LeFlur: A Biomolecular Design Model with Latent Structure Tokens

Sidney L. Lisanza [1] **Karina Zadorozhny** [1] **Frederic A. Dreyer** [1] **Kyunghyun Cho** [2]

Abstract

Protein design pipelines today are fragmented, separating backbone generation, sequence design, and structure prediction into bespoke models. We present LeFlur, a unified discrete-token model that integrates these tasks into a single standard text transformer. We offload Cartesian-space modeling to LatentGenerator, a Vision Transformer autoencoder that discretizes arbitrary 3D structures (proteins and small molecules alike) into a compact, shared-vocabulary (with modality-specific quantizers $\mathcal{Q}^{(p)}$ and $\mathcal{Q}^{(\ell)}$ feeding a standard transformer), using a Simple Linear Quantizer (SLQ), an application of Gumbel-Softmax categorical bottlenecks to per-residue 3D-coordinate latents,as a minimal-vocabulary alternative to Finite Scalar Quantization. On top of this representation, LeFlur supports both protein-only and ligand-conditioned design, a first for structure-token models. Because LeFlur is trained as a masked any-order model, it admits a sequence/structure/joint pseudo-likelihood (PLL). This PLL correlates strongly with downstream designability and structure-prediction accuracy and serves as a best-of-N ranker that re-uses the model itself. We also introduce Self-Reflection, an inference-time refinement loop that iteratively redesigns outputs against an internal forward-fold consistency check, lifting designability without any retraining or external scorer. Despite its architectural simplicity, LeFlur is competitive with specialized baselines across folding and generation tasks.

Code can be found at https://github.com/prescient-design/lobster/tree/main/src/lobster/model/latent_generator and https://github.com/prescient-design/lobster/tree/main/src/lobster/model/leflur. [1]Prescient Design, Genentech [2]New York University. Correspondence to: Sidney Lisanza <sidneylisanza@gmail.com>.

Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)

1. Introduction

Effective computational protein design requires navigating the joint probability distribution between a protein's amino acid sequence and its three-dimensional structure in order to ascertain if the intended function can be accomplished with the given sequence structure pair. Despite this intrinsic link, traditional computational approaches have largely treated these modalities in isolation. State-of-the-art predictive models are highly specialized—utilizing distinct architectures for folding (Jumper et al., 2021) versus sequence design (Dauparas et al., 2022). Likewise, pipelines for *de novo* generation often require three separate models: a structure generator, a sequence designer, and a structure verifier (Watson et al., 2023; Pacesa et al., 2025; Stark et al., 2025). This fragmentation bottlenecks design campaigns, which must often generate and filter hundreds of thousands of candidates to identify a fraction of self-consistent pairs.

While the need for joint sequence-structure modeling is apparent, a fundamental barrier to a unified protein design model has been the data type mismatch: sequences are discrete, while structures are continuous. One promising approach is to model sequence and structure within the same modality, as discrete tokens, to be processed by a single model. This strategy has been explored most notably by DPLM-2 (Wang et al., 2024) and ESM3 (Hayes et al., 2025), yet both systems introduce significant architectural complexity and computational overhead to handle structure, and are specific to protein-only systems.

We simplify protein structure tokenization by introducing **LatentGenerator**, a universal structure tokenizer that abstracts continuous 3D coordinates into a minimal discrete vocabulary. Unlike the complex equivariant architectures of prior works, LatentGenerator utilizes a standard Vision Transformer (Dosovitskiy et al., 2021) (ViT) autoencoder trained with a Simple Linear Quantizer (SLQ)—a direct application of the Gumbel-Softmax categorical bottleneck (Jang et al., 2017) to per-residue 3D-coordinate latents. Our contribution is the empirical finding that for protein backbones a learned linear projection plus a $K{=}256$ codebook is sufficient to recover high-fidelity geometry, orders of magnitude smaller than DPLM-2's 8,192-token codebook or ESM3's 4,096-token codebook. We further document

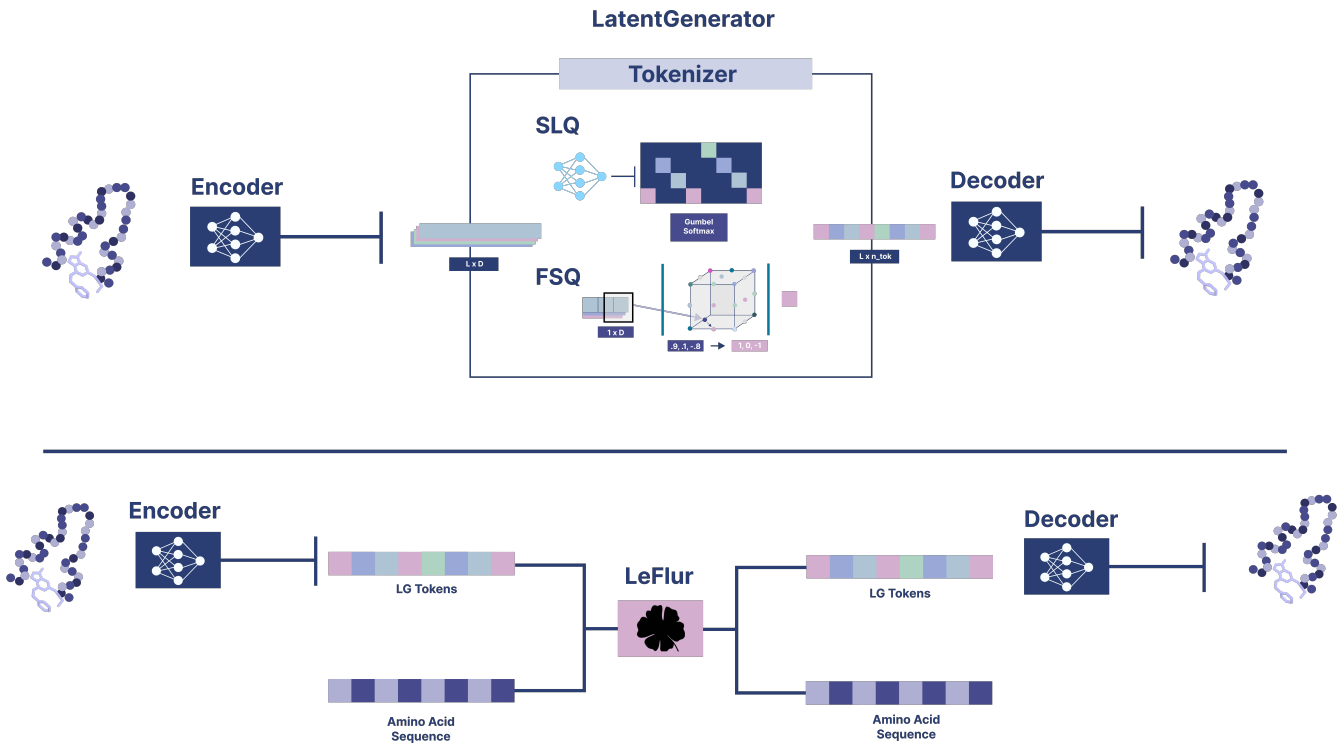


Figure 1. **LatentGenerator and LeFlur model overview**. LatentGenerator allows for the tokenization of protein, ligand, and protein-ligand complex structures. LeFlur allows for the generation of both protein sequence and structures with ligand conditioning, making it amenable for protein design tasks such as inverse folding, forward folding, and unconditional generation.

that this small-codebook regime does *not* scale to reconstructing protein-ligand complexes (Tables S1, Tables S2, and S3), where SLQ degrades sharply and we revert to FSQ (Mentzer et al., 2023). Crucially, because our architecture avoids domain-specific heuristics, it readily generalizes beyond proteins: we demonstrate the unified tokenization of protein-ligand complexes within a single model, preserving the relative geometry of both modalities without specialized small-molecule or protein models.

We then present **LeFlur** as the application of these structure tokens to the problem of generative protein design. By transforming the complex physical problem of 3D geometry into a unified sequence modeling task, LatentGenerator enables the seamless integration of structural data into standard, highly optimized architectures. We plug these discrete representations directly into NeoBERT (Breton et al., 2025), an encoder originally designed for text. This enables LeFlur to be *natively multimodal*: it processes amino acid sequences and structure tokens indistinguishably within a single attention stream. We train this backbone via Discrete Flow Matching (Campbell et al., 2024; Gat et al., 2024) to solve three distinct design tasks: unconditional generation, inverse folding, and forward folding without the need for massive pre-training or complex specialized architectures.

Our contributions are as follows:

- **modality-agnostic structure tokenisation.** We show that a vanilla Vision Transformer autoencoder, trained with $SE(3)$ data augmentation rather than equivariant layers, suffices to tokenise 3D biomolecular structure. The same encoder–decoder ingests proteins, small molecules, and protein-ligand complexes, removing the protein-specific inductive biases. More so, our tokens are informative enough to only require single step decoding rather than multi-step diffusion based decoding.
- **Simple Linear Quantization (SLQ).** We show that protein-backbone geometry can be compressed by a single learned linear projection followed by a Gumbel-Softmax categorical bottleneck (Jang et al., 2017), with a codebook of only 256 tokens.
- **LeFlur** By plugging LatentGenerator tokens directly into an off-the-shelf NLP encoder (NeoBERT (Breton et al., 2025)) and training with discrete flow matching over independently sampled sequence and structure timesteps, a single set of weights solves inverse folding, forward folding, and unconditional generation for both protein-only and protein-ligand systems while being competitive with specialized, task-specific baselines.
- **Built-in inference-time verifiers via discrete latents.** Discretising structure makes the joint distribution over

sequence and structure tractably scoreable: we use the model's own pseudo-likelihood as a ranker for generated designs, and a mutual-information-motivated *Self-Reflection* criterion (refold the generated sequence with the same model and accept only when the refold agrees with the originally sampled structure) as a learned-oracle-free designability filter, both improving downstream metrics without invoking an external folder.

2. Related Work

2.1. Co-Generation (Seq-Struc) Continuous Models

Traditional co-generative models produce protein backbones and sequences simultaneously but generally rely on architectures repurposed from structure prediction networks to account for the sequence-structure modality mismatch (Lisanza et al., 2024; Didi et al., 2025; Watson et al., 2023; Stark et al., 2025; Campbell et al., 2024). To manage the rigid physical and equivariant constraints of 3D geometry, these models require domain-specific modules, like triangular attention, that incur significant computational costs. LeFlur bypasses these complexities by offloading Cartesian modeling to a dedicated autoencoder. By discretizing 3D structures into a latent vocabulary, we decouple geometric constraints from the generative process, allowing a standard, highly optimized text-transformer to serve as the backbone. Furthermore, transforming both sequence and structure to the same modality (tokens) circumvents their modality mismatch and allows us to use the same losses (e.g. cross-entropy) for both representations.

2.2. Discrete Token-based Generative Models

The conversion of complex 3D biomolecular geometry into a discrete alphabet was pioneered by Foldseek's 3Di tokens (van Kempen et al., 2024a), which enabled the application of sequence-alignment tools to structural data. SaProt (Su et al., 2024) and ProstT5 (Heinzinger et al., 2024) later utilized these tokens to train Protein Language Models (PLMs) for structure-based homology search. However, while effective for retrieval, 3Di tokens are unsuitable for structure generation as they lack a dedicated decoder to reconstruct atomic coordinates. We further show by training a 3Di conditioned flow matching structure decoder (Appendix E, Table S4) that the 3Di tokens are insufficient to encode global structure but better serve for encoding local environments.

Recent architectures like ESM3 (Hayes et al., 2025) have advanced this paradigm by using multi-track Transformers that reason over discrete structure, sequence, and functional tokens via masked language modeling. Similarly, DPLM-2 (Wang et al., 2024) employs a multimodal discrete diffusion framework with structure tokens integrated into a pretrained PLM backbone. While these models demonstrate the power of discretization, they rely on complex, protein-specific equivariant encoders. Without the limitations of protein-only encoders; LeFlur with LatentGenerator enables the seamless extension to the discretization of protein-ligand complexes.

3. Molecular Encoding and Structure Tokenization with LatentGenerator

3.1. Overview

LatentGenerator supports protein-only, ligand-only, and protein-ligand tokenization. LatentGenerator consists of three components: (1) a ViT encoder $\mathcal{E}_\theta$ that maps 3D coordinates to continuous embeddings, (2) a quantizer $\mathcal{Q}_\phi$ that discretizes embeddings into tokens, and (3) a ViT decoder $\mathcal{D}_\psi$ that reconstructs coordinates from tokens. Further details can be found in Appendix A and schematics are in Fig S7 for protein only and Fig S8 for protein-ligand auto encoding.

Simple Linear Quantization (SLQ). SLQ employs a learned linear projection combined with the Gumbel-Softmax estimator (Jang et al., 2017) for differentiable discrete quantization, schematics comparing this approach to FSQ are provided in Fig S9. Given continuous embeddings $\mathbf{Z} \in \mathbb{R}^{L\times d}$, we first compute logits by projecting to the intended codebook size K (typically $K{=}256$) via $\boldsymbol{\ell} = \mathrm{LayerNorm}(\mathbf{Z})\,\mathbf{W}_q^\top \in \mathbb{R}^{L\times K}$, where $\mathbf{W}_q \in \mathbb{R}^{K\times d}$ is a learned projection matrix. To enable gradient flow through the discrete bottleneck, we apply the Gumbel-Softmax (Jang et al., 2017) estimator for differentiable sampling, defined as $\mathbf{C}_i = \mathrm{softmax}((\boldsymbol{\ell}_i{+}\mathbf{g})/\tau) \in \Delta^{K-1}$. Here, $\mathbf{g} = -\log(-\log(\mathbf{u}))$ represents Gumbel noise samples derived from $\mathbf{u} \sim \mathrm{Uniform}(0,1)$, and τ is the temperature parameter (default $\tau = 0.5$). This process generates soft assignments that approach one-hot vectors as $\tau \to 0$. During inference and for all downstream evaluations, we utilize the argmax of these logits to obtain discrete tokens. We also provided further details on FSQ (Mentzer et al., 2023) and how it compares to SLQ in Appendix B.

3.2. Training Objective

LatentGenerator is trained end-to-end with coordinate reconstruction losses. We apply a L^2 reconstruction loss after Kabsch alignment (Kabsch, 1976):

$$\mathcal{L}_{\mathrm{L}^2} = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \|\hat{\mathbf{Y}}_i - \mathbf{R}^*\mathbf{Y}_i - \mathbf{t}^*\| \tag{1}$$

where $(\mathbf{R}^*, \mathbf{t}^*)$ is the optimal rigid alignment and $\mathcal{M}$ is the set of valid positions.

We apply a pairwise distance loss for rotation-invariant supervision, and empirically find that it speeds up training:

$$\mathcal{L}_{\text{pw}} = \frac{1}{|\mathcal{M}|^2} \sum_{i,j \in \mathcal{M}} \text{clip}\Big(\big(\text{clip}(\|\hat{\mathbf{Y}}_i - \hat{\mathbf{Y}}_j\|, 20) - \|\mathbf{Y}_i - \mathbf{Y}_j\|\big)^2, 25\Big) \tag{2}$$

where the inner $\text{clip}(\cdot, 20)$ caps the predicted distance at 20Å so that beyond-cutoff pairs do not dominate the gradient (*distance clamping*), and the outer $\text{clip}(\cdot, 25)$ caps the squared error at 25 (*error clamping* at $\sqrt{25} = 5$Å).

For unified protein-ligand models, we apply losses to both modalities with weighting λ_ℓ:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{L}^2}^{(p)} + \mathcal{L}_{\text{pw}}^{(p)} + \lambda_\ell \left(\mathcal{L}_{\text{L}^2}^{(\ell)} + \mathcal{L}_{\text{pw}}^{(\ell)}\right) \tag{3}$$

Importantly, for protein-ligand complexes, Kabsch alignment is performed on the *concatenated* complex coordinates:

$$(\mathbf{R}^*, \mathbf{t}^*) = \text{Kabsch}([\hat{\mathbf{Y}}_p; \hat{\mathbf{Y}}_\ell], [\mathbf{Y}_p; \mathbf{Y}_\ell]) \tag{4}$$

This preserves relative protein-ligand positioning in the reconstruction loss.

3.3. Reconstruction Evaluation

We benchmark LatentGenerator on three modalities to verify that SLQ and FSQ retain enough geometry for downstream generation; full tables (CASP15, GEOM, PDBbind) are in Appendix D (Tables S1, S2, S3). On CASP15 the continuous baseline reaches 0.462 Å RMSD and the FSQ Prot-Lig tokenizer (4,375 tokens) closes most of the gap at 1.260 Å, while the specialist SLQ-256 used for LeFlur-p reaches 1.647 Å. On GEOM, FSQ scales to sub-0.3 Å (0.291 Å after MMFF94 minimisation, vs. 0.395 Å pre-minimisation), whereas SLQ saturates and degrades to 1.239 Å at 4,096 tokens. On PDBbind protein-ligand complexes, FSQ Prot-Lig (4,375/4,375) attains joint-aligned RMSD of 1.013 Å (protein) and 1.011 Å (ligand); the compact SLQ-256/512 remains competitive (1.507/2.306 Å), while increasing SLQ to 4,096 collapses joint-aligned RMSD to 4.761/3.589 Å. We therefore use SLQ-256 for LeFlur-p and FSQ-4,375/4,375 for LeFlur-pl. We further find that SLQ-256 without binder specific training extends to protein complexes well passed its original training length cutoff of 512. As detailed in Figure S1 it achieves a 3.17 Å RMSD and a TM score of 0.897 for complexes ranging from length 100 to 1000 residues. Finetuning just the decoder improves reconstruction to 2.78 Å and TM score of 0.912.

Decoding from 3Di tokens with a flow-matching backbone decoder. LatentGenerator's tokens are specifically trained for global atomistic reconstruction. To probe whether the Foldseek 3Di alphabet (van Kempen et al., 2024a), a local environment trained, 20-token structural vocabulary —admits an analogous token to global atom level reconstruction pipeline, we train a 3Di-conditioned flow-matching decoder (single step decoding was insufficient) that reconstructs backbone $N/C_\alpha/C$ coordinates from 3Di tokens. The decoder is a 768-dim, 12-layer U-ViT trained flow matching model with classifier-free guidance over per-residue 3Di-token conditioning. At inference the model integrates a 200-step SDE schedule at guidance scale w=2. We evaluate this decoder against LatentGenerator on the CASP15 monomers ($L \leq 512$) and on the 474-protein ProstT5 test split (Heinzinger et al., 2024) (Table S4). Despite the 3Di-flow decoder achieving high token recovery (71.1% on CASP15, 82.0% on ProstT5) it leaves a large reconstruction gap relative to the LatentGenerator (9.02 vs 1.33 Å Kabsch RMSD on CASP15; 0.644 vs 0.945 TM-score). The 3Di alphabet's 20 states encode local geometry only, and many globally-distinct folds map to the same 3Di string; the decoder must therefore generate a plausible global fold consistent with local descriptors, which the local only tokens cannot fully constrain. This motivates LatentGenerator's joint encoder-decoder global reconstruction training, which together preserve the global geometric information that the local-descriptor alphabets discard.

4. Biomolecular Prediction and Design with Latent Structure Tokens

4.1. Architecture

LeFlur consists of three main components: LatentGenerator our pre-trained structure tokenizer that converts 3D coordinates $\leftrightarrow$ discrete tokens, a bidirectional transformer that processes our sequence-structure tokens, and a discrete generative paradigm.

