MULTIMODAL LATENT FLOW MATCHING FOR DE NOVO NANOBODY DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative protein design must leverage heterogeneous sequence and structural data while supporting precise control over molecular interactions. We introduce Proteina Mixta, a multimodal latent flow-matching model for all-atom co-design and sequence-only generation. A shared autoencoder maps sequence-only, backbone–sequence, and all-atom inputs into continuous residue-level representations, enabling one model to generate across modalities. By modeling discrete amino acid sequences through continuous latent dynamics, Proteina Mixta achieves strong performance in both all-atom monomer co-design and sequence-only generation. We extend Proteina Mixta to target-conditioned nanobody (a single-domain antibody) design by assigning synthetic annotations of antibody framework and complementarity-determining region (CDR) to general protein interfaces. Antibody-specific conditioning is then learned entirely from these synthetic examples: true CDR and framework labels are never seen during representation learning or generative training. The multimodal formulation further enables direct co-training of the structure-based generator with sequence-only antibody repertoires from the Observed Antibody Space (OAS) database. Training on liability-filtered OAS sequences increases effective CDR3 diversity by 26% and reduces sequence-liability rates by 13%. In a controlled comparison with common targets, VHH frameworks, CDR lengths, and sampling budgets, Proteina Mixta achieves up to 50-fold higher unique in silico success rates than prior open models, without sequence redesign. These results suggest that a shared latent space lets structure-free data supervise all-atom generation, opening a route to de novo structure design at the scale of sequence data.

1 INTRODUCTION

De novo protein design models combine heterogeneous sequence and structural data to design desired molecular interactions. Sequence diffusion models learn from large scale sequence datasets (Alamdari et al., 2023; Wang et al., 2024), whereas structural diffusion models learn from limited structure data (Watson et al., 2023). Multimodal models increasingly connect these capabilities: ESM3 and DPLM-2 use discrete sequence and structure representations (Hayes et al., 2025; Wang et al., 2025b), while PLAID and La-Proteina demonstrate continuous latent approaches to sequence–structure generation (Lu et al., 2026; Geffner et al., 2026). These advances motivate a unified generator that learns across all levels of biological information and supports both sequence-only generation and all-atom design.

We introduce **Proteina Mixta**, a multimodal latent flow-matching model that maps sequence-only, backbone–sequence, and all-atom inputs into continuous residue-level representations. A single generative model learns across all three regimes, generating discrete amino acid sequences through continuous latent dynamics and jointly modeling backbone geometry when structure is required.

We extend Proteina Mixta from monomer to target-conditioned nanobody de novo design. A nanobody is the variable domain of a heavy-chain-only antibody (VHH) that combines a conserved framework with three complementarity-determining regions (CDRs), whose sequences and conformations govern molecular recognition (Muyldermans, 2013). Solved nanobody–antigen structures remain limited ($\sim$4K Schneider et al. (2022)), so rather than train on them directly, we build on the general interface modeling of Proteina-Complexa (Didi et al., 2026) and construct pseudo-VHH training examples by assigning synthetic framework and CDR annotations to general protein interfaces. Antibody-specific conditioning is thus learned from synthetic annotations.

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

Proteina Mixta's multimodal formulation further enables co-training of the structure-based generator with sequence-only antibody repertoires from Observed Antibody Space (OAS) (Olsen et al., 2022). We show the impact of sequence data by training Proteina Mixta on the combination of liability-filtered OAS heavy-chain sequences and existing protein structure data.

