Equation 2, and jointly fine-tune all model parameters. During generation, sequence lengths are sampled uniformly from 200 to 400 residues, and denoising follows the annealed sampling procedure with the delayed OAFC specificity schedule, supported by its highest HS and ER in the validation comparison (Table 7). OAFC, CARE, and the expanded GO embeddings introduce only 23.49M additional parameters, increasing the model size from 1.866B to 1.889B (1.26%). Further training and sampling details are provided in Appendix C.

**Baselines.** We benchmark against seven representative methods spanning the two function-representation strategies discussed in Section 1. Target-centric baselines include ProGen2 (Nijkamp et al., 2023) and CFP-Gen (Yin et al., 2025), which condition directly on target GO terms, as well as Chroma (Ingraham et al., 2023) and Pinal (Dai et al., 2024), which use natural-language descriptions. Context-enriched baselines include ProteoGAN (Kucera et al., 2022), which augments annotations with inherited GO ancestors; Func2Seq (Zhu et al., 2026), which combines GO descriptions with graph topology; and CodeFP (Chen et al., 2026), which enriches GO terms with retrieved hierarchy-aware structural-motif representations. Among these methods, CodeFP is the strongest baseline and shares the training data and DPLM-2 backbone with CodeFP2.

**Metrics.** We evaluate generated proteins along three dimensions: *functional recovery*, *foldability*, and *sequence-space quality*. For functional recovery, we report MRR (Yin et al., 2025; Chen et al., 2026) to measure distributional consistency with natural proteins of matching functions, and use DeepGO-SE (Kulmanov et al., 2023) to compute F1-Micro, F1-Macro, AUPR, and AUC-ROC from predicted GO terms. Because standard label-based metrics ignore the GO hierarchy, we additionally report *Exact Recall* (ER) and the *Hierarchy Score* (HS). ER measures the fraction of requested GO terms recovered exactly, whereas HS gives decaying credit to recovered ancestors, distinguishing near-hierarchical recovery from complete failure. For multi-function requests, we further report *min-HS*, the minimum per-target hierarchy score, which reveals whether any requested function is left behind, and *All-Exact Rate*, a strict all-or-nothing criterion requiring all requested endpoints to be recovered simultaneously. Together, ER, HS, min-HS, and All-Exact provide a graded evaluation, from hierarchy-aware recovery of individual targets to exact realization of the full requested combination. For foldability, we report ESMFold (Lin et al., 2022) success rates at pLDDT $\geq 70$ and pTM $\geq 0.5$. For sequence-space quality, MMD and MMD-G measure alignment with natural proteins, while novelty and diversity are computed from sequence identity using MMseqs2 (Steinegger & Söding, 2017). Full details are provided in Appendix A.

Table 1: Performance on the full test distribution for GO-conditioned protein design. Function recovery is evaluated with DeepGO-SE, and foldability with ESMFold. ↑/↓ indicate that higher/lower values are better, respectively. The best results among generative models are shown in bold, and the second best are underlined. Positive Control denotes natural proteins from the test set.

| Model | Function recovery | | | | | Distribution | | Foldability (%) | |
|---|---|---|---|---|---|---|---|---|---|
| | F1 Micro ↑ | F1 Macro ↑ | AUPR ↑ | AUC ROC ↑ | MRR ↑ | MMD ↓ | MMD G ↓ | pLDDT $\geq 70$ | pTM $\geq 0.5$ |
| Positive control | 0.543 | 0.522 | 0.402 | 0.775 | 0.939 | 0.000 | 0.000 | 91.3 | 93.5 |
| ProGen2 | 0.414 | 0.355 | 0.240 | 0.663 | 0.545 | 0.109 | 0.064 | 64.2 | 67.2 |
| CFP-Gen | 0.429 | 0.370 | 0.245 | 0.674 | 0.601 | 0.112 | <u>0.060</u> | 75.5 | 72.3 |
| Chroma | 0.262 | 0.067 | 0.076 | 0.501 | 0.018 | 0.313 | 0.183 | 23.5 | 66.8 |
| Pinal | 0.452 | 0.369 | 0.229 | 0.663 | 0.379 | 0.223 | 0.131 | 74.2 | 82.2 |
| ProteoGAN | 0.376 | 0.093 | 0.121 | 0.510 | 0.277 | **0.095** | **0.055** | 0.01 | 0.00 |
| Func2Seq | 0.202 | 0.045 | 0.041 | 0.509 | 0.020 | 0.135 | 0.077 | 8.70 | 2.30 |
| CodeFP | <u>0.496</u> | <u>0.446</u> | <u>0.321</u> | <u>0.724</u> | <u>0.658</u> | <u>0.106</u> | 0.063 | <u>80.7</u> | <u>83.5</u> |
| CodeFP2 | **0.506** | **0.473** | **0.347** | **0.740** | **0.684** | 0.108 | 0.065 | **89.2** | **88.3** |

Table 2: OOD performance on natural and hypothetical function combinations. Positive control denotes all 2,873 natural proteins retrieved from UniProt for the 200 natural-combination conditions, establishing the evaluator ceiling on this subset.

| Subset | Model | Function recovery | | | | | | Foldability (%) | |
|---|---|---|---|---|---|---|---|---|---|
| | | F1-Micro ↑ | F1-Macro ↑ | HS ↑ | min-HS ↑ | ER ↑ | All-Exact ↑ | pLDDT ↑ | pTM ↑ |
| Natural | Positive control | 0.671 | 0.564 | 0.825 | 0.753 | 0.725 | 0.635 | 81.9 | 82.3 |
| | CodeFP | 0.526 | **0.395** | 0.395 | 0.281 | 0.204 | 0.105 | 71.0 | 62.3 |
| | CodeFP2 | **0.534** | 0.382 | **0.457** | **0.322** | **0.261** | **0.117** | **90.7** | **86.2** |
| Hypothetical | CodeFP | 0.428 | 0.263 | 0.333 | 0.190 | 0.081 | 0.000 | 62.5 | 67.3 |
| | CodeFP2 | **0.452** | **0.299** | **0.396** | **0.237** | **0.129** | **0.001** | **87.1** | **88.4** |

### 4.2 Main Results

**Improved functional recovery with strong foldability and comparable distribution alignment.** On the full test distribution, CodeFP2 improves functional recovery over CodeFP across all reported metrics, including F1-Macro (0.446→0.473) and MRR (0.658→0.684). Foldability improves more substantially, with pLDDT $\geq 70$ increasing from 80.65% to 89.23% and pTM $\geq 0.5$ from 83.48% to 88.31%, while MMD and MMD-G remain close to those of CodeFP. These results suggest that hierarchy-aware conditioning improves functional control without materially shifting the generated sequence distribution, while substantially increasing structural confidence. The foldability gain is consistent with our hypothesis that broader ancestral guidance early in denoising allows plausible scaffolds to form before more specific functional constraints are introduced.