For each residue position i, **LeFlur** linearly embeds the sequence tokens $x_i \in \{0, \ldots, 21\}$, representing a 20-amino acid vocabulary plus mask and padding tokens, and the structure tokens $c_i \in \{0, \ldots, K+1\}$, which includes K codebook tokens plus mask and padding symbols. These are projected as $\mathbf{e}_i^{\text{seq}} = \text{Embed}_{\text{seq}}(x_i) \in \mathbb{R}^d$ and $\mathbf{e}_i^{\text{struc}} = \text{Embed}_{\text{struc}}(c_i) \in \mathbb{R}^d$, respectively. The modalities are fused into a combined embedding $\mathbf{e}_i = \mathbf{W}_{\text{combine}} \cdot [\mathbf{e}_i^{\text{seq}}; \mathbf{e}_i^{\text{struc}}]$, where $[\cdot;\cdot]$ denotes concatenation.

These fused representations are then processed by a NeoBERT (Breton et al., 2025) backbone to yield hidden states $\mathbf{h}_1, \ldots, \mathbf{h}_L = \text{NeoBERT}(\mathbf{e}_1, \ldots, \mathbf{e}_L)$. Finally, separate linear heads project these hidden states to predict the respective output logits: $\boldsymbol{\ell}_i^{\text{seq}} = \mathbf{W}_{\text{seq}} \cdot \mathbf{h}_i \in \mathbb{R}^{22}$ for the sequence modality and $\boldsymbol{\ell}_i^{\text{struc}} = \mathbf{W}_{\text{struc}} \cdot \mathbf{h}_i \in \mathbb{R}^{K+2}$ for the structure tokens.

Ligand modality (LeFlur-pl). For protein-ligand systems we extend the same 1D token stream by appending ligand atom tokens (atom-type embeddings) and ligand structure tokens to the protein tokens, so the full sequence processed by NeoBERT is $[\mathbf{e}_1^p, \ldots, \mathbf{e}_{L_p}^p; \mathbf{e}_1^\ell, \ldots, \mathbf{e}_{L_\ell}^\ell]$ (Fig. S13). In addition we also encode a ligand bond matrix $\mathbf{B} \in \{0, \ldots, 5\}^{L_\ell \times L_\ell}$ (encoding none/single/double/triple/aromatic/other).

4.2. Discrete Flow Matching

LeFlur employs Discrete Flow Matching (Campbell et al., 2024), enabling flow-based generation on categorical data. We define a probability path interpolating between a masked prior $p_0(\mathbf{x})$ and the data distribution $p_1(\mathbf{x})$.

For a data sample $\mathbf{x}_1 \sim p_1$ and time $t \in [0, 1]$, we construct noisy samples $\mathbf{x}_t$ via a linear masking schedule where $x_{t,i} = x_{1,i}$ with probability t and [MASK] otherwise. The model is trained to predict the clean data $\mathbf{x}_1$ from $\mathbf{x}_t$ by minimizing the expected cross-entropy:

$$\mathcal{L}_{\text{DFM}} = \mathbb{E}_{\mathbf{x}_1, t, \mathbf{x}_t} \left[\text{CE}(p_\theta(\mathbf{x}_1 | \mathbf{x}_t, t), \mathbf{x}_1) \right] \tag{5}$$

LeFlur samples independent timesteps $t_{\text{seq}}, t_{\text{struc}} \sim \mathcal{U}(0, 1)$ during training. By applying the flow matching loss independently to each modality ($\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seq}} + \mathcal{L}_{\text{struc}}$), the model learns to denoise one modality given arbitrary noise levels in the other. This decoupling enables flexible inference modes (unconditional generation, inverse folding, and forward folding) using a single weight set (Figure S10).

4.3. Generation

4.3.1. Inverse Folding

Protein only inverse folding. In the inverse folding task, the objective is to design an amino acid sequence $\mathbf{X}$ that folds into a fixed target backbone structure $\mathbf{Y}$. Within the LeFlur framework, this is achieved by fixing the structure tokens $\mathbf{c}_1$ (effectively setting $t_{\text{struc}} = 1$) and sampling the sequence trajectory t_{seq} from 0 to 1. We evaluated performance on the Campbell et al. benchmark (Campbell et al., 2024) and the CAMEO 2022 dataset (Haas et al., 2018) which are both past our PDB training cutoff date. We measure Native Sequence Recovery (AAR) and structural self-consistency by refolding generated sequences with ESMFold and calculating TM-score, RMSD, and pLDDT.

As shown in Table S5, on the Campbell et al. benchmark LeFlur Protein (LeFlur-p) exhibits strong performance. LeFlur-p outperforms the specialized ProteinMPNN (Dauparas et al., 2022) in AAR (53.60% vs 46.54%) While comparable to DPLM-2 (55.56%), highlighting the model's proficiency in sequence-structure understanding. All models benchmarked maintain the native structure after folding with ESMFold. TM scores are above 0.80 and pass rates (percent of structures with RMSD$\leq$2) are greater than 70%. On the more recent CAMEO 2022 benchmark (Table 1) LeFlur-p maintains an acceptable ESMFold-refold TM-score (≈ 0.76) while its AAR drops to $\sim 34\%$ and its pass rate to 37.0% (vs. ProteinMPNN at 55.9% and DPLM-2 at 41.7%). This AAR drop may reflect alternative sequences that still fold to the same backbone.

Protein-ligand inverse folding. In the protein-ligand inverse folding task, the objective is to design an amino acid sequence $\mathbf{X}$ that folds into a fixed target backbone structure $\mathbf{Y_p}$ binding onto the ligand, $\mathbf{Y_l}$ of interest. Within LeFlur Protein Ligand (LeFlur-pl), this is achieved by fixing the structure tokens for both the protein and the ligand $\mathbf{c}_{p1}$ and $\mathbf{c}_{l1}$ (effectively setting $t_{\text{struc}} = 1$) and sampling the sequence trajectory t_{seq} from 0 to 1.

We evaluated performance on the Posebusters benchmark (Buttenschoen et al., 2024) in Table 2. We measure overall AAR to determine if the model captures the global structure while measuring pocket AAR to see if the model captures ligand interactions. We use Boltz-2 to see if we recover the native structure and correct ligand placement. We measure correct ligand placement with GF+IP (good fold + in pocket). We consider a pass if the predicted structure has a TM-score $\geq$0.5 and at least one atom of the ligand within 6Å of a residue in the binding pocket. We outperform the specialized inverse folding model LigandMPNN (Dauparas et al., 2025) on ligand pocket AAR both with (74.80% vs 59.40%) while being comparable in protein structure and ligand pocket placement recovery as measured by Boltz-2 (0.603 TM vs. 0.647 TM) and GF+IPs of 41.6% vs 41.5%.

4.3.2. Forward Folding

Protein only forward folding. In the forward folding task, the model must predict the 3D structure $\mathbf{Y}$ given a fixed amino acid sequence $\mathbf{X}$. In our framework, this corresponds to fixing the sequence tokens $\mathbf{x}_1$ ($t_{\text{seq}} = 1$) and sampling the structure trajectory t_{struc} from 0 to 1. We measure performance using TM-score, RMSD, and Pass Rate (defined as the percentage of samples with RMSD < 2.0 Å).

Table S6 presents results on the Campbell et al. benchmark. The LeFlur-p model achieves a TM-score of 0.76 comparable to the performance of the DPLM-2 650M (0.77 TM-score). This parity is achieved using a standard ViT backbone and a minimal 256-token vocabulary, compared to DPLM-2's 8,192-token codebook. While the predictive specialist ESMFold 3B remains the upper bound (0.91 TM-score), LeFlur demonstrates that discrete generative models can approach predictive accuracy. Results on the CAMEO 2022 dataset (Table 3) demonstrate stronger Pass Rate (17.3%) relative to DPLM-2 (11.8%), while the average TM-score for LeFlur-p (0.67) is competitive to that of

Table 1. Inverse Folding Performance on the CAMEO 2022 Benchmark. Bold indicates the best model for that column; underline indicates the best token-based model. Suffix legend: `N30` = best-of-30 candidates per target; `PLL` = candidates ranked by joint pseudo-NLL (Section 4.5); `SR8`/`SR9` = Self-Reflection with forward-folding TM-score cutoff τ_{TM}=0.833/0.9; `oracle` = candidates ranked by ESMFold TM-score (upper bound for the ranker).

Model	Tokens	AAR (%)	TM	RMSD (Å)	Pass (%)	pLDDT
ProteinMPNN		42.93	0.85	4.18	**55.9**	**0.75**
DPLM-2 650M	8192	**49.22**	0.81	4.84	41.7	0.72
LeFlur-P 470M SLQ	256	34.00	0.76	4.60	37.0	0.66
LeFlur-P N30 PLL 470M SLQ	256	34.90	0.80	4.73	39.4	0.70
LeFlur-P N30 SR8 470M SLQ	256	35.10	0.81	4.49	39.4	0.70
LeFlur-P N30 SR9 470M SLQ	256	34.68	0.81	4.44	40.2	0.70
LeFlur-P N30 oracle 470M SLQ	256	35.11	**0.86**	**3.18**	52.0	0.71

Table 2. Protein Ligand Inverse Folding Performance on the PoseBusters Benchmark (Buttenschoen et al., 2024). AAR P represents the binding pocket measuring AAR at the pocket residues within a 5 Å radius of ligand atoms. TM-score and GF+IP are computed from a Boltz-2 (Passaro et al., 2025) cofold of (predicted-sequence, GT-ligand-SMILES) against the GT crystal structure; GF+IP requires $TM \geq 0.5$ AND at least one ligand atom within 6 Å of a pocket residue. Suffix legend: `N30` = best of 30 candidates per target; `PLL` = candidates ranked by joint protein pseudo-NLL (`joint_protein`, sum of seq and struc PLL with the ligand held clean, see Section 4.5); `oracle` = candidates ranked by GF+IP (upper bound for the ranker).

Model	Tokens	AAR (%)	AAR P (%)	TMscore	GF+IP (%)
LigandMPNN		52.87	59.40	**0.647**	41.5
LeFlur-pl 470M	4375	**68.20**	74.80	0.603	41.6
LeFlur-pl N30 PLL 470M	4375	67.47	**75.37**	0.595	41.6
LeFlur-pl N30 oracle 470M	4375	66.60	75.18	0.640	**70.4**

DPLM-2 (0.70) on this split. ESMFold maintains its lead while also dropping to 0.85 TM-score.

Protein-ligand forward folding. In the forward folding task with conditioning ligand information the model must predict the protein ligand complex 3D structure, $\mathbf{Y}_{\mathbf{pl}}$, given a fixed amino acid sequence $\mathbf{X}$, ligand atoms, ligand bond matrix, and ligand structure tokens. In our framework, this corresponds to fixing the sequence tokens $\mathbf{x}_1$ ($t_{\text{seq}} = 1$), ligand structure tokens and sampling the structure trajectory t_{struc} from 0 to 1.

We measure performance on the Posebusters benchmark (Buttenschoen et al., 2024) using TM-score and RMSD of the overall fold, in addition we score ligand pocket placement with GF+IP. We compare against RF3 (Corley et al., 2025) and Boltz-2 in single sequence mode. We find that we outperform all of the specialized models on single sequence folding TM-score where we achieve a score of 0.70. However, Boltz-2 outperforms LeFlur-pl on ligand placement where it achieves GF+IP of 49.6% vs. 26.4%, while RF3 lags with 20.8%.

4.4. Unconditional Generation

Protein only unconditional generation. In the unconditional generation task, the model must jointly sample an amino acid sequence $\mathbf{X}$ and its corresponding 3D structure $\mathbf{Y}$ from the learned joint distribution $P_\theta(X, Y)$. We evaluate the quality of these generations by assessing their structural viability (pass rate), and topological diversity (number of fold clusters), secondary structure diversity (helix-strand-coil percentages) across lengths ranging from 100 to 500 residues.

As detailed in Table 5, we compare against state-of-the-art models in continuous (Didi et al. (2025), Qu et al. (2024), Chen et al. (2025), Lu et al. (2024), Lisanza et al. (2024), Lin et al. (2024)) and discrete generation (Wang et al., 2024). Our standard LeFlur-p model outperforms them in pass rate where we achieve 84.8% vs. La Proteina's 79.6%, while maintaining a higher cluster count of 301 vs. 169. DPLM-2, although it has a lower pass rate, is very diverse in terms of secondary structures with a H/S/C of 38.2/17.2/44.6. We find that adding bias to the valine logits allows us to increase our strand percentage going from H/S/C of 84.3/0.2/15.5 to 69.1/7.0/24.0 while maintaining a pass rate of 75.8%.

Ligand conditioned generation. In the ligand conditioned generation task the model must generate a protein ligand complex 3D structure, $\mathbf{Y}_{\mathbf{pl}}$, given fixed ligand atoms and ligand bond matrix, but with masked ligand, structure, and sequence tokens. We take 4 ligands: IAI, FAD, SAM, and OQO from Didi et al. (2026) (the Proteina-Complexa

Table 3. Forward Folding Performance on the CAMEO 2022 Benchmark. Bold indicates the best model for that column; underline indicates the best token-based model. Suffix legend: `N30` = best-of-30 candidates per target; `PLL` = ranked by structure-token pseudo-NLL; `Oracle` = ranked by ESMFold TM-score (upper bound).

Model	Tokens	TM-Score	RMSD (Å)	Pass Rate (%)
ESMFold 3B		**0.85**	**4.34**	**49.6**
DPLM-2 650M	8192	0.70	7.40	11.8
LeFlurP 470M SLQ	256	0.67	11.97	17.3
LeFlurP N30 PLL 470M SLQ	256	0.69	12.34	17.3
LeFlurP N30 Oracle 470M SLQ	256	<u>0.75</u>	<u>6.73</u>	<u>26.8</u>

Table 4. Protein Ligand Forward Folding Performance on the PoseBusters Benchmark (Buttenschoen et al., 2024). GF+IP represents percentage of predictions with a TM-score $\geq$0.5 and at least one atom of the ligand within 6Å of a residue in the binding pocket. Suffix legend: `N30` = best of 30 candidates per target; `PLL` = candidates ranked by joint protein pseudo-NLL (`joint_protein` = sum of seq and struc PLL with the ligand held clean, see Section 4.5); `oracle TM` / `oracle GF+IP` = candidates ranked by the GT-task quality metric (upper bounds).

Model	Tokens	TM-Score	RMSD (Å)	GF+IP (%)
RF3		0.437	17.14	20.8
Boltz2		0.651	11.73	49.6
LeFlur-pl 470M	4375	0.703	12.19	26.4
LeFlur-pl N30 PLL 470M	4375	0.755	10.00	28.5
LeFlur-pl N30 oracle TM 470M	4375	<u>**0.793**</u>	<u>**7.59**</u>	30.1
LeFlur-pl N30 oracle GF+IP 470M	4375	0.696	11.22	<u>**56.9**</u>

benchmark setup) and follow the Boltz-design evaluation protocol of Cho et al. (2025); we generate 100 designs per ligand, and consider a successful design if Boltz predicts the sequence-ligand pair with high confidence (ipTM$\geq$0.9 and iPDE$\leq$1).

As detailed in Table 6, we compare against the state-of-the-art model Proteina-Complexa (Didi et al., 2026). We find both models to have low success rates with Proteina Complexa getting a 7.2% success rate while LeFlur-pl lags with 2%. Future work is needed to improve the success rate for both models.

4.5. Inference time scaling

Log Likelihood as a confidence module. Given that we operate in discrete space we can use the model's own likelihood as a proxy for a confidence head (Jumper et al., 2021). We calculate pseudo log likelihood (PLL) either for the sequence tokens, structure tokens, or both. We find a strong correlation (Table S7,Table S8,Table S9) in inverse, forward folding, and unconditional generation. Interestingly, we find structure token PLL to be the most predictive for forward folding accuracy (tm score of prediction vs gt structure) (Table S7), and sequence PLL for inverse folding accuracy (percent identity towards the gt sequence)(Table S8). For unconditional generation we find the structure token PLL is more predictive of ESMFold Tm score, but only at longer lengths where the model struggles generating (Table S9). We apply these finding to improve our forward and inverse folding results in the Cameo dataset. Given that our model is generative we can run it N times per sequence (or backbone) and get diverse results. We take advantage of this and our ability to predict a correct fold (or sequence) per PLL, to rank and select. Doing this we can increase our tm-score from 0.67 to 0.69 in forward folding (Table 3, Table S10). There is still room to grow as a perfect oracle would increase it to 0.75. For inverse folding we can increase the pass rate from 37.0 to 39.4, while a perfect oracle optimizing tm score would increase it to 52 (Table 1, Table S11). For unconditional generation we see a substantial improvements of pass rate from 75.8 to 84.8 for the valine logit-bias variant, at the cost of secondary structure (strand percentage goes from 7% to 3.6% (Table 5, Table S12).

PLL ranking transfers to the 4-modality protein-ligand model. The same PLL recipe extends to LeFlur-pl by scoring each of the four modalities (protein seq, protein struc, ligand atom, ligand struc) and combining them. the per-task PLL $\leftrightarrow$ quality Spearman correlations (analogous to Tables S7 and S8) are reported for the protein-ligand model in Tables S13 and S14. On the PoseBusters protein-ligand benchmark with N=30 candidates per target, the joint protein PLL ranker (`joint_protein`, sum of seq and struc PLL with the ligand held clean) significantly improves forward folding (Table 4): mean TM rises from 0.654 (`random_pick`, Table S15) to 0.755 (+0.101, paired

Table 5. Unconditional Generation Performance averaged across lengths 100–500. H/S/C are the helix, strand, and coil percentage of generated structures (DSSP). Suffix legend: `-val` = valine logit-bias variant (+1 on the valine logit for the first 25 sampling steps, see Appendix L); `PLL` = candidates ranked by joint pseudo-NLL; `sr8`/`sr9` = Self-Reflection with TM cutoff 0.833/0.9.

Method	Tokens	Pass (%)	Avg TM	H/S/C (%)	Clusters
P(all-atom)*		37.9		56/17/27	134
Protpardelle-1c*		44.8		63/14/23	41
APM*		32.2		73/8/19	32
PLAID*		19.2		44/14/42	25
ProteinGenerator*		17.8		78/5/17	12
Protpardelle*		35.2		65/14/21	10
Genie2 + ProteinMPNN		52.0	0.808	67.6/5.5/26.9	223
Proteina + ProteinMPNN		66.4	0.909	55.6/11.7/32.8	230
La Proteina 650M		79.6	0.922	70.7/6.2/23.1	169
DPLM-2 650M	8192	60.4	0.85	38.2/**17.2**/**44.6**	141
LeFlur-p 470M SLQ	256	84.8	0.937	**84.3**/0.2/15.5	301
LeFlur-p-val 470M SLQ	256	75.8	0.895	69.1/7.0/24.0	288
LeFlur-p-val PLL 470M SLQ	256	84.8	0.887	76.8/3.6/19.6	310
LeFlur-p-val-sr8 470M SLQ	256	81.8	0.937	74.2/6.1/19.7	310
LeFlur-p-val-sr9 470M SLQ	256	**85.2**	**0.938**	74.8/5.9/19.3	**312**

* Numbers baseline obtained from (Didi et al., 2025).