We make the following contributions:

- **Proteina Mixta, a multimodal latent flow-matching model.** A shared autoencoder maps sequence-only, backbone–sequence, and all-atom inputs into one continuous residue-level latent space, and a single flow model independently generates across all three regimes, producing discrete amino acid sequences through continuous latent dynamics.
- **Antibody conditioning learned from pseudo-VHH supervision.** Synthetic framework and CDR annotations on general protein interfaces yield framework-constrained VHH design without true CDR or framework annotations at any stage of training.
- **A structure generator co-trained on sequence-only repertoires.** Curated sequence data shifts generated designs toward its own distribution, improving CDR3 diversity and reducing sequence liabilities with no refolding and no inferred structural labels. The same sequence-only mode supports post-training against a structural reward, with no architectural change.
- **A matched comparison against open models.** Holding targets, VHH framework, CDR lengths, sampling budget, and evaluation criteria fixed, Proteina Mixta achieves up to 50-fold higher unique *in silico* success rates than prior open models, without sequence redesign.

2 Background and Related Work

Partially latent protein generation. Proteina Mixta builds directly on La-Proteina (Geffner et al., 2026), which pairs an explicit Cα coordinate with a continuous per-residue latent. Mixta replaces its all-atom-only autoencoder with one that encodes sequence-only, backbone–sequence, and all-atom inputs into a single latent space. A single flow model then trains across all three regimes, so structure-free sequences update the same generative field used for all-atom co-design (Fig. 1).

Antibody and nanobody generation. Structure-based VHH design requires the generation of CDR sequences, loop conformations, and target-binding poses while preserving a supplied framework. Generative methods learn this mapping directly: RFantibody produces CDR backbones and poses with an antibody-specialized RFdiffusion, followed by ProteinMPNN sequence design (Bennett et al., 2026), and BoltzGen jointly generates binder sequence and all-atom structure with optional sequence redesign (Stark et al., 2025). Hallucination methods instead optimize against a structure predictor: Germinal refines CDRs using AlphaFold-Multimer and the antibody language model IgLM (Mille-Fragoso et al., 2026), BindCraft2 optimizes binder sequences through AlphaFold2 before ProteinMPNN redesign (Pacesa Lab, 2026), and ESMFold2 supports gradient-guided hallucination, optimizing CDR logits by backpropagating distogram losses through a differentiable forward pass (Candido et al., 2026). These families differ in sequence redesign, internal filtering, and sampling budget, which motivates our controlled comparison.

Antibody structure prediction and generalization. Predicting antibody–antigen complexes requires resolving both flexible CDR conformations and their binding poses. FoldBench reports substantial antibody–antigen prediction failures on held-out structures and highlights dependencies between interface prediction accuracy and training-set similarity (Xu et al., 2026). Temporal holdouts therefore benefit from additional similarity controls when assessing generalization. Protenix-v2 substantially improves antibody–antigen prediction on post-cutoff benchmarks (Zhang et al., 2026), yet predicted complexes remain imperfect substitutes for experimentally determined structures. These limitations motivate learning from observed antibody sequences without requiring reliable structures.

Antibody re-design and docking-aware co-design. Antibody engineering spans two complementary problems: improving a binder that already works, and creating one where none exists. Re-design addresses the first, starting from an antibody known to bind the target. DiffAb jointly generates CDR sequences and structures inside a supplied complex (Luo et al., 2022), and AbDPO refines diffusion-based designs through energy-based preference optimization (Zhou et al., 2024). Docking-aware methods relax the pose assumption, predicting the binding geometry rather than consuming