**Further gains on the OOD benchmark.** On the natural OOD subset, CodeFP2 improves HS from 0.395 to 0.457 and ER from 0.204 to 0.261, despite a modest decrease in F1-Macro from 0.395 to 0.382 associated with increased false-positive predictions. Importantly, the HS and ER gains persist under matched prediction counts, suggesting that they are not driven solely by a larger prediction budget (Table 9). Min-HS also increases from 0.281 to 0.322, suggesting that the gains are not confined to the easiest targets but extend to weaker-recovered targets within each requested combination. The same trend holds for hypothetical combinations, with HS/min-HS/ER improving from 0.333/0.190/0.081 to 0.396/0.237/0.129. However, recovering all requested functions simultaneously remains challenging: All-Exact increases from 0.105 to 0.117 on natural combinations and remains near zero on hypothetical ones. Together with the large foldability gains on both subsets, these results indicate that ontology-guided conditioning improves compositional OOD generation, while reliable realization of complete unseen function combinations remains an open challenge.

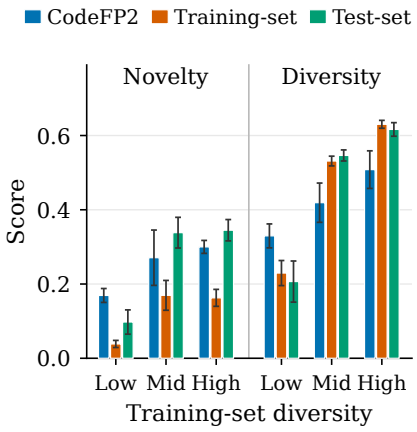



Figure 2: Novelty and diversity across function combinations with different levels of training-set diversity.

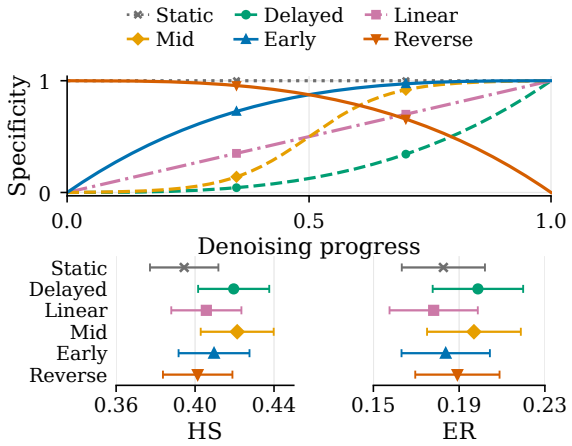



Figure 3: Effect of OAFC inference schedules on OOD function recovery. Top: scheduled specificity trajectories. Bottom: HS and ER over 350 OOD conditions.

**Novelty and diversity.** We evaluate sequence-space exploration on 15 training-observed function combinations stratified by training-set diversity into low-, medium-, and high-diversity regimes, comparing 24 generated sequences per combination with equally sized training and test references (Appendix B.3). Across the three regimes, proteins generated by CodeFP2 show higher novelty than the training references and comparable novelty and diversity to the test references (Figure 2), indicating consistent sequence-space exploration.

### 4.3 Effect of the Specificity Schedule

To study when functional specificity should be introduced, we vary only the OAFC inference schedule $\eta_{\text{inf}}(t)$ while fixing the checkpoint and evaluate the resulting generations on the OOD test set. Coarse-to-specific schedules consistently improve HS over static endpoint conditioning (Figure 3), while the reverse schedule underperforms the delayed schedule, indicating that the gain depends not only on varying specificity over time but also on the direction of the semantic transition. The delayed and mid-transition schedules are the strongest variants, consistent with broad functional guidance remaining useful until the emerging scaffold can support more specific constraints. Among the coarse-to-specific schedules, linear annealing is the weakest, suggesting that a concentrated shift in specificity is preferable to a uniform transition across denoising.

### 4.4 Case Study

We trace an illustrative generation conditioned on the single GO term homoserine O-acetyltransferase activity. At selected reverse steps, we decode the denoiser's current clean-sequence prediction and evaluate its functional and structural properties (Figure 4). The functional predictions follow a clear coarse-to-specific progression, with broad GO ancestors emerging before the requested activity. In parallel, the approximate location of the functional domain is established early, while its detailed conformation continues to evolve as the predicted fold becomes more reliable. Although qualitative, this coupled progression is consistent with our hypothesis that broad functional semantics guide early sequence–structure organization before more specific refinement.

### 4.5 Ablation Study

We ablate OAFC and CARE on the OOD test set under the same training protocol (Table 3). Removing both OAFC and CARE yields a retrained baseline using the CodeFP conditioning architecture.

**OAFC accounts for most of the OOD recovery gain.** Adding OAFC alone improves HS/min-HS/ER from 0.343/0.234/0.131 to 0.400/0.278/0.181, while removing OAFC causes a substantially larger drop than removing CARE, indicating that OAFC contributes most of the functional-recovery

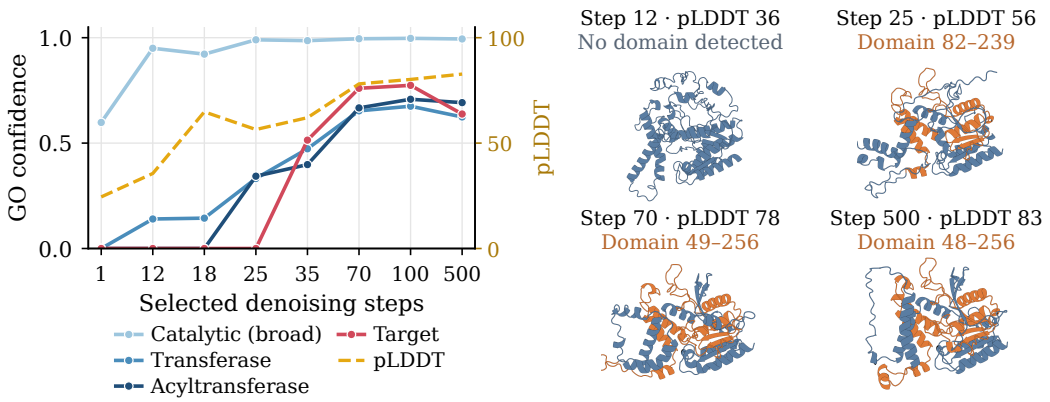



Figure 4: Functional and structural progression of a CodeFP2 generation conditioned on homoserine O-acetyltransferase activity (GO:0004414). Left: predicted GO confidence and pLDDT across denoising steps. Right: ESMFold-predicted structures at selected steps.