Table 6. Ligand-conditioned generation. Length 100, 100 designs per ligand, ligands: IAI, FAD, SAM, OQO. A design passes if Boltz (Passaro et al., 2025) predicts ipTM ≥ 0.9 *and* iPDE ≤ 1.0.

Method	Tokens	Pass (%)	ipTM	iPDE	unique ligands
Proteina Complexa 650M		**7.2**	**0.750**	**1.669**	**4**
LeFlur-pl 470M	4375	2	0.678	2.431	2

Wilcoxon $p < 10^{-4}$). The same ranker raises GF+IP from 24.4% to 28.5%, but the GF+IP oracle reaches 56.9%. For protein-ligand inverse folding (Table 2, Table S16), evaluated by Boltz-2 cofolds of the predicted sequences), our results are not significant, suggesting our previous correlation signals indicate the difficulty in designing a sequence for that target, but not being able to find a sequence for that target. On ligand-conditioned generation using the PoseBusters ligands (Table S17), the joint-true-4 pseudo-NLL ranker more than doubles the strict ipTM $\geq$ 0.9 *and* iPDE $\leq$ 1 pass-rate from 6.4% (`random_pick`) to 15.2% (McNemar p=0.019), while the per-target Boltz-2 oracle reaches 32.8%.

Self Reflection. Recent advances in Large Language Models (LLMs) have demonstrated that self-verification significantly improves reasoning capabilities (Weng et al., 2023). We extend this paradigm to *de novo* protein design. However, unlike standard LLM tasks which are conditional (Given Question → Generate Answer), our setting is *unconditional*: we jointly generate both the amino acid sequence and the 3D structure. We propose a verification mechanism based on *Mutual Information Maximization*. By enforcing that the generated sequence must accurately predict the generated structure (folding), we minimize the conditional entropy of the pair.

Let $\mathcal{M}_\theta$ be a unified probabilistic model defined over the joint space of protein sequences $\mathbf{X}$ and structures $\mathbf{Y}$. Our objective is to sample valid protein pairs (X, Y) from the joint distribution $P_\theta(X, Y)$ such that the pair is physically realizable: the sequence X must intrinsically encode the structure Y as its global free energy minimum. We posit that the physical validity of a generated pair is quantifiable by the *Pointwise Mutual Information (PMI)* between the sequence and structure tokens. We seek to maximize:

$$I(X;Y) = \mathbb{E}_{P(X,Y)}\left[\log \frac{P(X,Y)}{P(X)P(Y)}\right] = H(Y) - H(Y|X) \tag{6}$$

where $H(Y)$ is the marginal entropy (structural diversity) and $H(Y|X)$ is the conditional entropy (folding uncertainty).

To maximize $I(X;Y)$ while maintaining diversity (high $H(Y)$), we must minimize the conditional entropy $H(Y|X)$. In practice we operationalise this with a forward-folding self-consistency check: given the generated sequence $\hat{x}$, we re-sample a structure $\tilde{y} \sim P_\theta(\cdot \mid \hat{x})$ from the model's own conditional distribution and accept the pair

$(\hat{x}, \hat{y})$ if and only if the re-folded structure agrees with the originally generated structure within a TM-score cutoff τ_{TM}:

$$\text{TM}(\hat{y}, \tilde{y}) \geq \tau_{\text{TM}}, \qquad \tilde{y} \sim P_\theta(y \mid \hat{x}). \tag{7}$$

A high TM-score implies that $\hat{x}$ approximately determines $\hat{y}$ under P_θ, i.e. that $\hat{x}$ acts as a *sufficient statistic* for $\hat{y}$ and that $H(Y \mid X)$ is small for this pair.

Self Reflection Boosts Performance where model struggles. To further motivate the use of LeFlur as a verifier we confirm that there is a positive correlation between the model's predictions and ESMFold (Fig. S2). We then run self reflection on LeFlur-p and LeFlur-p-val using a forward-folding TM-score cutoff τ_{TM} of 0.833 (denoted SR8) and 0.9 (denoted SR9). We find self reflection is particularly helpful in the high length ranges of LeFlur-p-val where it struggles relative to LeFlur-p, and is negligible elsewhere (Fig. S3). As demonstrated in Table 5, we can take LeFlur-p-val and apply self reflection to length 400 and 500 and increase the pass rate from 75.8 to 85.2, while also increasing the number of clusters from 288 to 312. We also apply it to inverse folding and boost the pass rate from 37.0% to 40.2%.

5. Conclusion

We introduced **LeFlur**, a unified generative framework built on the **LatentGenerator**, a universal tokeniser that compresses proteins, small molecules, and protein-ligand complexes into a shared discrete vocabulary while replacing equivariant backbones with standard transformers and $SE(3)$ data augmentation, achieving reconstruction RMSDs of 1.26 Å on proteins, 0.29 Å on small molecules, and 1.00 Å on protein-ligand complexes. Discretisation lets us apply optimized text transformers directly: LeFlur-p matches or exceeds protein-only baselines on inverse folding (53.60% AAR), forward folding (0.76 TM), and unconditional generation (84.8% pass), and LeFlur-pl extends the same recipe to protein-ligand inverse folding (74.80% pocket AAR), forward folding (0.70 TM, 26.4% correct pocket placement), and ligand-conditioned generation. Because the model operates in discrete space, pseudo-likelihoods and self-consistency act as built-in verifiers: PLL ranking lifts CAMEO forward-fold TM from 0.67 to 0.69 and valine biased unconditional pass-rate from 75.8% to 84.8%. We can further easily extend PLL ranking to protein ligands, for instance we lift ligand conditioned forward folding from 0.703 TM and 26.4% correct pocket placement to 0.755 TM and 28.5% correct pocket placement. Self-Reflection raises valine biased unconditional pass-rate from 75.8% to 85.2% and CAMEO inverse folding pass rate from 37.0% to 40.2% without an external oracle. LeFlur thus offers a single tokenised latent space for joint sequence-structure design, with natural extensions to protein-DNA and protein-protein complexes. Furthermore because we model atomic systems within a discrete token space, these tokens can be fed directly into standard text only LLMs allowing them to reason about bimolecular structure, bestowing them the capability to be natural language conditioned bimolecular design models. Future work will seek to investigate how to best incorporate structure tokens into frontier LLM training.

Limitations. LeFlur-pl predicts the protein backbone of a complex accurately but does not co-localise the ligand as well as specialised co-folders (GF+IP 26.4% vs. 49.6% for Boltz-2; Table 4), and ligand-conditioned design success (2%) trails Proteina-Complexa (Didi et al., 2026) (7.2%, Table 6). The small-codebook SLQ regime does not scale to protein-ligand vocabularies (SLQ-4,096 degrades complex RMSD to 4.680 Å; Table S3), so we use FSQ for LeFlur-pl; even so, ligand outputs require a constrained MMFF94 relaxation to reach sub-0.3 Å geometry (0.395 → 0.291 Å; Table S2).

Acknowledgements

We thank Nathan Frey, Hannah Lawrence, Patricia Suriana, Zach Wang, and Jacob Gershon for useful discussions. We thank the Prescient Design engineering team for infra and training support.

Impact Statement

LeFlur is a generative model for joint sequence-structure design of proteins and protein-ligand complexes. By unifying inverse folding, forward folding, and unconditional/ligand-conditioned generation under a single set of weights, LeFlur lowers the engineering and compute barriers to running protein-design pipelines that previously required orchestrating multiple specialised models (Watson et al., 2023; Pacesa et al., 2025; Stark et al., 2025). Its built-in pseudo-likelihood and Self-Reflection verifiers further reduce reliance on heavy external structure-prediction oracles. These properties may help accelerate research in protein engineering, enzyme design, antibody and small-molecule binder discovery, biosensor development, and basic structural biology, particularly in academic and resource-constrained settings. Like other generative biomolecular design tools, LeFlur could in principle be misused to design biomolecules with harmful function. We mitigate this risk by training only on public protein-monomer and small-molecule complex data (PDB/OpenProteinSet, AFDB SwissProt, GEOM, PDBbind, PLINDER, SAIR; see Appendix H) with no enrichment for hazardous targets, and discuss broader impact considerations in detail in Appendix O.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024. doi: 10.1038/s41586-024-07487-w.

Ahdritz, G. et al. Openproteinset: Training data for structural biology at scale. *arXiv preprint arXiv:2308.05326*, 2023.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2107.03006.

Axelrod, S. and Gomez-Bombarelli, R. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The Protein Data Bank. *Nucleic Acids Research*, 28 (1):235–242, 2000. doi: 10.1093/nar/28.1.235. URL https://doi.org/10.1093/nar/28.1.235.

Breton, L. L., Fournier, Q., Mezouar, M. E., Morris, J. X., and Chandar, S. Neobert: A next-generation bert, 2025. URL https://arxiv.org/abs/2502.19587.

Buttenschoen, M., Morris, G. M., and Deane, C. M. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024. doi: 10.1039/D3SC04185A.

Campbell, A., Yim, J., Barzilay, R., Rainforth, T., and Jaakkola, T. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2402.04997.

Chen, R., Xue, D., Zhou, X., Zheng, Z., Zeng, X., and Gu, Q. An all-atom generative model for designing protein complexes. In *International Conference on Machine Learning (ICML)*, 2025. URL https://openreview.net/forum?id=1bf7d7d488357096aa06035954c6ec6a70a09ce2.

Cho, Y., Pacesa, M., Zhang, Z., Correia, B. E., and Ovchinnikov, S. Boltzdesign1: Inverting all-atom structure prediction model for generalized biomolecular binder design. *bioRxiv*, 2025. doi: 10.1101/2025.04.06.647261. URL https://www.biorxiv.org/content/10.1101/2025.04.06.647261v1.

Corley, N., Mathis, S., Krishna, R., Bauer, M. S., Thompson, T. R., Ahern, W., Kazman, M. W., Brent, R. I., Didi, K., Kubaney, A., McHugh, L., Nagle, A., Baker, D., DiMaio, F., et al. Accelerating biomolecular modeling with AtomWorks and RF3. *bioRxiv*, 2025. doi: 10.1101/2025.08.14.670328. URL https://www.biorxiv.org/content/10.1101/2025.08.14.670328v2.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378 (6615):49–56, 2022. doi: 10.1126/science.add2187.

Dauparas, J., Lee, G. R., Pecoraro, R., et al. Atomic context-conditioned protein sequence design using LigandMPNN. *Nature Methods*, 22(4):717–723, 2025. doi: 10.1038/s41592-025-02626-1.

Didi, K., Jendrusch, M., Terayama, K., Schölkopf, B., and Steinegger, M. La-proteina: Atomistic protein generation via partially latent flow matching. *arXiv preprint arXiv:2507.09466*, 2025. URL https://arxiv.org/abs/2507.09466.

Didi, K., Zhang, Z., Zhou, G., Reidenbach, D., Cao, Z., Cha, S., Geffner, T., Dallago, C., Tang, J., Bronstein, M. M., Steinegger, M., Kucukbenli, E., Vahdat, A., and Kreis, K. Scaling atomistic protein binder design with generative pretraining and test-time compute. *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=qmCpJtFZra. Oral Presentation.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Durairaj, J., Adeshina, Y., Cao, Z., Zhang, X., Oleinikovas, V., Duignan, T., McClure, Z., Robin, X., Studer, G., Kovtun, D., Rossi, E., Zhou, G., Veccham, S., Isert, C., Peng, Y., Sundareson, P., Akdel, M., Corso, G., Stärk, H., Tauriello, G., Carpenter, Z., Bronstein, M., Kucukbenli, E., Schwede, T., and Naef, L. PLINDER: The protein-ligand interactions dataset and evaluation resource. *bioRxiv*, pp. 2024–07, 2024. doi: 10.1101/2024.07.17.603955. URL https://www.biorxiv.org/content/10.1101/2024.07.17.603955v3.

Gat, I., Remez, T., Shaul, N., Kreuk, F., Chen, R. T. Q., Synnaeve, G., Adi, Y., and Lipman, Y. Discrete flow

matching, 2024. URL https://arxiv.org/abs/2407.15595.

Geffner, T., Didi, K., Zhang, Z., Reidenbach, D., Cao, Z., Yim, J., Geiger, M., Dallago, C., Kucukbenli, E., Vahdat, A., and Kreis, K. Proteina: Scaling flow-based protein structure generative models. *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=TVQLu34bdw. Oral Presentation.

Haas, J., Barbato, A., Behringer, D., Studer, G., Roth, S., Bertoni, M., Mostaguir, K., Gumienny, R., and Schwede, T. Continuous automated model evaluation (CAMEO) complementing the critical assessment of structure prediction in CASP12. *Proteins: Structure, Function, and Bioinformatics*, 86(S1):387–398, 2018. doi: 10.1002/prot.25431.

Hayes, T., Rao, R., Akin, H., Sofroniew, N. J., Oktay, D., Lin, Z., Verkuil, R., Tran, V. Q., Deaton, J., Wiggert, M., Badkundri, R., Shafkat, I., Gong, J., Derry, A., Molina, R. S., Thomas, N., Khan, Y. A., Mishra, C., Kim, C., Bartie, L. J., Nemeth, M., Hsu, P. D., Sercu, T., Candido, S., and Rives, A. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018. URL https://www.science.org/doi/10.1126/science.ads0018.

Heinzinger, M., Weissenow, K., Sanchez, J. G., Henkel, A., Mirdita, M., Steinegger, M., and Rost, B. Bilingual language model for protein sequence and structure. *NAR Genomics and Bioinformatics*, 6(4):lqae150, 2024. doi: 10.1093/nargab/lqae150.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1611.01144.

Jendrusch, M., Didi, K., Schölkopf, B., and Steinegger, M. Scaling atomistic protein binder design with generative pretraining and test-time compute. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=qmCpJtFZra.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. doi: 10.1107/S0567739476001873.

Kovtun, D., Akdel, M., Goncearenco, A., Zhou, G., Holt, G., Baugher, D., Lin, D., Adeshina, Y., Castiglione, T., Wang, X., Marquet, C., McPartlon, M., Geffner, T., Rossi, E., Corso, G., Stärk, H., Carpenter, Z., Kucukbenli, E., Bronstein, M., and Naef, L. Pinder: The protein interaction dataset and evaluation resource. *bioRxiv*, 2024. doi: 10.1101/2024.07.17.603980. URL https://www.biorxiv.org/content/10.1101/2024.07.17.603980v4.

Lau, A. M., Bordin, N., Kandathil, S. M., Sillitoe, I., Waman, V. P., Wells, J., Orengo, C. A., and Jones, D. T. Exploring structural diversity across the protein universe with the encyclopedia of domains. *Science*, 386(6721):eadq4946, 2024. doi: 10.1126/science.adq4946. URL https://www.science.org/doi/10.1126/science.adq4946.

Lemos, P., Beckwith, Z., Bandi, S., van Damme, M., Crivelli-Decker, J., Shields, B. J., Merth, T., Jha, P. K., De Mitri, N., Callahan, T. J., et al. Sair: Enabling deep learning for protein-ligand interactions with a synthetic structural dataset. *bioRxiv*, pp. 2025.06.17.660168, 2025.

Lin, Y., Lee, M., Zhang, Z., and AlQuraishi, M. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with Genie 2. *arXiv preprint arXiv:2405.15489*, 2024. URL https://arxiv.org/abs/2405.15489.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/10.1126/science.ade2574.

Lisanza, S. L., Gershon, J. M., Tipps, S. W. K., Sims, J. N., Arnoldt, L., Hendel, S. J., Simma, M. K., Liu, G., Yase, M., Wu, H., et al. Multistate and functional protein design using rosettafold sequence space diffusion. *Nature Biotechnology*, 2024. doi: 10.1038/s41587-024-02395-w.

Liu, Z., Li, Y., Han, L., Li, J., Liu, J., Zhao, Z., Nie, W., Liu, Y., and Wang, R. Pdb-wide collection of binding data: current status of the pdbbind database. *Bioinformatics*, 31(3):405–412, 2015.

Lu, A. X., Zhang, H., Ghassemi, M., and Moses, A. Self-supervised contrastive learning of protein representations

by mutual information maximization. In *Machine Learning for Computational Biology Workshop, NeurIPS*, 2020. URL https://www.biorxiv.org/content/10.1101/2020.09.04.283929v1.

Lu, A. X., Yang, J., Long, Y., Lambert, T. P., Yan, W., Robinson, S. A., Yang, K. K., Gligorijevic, V., Cho, K., Bonneau, R., Frey, N. C., Arnold, F., and Abbeel, P. Controllable all-atom protein generation with latent diffusion. *bioRxiv*, 2024. doi: 10.1101/2024.12.02.626353.

Mentzer, F., Minnen, D., Agustsson, E., and Tschannen, M. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023. URL https://arxiv.org/abs/2309.15505.

O'Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., and Hutchison, G. R. Open babel: An open chemical toolbox. *Journal of Cheminformatics*, 3(1):33, Oct 2011. ISSN 1758-2946. doi: 10.1186/1758-2946-3-33. URL https://doi.org/10.1186/1758-2946-3-33.

Pacesa, M., Nickel, L., Schellhaas, C., et al. One-shot design of functional protein binders with bindcraft. *Nature*, 646(8084):483–492, Oct 2025. doi: 10.1038/s41586-025-09429-6. URL https://www.nature.com/articles/s41586-025-09429-6.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., Kwabi-Addo, D., Beaini, D., Jaakkola, T., and Barzilay, R. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, pp. 2025–06, 2025. doi: 10.1101/2025.06.14.659707. URL https://www.biorxiv.org/content/10.1101/2025.06.14.659707v1.

Qu, W., Guan, J., Ma, R., Zhai, K., Wu, W., and Wang, H. P(all-atom) is unlocking new path for protein design. *bioRxiv*, 2024. doi: 10.1101/2024.08.16.608235.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.07524.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2303.11366.

Shuai, R. W., Lu, T., Bhatti, S., Kouba, P., and Huang, P.-S. Ensemble-conditioned protein sequence design with Caliby. *bioRxiv*, 2025a. doi: 10.1101/2025.09.30.679633. URL https://www.biorxiv.org/content/10.1101/2025.09.30.679633v4.

Shuai, R. W., Lu, T., Bhatti, S., Kouba, P., and Huang, P.-S. Ensemble-conditioned protein sequence design with Caliby. *bioRxiv*, pp. 2025–09, 2025b. doi: 10.1101/2025.09.30.679633. URL https://www.biorxiv.org/content/10.1101/2025.09.30.679633v4.

Sillitoe, I., Bordin, N., Dawson, N., Waman, V. P., Ashford, P., Scholes, H. M., Pang, C. S. M., Woodridge, L., Rauer, C., Sen, N., Abbasian, M., Le Cornu, S., Lam, S. D., Berka, K., Hutařová Varekova, I., Svobodova, R., Lees, J., and Orengo, C. A. CATH: increased structural coverage of functional space. *Nucleic Acids Research*, 49(D1):D266–D273, 2021. doi: 10.1093/nar/gkaa1079. URL https://academic.oup.com/nar/article/49/D1/D266/6006195.

Stark, H., Faltings, F., Choi, M., Xie, Y., Hur, E., O'Donnell, T., Bushuiev, A., Uçar, T., Passaro, S., Mao, W., et al. Boltzgen: Toward universal binder design. *bioRxiv*, 2025. doi: 10.1101/2025.11.20.689494. URL https://www.biorxiv.org/content/10.1101/2025.11.20.689494v1.