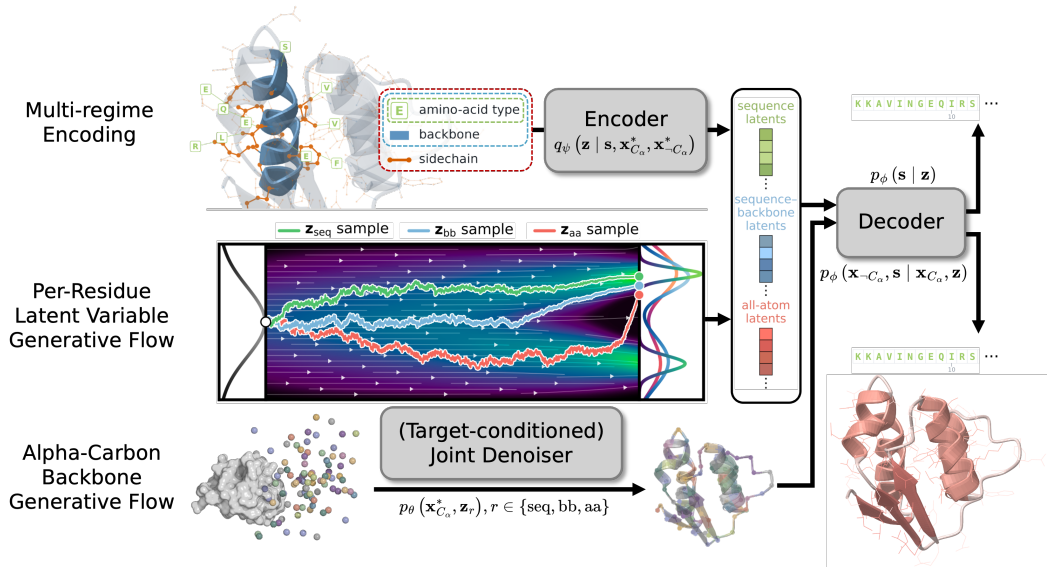


Figure 1: **Proteina Mixta overview. (a) Multi-regime autoencoder.** A shared encoder–decoder is trained with modality dropout (indicated by the * symbol) over sequence-only, backbone–sequence, and all-atom inputs. This yields compatible latents $\mathbf{z}_{\text{seq}}$, $\mathbf{z}_{\text{bb}}$, and $\mathbf{z}_{\text{aa}}$ in one latent space. **(b) Multi-regime product-space flow matching.** The flow model is trained on a mixture of regimes $(\text{MASK}, \mathbf{z}_{\text{seq}})$, $(\mathbf{x}_{C_\alpha}, \mathbf{z}_{\text{bb}})$, and $(\mathbf{x}_{C_\alpha}, \mathbf{z}_{\text{aa}})$, so sequence and all-atom data update a shared generative field. Decoder either takes z to build discrete sequences or $(\mathbf{x}_{C_\alpha}, z)$ for all-atom structures.

it: hierarchical refinement couples paratope generation with docking (Jin et al., 2022), dyMEAN designs full-atom antibodies end-to-end from an epitope and an incomplete antibody sequence (Kong et al., 2023), Igformer improves antibody–antigen interaction modeling (Wu et al., 2025), and dyAb accounts for antigen conformational change (Tan et al., 2025). Because a binding antibody is given, these methods are scored by sequence recovery, structural reconstruction, and docking accuracy. Mixta instead starts from a framework with no known affinity for the target, so the interface must be created rather than refined. De novo design focuses on the creation of novel binding interfaces different from what exists in the current PDB.

Learning from antibody sequences without structures. OAS provides large antibody repertoires without paired structural observations (Olsen et al., 2022). IgLM, AbLang2, and PRISM demonstrate the value of repertoire learning for antibody sequence modeling, with AbLang2 explicitly addressing germline bias and diversity (Kim et al., 2026; Shuai et al., 2023; Olsen et al., 2024). Mixta directly co-trains its structure-based generator on sequence-only repertoire data alongside pseudo-VHH interface examples. Observed sequences supervise antibody sequence properties without treating inferred structures as ground truth, while structural examples supply geometry and target conditioning.

3 Methods

3.1 Multi-Regime Protein Generation

Proteina Mixta extends the partially latent representation of La-Proteina (Geffner et al., 2026) from all-atom co-design to three regimes of sequence-only, backbone–sequence, and all-atom (Fig. 1). For a protein of length L, we represent each residue by a backbone C_α coordinate and a continuous latent, written jointly as $(\mathbf{x}_{C_\alpha}, \mathbf{z})$ with $\mathbf{x}_{C_\alpha} \in \mathbb{R}^{L\times 3}$ and $\mathbf{z} \in \mathbb{R}^{L\times d}$. A shared autoencoder maps whichever sequence and structural inputs are available into $\mathbf{z}$ and decodes $\mathbf{z}$ back to sequence and (when conditioned on C_α) all-atom geometry. A shared flow-matching network then generates $(\mathbf{x}_{C_\alpha}, \mathbf{z})$ across regimes, so the same denoiser can produce sequence-only samples or full all-atom structures depending on which channels are active.