Table 3: Ablation of OAFC and CARE on the OOD test set.

| Model | HS ↑ | min-HS ↑ | ER ↑ | pLDDT $\geq 70$ ↑ | pTM $\geq 0.5$ ↑ |
|---|---|---|---|---|---|
| CodeFP2 | **0.420** | **0.285** | **0.199** | 89.1 | **87.1** |
| w/o CARE path (endpoint-only) | 0.407 | 0.272 | 0.190 | **90.7** | 83.2 |
| w/o OAFC annealing (mean-pool) | 0.373 | 0.241 | 0.159 | 87.4 | 79.0 |
| w/o CARE | 0.400 | 0.278 | 0.181 | 83.0 | 76.2 |
| w/o OAFC | 0.352 | 0.243 | 0.140 | 73.7 | 70.1 |
| w/o OAFC & CARE | 0.343 | 0.234 | 0.131 | 64.9 | 59.8 |

gain. To disentangle the effect of hierarchical context from that of timestep-dependent annealing, we further consider a static mean-pooling variant over the target and its GO ancestors. The static variant slightly outperforms w/o OAFC, showing the benefit of hierarchical information. Full OAFC further improves HS/min-HS/ER from 0.373/0.241/0.159 to 0.420/0.285/0.199, highlighting the substantial benefit of timestep-dependent annealing over static hierarchy encoding.

**CARE provides complementary gains.** Without OAFC, CARE improves HS/min-HS/ER from 0.343/0.234/0.131 to 0.352/0.243/0.140, while on top of OAFC it further improves them from 0.400/0.278/0.181 to 0.420/0.285/0.199, yielding larger incremental gains in HS and ER when combined with OAFC. To isolate the contribution of ontology paths, we evaluate an endpoint-only variant that retains the symmetric target-pair embedding but removes the shared-ancestor paths. The endpoint-only variant reaches 0.407/0.272/0.190 in HS/min-HS/ER, improving HS and ER over OAFC alone but not min-HS. Full CARE further improves all three metrics, indicating that explicit ontology-path modeling is a key contributor to CARE's gains. For foldability, both OAFC-only and CARE-only outperform the baseline, while their combination further improves pLDDT and pTM.

## 5 Conclusion and Future Work

We introduced **CodeFP2**, an ontology-guided conditional diffusion framework that exploits biological function hierarchies to improve *de novo* functional protein design. To this end, we developed two complementary modules: **OAFC** anneals each functional condition from broad GO ancestors toward its requested term during denoising, while **CARE** encodes hierarchical relations among co-requested functions through shared ancestry. Experiments demonstrate state-of-the-art functional recovery and foldability on the standard test set, together with improvements on the OOD benchmark.

While CodeFP2 advances OOD generation, reliably realizing unseen function combinations remains challenging. We see opportunities for future work in (1) expanding datasets for broader functional coverage and OOD evaluation; (2) investigating how semantic annealing shapes protein structure and function during denoising; and (3) conducting wet-lab validation of generated proteins.

## Ethics Statement

This paper presents work intended to advance the field of machine learning for protein design. There may be broader societal and dual-use implications associated with this research area, but we do not identify additional ethical concerns specific to the proposed method.

## Reproducibility Statement

We will release the datasets used in this work, the CodeFP2 model checkpoint, and the code. The paper and supplementary material specify the versions of the ontology and annotation mappings, the training and decoding configurations, and the implementation details and hyperparameters needed to reproduce our experiments. All ablations use matched initialization and optimization settings for controlled comparison. Equations 6–14 specify the proposed conditioning modules.

## AI Use Statement

Generative AI tools were used to assist with manuscript editing and the development of implementation and analysis code. All AI-assisted code and analyses were reviewed and verified by the authors. Numerical results reported in the paper were computed from stored model outputs using the corresponding evaluation procedures, rather than generated by AI tools. The authors take full responsibility for the final manuscript, results, claims, and supporting artifacts.

## References

Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris, et al. The gene ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031, 2023.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Alex Bateman, Maria-Jesus Martin, Sandra Orchard, Michele Magrane, Aduragbemi Adesina, Shadab Ahmad, Emily H Bowler-Barnett, Hema Bye-A-Jee, David Carpentier, Paul Denny, et al. UniProt: the universal protein knowledgebase in 2025. *Nucleic acids research*, 53(D1), 2025.

Matthias Blum, Antonina Andreeva, Laise Cavalcanti Florentino, Sara Rocio Chuguransky, Tiago Grego, Emma Hobbs, Beatriz Lazaro Pinto, Ailsa Orr, Typhaine Paysan-Lafosse, Irina Ponamareva, et al. InterPro: the protein sequence classification resource in 2025. *Nucleic acids research*, 53 (D1):D444–D456, 2025.

Longxing Cao, Inna Goreshnik, Brian Coventry, James Brett Case, Lauren Miller, Lisa Kozodoy, Rita E Chen, Lauren Carter, Alexandra C Walls, Young-Jun Park, et al. De novo design of picomolar sars-cov-2 miniprotein inhibitors. *Science*, 370(6515):426–431, 2020.

Longxing Cao, Brian Coventry, Inna Goreshnik, Buwei Huang, William Sheffler, Joon Sung Park, Kevin M Jude, Iva Marković, Rameshwar U Kadam, Koen HG Verschueren, et al. Design of protein-binding proteins from the target structure alone. *Nature*, 605(7910):551–560, 2022.

Xinrui Chen, Yizhen Luo, Siqi Fan, and Zaiqing Nie. Co-generative de novo functional protein design. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=O4BmkfXv1f.

Zibo Chen, Ryan D Kibler, Andrew Hunt, Florian Busch, Jocelynn Pearl, Mengxuan Jia, Zachary L VanAernum, Basile IM Wicky, Galen Dods, Hanna Liao, et al. De novo design of protein logic gates. *Science*, 368(6486):78–84, 2020.

Aaron Chevalier, Daniel-Adriano Silva, Gabriel J Rocklin, Derrick R Hicks, Renan Vergara, Patience Murapa, Steffen M Bernard, Lu Zhang, Kwok-Ho Lam, Guorui Yao, et al. Massively parallel de novo protein design for targeted therapeutics. *Nature*, 550(7674):74–79, 2017.

Fengyuan Dai, Shiyang You, Yudian Zhu, Yuan Gao, Lihao Fu, Xibin Zhou, Jin Su, Chentong Wang, Yuliang Fan, Xiaoxiao Ma, et al. Toward de novo protein design from natural language. *BioRxiv*, pp. 2024–08, 2024.

Hyunuk Eom and Woon Ju Song. Emergence of metal selectivity and promiscuity in metalloenzymes. *JBIC Journal of Biological Inorganic Chemistry*, 24(4):517–531, 2019.