Su, J., Han, C., Zhou, Y., Shan, J., Zhou, X., and Yuan, F. Saprot: Protein language modeling with structure-aware vocabulary. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=66S78967S0.

Uria, B., Murray, I., and Larochelle, H. A deep and tractable density estimator. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014. URL https://arxiv.org/abs/1310.1757.

van Kempen, M., Kim, S. S., Tumescheit, C., Mirdita, M., Lee, J., Gilchrist, C. L. M., Söding, J., and Steinegger, M. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, 42(2):243–246, 2024a. doi: 10.1038/s41587-023-01773-0.

van Kempen, M., Kim, S. S., Tumescheit, C., Mirdita, M., Lee, J., Gilchrist, C. L. M., Söding, J., and Steinegger, M. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42(2):243–246, 2024b. doi: 10.1038/s41587-023-01773-0.

Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., Žídek, A., Green, T., Tunyasuvunakool, K., Petersen, S., Jumper, J., Clancy, E., Green, R., Vora, A., Lutfi, M., Figurnov, M., Cowie, A., Hobbs, N., Kohli, P., Kleywegt, G., Birney, E., Hassabis, D., and Velankar, S. AlphaFold Protein Structure Database:

massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50(D1):D439–D444, 11 2021. ISSN 0305-1048. doi: 10.1093/nar/gkab1061. URL https://doi.org/10.1093/nar/gkab1061.

Wang, X., Zheng, Z., Ye, F., Xue, D., Huang, S., and Gu, Q. Dplm-2: A multimodal diffusion protein language model. *arXiv preprint arXiv:2410.13782*, 2024. URL https://arxiv.org/abs/2410.13782.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976): 1089–1100, 2023. doi: 10.1038/s41586-023-06415-8.

Weng, Y., Zhu, M., Xia, F., Li, B., He, S., Liu, K., and Zhao, J. Large language models are better reasoners with self-verification. *arXiv preprint arXiv:2212.09561*, 2023.

Zhang, Y. and Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004. doi: 10.1002/prot.20264.

Supplementary Information

A. LatentGenerator Further details

To ensure that our discrete token reconstruction capabilities are invariant to the global orientation of the input molecule, we perform data augmentation during training. For every training sample, we apply a random rigid transformation sampled from the special Euclidean group $SE(3)$, defined as $\mathbf{Y}' = \mathbf{R}\mathbf{Y} + \mathbf{t}$, where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$. The encoder–decoder is then trained to reconstruct the *internal/relative geometry* of the original structure rather than its absolute coordinates: the predicted coordinates $\hat{\mathbf{Y}}$ live in an arbitrary global frame, and the L^2 reconstruction loss (Eq. 1) compares them to the ground truth only after a Kabsch rigid alignment. The pairwise-distance loss (Eq. 2) is itself $SE(3)$-invariant. This way the model learns reconstruction equivariance up to a global rigid transformation without requiring specialized equivariant architectures.

Encoder. Given protein backbone coordinates $\mathbf{Y}_p \in \mathbb{R}^{L_p \times 3 \times 3}$ (representing N, C_α, and C atoms per residue) and a residue mask $\mathbf{m}_p \in \{0,1\}^{L_p}$, we form two complementary inputs: a per-residue patch embedding $\mathbf{P}_p \in \mathbb{R}^{L_p \times d}$ (a linear projection of the flattened nine backbone coordinates per residue) and a pairwise inter-atomic distance tensor $\mathbf{D}_p \in \mathbb{R}^{L_p \times L_p \times 9}$ (the 3×3 atom-atom distance matrix between residue pairs). The pairwise tensor is reduced along its second axis via a learned MLP followed by mean-pooling over the L_p context dimension to yield a 1D summary $\bar{\mathbf{D}}_p \in \mathbb{R}^{L_p \times d}$, which is concatenated with $\mathbf{P}_p$ and projected back to dimension d by an MLP, giving $\mathbf{Z}_p \in \mathbb{R}^{L_p \times d}$. These are then processed into the protein latent representation $\mathbf{Z} = \mathcal{E}_\theta(\mathbf{Z}_p, \mathbf{m}_p) \in \mathbb{R}^{L_p \times d}$, where d is the latent dimension. For the protein-ligand modality, ligand atom coordinates $\mathbf{Y}_\ell \in \mathbb{R}^{L_\ell \times 3}$ and an atom mask $\mathbf{m}_\ell \in \{0,1\}^{L_\ell}$ are jointly encoded by first embedding the ligand as $\mathbf{Z}_\ell = \text{Embed}(\mathbf{Y}_\ell)$ and then computing $\mathbf{Z} = \mathcal{E}_\theta([\mathbf{Z}_p; \mathbf{Z}_\ell], [\mathbf{m}_p; \mathbf{m}_\ell])$, where $[\cdot;\cdot]$ represents concatenation along the sequence dimension.

Quantizer. After joint encoding, the combined embeddings are split and quantized separately to account for the distinct structural characteristics of each molecular type. Specifically, we partition the joint latent representation into protein and ligand components. These are then mapped to their discrete representations via modality-specific quantization functions, yielding $\mathbf{C}_p = \mathcal{Q}_\phi^{(p)}(\mathbf{Z}_p, \mathbf{m}_p)$ for the protein and $\mathbf{C}_\ell = \mathcal{Q}_\phi^{(\ell)}(\mathbf{Z}_\ell, \mathbf{m}_\ell)$ for the ligand.

Decoder. The decoder mirrors the encoder architecture, tasked with reconstructing the original Cartesian coordinates from the quantized tokens. Given the discrete representations $\mathbf{C}_p$ and $\mathbf{C}_\ell$, the decoder $\mathcal{D}_\psi$ predicts the reconstructed protein backbone coordinates $\hat{\mathbf{Y}}_p = \mathcal{D}_\psi(\mathbf{C}_p, \mathbf{m}_p) \in \mathbb{R}^{L_p \times 3 \times 3}$ and the ligand atom coordinates $\hat{\mathbf{Y}}_\ell = \mathcal{D}_\psi(\mathbf{C}_\ell, \mathbf{m}_\ell) \in \mathbb{R}^{L_\ell \times 3}$. This architecture ensures that the structural information is effectively bottlenecked within the discrete latent space before reconstruction.

B. Quantization with Finite Scalar Quantization (FSQ)

FSQ (Mentzer et al., 2023) provides an alternative codebook-free quantization approach by discretizing each dimension independently using fixed scalar levels; schematics comparing this to SLQ are provided in Fig S9. We first define a vector of levels $\mathbf{L} = (L_1, \ldots, L_D)$, which determines an implicit codebook size $K = \prod_{d=1}^{D} L_d$. For example, a level configuration of $\mathbf{L} = (8, 6, 5)$ yields $K = 240$ discrete tokens. To quantize the latent representation, the encoder embeddings are projected to D dimensions, $\mathbf{z} \in \mathbb{R}^D$, and subjected to a three-step transformation. First, we bound the continuous values using a scaled tanh function, $\tilde{z}_d = \frac{L_d - 1}{2} \cdot \tanh(z_d + \delta_d) - o_d$, where δ_d and o_d are offsets that handle even or odd level counts. Second, we round these values using a straight-through estimator to maintain gradient flow, $\hat{z}_d = \tilde{z}_d + \text{sg}(\lfloor \tilde{z}_d \rceil - \tilde{z}_d)$, where sg denotes the stop-gradient operator (so the forward pass evaluates to the rounded scaled projection $\lfloor \tilde{z}_d \rceil$ while gradients flow through $\tilde{z}_d$). Finally, the values are normalized to the range $[-1, 1]$ via $\bar{z}_d = \hat{z}_d / (L_d // 2)$. This process ensures a fixed, deterministic discretization without the need for learned codebook embeddings.

C. Constrained Geometrical Refinement

To ensure optimal structural fidelity, ligand molecular geometries were refined after decoding using a constrained force field minimization protocol within the Open Babel framework (O'Boyle et al., 2011). Potential energy minimization was performed using the MMFF94 force field coupled with a robust set of geometric constraints applied via a harmonic penalty function with a scaling factor of $k = 10^4$. Ideal bond lengths, d_{ij}, were calculated based on the sum of atomic covalent radii r_i and r_j, adjusted for bond order (BO) and aromaticity as follows:

$$d_{ij} = (r_i + r_j) \times \alpha(BO) \tag{8}$$

where the scaling factor α was assigned values of 1.00 for single bonds ($BO = 1$), 0.87 for double bonds ($BO = 2$), 0.78 for triple bonds ($BO = 3$), and 0.91 for aromatic bonds. Valence angles θ_{ijk} were constrained to values determined by the hybridization state (hyb) of the central vertex atom; for sp hybridization $\theta_{ideal} = 180°$, for sp^2 hybridization $\theta_{ideal} = 120°$, and for sp^3 hybridization $\theta_{ideal} = 109.47°$. Structural convergence was achieved using the Conjugate Gradients algorithm. This ensures that the local chemical geometry is idealized to high precision while the global conformation and binding pose—captured by the model's latent representation—remain largely unperturbed.

D. Reconstruction Tables

Table S1. **LatentGenerator Structure Reconstruction Performance.** Evaluated on CASP15 proteins ($\leq$ 512 residues). Token counts denote codebook sizes for protein and ligand components respectively. RMSD values are reported as Average $\pm$ Standard Deviation.

		Tokens		RMSD (Å)
Model	Quant.	Prot	Lig	Avg $\pm$ Std
LG Protein (cont.)	None			0.462 $\pm$ 0.322
LG Prot-Lig (cont.)	None			0.651 $\pm$ 0.339
LG Protein SLQ	SLQ	256		1.647 $\pm$ 0.535
LG Prot-Lig SLQ	SLQ	256	512	1.873 $\pm$ 1.054
LG Prot-Lig SLQ	SLQ	4096	4096	3.097 $\pm$ 2.009
LG Protein FSQ	FSQ	240		1.848 $\pm$ 1.194
LG Prot-Lig FSQ	FSQ	4375	4375	**1.260** $\pm$ **0.632**
LG Prot-Lig FSQ	FSQ	4375	15360	1.418 $\pm$ 0.810

Table S2. **LatentGenerator Ligand Reconstruction Performance.** Evaluated on ligand structures from the GEOM dataset. Token counts denote protein and ligand codebook sizes respectively. RMSD values are reported as Average $\pm$ Standard Deviation.

		Tokens		RMSD (Å)
Model	Quant.	Prot	Lig	Avg $\pm$ Std
LG Prot-Lig (cont.)	None			0.043 $\pm$ 0.011
LG Ligand SLQ	SLQ		512	0.752 $\pm$ 0.305
LG Prot-Lig SLQ	SLQ	256	512	0.920 $\pm$ 0.236
LG Prot-Lig SLQ	SLQ	4096	4096	1.239 $\pm$ 0.335
LG Prot-Lig FSQ	FSQ	4375	4375	0.395 $\pm$ 0.059
LG Prot-Lig FSQ	FSQ	4375	15360	0.295 $\pm$ 0.052
LG Prot-Lig FSQ (min.)	FSQ	4375	4375	**0.291** $\pm$ **0.056**

Table S3. Protein-Ligand Complex Reconstruction Performance on PDBbind. Token counts denote protein and ligand codebook sizes. RMSD values are reported as Average ± Standard Deviation.

		Tokens			RMSD (Å)
Model	Metric	Prot	Lig	Align	Avg ± Std
LG Prot-Lig (cont.)	Protein	–	–	Indiv.	0.496 ± 0.019
LG Prot-Lig SLQ	Protein	256	512	Indiv.	1.483 ± 0.232
LG Prot-Lig SLQ	Protein	4096	4096	Indiv.	4.740 ± 3.010
LG Prot-Lig FSQ	Protein	4375	4375	Indiv.	**1.008 ± 0.107**
LG Prot-Lig FSQ	Protein	4375	15360	Indiv.	1.010 ± 0.107
LG Prot-Lig (cont.)	Ligand	–	–	Indiv.	0.499 ± 0.046
LG Prot-Lig SLQ	Ligand	256	512	Indiv.	1.411 ± 0.593
LG Prot-Lig SLQ	Ligand	4096	4096	Indiv.	1.620 ± 0.711
LG Prot-Lig FSQ	Ligand	4375	4375	Indiv.	0.705 ± 0.139
LG Prot-Lig FSQ	Ligand	4375	15360	Indiv.	0.657 ± 0.146
LG Prot-Lig FSQ (min.)	Ligand	4375	4375	Indiv.	**0.468 ± 0.139**
LG Prot-Lig (cont.)	Complex	–	–	Joint	0.507 ± 0.019
LG Prot-Lig SLQ	Complex	256	512	Joint	1.567 ± 0.343
LG Prot-Lig SLQ	Complex	4096	4096	Joint	4.680 ± 2.962
LG Prot-Lig FSQ	Complex	4375	4375	Joint	1.011 ± 0.127
LG Prot-Lig FSQ	Complex	4375	15360	Joint	1.009 ± 0.138
LG Prot-Lig FSQ (min.)	Complex	4375	4375	Joint	**1.004 ± 0.119**
LG Prot-Lig (cont)	Protein	–	–	Joint (c)	0.496 ± 0.019
LG Prot-Lig SLQ	Protein	256	512	Joint (c)	1.507 ± 0.294
LG Prot-Lig SLQ	Protein	4096	4096	Joint (c)	4.761 ± 3.033
LG Prot-Lig FSQ	Protein	4375	4375	Joint (c)	**1.013 ± 0.116**
LG Prot-Lig FSQ	Protein	4375	15360	Joint (c)	1.017 ± 0.139
LG Prot-Lig (cont)	Ligand	–	–	Joint (c)	0.607 ± 0.103
LG Prot-Lig SLQ	Ligand	256	512	Joint (c)	2.306 ± 0.758
LG Prot-Lig SLQ	Ligand	4096	4096	Joint (c)	3.589 ± 2.151
LG Prot-Lig FSQ	Ligand	4375	4375	Joint (c)	1.011 ± 0.271
LG Prot-Lig FSQ	Ligand	4375	15360	Joint (c)	0.998 ± 0.299
LG Prot-Lig FSQ (min.)	Ligand	4375	4375	Joint (c)	**0.854 ± 0.290**

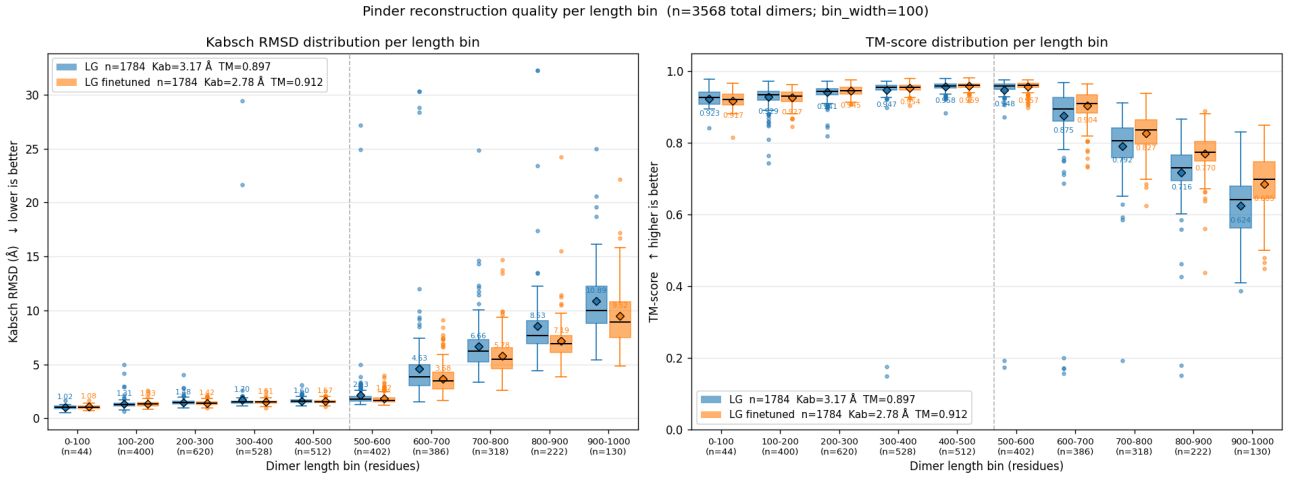


Figure S1. **LatentGenerator reconstructs protein complexes well past its training length cap, and decoder-only finetuning further closes the long-complex gap.** Kabsch RMSD (left) and TM-score (right) of LG codec reconstructions on $n = 1{,}784$ Pinder (Kovtun et al., 2024) test dimers, binned by total complex length (100-residue buckets); per-bin sample counts annotated under each x-tick. *LG* (blue) is the published *LG full attention* checkpoint trained at $\max L = 512$; *LG finetuned* (orange) is the same architecture finetuned at $\max L = 1024$ on Pinder dimers + PDB monomers. Boxes show the interquartile range; whiskers extend to $1.5\times$ IQR; fliers are individual dimers; diamond markers are per-bin means with numeric annotations. Dashed vertical line marks the original training cap at $L = 512$. Both models maintain near-codec-ceiling fidelity (TM ≥ 0.92, RMSD ≤ 2 Å) for $L \leq 500$, and degrade gracefully for longer complexes – LG sustains TM ≥ 0.62 at $L = 900$–1000 without ever having seen lengths beyond 512 in training. Finetuning at $\max L = 1024$ reduces the long-complex gap substantially: RMSD drops from $4.63 \rightarrow 3.68$ Å at 600–700 residues and from $10.89 \rightarrow 9.52$ Å at 900–1000, while TM rises from $0.875 \rightarrow 0.904$ and $0.624 \rightarrow 0.685$ respectively. Aggregate: LG Kab $= 3.17$ Å/ TM $= 0.897$ vs LG finetuned Kab $= 2.78$ Å/TM $= 0.912$.

E. 3Di-flow decoder reconstruction

We compare two decoders on protein-only reconstruction: (i) the LatentGenerator full-attention autoencoder, which encodes ground-truth coordinates into discrete SLQ-256 tokens and decodes them back, and (ii) a 3Di-conditioned flow-matching backbone decoder that reconstructs $N/C_\alpha/C$ coordinates from the Foldseek 3Di alphabet alone. Two test sets are used: the CASP15 monomers with $L \leq 512$ that we already report on in Table S1 (n=20 after dropping the T1137s1–s6 entries that were not compact enough leading to lever arm effects for the local environment centric 3di tokens), and the 474-protein ProstT5 test split (Heinzinger et al., 2024) (n=390 after restricting to $L \leq 512$ and excluding the AlphaFoldDB IDs that have been retired since the split was published; the paper's pLDDT $\geq$ 70 filter was applied at the original train/test split, so no further pLDDT filtering is needed). Sampling for the 3Di-flow decoder uses a 200-step power-exponent-2 SDE schedule with classifier-free guidance scale w=2. Reported numbers are Avg $\pm$ Std over the per-protein metrics; AAR is the per-residue 3Di token recovery (predicted backbone $\rightarrow$ 3Di (van Kempen et al., 2024b) $\rightarrow$ argmax) and TM is the TM-score from `tmtools` (Zhang & Skolnick, 2004).

Table S4. **LatentGenerator vs 3Di-flow reconstruction on CASP15 and the ProstT5 test split.** The LatentGenerator autoencoder sees ground-truth coordinates during encoding; the 3Di-flow decoder sees only the per-residue 3Di tokens. Token recovery (AAR) is the percentage of 3di-decoded predicted-coord tokens that match the GT 3Di tokens. Avg $\pm$ Std over chains.