Multi-regime autoencoder. Let $\mathbf{s}$ denote the amino acid sequence and $\mathbf{x}$ the all-atom coordinates, including backbone $\mathbf{x}_{C_\alpha}$ and non C_α and sidechain atoms $\mathbf{x}_{\neg C_\alpha}$. The encoder operates under three input masks:

$$\mathbf{z}_r = q_\psi(\mathcal{M}_r(\mathbf{s}, \mathbf{x}_{C_\alpha}, \mathbf{x}_{\neg C_\alpha})), \qquad r \in \{\text{seq}, \text{bb}, \text{aa}\}, \tag{1}$$

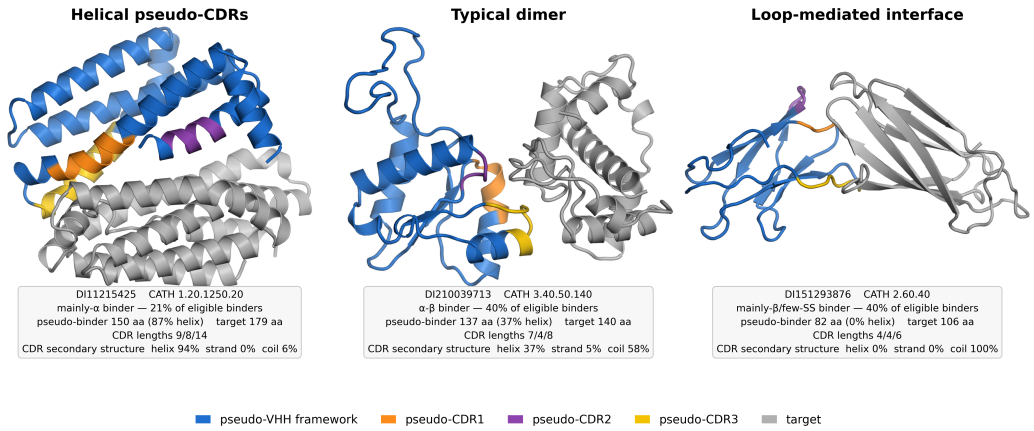


Figure 2: Representative PseudoVHH annotations applied to Teddymer protein–protein dimers. One chain is assigned as the pseudo-VHH, with framework residues shown in blue and pseudo-CDR1, pseudo-CDR2, and pseudo-CDR3 shown in orange, purple, and yellow, respectively.

where $\mathcal{M}_{\text{seq}}$ retains sequence, $\mathcal{M}_{\text{bb}}$ retains sequence and backbone geometry, and $\mathcal{M}_{\text{aa}}$ retains all-atom inputs. Here, q_ψ denotes latent encoding, including posterior sampling where applicable. Independently, the decoder receives either $\mathbf{z}_r$ alone for sequence reconstruction or $(\mathbf{x}_{C_\alpha}, \mathbf{z}_r)$ for sequence and structural reconstruction. These choices produce six training configurations, including side-chain packing from reduced encoder context (Table 3). Shared reconstruction tasks make the latent representations usable across modalities . The autoencoder is frozen during generative training.