Ada Görgün, Fawaz Sammani, Nikos Deligiannis, Bernt Schiele, and Jonas Fischer. Temporal concept dynamics in diffusion models via prompt-conditioned interventions. In *International Conference on Learning Representations*, volume 2026, pp. 66640–66669, 2026.

John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

Bowen Jing, Anna Sappington, Mihir Bafna, Ravi Shah, Adrina Tang, Rohith Krishna, Adam Klivans, Daniel J Diaz, and Bonnie Berger. Generating functional and multistate proteins with a multimodal diffusion transformer. *bioRxiv*, 2025.

Tanja Kortemme. De novo protein design—from new structures to programmable functions. *Cell*, 187(3):526–544, 2024.

Tim Kucera, Matteo Togninalli, and Laetitia Meng-Papaxanthos. Conditional generative modeling for de novo protein design with hierarchical functions. *Bioinformatics*, 38(13):3454–3461, 2022.

Maxat Kulmanov, Francisco J Guzmán-Vega, Paula Duek Roggli, Lydie Lane, Stefan T Arold, and Robert Hoehndorf. Deepgo-se: Protein function prediction as approximate semantic entailment. *bioRxiv*, pp. 2023–09, 2023.

Robert A Langan, Scott E Boyken, Andrew H Ng, Jennifer A Samson, Galen Dods, Alexandra M Westbrook, Taylor H Nguyen, Marc J Lajoie, Zibo Chen, Stephanie Berger, et al. De novo design of bioactive protein switches. *Nature*, 572(7768):205–210, 2019.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

Shengchao Liu, Yanjing Li, Zhuoxinran Li, Anthony Gitter, Yutao Zhu, Jiarui Lu, Zhao Xu, Weili Nie, Arvind Ramanathan, Chaowei Xiao, et al. A text-guided protein design framework. *Nature Machine Intelligence*, 7(4):580–591, 2025.

Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41(8): 1099–1106, 2023.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

Misha Soskine and Dan S Tawfik. Mutational effects and the evolution of new protein functions. *Nature Reviews Genetics*, 11(8):572–582, 2010.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Olga Khersonsky Tawfik and Dan S. Enzyme promiscuity: a mechanistic and evolutionary perspective. *Annual review of biochemistry*, 79(1):471–505, 2010.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Dplm-2: A multimodal diffusion protein language model. In *International Conference on Learning Representations*, volume 2025, pp. 35463–35490, 2025.

Andy Hsien-Wei Yeh, Christoffer Norn, Yakov Kipnis, Doug Tischer, Samuel J Pellock, Declan Evans, Pengchen Ma, Gyu Rie Lee, Jason Z Zhang, Ivan Anishchenko, et al. De novo design of luciferases using deep learning. *Nature*, 614(7949):774–780, 2023.

Mingyang Yi, Aoxue Li, Yi Xin, and Zhenguo Li. Towards understanding the working mechanism of text-to-image diffusion model. *Advances in Neural Information Processing Systems*, 37:55342–55369, 2024.

Junbo Yin, Chao Zha, Wenjia He, Chencheng Xu, and Xin Gao. CFP-gen: Combinatorial functional protein generation via diffusion language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72485–72504. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/yin25g.html.

Shaowen Zhu, Yue Cao, Yihong Yang, and Yang Shen. Protein function-conditioned language models for variant effect prediction and controllable design. *Biophysical journal*, 2026.

# A Evaluation Metrics

We evaluate functional recovery, foldability, and sequence-space quality. Let $\mathcal{Y}_i$ denote the requested GO terms for protein $i$, and $\widehat{\mathcal{Y}}_i$ denote its predicted terms.

**GO hierarchy.** We construct the functional hierarchy from the 2025-10-10 release of `go-basic.obo`, distributed by the Gene Ontology Consortium (Aleksander et al., 2023).[1] We retain molecular-function terms as nodes and *is-a* relations as directed edges from a specific function to its broader parent. The resulting directed acyclic graph is rooted at molecular function (GO:0003674); a term may have multiple parents. Traversing these edges yields the ancestors used for hierarchical conditioning and evaluation.

## A.1 Functional fidelity

**MRR.** Following CodeFP (Chen et al., 2026), we represent each sequence by an $\ell_2$-normalized vector of amino-acid 3-mer counts. For each GO term $c$, generated and natural proteins carrying that annotation form groups $G_c$ and $R_c$. We rank the natural groups by the Euclidean distance between their mean vectors and that of $G_c$. If the matching group $R_c$ has rank $r_c$, mean reciprocal rank is

$$\mathrm{MRR} = \frac{1}{|\mathcal{V}_{\mathrm{eval}}|} \sum_{c \in \mathcal{V}_{\mathrm{eval}}} \frac{1}{r_c}, \tag{15}$$

where $\mathcal{V}_{\mathrm{eval}}$ contains labels with nonempty groups. Higher MRR indicates closer agreement with natural proteins of matching functions.

**F1, AUPR, and AUC-ROC.** We average the six DeepGO-SE ensemble predictions and retain GO terms with scores at least 0.1 (Kulmanov et al., 2023). The benchmark evaluator expands target annotations to their ancestors and evaluates targets and predictions over a shared label universe $\mathcal{U}$. For label $c$, let $\mathrm{TP}_c$, $\mathrm{FP}_c$, and $\mathrm{FN}_c$ denote true positives, false positives, and false negatives. Then

$$\begin{aligned} \mathrm{F1}_{\mathrm{Micro}} &= \frac{2\sum_c \mathrm{TP}_c}{2\sum_c \mathrm{TP}_c + \sum_c (\mathrm{FP}_c + \mathrm{FN}_c)}, \\ \mathrm{F1}_{\mathrm{Macro}} &= \frac{1}{|\mathcal{U}|} \sum_{c \in \mathcal{U}} \frac{2\mathrm{TP}_c}{2\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c}. \end{aligned} \tag{16}$$

Undefined F1 values are set to zero. We also report the area under the precision-recall curve (AUPR) and the area under the receiver operating characteristic curve (AUC-ROC), averaged across classes using the benchmark implementation inherited from CodeFP.