Decoder	Test set	n	Kab RMSD (Å)	3Di AAR (%)	TM-score
LatentGenerator	CASP15 ($L \leq 512$)	20	**1.33** $\pm$ **0.25**	61.8 $\pm$ 11.8	**0.945** $\pm$ **0.016**
3Di-flow decoder	CASP15 ($L \leq 512$)	20	9.02 $\pm$ 6.51	**71.1** $\pm$ **11.3**	0.644 $\pm$ 0.195
LatentGenerator	ProstT5 test ($L \leq 512$)	390	**1.31** $\pm$ **1.06**	69.5 $\pm$ 11.6	**0.895** $\pm$ **0.059**
3Di-flow decoder	ProstT5 test ($L \leq 512$)	390	12.10 $\pm$ 13.24	**82.0** $\pm$ **8.8**	0.535 $\pm$ 0.162

The 3Di-flow decoder dominates on per-residue 3Di token recovery (its training objective directly optimizes 3di consistency), while the LatentGenerator autoencoder dominates on Kabsch RMSD and TM-score by an order of magnitude on Kab. We conclude that the Foldseek 3Di alphabet is sufficient to specify local geometry (the AAR signal) but underspecifies the global fold. A single 3Di string is consistent with many distinct backbone configurations, and the flow-matching decoder must commit to one of them with no extra information. LatentGenerator's 256-state codebook, learned jointly with its decoder, instead encodes the geometric content the model needs for high-fidelity reconstruction. We use this as evidence that pairing a frozen, off-the-shelf token alphabet with a downstream decoder is not enough on its own and that joint training of the tokenizer with its decoder remains necessary for the small-codebook regime targeted by LeFlur.

F. Campbell et al. ICML 2024 Benchmark Results

Table S5. Inverse Folding Performance on the Campbell et al. ICML 2024 Benchmark. Bold indicates best model for that column while underline means best token based model.

Model	Tokens	AAR (%)	TM	RMSD (Å)	Pass (%)	pLDDT
ProteinMPNN		46.54	**0.93**	**1.31**	**84.6**	**0.77**
DPLM-2 650M	8192	**55.56**	0.88	1.90	77.1	0.74
LeFlur-p 470M SLQ	256	53.60	0.85	2.15	71.1	0.71

Table S6. Forward Folding Performance on the Campbell et al. ICML 2024 Benchmark. Bold indicates best model for that column while underline means best token based model.

Model	Tokens	TM-Score	RMSD (Å)	Pass Rate (%)
ESMFold 3B		**0.91**	**2.88**	**64.6**
DPLM-2 650M	8192	0.77	5.29	24.9
LeFlurP 470M SLQ	256	0.76	6.41	28.7

G. Inference time scaling

Calculating PLL with sequence and structure. Leflur-p is trained as an absorbing-state discrete flow / any-order autoregressive (AO-ARM) model on the joint sequence–structure tokens $x = (x^{\text{seq}}, x^{\text{str}})$ of length L.[1] For each modality $m \in \{\text{seq}, \text{str}\}$, the corruption process masks each position independently with probability $1 - t$, producing $x_t^{(m)}$ with masked set $\mathcal{M}_t^{(m)}$. The standard AO-ARM identity (Austin et al., 2021; Uria et al., 2014; Sahoo et al., 2024) states that the expected denoising cross-entropy is the per-position negative log-likelihood of $x^{(m)}$ under the random-permutation factorisation, $\mathbb{E}_{t\sim U(0,1)}[\mathcal{L}_m(t)] = -L^{-1} \log p_\theta(x^{(m)})$, so we estimate the per-position NLL of a sample $(x^{\text{seq}}, x^{\text{str}})$ with K=32 stratified-t Monte-Carlo draws[2] as

$$\widehat{\text{PLL}}(x) \;=\; \sum_{m\in\{\text{seq},\,\text{str}\}} \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{M}_{t_k}^{(m)}|} \sum_{i\in\mathcal{M}_{t_k}^{(m)}} -\log p_\theta\Big(x_{0,i}^{(m)} \,\big|\, x_{t_k}^{(m)}\Big) \;\xrightarrow{K\to\infty}\; -\frac{\log p_\theta(x^{\text{seq}},\, x^{\text{str}})}{L}\,, \tag{9}$$

which we refer to as JOINT_SCORE_UNIF; the per-modality estimators SEQ_SCORE_UNIF and STRUC_SCORE_UNIF are defined by restricting the outer sum to m=seq and m=str respectively, and when scoring one modality the other is held clean (t=1). This makes the choice of head task-dependent: in *forward folding* (seq given) the sequence head sees a fixed input and contributes little selection signal, so STRUC_SCORE_UNIF dominates ($\rho = -0.77$ vs. TM, vs. $\rho = -0.60$ for the seq head); in *inverse folding* (str given) the structure head is fixed and the two heads carry complementary signal — SEQ_SCORE_UNIF is the strongest predictor of %-identity ($\rho = -0.80$) while JOINT_SCORE_UNIF is the most stable predictor of predicted-structure quality ($\rho = -0.67$ vs. pLDDT). In *unconditional* generation the signal is real but markedly weaker (best $|\rho| \approx 0.25$), reflecting the compressed dynamic range of Leflur-p's better-trained candidate distribution relative to its noisier early checkpoints; here JOINT_SCORE_UNIF is the most consistent ranker across pLDDT and RMSD, while STRUC_SCORE_UNIF alone is the best predictor of ESMFold TM-score.

Inference-time cost. Self-Reflection adds wall-clock cost on top of single-pass generation but stays entirely within LeFlur and avoids any external folding oracle. Our default configuration runs T=400 steps per generation, and for each that fails the TM cutoff, performs at most 30 generation and forward-folding retries (100 inference steps each for forward-folding) using LeFlur's own conditional structure sampler; in the worst case a single design therefore requires $30\times400 + 30\times100$ NeoBERT forward passes. In practice the retry budget is rarely exhausted because most designs that pass on the first attempt incur no retries at all, so the effective per-design cost is much closer to the single-pass figure than to this worst case. The PLL-based best-of-N ranker is the more reasonable in worst case: if we use best-of-N=30 each candidate needs K=32 stratified Monte-Carlo draws (Appendix G), so would require $30\times400 + 30\times32$ NeoBERT forward passes. We emphasise that all of this compute is forward-only, batched, and shares its weights with generation; it requires no separate folding (and inverse folding) model and replaces the standard RFdiffusion-then-ProteinMPNN-then-AlphaFold pipeline (Watson et al., 2023). Our verification approach is conceptually closest to verbal-self-correction methods in language models such as Reflexion (Shinn et al., 2023) and self-verification reasoners (Weng et al., 2023), recast as a check on $H(Y \mid X)$ in a multimodal sequence-structure model rather than on free-form text, and to mutual-information-based representation-learning objectives (Lu et al., 2020) adapted here to the generative setting. Using a model-internal confidence signal as a quality filter is also the design choice taken by the AlphaFold 2 and AlphaFold 3 confidence modules (pLDDT/pTM/PAE in Jumper et al. (2021) and Abramson et al. (2024)).

Protein-only best-of-N=30 per-picker breakdown. The N30 `PLL` and `oracle` rows reported in the main-body protein-only Tables 3, 1, and 5 are best-of-N=30 results for forward folding, inverse folding, and unconditional generation. Tables S10, S11, and S12 expand each row with the full per-picker breakdown in the same formulation as Tables S15, S16, and S17: the `random_pick` single-uniform-draw reference, every PLL variant available for the task, and the per-target oracles. We see strong selection and significance from FF and UG, but not for IF, suggesting our previous correlation signals indicates the difficulty in designing a sequence for that target, but not being able to find a sequence for that target.

[1] Notation: in this appendix we use the AO-ARM convention where x_t denotes the noisy/partially masked sample at time t and x_0 denotes the clean target. This corresponds to the discrete-flow-matching convention in Section 4.2 (where $\mathbf{x}_1$ is the clean data and $\mathbf{x}_t$ is noisy) under the identification $x_0^{\text{AO}} \leftrightarrow \mathbf{x}_1^{\text{DFM}}$.

[2] Stratified t: K equal-width bins on $(\epsilon, 1-\epsilon)$ with one uniform draw per bin; this removes the dominant t-selection variance that otherwise overwhelms the estimator at small K.

Table S7. Forward folding (CAMEO, n=127). Spearman ρ vs. external quality metric. Lower PLL should imply higher quality, so $\rho \leq 0$ is desired; bold marks the strongest predictor in each column.

PLL variant	vs. TM-score	vs. RMSD
SEQ_SCORE_UNIF	-0.601	$+0.583$
STRUC_SCORE_UNIF	$\mathbf{-0.765}$	$\mathbf{+0.725}$
JOINT_SCORE_UNIF	-0.676	$+0.646$
JOINT_TRUE_SCORE_UNIF	-0.684	$+0.639$

Table S8. Inverse folding (CAMEO, n=127). Spearman ρ vs. %-identity, ESMFold pLDDT, ESMFold TM-score, and ESMFold RMSD.

PLL variant	vs. %identity	vs. pLDDT	vs. ESM-TM	vs. ESM-RMSD
SEQ_SCORE_UNIF	$\mathbf{-0.803}$	-0.588	-0.498	$+0.418$
STRUC_SCORE_UNIF	-0.513	-0.589	-0.600	$+0.640$
JOINT_SCORE_UNIF	-0.742	-0.671	-0.638	$+0.618$
JOINT_TRUE_SCORE_UNIF	-0.676	$\mathbf{-0.758}$	$\mathbf{-0.672}$	$\mathbf{+0.640}$

Calculating PLL with protein and ligand tokens. LeFlur-pl extends the protein-only AO-ARM construction in Eq. 9 to a complex represented by four token streams: protein sequence x^{seq} and protein structure x^{str} as before, plus ligand atom-type tokens $x^{\text{lig-atom}}$ and ligand structure tokens $x^{\text{lig-str}}$. Training masks all four modalities under a shared $t \sim U(0,1)$, so the same AO-ARM identity holds over the joint state $x = (x^{\text{seq}}, x^{\text{str}}, x^{\text{lig-atom}}, x^{\text{lig-str}})$:

$$-\frac{1}{L}\log p_\theta(x) \;=\; \mathbb{E}_{t\sim U(0,1)} \sum_{m\in\mathcal{M}} \mathcal{L}_m(t), \qquad \mathcal{M} = \{\text{seq}, \text{str}, \text{lig-atom}, \text{lig-str}\}. \tag{10}$$

Each per-modality term is estimated with the same $K = 32$ stratified-t Monte-Carlo draws as in Eq. 9, restricted to the masked positions of that modality. The 8 PLL variants reported in Tables S13–S14 and used by the pickers in Tables S15, S16, and S17 are:

- **SEQ, STRUC, LIG_ATOM, LIG_STRUC**: The four single-modality scores, computed by restricting Eq. 10 to one modality and holding the other three clean ($t = 1$).
- **JOINT_PROTEIN = SEQ + STRUC**: The protein-only sum with the ligand held clean. Mirrors LeFlur-p's JOINT_SCORE_UNIF.
- **JOINT_LIGAND = LIG_ATOM + LIG_STRUC**: The ligand-only sum with the protein held clean.
- **JOINT_ALL = SEQ + STRUC + LIG_ATOM + LIG_STRUC**: The additive sum of all four single-modality scores. Each summand is computed with the other three modalities held clean; this is convenient and modality-decomposable but is *not* an unbiased estimator of the 4-modality joint PLL.
- **JOINT_TRUE_4**: Applies Eq. 10 directly—one shared t draw masks all four modalities simultaneously, and the score is averaged over the union of masked positions across modalities. This is the unbiased $K \to \infty$ estimator of the 4-modality joint $-\log p_\theta(x)/L$.

The `_unif` suffix everywhere denotes the uniform-t stratified estimator described above; we use `_unif` throughout the paper because the AO-ARM identity holds in expectation under uniform t. As in the protein-only setting, lower PLL should imply higher quality, so $\rho \leq 0$ is the desired sign for "higher-is-better" metrics (TM, AAR, ligand-ipTM, GF+IP) and $\rho \geq 0$ for "lower-is-better" metrics (RMSD, iPDE, ligand-RMSD, ligand-centroid distance, ligand-pocket min. distance).

Protein-ligand pseudo-NLL correlations. Tables S13 and S14 report the protein-ligand analogues of Tables S7 and S8: Spearman ρ between each LeFlur-pl pseudo-NLL variant (K=32 stratified-t Monte-Carlo draws) and the corresponding external quality metric on PoseBusters (n=123 FF / n=125 IF, CG; $L \leq 512$ filter). As in the protein-only setting, lower PLL should imply higher quality, so $\rho \leq 0$ is the desired sign for "higher-is-better" metrics (TM-score, AAR) and $\rho \geq 0$ for

Table S9. Per-length PLL ↔ ESMFold TM-score correlations on LEFLUR-P-VAL unconditional generation. PLL = Monte-Carlo estimate of the any-order AR log-likelihood with $K = 32$ stratified-t random draws per modality. Each cell shows Pearson r / Spearman ρ. Lower (more negative) is better. $n = 100$ per length cell, $n = 500$ in the aggregate column. Strongest variant per column in **bold**.

PLL variant	$L = 100$	$L = 200$	$L = 300$	$L = 400$	$L = 500$	Agg.
seq	−0.17 / −0.12	−0.06 / −0.22	−0.24 / −0.20	−0.16 / −0.17	−0.20 / −0.22	−0.10 / −0.16
struc	**−0.29** / −0.46	**−0.42** / −0.44	**−0.43 / −0.56**	**−0.38 / −0.43**	**−0.68 / −0.73**	**−0.45 / −0.60**
joint	−0.30 / −0.38	−0.24 / −0.39	−0.37 / −0.37	−0.29 / −0.32	−0.50 / −0.54	−0.33 / −0.44
joint_true	−0.28 / **−0.50**	−0.30 / **−0.52**	−0.25 / −0.31	−0.33 / −0.39	−0.56 / −0.62	−0.38 / −0.55

Table S10. Protein-only Forward Folding best-of-N=30 per-picker breakdown (CAMEO 2022, n=127 targets). `random_pick` = single uniform draw per target. `struc_pll` matches the `LeFlurP N30 PLL 470M` row in Table 3. `oracle TM` matches the `LeFlurP N30 Oracle 470M` row. Pass-rate is RMSD < 2 Å (the main-table convention). Bold = best PLL picker per column; underline = overall best (oracle). p = paired Wilcoxon vs. `random_pick` on per-target TM-score.

Selector	Tokens	TM-Score	RMSD (Å)	Pass<2 Å (%)	p (Wilc. TM)
`random_pick`	256	0.653	11.53	13.4	—
`seq_pll`	256	0.660	**10.53**	13.4	0.530
`struc_pll`	256	**0.693**	12.34	**17.3**	$<10^{-4}$
`joint_pll`	256	0.673	13.45	15.7	0.023
`oracle TM`	256	<u>**0.751**</u>	6.73	26.8	$<10^{-4}$
`oracle RMSD`	256	0.734	<u>**5.78**</u>	<u>**27.6**</u>	$<10^{-4}$

"lower-is-better" metrics (RMSD, ligand RMSD, ligand centroid distance, ligand pocket min. distance). Bold marks the strongest predictor in each column. The per-task hierarchy mirrors protein-only LeFlur-p exactly: `struc_pll` dominates FF protein-quality columns (TM, RMSD), `seq_pll` dominates IF (AAR), and ligand-aware variants (`lig_struc_pll`, `joint_ligand_pll`, `joint_all_pll`) dominate ligand-pose columns. The 4-way `joint_true_4` estimator is competitive but not universally best.

Best-of-N=30 per-picker breakdown. The N30 `PLL` and `oracle` rows reported in Tables 4 and 2 are best-of-N=30 results from sampling 30 independent candidates per target and then selecting with the given selector. Tables S15 and S16 expand this with the full per-picker breakdown (mirroring Table S17): the `random_pick` single-uniform-draw reference, all 8 PLL variants, and the per-target oracles for each main-table quality metric.

Best-of-N=30 ligand-conditioned generation on PoseBusters. The CG benchmark in Table 6 (4 ligands × 100 designs from Didi et al. (2026)) is a separate, smaller suite focused on whether *any* sample passes a strict cofold filter. Table S17 reports the analogous best-of-N=30 study run on the full PoseBusters suite. Each candidate is then folded with Boltz-2 against the GT ligand SMILES (3,750 cofolds total), and *both_pass* is iptm ≥ 0.9 AND ipde ≤ 1.0. The PLL ranker more than doubles the strict pass-rate over `random_pick`, but a per-target Boltz-2 oracle would recover 32.8%.

Table S11. Protein-only Inverse Folding best-of-N=30 per-picker breakdown (CAMEO 2022, n=127 targets). For each backbone, generate 30 candidate sequences and select one by the indicated ranker; ESMFold is then used to score the picked sequence. `joint_true_pll` matches the `LeFlur-P N30 PLL 470M` row in Table 1. `oracle TM` matches the `LeFlur-P N30 oracle 470M` row. Pass-rate is RMSD < 2 Å. Bold = best pll picker per column; underline = overall best (oracle). p = paired Wilcoxon vs. `random_pick` on per-target ESMFold sc-TM.

Selector	Tokens	AAR (%)	sc-TM	RMSD (Å)	Pass<2 Å (%)	p (Wilc. sc-TM)
`random_pick`	256	35.0	0.803	4.59	33.9	—
`seq_pll`	256	35.2	0.795	4.76	35.4	0.672
`struc_pll`	256	34.9	**0.805**	**4.48**	36.2	0.143
`joint_pll`	256	**35.3**	0.800	4.70	33.1	0.584
`joint_true_pll`	256	34.9	0.799	4.73	**39.4**	0.198
`oracle TM`	256	35.1	**0.860**	**3.18**	**52.0**	$<10^{-4}$
`oracle AAR`	256	**38.7**	0.811	4.23	31.5	0.130

Table S12. Protein-only Unconditional Generation best-of-N=30 per-picker breakdown (LEFLUR-P-VAL, $L \in \{100, 200, 400, 500\}$, 400 slots = 4 lengths × 100 slots/length, 30 candidates per slot; 12,000 candidates total). For each slot, generate 30 candidate (sequence, structure) pairs and select one by the indicated ranker; ESMFold then scores the picked sample. Pass-rate is RMSD < 2 Å (the main-table convention). Bold = best PLL picker; underline = overall best (oracle). p = paired Wilcoxon vs. `random_pick` on the per-slot binary Pass < 2 Å indicator. `struc_pll` increases Pass-rate from 69.0% to 77.8%, +8.8 pp.

Selector	TM	RMSD (Å)	pLDDT	Pass<2 Å (%)	p (Wilc. Pass)
`random_pick`	0.840	15.29	0.739	69.0	—
`seq_pll`	0.572	29.97	0.622	49.2	$<10^{-4}$
`struc_pll`	**0.803**	**12.53**	**0.717**	**77.8**	1.2×10^{-4}
`joint_pll`	0.632	29.54	0.650	58.0	$<10^{-4}$
`joint_true_pll`	0.577	32.20	0.624	51.5	$<10^{-4}$
`oracle TM`	**0.971**	**1.04**	**0.786**	**100.0**	$<10^{-4}$

Table S13. Protein-ligand forward folding (PoseBusters, n=123). Spearman ρ vs. external quality metric. Lower PLL ⇒ higher quality, so the desired sign is ≤ 0 for TM-score and ≥ 0 for the three distance/RMSD metrics; bold = strongest predictor per column.