**Exact Recall and All-Exact Rate.** Throughout the paper, ER and HS denote GO-macro averages: we first average recovery across all generated proteins requesting a given GO term, then average equally over the distinct requested terms. For an evaluation set of proteins, let $\mathcal{G} = \bigcup_i \mathcal{Y}_i$ and $\mathcal{I}_y = \{i : y \in \mathcal{Y}_i\}$. ER measures exact target recovery with equal weight for each requested GO term, whereas All-Exact requires all requested terms to be recovered together:

$$\begin{aligned} \mathrm{ER} &= \frac{1}{|\mathcal{G}|} \sum_{y \in \mathcal{G}} \frac{1}{|\mathcal{I}_y|} \sum_{i \in \mathcal{I}_y} \mathbf{1}[y \in \widehat{\mathcal{Y}}_i], \\ \mathrm{AllExact}_i &= \mathbf{1}[\mathcal{Y}_i \subseteq \widehat{\mathcal{Y}}_i]. \end{aligned} \tag{17}$$

We use the exported predictions without additional ancestor expansion; additional predicted terms do not invalidate an exact hit. All-Exact is averaged over proteins. For HS and ER, each evaluation subset defines its own $\mathcal{G}$, including terms with zero recovery, and all five generated proteins per condition are included. For a single-protein case study, the same definitions reduce to averaging over that protein's requested terms.

[1] `https://geneontology.org/docs/download-ontology/`

**Hierarchy Score.** HS gives partial credit when the predictor recovers an ancestor of a requested term. Let $\mathrm{Anc}(y)$ include $y$ and its *is-a* ancestors, excluding the molecular-function root, and let $D(y, a)$ be the shortest upward distance from $y$ to $a$. Using the same GO-macro aggregation as ER, we compute

$$\mathrm{HS} = \frac{1}{|\mathcal{G}|} \sum_{y \in \mathcal{G}} \frac{1}{|\mathcal{I}_y|} \sum_{i \in \mathcal{I}_y} \max_{a \in \widehat{\mathcal{Y}}_i \cap \mathrm{Anc}(y)} 2^{-D(y,a)}, \tag{18}$$

with zero for an empty intersection. An exact hit contributes 1, a parent hit contributes $1/2$, and a two-hop ancestor hit contributes $1/4$.

**Minimum Hierarchy Score.** Whereas HS averages recovery across requested GO terms, joint functional recovery additionally requires that no requested term is left behind. min-HS therefore scores a request by its weakest term:

$$\text{min-HS}_i = \min_{y \in \mathcal{Y}_i} \max_{a \in \widehat{\mathcal{Y}}_i \cap \mathrm{Anc}(y)} 2^{-D(y,a)}, \tag{19}$$

with zero for an empty intersection. It is the soft counterpart of the all-exact criterion: min-HS reaches one only when every requested term is recovered exactly, but partial ancestor credit still distinguishes requests whose weakest term is close to being recovered.

**Closure precision and recall.** For the threshold analysis, we expand both requested and predicted terms to ancestors and restrict predictions to the union of all OOD target closures, excluding the root. This gives a fixed universe of 341 terms. Precision and recall are the fraction of predicted terms that match the target closure and the fraction of the target closure recovered, respectively. We report the mean per-protein harmonic mean as closure F1, assigning zero to empty predictions.

### A.2 FOLDABILITY

We fold each generated amino-acid sequence with ESMFold (Lin et al., 2022). For $N$ proteins, the pLDDT and pTM success rates are

$$\begin{aligned} \mathrm{SR}_{\mathrm{pLDDT}} &= \frac{100}{N} \sum_i \mathbf{1}[\overline{\mathrm{pLDDT}}_i \geq 70], \\ \mathrm{SR}_{\mathrm{pTM}} &= \frac{100}{N} \sum_i \mathbf{1}[\mathrm{pTM}_i \geq 0.5]. \end{aligned} \tag{20}$$

The two criteria are evaluated separately and measure predicted structural confidence.

### A.3 SEQUENCE-SPACE QUALITY

**MMD and MMD-G.** We compare generated features $x_1, \ldots, x_n$ with natural-reference features $z_1, \ldots, z_m$, using the same normalized 3-mer representation as MRR:

$$\begin{aligned} \widehat{\mathrm{MMD}}_k^2 = \frac{1}{n^2} \sum_{i,i'} k(x_i, x_{i'}) + \frac{1}{m^2} \sum_{j,j'} k(z_j, z_{j'}) \\ - \frac{2}{nm} \sum_{i,j} k(x_i, z_j). \end{aligned} \tag{21}$$

We report the square root of the quantity in Equation 21, with diagonal terms included in the sums. MMD uses $k(x, z) = x^\top z$; MMD-G uses $k(x, z) = \exp(-\|x - z\|_2^2/(2\sigma^2))$, where $\sigma$ is the median pairwise distance in the pooled feature set. A zero median is replaced by one. Lower values indicate closer sequence distributions.

**Novelty.** Novelty measures how different a sequence is from its closest match in the training database. For a set $Q = \{s_1, \ldots, s_K\}$, let $\mathcal{H}(s_i)$ be the training sequences returned by MMseqs2 for query $s_i$ (Steinegger & Söding, 2017). We compute

$$\mathrm{Novelty}(Q) = 1 - \frac{1}{K} \sum_{i=1}^{K} \max_{r \in \mathcal{H}(s_i)} \mathrm{Id}_{\mathrm{MMseqs2}}(s_i, r), \tag{22}$$

where identity is the reported percentage divided by 100, and an empty hit set contributes zero identity. We use the sensitive search setting (`-s 7.5`) and allow up to 200 candidate matches per query to pass the initial filter for alignment (`--max-seqs 200`). Matches need not cover a minimum fraction of either sequence. When evaluating training references, we remove each query's own database entry from its hit set to avoid a trivial identity of one.

**Diversity.** Diversity measures variation within a set by averaging global-alignment identity over all unordered sequence pairs:

$$\text{Diversity}(Q) = 1 - \frac{2}{K(K-1)} \sum_{1 \le i < j \le K} \text{Id}_{\text{global}}(s_i, s_j). \tag{23}$$

Here $\text{Id}_{\text{global}}(s_i, s_j) = M_{ij} / \max(|s_i|, |s_j|)$, where $M_{ij}$ counts identical residues in the global alignment. Alignment uses match/mismatch scores $1/0$ and gap-open/extension scores $-1/-0.5$. We use $K = 24$ for the generated, training-reference, and test-reference sets. Both novelty and diversity lie in $[0, 1]$, with larger values indicating greater sequence differences. Condition selection is described in Appendix B.3.

### A.4 AGGREGATION AND ADDITIONAL RESULTS

For per-protein OOD metrics, we average the five candidates within each condition, then average conditions equally within each subset. HS and ER use GO-macro averaging; schedule-plot error bars are standard errors across the 169 per-GO means.

Table 4 separates exact recovery from ancestor matches. Exact hits account for most of the HS gain on both subsets.