PLL variant	vs. TM	vs. RMSD	vs. lig. RMSD	vs. lig. centroid
SEQ_SCORE_UNIF	−0.608	+0.553	−0.225	+0.007
STRUC_SCORE_UNIF	**−0.691**	**+0.620**	+0.022	+0.058
LIG_ATOM_SCORE_UNIF	+0.149	−0.073	−0.007	+0.218
LIG_STRUC_SCORE_UNIF	−0.040	+0.044	**+0.273**	+0.189
JOINT_PROTEIN_SCORE_UNIF	−0.674	+0.603	−0.150	+0.031
JOINT_LIGAND_SCORE_UNIF	+0.010	+0.011	+0.244	**+0.225**
JOINT_ALL_SCORE_UNIF	−0.601	+0.558	−0.034	+0.135
JOINT_TRUE_4_SCORE_UNIF	−0.661	+0.573	−0.115	+0.017

Table S14. Protein-ligand inverse folding (PoseBusters, n=125 targets $\times$ 30 candidates = 3,750 rows). Spearman ρ between each PLL variant and the indicated quality metric. AAR is the design-time amino-acid-recovery metric used in the main body; the remaining four columns are post-hoc Boltz-2 cofolds of the picked (predicted seq, GT ligand SMILES) pair: ligand-ipTM, complex iPDE, TM-to-GT, and the GF+IP pass indicator (TM $\geq$ 0.5 AND ligand inside the pocket). Lower PLL $\Rightarrow$ higher quality, so the desired sign is $\rho \leq 0$ for higher-is-better metrics (AAR, lig. ipTM, TM, GF+IP) and $\rho \geq 0$ for the lower-is-better iPDE. Bold = strongest predictor per column.

PLL variant	vs. AAR	vs. lig. ipTM	vs. iPDE	vs. cofold TM	vs. GF+IP
SEQ_SCORE_UNIF	$\mathbf{-0.851}$	-0.408	$+0.455$	-0.471	-0.350
STRUC_SCORE_UNIF	-0.660	-0.474	$\mathbf{+0.606}$	-0.577	$\mathbf{-0.488}$
LIG_ATOM_SCORE_UNIF	$+0.198$	$+0.052$	-0.053	$+0.091$	$+0.020$
LIG_STRUC_SCORE_UNIF	-0.082	-0.135	$+0.120$	-0.104	-0.102
JOINT_PROTEIN_SCORE_UNIF	-0.810	$\mathbf{-0.503}$	$+0.602$	$\mathbf{-0.589}$	-0.481
JOINT_LIGAND_SCORE_UNIF	-0.013	-0.082	$+0.064$	-0.024	-0.042
JOINT_ALL_SCORE_UNIF	-0.564	-0.400	$+0.466$	-0.421	-0.374
JOINT_TRUE_4_SCORE_UNIF	-0.790	-0.496	$+0.565$	-0.585	-0.464

Table S15. Protein Ligand Forward Folding best-of-N=30 per-picker breakdown (n=123 PoseBusters targets after $L \leq 512$ filtering). `random_pick` = single uniform draw per target (no selector). PLL pickers select the candidate with minimum value of the indicated pseudo-NLL (`joint_protein` = seq + struc with ligand held clean). Oracles are upper bounds chosen by the GT-task quality metric (`TM`, `GF+IP`, or pocket-RMSD). The `joint_protein`, `oracle TM`, and `oracle GF+IP` rows match the corresponding rows in Table 4. Bold = best PLL picker per column; underline = overall best (oracle). p = paired Wilcoxon vs. `random_pick` on the per-target TM-score (the headline continuous metric the main-body PLL-transfer paragraph quotes).

Selector	Tokens	TM-Score	RMSD (Å)	GF+IP (%)	p (Wilc. TM)
`random_pick`	4375	0.654	13.32	24.4	—
`seq_pll`	4375	0.699	10.55	24.4	1.2×10^{-3}
`struc_pll`	4375	0.742	13.06	24.4	$<10^{-4}$
`lig_atom_pll`	4375	0.675	11.90	22.8	0.073
`lig_struc_pll`	4375	0.640	16.07	21.1	0.184
`joint_protein_pll`	4375	**0.755**	**10.00**	**28.5**	$<10^{-4}$
`joint_ligand_pll`	4375	0.651	16.07	18.7	0.383
`joint_all_pll`	4375	0.703	13.26	22.0	7.1×10^{-3}
`joint_true_4_pll`	4375	0.743	11.24	26.0	$<10^{-4}$
`oracle TM`	4375	<u>**0.793**</u>	<u>**7.59**</u>	30.1	$<10^{-4}$
`oracle GF+IP`	4375	0.696	11.22	<u>**56.9**</u>	2.1×10^{-3}
`oracle pocket RMSD`	4375	0.757	8.04	32.5	$<10^{-4}$

Table S16. Protein Ligand Inverse Folding best-of-N=30 per-picker breakdown (n=125 PoseBusters targets). `random_pick` = single uniform draw per target. TM-score and GF+IP are computed from a Boltz-2 cofold of the picked (predicted sequence, GT ligand SMILES) pair against the GT crystal structure (3,750 cofolds total; same harness as the IF row in Table 2). The `joint_protein` and `oracle GF+IP` rows match the corresponding rows in Table 2. Bold = best PLL picker per column; underline = overall best (oracle). p = paired McNemar vs. `random_pick` on the binary GF+IP pass-rate.

Selector	Tokens	AAR (%)	AAR P (%)	TM-Score	GF+IP (%)	p (McN. GF+IP)
random_pick	4375	65.88	73.93	0.594	37.6	—
seq_pll	4375	66.01	73.33	0.590	37.6	1.00
struc_pll	4375	67.36	**75.98**	**0.596**	40.0	0.678
lig_atom_pll	4375	67.08	73.77	0.588	39.2	0.815
lig_struc_pll	4375	67.02	75.69	0.583	35.2	0.607
joint_protein_pll	4375	**67.47**	75.37	0.595	**41.6**	0.359
joint_ligand_pll	4375	66.98	75.67	0.584	36.0	0.774
joint_all_pll	4375	67.29	75.54	0.590	37.6	1.00
joint_true_4_pll	4375	66.48	72.52	0.586	37.6	1.00
oracle AAR	4375	**69.47**	76.64	0.594	38.4	1.00
oracle AAR P	4375	67.67	**82.85**	0.588	39.2	1.00
oracle ipTM	4375	67.12	74.43	0.604	40.8	0.503
oracle TM	4375	67.54	76.12	**0.709**	60.0	5.0×10^{-4}
oracle GF+IP	4375	66.60	75.18	0.640	**70.4**	$<10^{-4}$

Table S17. Ligand-conditioned generation, best-of-N=30 on the full PoseBusters benchmark (125 targets). Each candidate is folded with Boltz-2 against the GT ligand SMILES; *both_pass* is ipTM ≥ 0.9 AND ipde ≤ 1.0. Mean cofold ligand iptm is the average over the 125 picked candidates. The `joint_true_4` selector is the additive 4-modality pseudo-NLL with the true masked-only AO-ARM mask; `oracle_ipTM` picks the candidate with maximum cofold ligand iptm per target (upper bound for any post-hoc selector). All p-values are paired McNemar against `random_pick`.

Selector	Tokens	both_pass (%)	ipTM≥0.9 (%)	mean ipTM	p (McN. pass)
random_pick	4375	6.4	8.0	0.599	—
joint_protein	4375	12.8	16.0	0.612	0.077
joint_ligand	4375	13.6	19.2	0.622	0.035
joint_all	4375	14.4	20.0	0.613	0.021
joint_true_4	4375	**15.2**	16.0	0.577	0.019
oracle_ipTM	4375	**32.8**	**60.0**	**0.902**	$<10^{-4}$

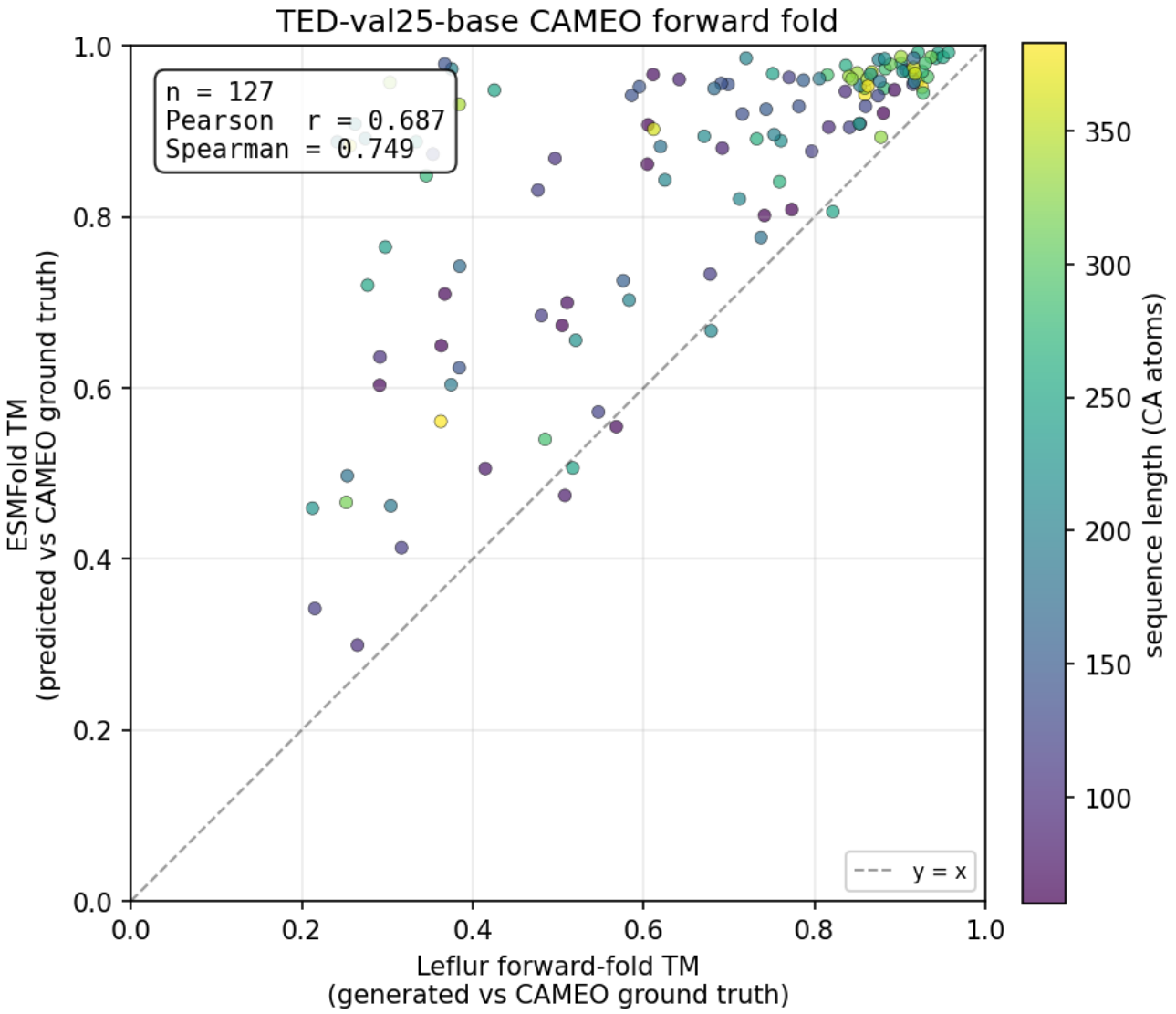


Figure S2. **LeFlur forward folding correlates with ESMFold.** Per-target scatter plot of LeFlur-p's forward-folded TM-score (against ground truth) versus ESMFold's TM-score on the same target. Each point is one CAMEO 2022 target. The positive correlation supports using LeFlur-p as its own forward-fold verifier in Self-Reflection.

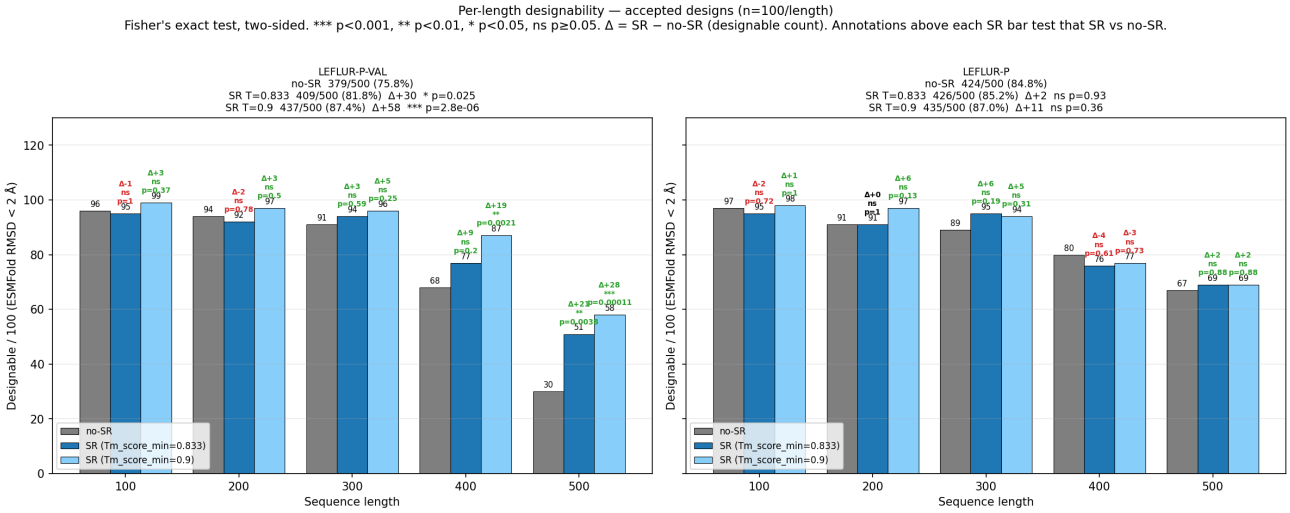


Figure S3. **Self-Reflection improves designability at longer lengths where the model struggles.** Per-length unconditional designability (ESMFold pass rate) for LeFlur-p, LeFlur-p-val, and the Self-Reflection variants `sr8`/`sr9`. Self-Reflection contributes most of its lift at $L \geq 400$, where LeFlur-p-val's single-pass pass rate degrades; below $L = 300$ the lift is negligible.

H. Datasets

To train LatentGenerator and LeFlur across proteins, ligands, and joint modalities, we curated a multi-scale dataset combining experimental structures with high-quality synthetic data. For the protein only models we used crystal and predicted protein structures. For the protein-ligand models we used both these and ligand, and protein-ligand dataset.

Protein Structures. For the protein-only modality, we utilized the curated Protein Data Bank (PDB) (Berman et al., 2000) snapshot provided by OpenProteinSet (Ahdritz et al., 2023). This dataset provides a standardized, high-quality split of experimental protein structures filtered for resolution and chain length.

Synthetic Protein Structures. For protein monomer synthetic data we utilize the AlphaFold (Jumper et al., 2021) predicted structures of SwissProt (Varadi et al., 2021). This dataset provides higher diversity than the PDB and should allow the model to better generalize. To maintain high quality prediction we filter for structures with pLDDT > 85.

Protein only Distillation Set. Modern models train on a far larger corpus of data than we have in LeFlur, for instance Proteina (Geffner et al., 2025) is trained on 21M AFDB samples while La Proteins is trained on $\sim$46M AFDB samples (Didi et al., 2025). Rather than spending the compute on training our models on this corpus we decide to derive the relevant learning signals from pretrained models through distilling sequence-structure pairs (along with inverse folding augmentation). For each length $L \in \{100, 200, 300, 400, 500\}$ we combine samples from four generators: La-Proteina (Didi et al., 2025) and DPLM-2 (Wang et al., 2024), which directly co-design structure and sequence; and Proteina (Geffner et al., 2025) and Genie2 (Lin et al., 2024) which only generate structure. Each backbone is paired with designed sequences from both LigandMPNN (Dauparas et al., 2022; 2025) and Caliby (Shuai et al., 2025b). Every design is then subjected to ESMFold (Lin et al., 2023) self-consistency filtering: we retain only designs whose ESMFold-refolded structure satisfies Cα RMSD < 2 Å and pLDDT > 70 against the generated backbone. The final distillation corpus contains 26,390 cluster representatives drawn from 772,439 passing designs, providing a high-diversity, length-balanced supplement to the natural structure data used for training.

Protein only Redesigns set. To broaden coverage of structural diversity beyond what PDB-derived training data offer, we add two re-designed domain datasets. Backbones are taken from (i) The Encyclopedia of Domains (TED) (Lau et al., 2024), restricted to the *novel folds* (7427 domains with topologies absent from CATH) and *high-symmetry folds* (6433 domains) subsets released for AFDB, and (ii) the CATH non-redundant S40 release (Sillitoe et al., 2021) (34 653 experimental domains clustered at 40 % sequence identity). For every backbone we generate sequences from LigandMPNN (Dauparas et al., 2025) and from Caliby (Shuai et al., 2025a), fold each with ESMFold (Lin et al., 2023), and keep candidates with C_α RMSD < 2 Å to the input backbone, mean pLDDT > 70, and no single amino acid exceeding 20 % of the sequence. Each retained backbone contributes all of its sequences which are sampled uniformly during training if that backbone is chosen as a training example. Backbones are then clustered with Foldseek (van Kempen et al., 2024a). This yields 6025 backbones in 4386 SS-balanced clusters for TED and 27 136 backbones in 4735 SS-balanced clusters for CATH S40 (78.3%). Cluster-level minority up-sampling makes both datasets contribute equally per epoch (20 % of effective training mass each).

Synthetic-sequence data drives generation quality. Tables S20 and S19 ablate LeFlur-p's training mix while holding the model recipe fixed, isolating the effect of synthetic-sequence data. The native-only model (PDB+AFDB) collapses on unconditional generation: designability falls from 60% at length 100 to 1–6% at lengths 400–500, yielding only 39 distinct Foldseek clusters across all lengths. Adding synthetic-sequence data restores generation: the de novo-only variant reaches 74% mean designability and 197 clusters, and the full de novo+TED+CATH mix is best, at 84% mean designability with consistently high self-consistency (scTM ≈ 0.94). The two synthetic variants trade off quality against diversity—de novo+TED+CATH is the most designable but its samples collapse onto fewer folds (173 clusters), whereas de novo-only is less designable but spans more distinct topologies. The same ordering holds for inverse folding (de novo+TED+CATH best, scTM 0.774), while forward folding remains weak for both synthetic variants, indicating the native sequence space is necessary for forward folding. We emphasize these are mid-training snapshots taken at unequal progress (native and synthetic-all $\approx$76% of the 50k-step schedule, de novo-only only $\approx$27%), so the de novo-only numbers in particular could improve with further training.

Table S18. Training data partition for LeFlur-P. Sampling is done with a minority-upsampling strategy so each dataset contributes equally per epoch (`balance_datasets: true`). Structures truncated to ≤ 512 residues.