Table 4: GO-macro nearest-predicted-ancestor distance distribution (%).

| Subset | Model | Exact | 1 hop | 2 hops | $\geq$ 3 hops | No hit |
|---|---|---|---|---|---|---|
| Natural | CodeFP | 20.41 | 24.26 | 18.78 | 24.27 | 12.28 |
| Natural | CodeFP2 | 26.12 | 25.99 | 20.05 | 17.64 | 10.20 |
| Hypothetical | CodeFP | 8.12 | 32.82 | 22.46 | 33.48 | 3.13 |
| Hypothetical | CodeFP2 | 12.91 | 38.59 | 19.87 | 25.32 | 3.31 |

Table 5 shows that HS, ER, and closure F1 gains persist at higher prediction thresholds, alongside a precision–recall trade-off.

Table 5: Prediction-threshold sensitivity over 350 conditions. P, R, and F1 use the fixed target-closure universe; the *Terms* column reports the number of directly predicted labels.

| Model | Threshold | HS | ER | P | R | F1 | Terms |
|---|---|---|---|---|---|---|---|
| CodeFP | 0.1 | 0.3598 | 0.1480 | 0.4053 | 0.5015 | 0.4195 | 13.96 |
| CodeFP2 | 0.1 | 0.4196 | 0.1988 | 0.3902 | 0.5552 | 0.4310 | 16.51 |
| CodeFP | 0.3 | 0.2479 | 0.0804 | 0.6057 | 0.3333 | 0.3855 | 5.57 |
| CodeFP2 | 0.3 | 0.2970 | 0.0996 | 0.5815 | 0.3898 | 0.4222 | 7.22 |
| CodeFP | 0.5 | 0.1753 | 0.0470 | 0.7025 | 0.2291 | 0.3038 | 3.07 |
| CodeFP2 | 0.5 | 0.2215 | 0.0583 | 0.6646 | 0.2883 | 0.3602 | 4.29 |

# B Datasets and Benchmark Construction

## B.1 Training and test data

We use the sequence–structure dataset provided by CodeFP (Chen et al., 2026), which augments the CFP-Gen functional-protein dataset (Yin et al., 2025) with structural data. The split contains 95,627 training, 831 validation, and 8,309 test proteins, with 375 molecular-function GO labels. We retain the CodeFP structural preprocessing and tokenization for joint sequence–structure training. The training annotations contain 541 distinct label combinations and 567 distinct unordered label pairs.

## B.2 Compositional OOD benchmark

**OOD definition.** We evaluate whether a model can generalize functional conditioning to combinations of functions whose individual terms have been observed during training but whose co-occurrence is unseen. We first list all unordered GO-term pairs annotated together on at least one training protein. A candidate combination is OOD if every term occurs in training, but at least one of its pairs is absent from this list. Here, unseen pairs are defined using the original training annotations. For example, $\{a, b, c\}$ qualifies if $a$, $b$, and $c$ are individually observed, but no training protein is annotated with both $a$ and $b$. This criterion therefore requires at least one previously unseen pairwise co-occurrence, rather than merely requiring the complete label set to be absent from training. Intersecting the training vocabulary with the function predictor's vocabulary gives 336 eligible terms. We construct two subsets: combinations found on natural proteins and hypothetical combinations obtained by replacing one term in an observed pair.

**Natural combinations.** We collect reviewed UniProt proteins of length 50–1,000 residues and remove training accessions. Each candidate consists of the training-vocabulary molecular-function annotations assigned to a single protein; we retain sets containing an unseen pair as defined above. Thus, the requested functions are known to occur together in a natural protein. To retain conditions recognizable by the evaluator, we run DeepGO-SE on up to three representatives per combination, choosing those closest to 300 residues. We retain the best-scoring representative and require a hierarchy-based selection score of at least 0.5.[2]

**Hypothetical combinations.** Starting from a training combination $\{a, b\}$, we replace $b$ with an eligible sibling term $b'$ that shares a direct *is-a* parent with $b$. This preserves a shared functional category while changing the specific function paired with $a$. We require $a \neq b'$, discard pairs already observed in training, and deduplicate candidates. We retain pairs for which our reviewed-UniProt search returns no protein annotated with both terms. For each constituent term, DeepGO-SE must give a nonzero score under the selection rule above on at least half of the natural examples annotated with that term. This checks that the evaluator recognizes the individual functions before testing their new combination.

**Functional-distance stratification.** We sample across levels of GO separation to cover function pairs ranging from closely related terms to terms in distant branches. This prevents selection from concentrating on one type of functional combination. We use the depth-based separation measure $\widetilde{D}(a, b) = d(a) + d(b) - 2d(c^*)$, where $d(v)$ is the shortest distance from $v$ to the root and $c^*$ is a deepest common ancestor of $a$ and $b$. For combinations with more than two terms, we average over all pairs. Near, medium, and far bins correspond to values below 4, from 4 to below 6, and at least 6, respectively.

**Final selection.** Within each bin, natural candidates are ranked by representative score, with at most eight combinations per GO term to limit the dominance of frequent functions. Hypothetical candidates are ranked by the training frequency of the source pair $\{a, b\}$, prioritizing substitutions from well-observed combinations. The final set has 200 natural and 150 hypothetical conditions

[2]The selection score averages $\max(0, 1 - k/d)$ over requested terms, where $d$ is the term's shortest distance to the root and $k$ its shortest distance to a predicted ancestor, including itself. Terms without a predicted ancestor receive zero.

(Table 6). We generate five proteins per condition, using GO labels rather than natural reference sequences as generation inputs.

Table 6: Composition of the OOD benchmark.

| Property | Natural | Hypothetical |
|---|---|---|
| Conditions | 200 | 150 |
| Generated proteins per method | 1,000 | 750 |
| Distinct requested GO terms | 109 | 93 |
| Near / medium / far conditions | 70 / 50 / 80 | 45 / 45 / 60 |
| Target count 2 / 3 / 4 / 5 / 7 | 128 / 53 / 15 / 3 / 1 | 150 / 0 / 0 / 0 / 0 |

### B.3 Novelty and diversity subset

We select training-observed combinations of two to four GO terms with at least 24 proteins in both training and test splits and a median training length between 180 and 420 residues. Candidate diversity is estimated from pairwise identity among up to 16 training proteins. The highest 15%, middle 40–60%, and lowest 15% of the identity distribution define low-, medium-, and high-diversity pools. We choose five combinations per pool, preferentially avoiding repeated GO terms. For each, we sample 24 training and 24 test references and generate 24 proteins. Selection and reference sampling use seed 42; results are summarized over five conditions per regime.

## C Model Details

### C.1 Architecture and model size

CodeFP2 uses the DPLM-2 650M backbone with 33 Transformer layers, hidden width 1,280, feed-forward width 5,120, and a joint vocabulary of 8,229 tokens. We retain CodeFP's diffusion modulation, functional cross-attention, training-derived structural-motif embeddings, and auxiliary motif head, and introduce OAFC and CARE for GO conditioning. The full model has 1.889B parameters, compared with 1.866B for CodeFP, including the auxiliary head in both counts. OAFC, CARE, and the expanded GO embeddings add 23.49M parameters, a 1.26% increase.

### C.2 OAFC and CARE implementation

**OAFC.** We construct the molecular-function *is-a* graph from `go-basic.obo`. Each target uses at most 12 nodes, including itself and excluding the root. Ancestors are sorted by root depth and GO identifier; when the support exceeds 12 nodes, we sample approximately uniformly across the sorted ancestors while always retaining the target. Targets without represented ancestors use their own embedding. The residual scorer is a $5 \to 32 \to 1$ MLP with SiLU activation. The learned temperature is clamped to $[0.02, 1]$; initial values are $\tau = 0.12$, $\lambda_p = 1$, and $\lambda_{\text{res}} = 0.1$.

**CARE paths and attributes.** For each path node, we use eight attributes describing its ontology position and role: normalized information content and depth, absolute and normalized distances to the nearer endpoint and to the shared center, and binary indicators for endpoint and center nodes. Both distances are normalized by $\max(L_{\text{branch}}, 1)$, where $L_{\text{branch}}$ is the corresponding endpoint-to-center branch length in edges. Information content is computed from ancestor-expanded training counts as $\log_2[n_{\text{min-parent}}(v)/n(v)]$ and normalized globally. For each target pair, we additionally use eight attributes describing its overall path geometry: the mean and absolute difference of endpoint depths, center depth, total branch length, the mean and difference of the two branch lengths, and indicators for ancestor–descendant and root-center relations.

**CARE networks.** The two branches share role embeddings and a LayerNorm–$1280 \to 64 \to 1$ attention scorer with SiLU. Geometry and path-composition MLPs have widths $8 \to 1280 \to 1280$ and $5120 \to 1280 \to 1280$. The endpoint projection uses the sum and absolute difference of target embeddings, with two zero-filled blocks preserving its input width. We use path weight $\alpha = 0.5$, pair-token scale $\lambda_{\text{pair}} = 0.35$, and pair dropout 0.1. CARE is independent of diffusion time.

### C.3 Training procedure and computational cost

We initialize directly from the pretrained DPLM-2 backbone and perform one 110,000-update task-training run with all model parameters enabled for fine-tuning. The diffusion and optimization protocol follows CodeFP: AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.98$, weight decay 0.01, FP16, seed 42, and a maximum length of 1,022 residues. The learning rate starts at $4 \times 10^{-5}$ with 2,000 warmup updates and follows polynomial decay to $10^{-5}$.

For clarity, Eq. 4 in the main text presents the multimodal denoising objective, whereas the full training objective follows CodeFP and additionally retains its motif-supervision loss:

$$\mathcal{L}_{\text{train}} = \tfrac{1}{2}\big(\mathcal{L}_{\text{aa}} + \mathcal{L}_{\text{str}} + 2\mathcal{L}_{\text{motif}}\big). \tag{24}$$

Single-modality, folding, inverse-folding, and joint denoising modes are sampled with equal probability. When the two modalities use different timesteps, OAFC conditions on the larger timestep during training, with linear semantic progress $1 - t_{\text{cond}}/T$.

**Ablation training protocol.** All ablations use the same initialization, training configuration, and 110,000-update budget as the full model, differing only in the removed conditioning components.

**CARE endpoint-only ablation.** To isolate the contribution of ontology-path information in CARE, we remove the path-based relation representation $p_{ij}$ while retaining the symmetric endpoint-pair representation $q_{ij}$. Recall that the full CARE relation token is

$$r_{ij} = \lambda_{\text{pair}} W_r \,\text{LN}\left(\alpha p_{ij} + q_{ij} + e_{\text{pair}}\right). \tag{25}$$

The endpoint-only variant sets the path contribution to zero:

$$r_{ij}^{\text{endpoint}} = \lambda_{\text{pair}} W_r \,\text{LN}\left(q_{ij} + e_{\text{pair}}\right). \tag{26}$$

Thus, this variant preserves explicit pairwise conditioning on the two requested functions but removes the shared-ancestor paths and their ontology geometry.

**OAFC mean-pooling ablation.** To separate the benefit of using ancestral information from that of annealing its specificity over denoising, we replace OAFC's timestep-dependent ancestor weighting with static mean pooling. For target $y$ with represented ontology support $S(y)$, the full OAFC hierarchy summary is

$$\bar{e}_y(t) = \sum_{a \in S(y)} w_{y,a}(t) e_a, \tag{27}$$

where $w_{y,a}(t)$ varies with denoising progress. The mean-pooling variant instead uses

$$\bar{e}_y^{\text{mean}} = \frac{1}{|S(y)|} \sum_{a \in S(y)} e_a, \qquad h_y^{\text{mean}} = W_h \bar{e}_y^{\text{mean}}, \tag{28}$$

which is constant across diffusion timesteps. This variant therefore retains the same target-and-ancestor ontology information but removes the coarse-to-specific annealing mechanism.

### C.4 Inference settings

OOD generation uses five candidates per condition, lengths sampled uniformly from 200–400 residues, 500 reverse steps, and sampling temperature annealed from 2.0 to 1.0. We use seed 42 plus the worker GPU index, without classifier-free guidance. The CodeFP amino-acid resampling rule is retained with ratio 0.25 and scale 1.0. For denoising progress $p \in [0, 1]$, the specificity schedules in Figure 3 are

$$\begin{aligned} \eta_{\text{static}}(p) &= 1, & \eta_{\text{linear}}(p) &= p, \\ \eta_{\text{early}}(p) &= 1 - (1-p)^3, & \eta_{\text{delayed}}(p) &= p^3, \\ \eta_{\text{mid}}(p) &= \frac{\sigma(12(p-1/2)) - \sigma(-6)}{\sigma(6) - \sigma(-6)}, & \eta_{\text{reverse}}(p) &= 1 - p^3. \end{aligned} \tag{29}$$

Here $\sigma$ is the sigmoid function, and the condition timestep is $\max(1, \text{round}[T(1-\eta)])$. Both the full-test and OOD main comparisons use delayed. Functional and structural metrics in each comparison use the same generated amino-acid sequences.

Table 7: Schedule selection on the validation set.

| Schedule | HS ↑ | ER ↑ |
|---|---|---|
| Early | 0.457 | 0.268 |
| Linear | 0.465 | 0.276 |
| Mid-transition | 0.469 | 0.283 |
| Delayed | **0.472** | **0.284** |

## D BASELINE IMPLEMENTATION

We retain the benchmark results previously reported for ProGen2, CFP-Gen, Chroma, Pinal, ProteoGAN, and CodeFP in (Chen et al., 2026). Func2Seq is the additional baseline implemented for this work. The OOD comparison focuses on CodeFP because of its leading overall and OOD recovery in the preceding benchmark (Chen et al., 2026).