Dataset	Structures	Clusters	Rep. factor	Balanced clusters
PDB ($\leq 40\%$ seq. id.)	280,586	50,124	1.69×	84,816
AFDB SwissProt	220,354	84,816	1.00×	84,816
Distillation	772,439	26,406	3.21×	84,816
TED	6025	4386	19.34×	84,816
CATH S40	27,136	4735	17.91×	84,816
Total	1,306,540	170,467		424,080

Table S19. LeFlur-p training-data ablation: conditional folding on CAMEO (n=127). Inverse folding (IF) reports ESMFold self-consistency of designed sequences; forward folding (FF) reports recovery of the input structure. All variants use the identical leflur-base recipe; only the training data differs. **Snapshots are mid-training** (no-distill/synthetic-all ≈76% of 50k steps; denovo-only ≈27%).

	Inverse folding				Forward folding	
Training data	SeqID (%)	scTM	RMSD (Å)	Pass$_{<2Å}$ (%)	scTM	RMSD (Å)
PDB+AFDB (native only)	**32.2**	0.729	6.40	**22.0**	**0.652**	**11.71**
de novo only	21.0	0.754	5.29	15.7	0.421	15.36
de novo+TED+CATH (synthetic)	23.0	**0.774**	**4.72**	**22.0**	0.462	14.82

Ligand Structures. Ligand geometries were sourced from the GEOM dataset (Axelrod & Gomez-Bombarelli, 2022), a large-scale collection of conformal geometries derived from QM-level calculations. We utilized the subset of drug-like molecules, which provides over 300,000 unique molecular graphs associated with millions of valid 3D conformers. This allows the model to learn valid local chemical geometry (bond lengths, angles, torsions) independent of protein context.

Experimental Protein-Ligand Structures. For joint training on experimental data, we utilized the PDBbind database (Liu et al., 2015) and PLINDER (Durairaj et al., 2024). PDBbind serves as the standard collection of experimentally measured binding data mapped to 3D structures in the Protein Data Bank. We employed the refined set, which filters for high-resolution crystal structures and valid binding data, providing a high-fidelity "gold standard" for intermolecular interactions. Plinder is a large-scale curated dataset of ∼60,000 protein-ligand complexes sourced from the Protein Data Bank. PLINDER provides standardized protein-ligand pairs with quality filtering, ligand SMILES clustering, and train/test splits designed to minimize data leakage. We used the training split with SMILES-based clustering (22,237 clusters) to ensure diverse ligand coverage during balanced sampling.

Synthetic Protein-Ligand Structures. To address the scarcity of experimental complex data, we augmented our training set with the Structurally Augmented IC50 Repository (SAIR) (Lemos et al., 2025). SAIR is a massive synthetic dataset comprising over 5 million protein-ligand structures derived from ChEMBL and BindingDB activity data. These complexes were computationally folded using the Boltz-1x model, providing a orders-of-magnitude expansion in chemical and structural diversity compared to PDBbind alone. We utilized high-confidence subsets of SAIR to scale the joint probability distribution learned by LeFlur.

Protein-Ligand Distillation Set. To transfer knowledge from specialized ligand-conditioned generative models, we constructed a distillation dataset using Proteina-Complexa (Jendrusch et al., 2025). We selected 240 unique drug-like ligands from the PLINDER database (Durairaj et al., 2024) and generated 100-residue protein binder designs for each using Proteina-Complexa's ligand-conditioned partially latent diffusion model. Protein sequences were obtained via two routes: direct generation by Proteina-Complexa and LigandMPNN (Dauparas et al., 2025) redesign of generated backbones. Each design was validated via RF3 co-folding (Corley et al., 2025), retaining 1,663 complexes that passed quality filters on interface predicted aligned error (iPAE < 3.1), backbone self-consistency RMSD, and pLDDT.

Protein-Ligand Redesign Set. To increase sequence diversity for known protein-ligand complexes, we created a redesign dataset using LigandMPNN (Dauparas et al., 2025). Starting from 1,315 experimentally determined protein-ligand complexes

Table S20. LeFlur-p training-data ablation: unconditional generation (100 samples per length). Des. = designability (ESMFold self-consistency scRMSD < 2 Å); scTM = mean self-consistency TM-score; #Clu = Foldseek structural clusters (single-linkage, TM ≥ 0.5) among designable samples.

	PDB+AFDB (native)			de novo only			de novo+TED+CATH		
Length	Des.%	scTM	#Clu	Des.%	scTM	#Clu	Des.%	scTM	#Clu
100	60	0.82	13	96	0.92	48	99	0.93	36
200	29	0.71	10	93	0.94	45	98	0.95	41
300	29	0.87	13	86	0.94	43	90	0.95	42
400	1	0.88	1	59	0.93	38	76	0.94	35
500	6	0.92	2	37	0.92	23	58	0.94	19
Mean / Total	25.0	—	39	74.2	—	197	84.2	—	173

sourced from PLINDER (Durairaj et al., 2024), we applied LigandMPNN to redesign the protein sequence conditioned on the crystal structure and bound ligand and used the same filtering criteria as the distillation set.

Table S21. Training data composition for LeFlur-pl model. Balanced sampling with (max cluster replicates 5) upsampled minority datasets to match the largest cluster count

Dataset	**Type**	**Samples**	**Clusters**	**Cluster Method**	**Balanced**	**% Epoch**
PDB	Protein	278,768	49,441	SeqID 40	1.7×	22.1
AFDB SwissProt	Protein	198,295	84,573	SwissProt	1.0×	22.1
PDBBind	Protein–Ligand	21,835	17,267	SMILES	4.9×	22.1
SAIR	Protein–Ligand	279,963	2,039	SeqID 40	5.0×	2.7
PLINDER	Protein–Ligand	60,276	22,237	SMILES	3.8×	22.1
Distillation	Protein–Ligand	1,663	240	SMILES	5.0×	0.3
Redesign	Protein–Ligand	1,315	1,246	SMILES	5.0×	1.6
Total		847,574			383,212	100.0

I. Benchmark Separation From Training Data

Protein-only benchmarks (Campbell et al., CAMEO 2022). Our PDB training data is the OpenProteinSet dated December 2021. Both protein-only evaluation sets lie strictly after this cutoff: CAMEO 2022 evaluates on the CAMEO targets from August through October 2022, and the Campbell et al. ICML 2024 split is drawn from PDB entries deposited after their 2023 cutoff.

Protein-ligand benchmark (PoseBusters). The PoseBusters (Buttenschoen et al., 2024) set is the only protein-ligand benchmark we evaluate against. The two real-PDB training sources that could overlap PoseBusters are PDBbind and the experimental subset of PLINDER. Posebusters by constructions does not include complexes within PDBbind General Set v2020. We filter the PLINDER training set to remove any PDBs found in PoseBusters.

J. Additional Unconditional Generation Results

In this section, we provide extended benchmarks for the unconditional generation capabilities of LeFlur-p. We find that the model maintains reasonable diversity from all of its training data sources, including the PDB, AFDB, and the distillation dataset. As reported in Table S22, LeFlur-p is comparable in novelty to La Proteina against the natural reference sets (LeFlur-p 0.788 vs. La Proteina 0.775 against the PDB; 0.732 vs. 0.742 against AFDB) and substantially more novel than DPLM-2 (0.932 / 0.948). The most-novel model in the table is Genie2+ProteinMPNN (0.689 / 0.661). LeFlur-p offers a competitive novelty/designability trade-off (84.8% pass rate vs. Genie2+ProteinMPNN at 52.0%, Table 5).

Table S22. **Unconditional Structural Novelty Analysis.** Reported values are the mean of the maximum TM-score of each generated structure against the indicated reference set (computed by Foldseek). *Lower is more novel*; higher means the model produces structures with closer matches in the reference. The reference sets are the natural PDB and AFDB Swiss-Prot training sources and the protein-only distillation set; we evaluate against the natural sources rather than the ESMFold-distilled synthetic set so as to avoid using ESMFold-filtered samples as the novelty reference.

Model	vs PDB	vs AFDB	vs Distillation
Genie2 + ProteinMPNN	**0.689**	**0.661**	
Proteina + ProteinMPNN	0.788	0.763	
La Proteina 650M	0.775	0.742	
DPLM-2 650M	0.932	0.948	
LeFlur-p 470M SLQ	0.788	0.732	**0.691**
LeFlur-p-val 470M SLQ	0.771	0.732	0.714

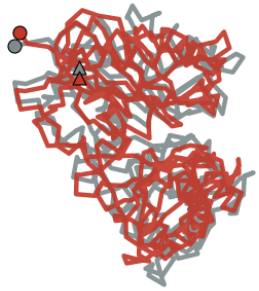
Designability is not an ESMFold artifact. Because part of LeFlur-p's training data is a distillation set filtered by ESMFold, one might worry that its high in-distribution pass rate simply reflects agreement with the same oracle used to curate the data. To rule this out, we re-evaluate the designed sequences with AlphaFold2 (AF2), a structurally and methodologically independent folding model (Table S23). The conclusions hold under this orthogonal check: LeFlur-p retains a high AF2 pass rate of 79.4% (vs. 84.8% under ESMFold) at an average AF2 TM of 0.926, and LeFlur-p-val reaches 69.0% (vs. 75.8%). The small, uniform gap is the expected consequence of AF2 being a marginally stricter folder—concentrated almost entirely at the longest backbones ($L \geq 400$)—rather than a collapse of designability. Foldseek cluster counts remain high under AF2 (272 and 253), preserving the diversity trends of Table 5. We therefore attribute LeFlur-p's designability to genuinely foldable generations rather than to overfitting the ESMFold oracle used during distillation.

Table S23. AlphaFold2 cross-validation of unconditional designability. Each model is evaluated by refolding its designed sequences with both ESMFold (the metric reported in Table 5) and AlphaFold2 (AF2), averaged over lengths 100–500 (100 designs each). Pass (%) uses scRMSD < 2 Å; Clusters is the number of Foldseek clusters (TM ≥ 0.5) summed over lengths on the passing designs. AF2 is a uniformly slightly stricter folder but preserves the ranking, and diversity.

Method	Folder	Pass (%)	Avg TM	Clusters
LeFlur-p 470M SLQ	ESMFold	84.8	0.937	301
	AF2	79.4	0.926	272
LeFlur-p-val 470M SLQ	ESMFold	75.8	0.895	288
	AF2	69.0	0.876	253

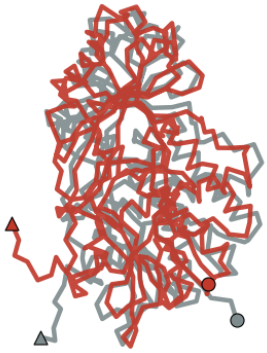
K. Sample Predictions

7q6d.A L=383
TM=0.86 Kabsch RMSD=3.24 Å



8dt6.C L=381
TM=0.84 Kabsch RMSD=4.42 Å

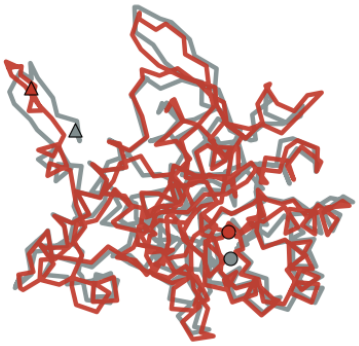
7fgp.A L=377
TM=0.83 Kabsch RMSD=4.85 Å



7x0q.A L=362
TM=0.94 Kabsch RMSD=1.87 Å

8b26.A L=359
TM=0.92 Kabsch RMSD=3.48 Å

7oj2.A L=353
TM=0.94 Kabsch RMSD=2.20 Å



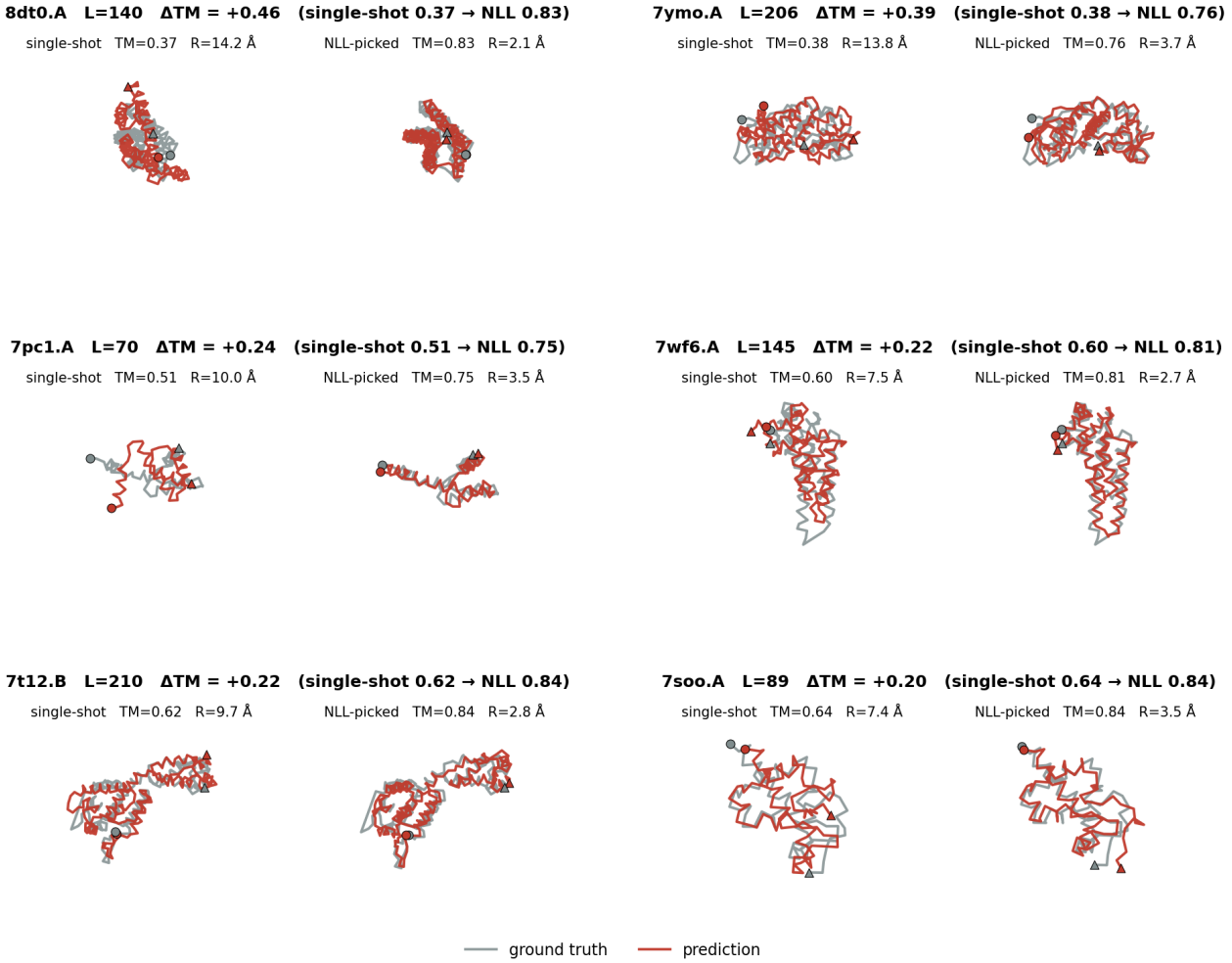
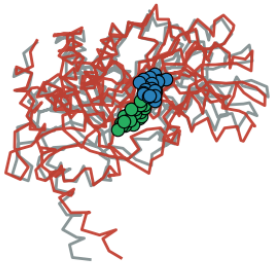


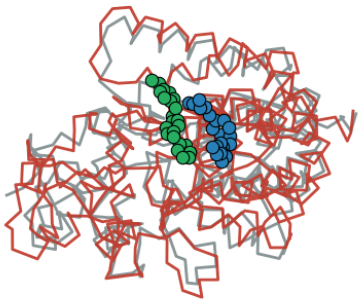
Figure S5. **Sample Leflur-p forward folding predictions with PLL improvements.**

7VC5_9SF L=485 TM=0.95
pkt RMSD=0.99 Å lig Δ=8.2 Å

7XEK_9YX L=413 TM=0.95
pkt RMSD=0.99 Å lig Δ=7.5 Å

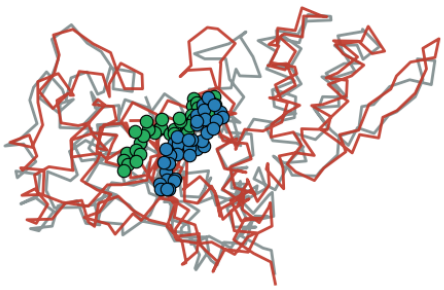


7P5T_5YG L=396 TM=0.92
pkt RMSD=0.11 Å lig Δ=7.7 Å



8BTI_RFO L=350 TM=0.94
pkt RMSD=0.70 Å lig Δ=6.8 Å

7ROU_66I L=335 TM=0.92
pkt RMSD=1.65 Å lig Δ=3.7 Å



6ZAE_ACV L=328 TM=0.92
pkt RMSD=0.94 Å lig Δ=5.2 Å

L. Sampling Hyperparameters

The generative performance of LeFlur-p is closely tied to the configuration of its iterative decoding process. Unlike standard autoregressive models, our framework utilizes an iterative refinement scheme where sequence and structure tokens are jointly updated over N steps. This process is governed by four primary categories of hyperparameters: sampling temperatures ($\tau_{\text{seq}}, \tau_{\text{struct}}$), which control the sharpness of the categorical distributions; stochasticity factors, which define the percentage of tokens randomly re-sampled at each step to prevent local minima; and transition schedules (Linear, Log, Power), which dictate the rate at which noise is annealed throughout the generation process. For the logit bias we add $+1.0$ to the valine logit along the entire protein length before adding temperature for the first 25 steps of generation.

Table S24. **Sampling Hyperparameters Across Protein only Models and Tasks.** Comparison of generation parameters for Unconditional Generation, Inverse Folding, and Forward Folding.

Mode	Model	N	τ_{seq}	τ_{struct}	$\text{Stoch}_{\text{seq}}$	$\text{Stoch}_{\text{struct}}$	$\text{Sched}_{\text{seq}}$	$\text{Sched}_{\text{struct}}$
UG	LeFlur-p	400	0.273	0.316	10	10	Log	Power
UG	LeFlur-p-v	400	0.273	0.316	20	60	Log	Power
IF	LeFlur-p	100	0.150	0.418	10	50	Log	Linear
FF	LeFlur-p	200	0.361	0.220	1	20	Log	Linear

For LeFlur-pl in addition to the protein sequence and structure sampling hyperparameters we also have ligand atom and structure sampling hyperparameters. Note that during inverse folding and forward folding we give the model the atom types and ligand structure tokens and keep them fixed. During ligand conditioned generation we do not provide the structure tokens, but give the atom types and bond matrix.

Table S25. **Sampling Hyperparameters Across Protein-Ligand Models and Tasks.** Comparison of generation parameters for Ligand Conditioned Generation, Inverse Folding, and Forward Folding.