**Func2Seq implementation.** We use the released meta-pretrained checkpoint `meta_lr0.00005.model` at step 330,000 (Zhu et al., 2026), conditioning each request on molecular-function GO embeddings from the released embedding table. We sample one sequence per request with temperature 1.0, top-$k = 21$, seed zero, and a maximum of 1,002 positions; sequence length is determined by the stop symbol. Outputs are evaluated with the same metrics as the other methods.

## E Additional Case Study

We examine a natural OOD request derived from UniProt protein Q9HXE3, which combines nucleotide binding (GO:0000166) with D-amino-acid dehydrogenase activity (GO:0008718). Figure 5 compares the natural reference with illustrative CodeFP2 and CodeFP outputs. The delayed-schedule CodeFP2 sample recovers both requested terms, with pLDDT 89.05 and pTM 0.945. InterProScan identifies a PF01266 family match in both the reference (residues 5–347) and the generated sample (4–372), providing independent domain-level support for the predicted functional assignment. The displayed CodeFP sample recovers partial ancestral functions, with HS 0.375 and ER zero.

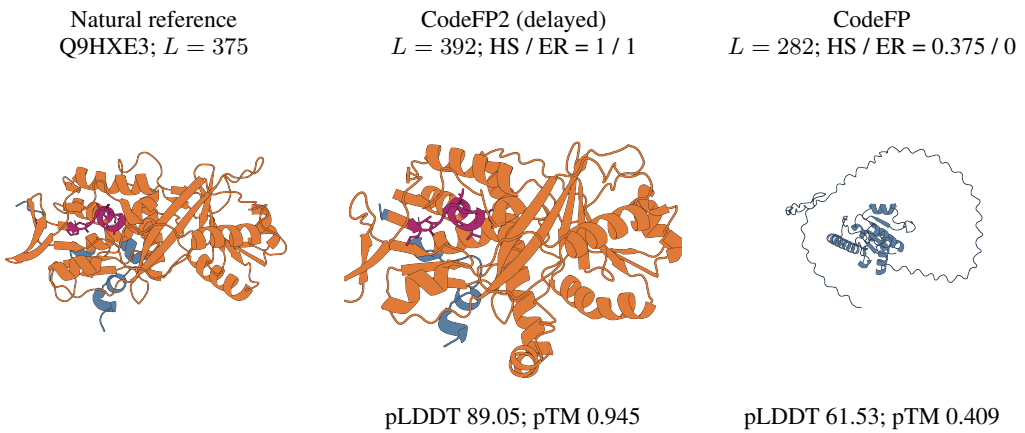



Figure 5: Natural OOD example. Structures are predicted by ESMFold. Orange regions mark PF01266 family-domain matches carrying the catalytic function; magenta sticks mark the glycine-rich Rossmann loop (GXGXXG motif) binding the FAD cofactor.

## F Additional Analysis of OOD Generations

We analyze the five generated proteins per condition from Table 2, covering 200 natural and 150 hypothetical requests. HS and ER use the GO-macro definitions above; other results are averaged within each condition and then across conditions.

### F.1 Recovery of individual and combined targets

Let $\mathcal{P}_{\text{new}}(\mathcal{Y}_i)$ contain the target pairs absent from the original training annotations. Unseen-pair recall measures how often both terms of such a pair are recovered in the same protein:

$$\text{UPR}_i = \frac{1}{|\mathcal{P}_{\text{new}}(\mathcal{Y}_i)|} \sum_{\{a,b\} \in \mathcal{P}_{\text{new}}(\mathcal{Y}_i)} \mathbf{1}[\{a, b\} \subseteq \widehat{\mathcal{Y}}_i]. \tag{30}$$

We also report the fraction of proteins that exactly recover at least one requested target.

Table 8: Target and unseen-pair recovery on OOD generations. Any exact target denotes the fraction of proteins that recover at least one requested function exactly, while unseen-pair recall directly measures recovery of function pairs unseen during training.

| Subset | Metric | CodeFP | CodeFP2 |
|---|---|---|---|
| Natural | Any exact target | 0.5120 | 0.6130 |
| Natural | Unseen-pair recall | 0.1211 | 0.1441 |
| Hypothetical | Any exact target | 0.1480 | 0.2440 |
| Hypothetical | Unseen-pair recall | 0.0000 | 0.0013 |

CodeFP2 increases the fraction of proteins that recover at least one target exactly from 51.2% to 61.3% on natural requests and from 14.8% to 24.4% on hypothetical requests. Natural unseen-pair

recall also increases, from 0.1211 to 0.1441. Despite these improvements, the low absolute recovery rates indicate that complete realization of unseen function combinations remains challenging.

### F.2 SPECIFICITY OF TARGET RECOVERY

**Recovery at matched prediction counts.** To control for differences in the number of predicted labels, we match the prediction count between CodeFP and CodeFP2 for each paired sample. Predictions are ranked by DeepGO-SE score, and both models retain the same number of top-scoring labels, capped by the smaller number available above the 0.1 threshold. We evaluate budgets from $k = 2$ to 16; Table 9 reports representative values.

Table 9: Recovery with matched prediction counts over all 350 OOD conditions. Both models retain the same number of predictions for each paired sample.

| | | ER | | HS | |
|---|---|---|---|---|---|
| Budget $k$ | Mean labels | CodeFP | CodeFP2 | CodeFP | CodeFP2 |
| 2 | 2.00 | 0.0138 | 0.0195 | 0.1273 | 0.1351 |
| 4 | 3.98 | 0.0325 | 0.0424 | 0.1979 | 0.2123 |
| 8 | 7.39 | 0.0717 | 0.0889 | 0.2695 | 0.2878 |
| 16 | 10.76 | 0.1122 | 0.1333 | 0.3224 | 0.3493 |

CodeFP2 achieves higher ER and HS across all evaluated budgets. The recovery advantage persists under matched prediction counts, indicating that the gains are not solely explained by predicting more labels or the associated increase in false-positive opportunities.

**Condition reassignment.** We keep each protein's predictions fixed and reassign it a condition from the same OOD subset with the same number of targets. Table 10 averages 1,000 such shuffles over 349 conditions; one condition has no eligible replacement.

Table 10: ER under the original and reassigned requests. The gap is original minus reassigned ER.

| Model | Original ER | Reassigned ER | Gap |
|---|---|---|---|
| CodeFP | 0.1482 | 0.0179 | 0.1303 |
| CodeFP2 | 0.1984 | 0.0229 | 0.1755 |

Both models recover more targets from the original requests. The larger gap for CodeFP2 supports greater specificity to the requested labels.