Mode	Model	N	τ_{seq}	τ_{struct}	$\text{Stoch}_{\text{seq}}$	$\text{Stoch}_{\text{struct}}$	$\text{Sched}_{\text{seq}}$	$\text{Sched}_{\text{struct}}$
CG	LeFlur-pl	200	0.5	0.5	20	20	Linear	Power
IF	LeFlur-pl	100	0.5	0.5	20	20	Log	Linear
FF	LeFlur-pl	100	0.5	0.5	20	20	Log	Linear

Table S26. **Ligand Specific Sampling Hyperparameters Across Protein-Ligand Models and Tasks.** Comparison of generation parameters for Ligand Conditional Generation, Inverse Folding, and Forward Folding.

Mode	Model	τ_{atom}	τ_{struc}	$\text{Stoch}_{\text{atom}}$	$\text{Stoch}_{\text{struc}}$	$\text{Sched}_{\text{atom}}$	$\text{Sched}_{\text{struc}}$
CG	LeFlur-pl	0.1	0.1	5	5	Power	Linear
IF	LeFlur-pl	0.5	0.5	20	20	Log	Linear
FF	LeFlur-pl	0.5	0.5	20	20	Log	Linear

M. Architecture Diagrams

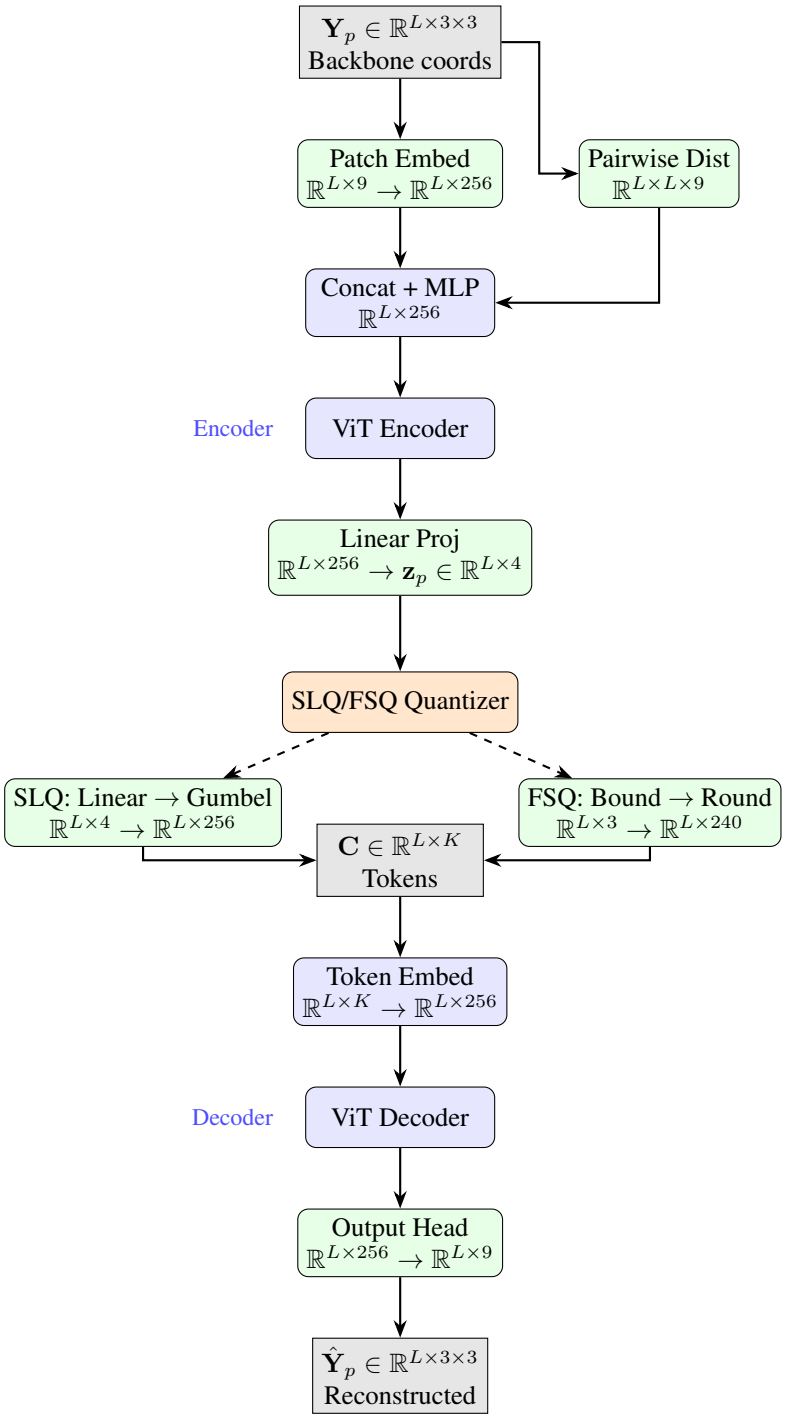


Figure S7. Information flow in protein-only LatentGenerator. Backbone coordinates are embedded with pairwise distance features, processed through a ViT encoder, quantized via SLQ or FSQ, and reconstructed through a ViT decoder.

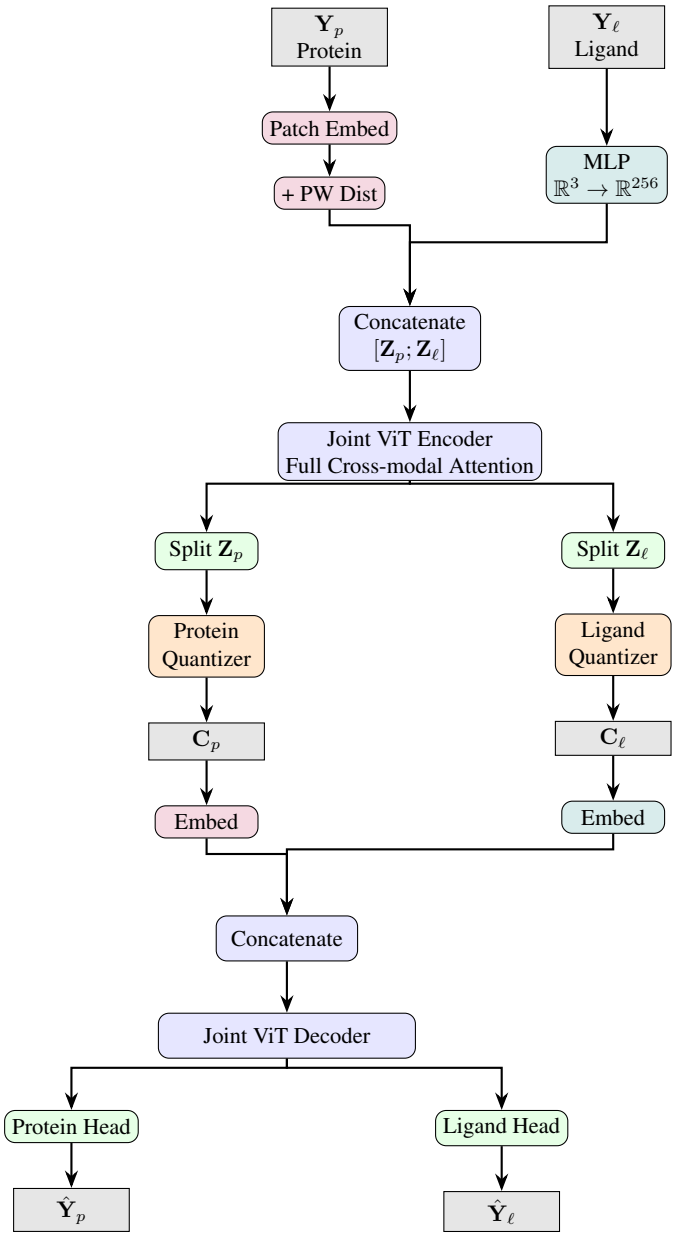


Figure S8. Information flow in protein-ligand LatentGenerator. Protein backbone and ligand atoms are embedded separately, concatenated for joint attention in the encoder, split for modality-specific quantization, and decoded jointly to preserve relative positioning.

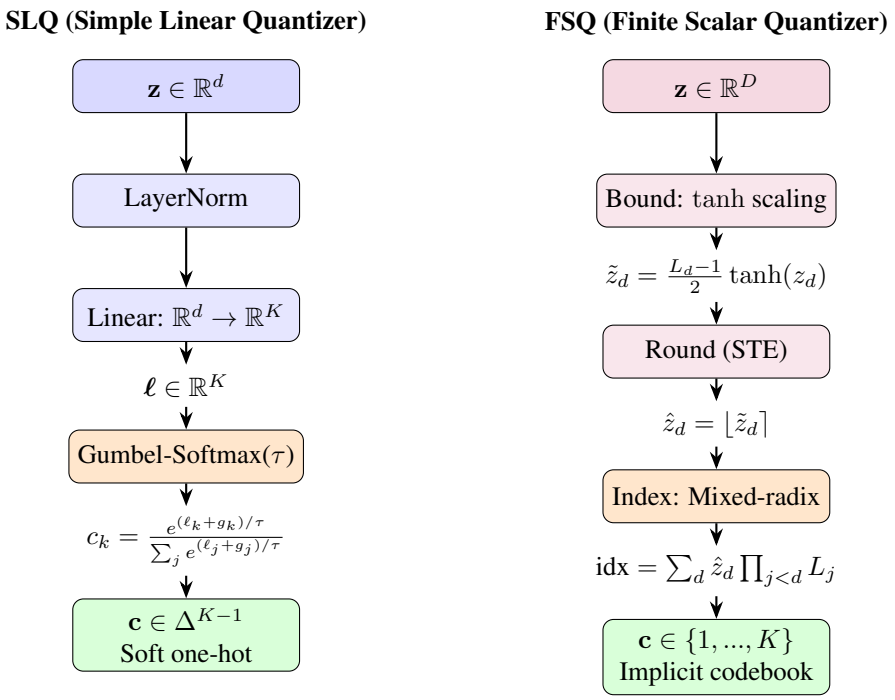


Figure S9. Comparison of SLQ and FSQ quantization mechanisms. SLQ uses learned projections with Gumbel-Softmax for differentiable soft quantization, while FSQ applies fixed scalar quantization per dimension to create an implicit codebook.

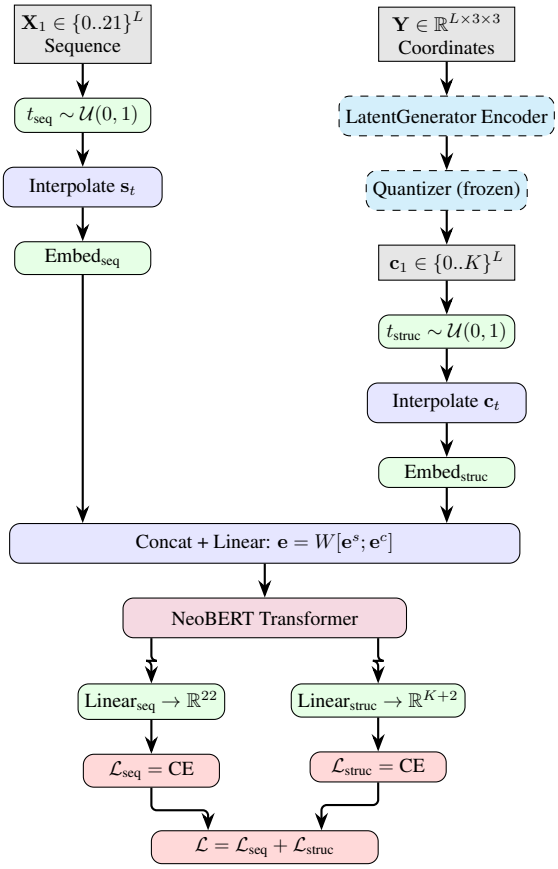


Figure S10. LeFlur training pipeline. Structure coordinates are tokenized by frozen LatentGenerator. Sequence and structure tokens are independently masked at sampled timesteps, embedded, combined, and processed by trainable NeoBERT. Separate heads predict logits for cross-entropy loss against ground truth.

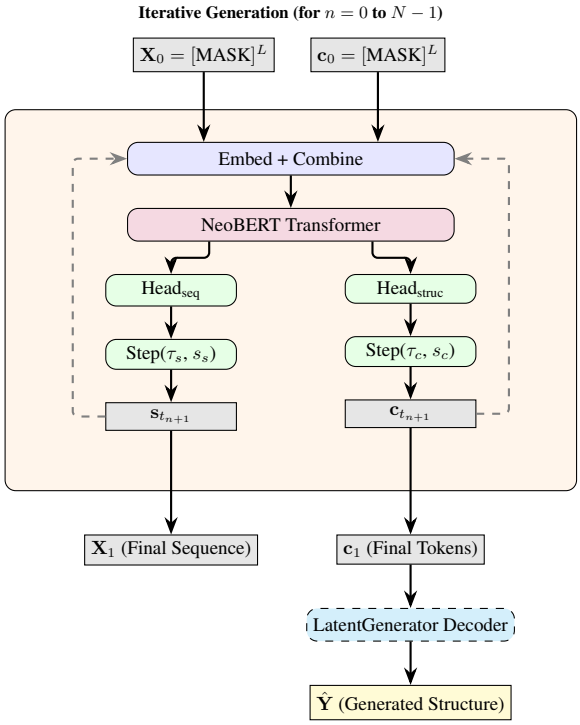


Figure S11. LeFlur generation pipeline. Starting from fully masked priors, the model iteratively unmasks tokens using predicted logits with temperature-controlled stochastic sampling. After N steps, structure tokens are decoded to 3D coordinates via frozen LatentGenerator decoder.

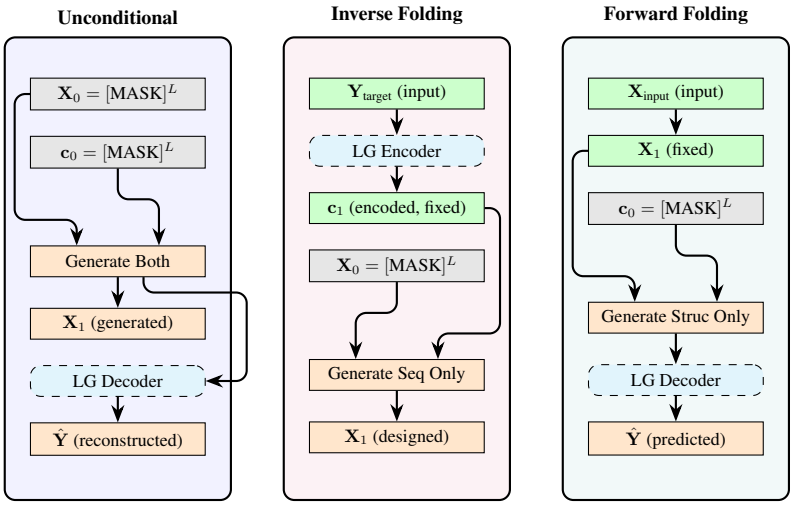


Figure S12. LeFlur generation modes including LatentGenerator components (dashed boxes). **Inverse Folding** uses the *Encoder* to tokenize the target structure. **Forward Folding** and **Unconditional** modes use the *Decoder* to reconstruct 3D coordinates from generated tokens.

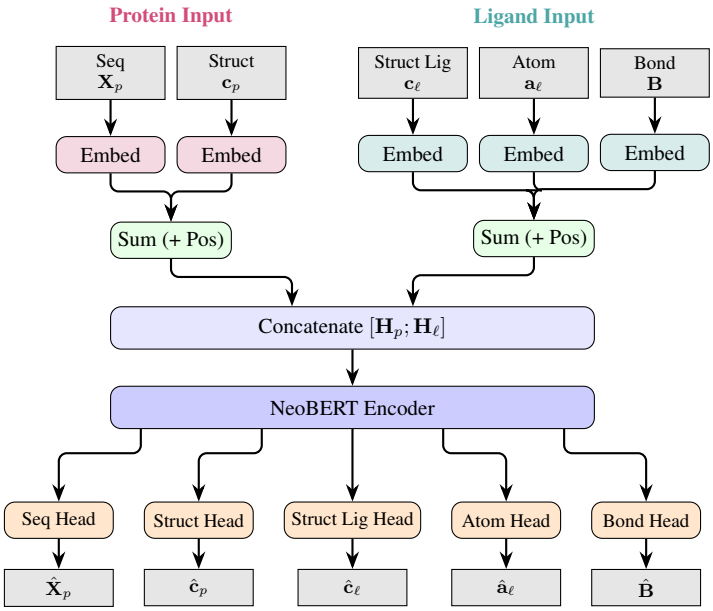


Figure S13. LeFlur Protein-Ligand architecture. Protein features (left) and ligand features (right) are embedded, summed with positional encodings, concatenated, and processed by a shared NeoBERT encoder. Five specialized heads predict the denoised modalities.

N. Compute Resources

All training was performed on a single GPU node (8 NVIDIA GPUs) using PyTorch Lightning with bfloat16 mixed-precision and a gradient-clipping value of 0.5. Table S27 summarises the headline training-budget settings used for each model; per-task inference-time cost is discussed separately in Section G.

Table S27. **Training compute settings.** "Effective batch" is computed as (per-GPU batch size) × (`accumulate_grad_batches`) × (8 GPUs). All runs used a single node, bf16-mixed precision, and gradient clipping at 0.5.

Model	Steps	Warmup	Per-GPU batch	Effective batch
LatentGenerator	100,000	10,000	40	–
LeFlur-p	50,000	2,500	48	7,680
LeFlur-pl	100,000	5,000	8	2,560

Inference compute. Single-pass design with LeFlur runs on a single GPU. The default sampling configurations in Tables S24, S25, and S26 use 100–400 NeoBERT forward passes per design. PLL best-of-N ranking and Self-Reflection multiply this cost as analysed in the "Inference-time cost" paragraph of Section G; both stay within LeFlur and avoid any external folding oracle.

O. Broader Impact

LeFlur is a generative model for joint sequence-structure design of proteins and protein-ligand complexes. We discuss the principal positive and negative societal impacts below, in keeping with the NeurIPS guidelines.

Potential positive impacts. By unifying inverse folding, forward folding, and unconditional/ligand-conditioned generation under a single set of weights, LeFlur lowers the engineering and compute barriers to running protein-design pipelines that previously required orchestrating multiple specialised models (Watson et al., 2023; Pacesa et al., 2025; Stark et al., 2025). Its built-in pseudo-likelihood and Self-Reflection verifiers further reduce reliance on heavy external structure-prediction oracles. These properties may help accelerate research in protein engineering, enzyme design, antibody and small-molecule binder discovery, biosensor development, and basic structural biology, particularly in academic and resource-constrained settings.

Potential negative impacts. Like other generative biomolecular design tools (e.g.RFdiffusion (Watson et al., 2023), BindCraft (Pacesa et al., 2025), BoltzDesign (Cho et al., 2025), and Proteina-Complexa (Didi et al., 2026)), LeFlur could in principle be misused to design biomolecules with harmful function. We mitigate this risk in several ways. The model is not trained on, conditioned on, or evaluated against any pathogen-specific or toxin-specific datasets; its training corpus consists of public protein-monomer and small-molecule complex data (PDB/OpenProteinSet, AFDB SwissProt, GEOM, PDBbind, PLINDER, SAIR; see Section H) with no enrichment for hazardous targets.

Use of public data and compute. All training data are drawn from publicly distributed structure databases and predicted-structure resources, used under their respective licenses. Compute requirements (Section N) are modest by foundation-model standards (single 8-GPU node), which we view as a positive: it broadens who can reproduce, audit, and extend the work, rather than concentrating capability in a small number of well-resourced groups.