Product-space flow matching. Mixta learns a shared generative field over the following regimes:

$$\mathbf{y}_r = \begin{cases} (\text{MASK},\, \mathbf{z}_{\text{seq}}), & r = \text{seq}, \\ (\mathbf{x}_{C_\alpha},\, \mathbf{z}_r), & r \in \{\text{bb}, \text{aa}\}. \end{cases} \tag{2}$$

The network receives the evolving state, noise-level information, and any design conditioning. Parameters are shared across regimes, allowing sequence-only examples to update the same network used for all-atom co-design. Structural losses are evaluated only where the corresponding observations exist, and loss terms are normalized over their active examples or residues. In particular, we assign coordinate placeholders on structure-free examples, which are excluded from coordinate supervision. Thus, we can train our structure-based model with sequence-only inputs.

Generation and latent warmup. Sequence-only generation evolves the latent channel with backbone information masked, then decodes the final latents into amino acid identities. All-atom generation evolves latents and backbone coordinates and decodes them jointly. We additionally evaluate a latent warm-up schedule, parameterized by sampling progress $\tau \in [0, 1]$:

$$\mathbf{y}(\tau) = \begin{cases} (\text{MASK},\, \mathbf{z}(\tau)), & 0 \le \tau < \tau_{\text{warm}}, \\ (\mathbf{x}(\tau),\, \mathbf{z}(\tau)), & \tau_{\text{warm}} \le \tau \le 1. \end{cases} \tag{3}$$

This exposes the coordinate generator to a partially denoised latent when joint generation begins. Eq. (3) specifies channel activation during denoising. Details are included in Appendix B.5.

3.2 Framework-Conditioned Nanobody Design

Pseudo-VHH construction. Given a target structure and a fixed nanobody framework, a large, biologically conserved scaffold, the task is to generate the three CDR loops that mediate binding together with the binder's global dock, i.e., its rigid-body orientation relative to the target. Building on Didi et al. (2026) we transform real and synthetic protein-protein interfaces into framework-conditioned design examples by assigning one chain as the binder and the other as the target. Three contiguous pseudo-CDRs are placed preferentially in surface-exposed, target-proximal binder regions,

with sampled lengths

$$L_{\mathrm{CDR1}} \in [5, 20], \qquad L_{\mathrm{CDR2}} \in [4, 16], \qquad L_{\mathrm{CDR3}} \in [6, 35]. \tag{4}$$

The region with the most target contacts is assigned as CDR3, followed by CDR1 and CDR2. Remaining residues constitute the pseudo-framework. Hotspots are selected from target residues contacted by pseudo-CDRs, with greater weight on CDR3 contacts. Both Teddymer (TED-derived dimers (Didi et al., 2026)) and PDB interface examples receive synthetic framework/CDR annotations. These labels define a design task on general interfaces without requiring the binder to have a native VHH fold (Fig. 2).

Independent framework and target conditioning. Following the target-aware architecture of Proteina-Complexa, we supply the framework without any indication of how it sits against the target. Each framework is centered and randomly reoriented,

$$\widetilde{\mathbf{x}}_i^{\mathrm{fw}} = \mathbf{R}_{\mathrm{fw}} \left(\mathbf{x}_i^{\mathrm{fw}} - \boldsymbol{\mu}_{\mathrm{fw}} \right), \qquad \mathbf{R}_{\mathrm{fw}} \sim \mathrm{Unif}(\mathrm{SO}(3)), \tag{5}$$

which preserves the scaffold's internal geometry while carrying no information about its native pose. Residue features hold the framework sequence, coordinates and masks, torsions, secondary structure, and region identities. The target is centered separately and enters as concatenated context tokens with its sequence, geometry, secondary structure, interface descriptors, and hotspot labels. The model must therefore recover both the variable-region conformations and the binder's docked orientation.

3.3 Training Details

Proteina Mixta Autoencoder and Monomer Training We train our mixed multimodal autoencoder with the ability to encode and decode each of the modalities following Table 3. We then train a monomer generative model following Geffner et al. (2026), where about half of data inputs are sequence only via random structure dropout. The autoencoder is frozen and used to obtain the latent targets for training the latent flow-matching model. Please see Appendix A-B for more details.

Complexa-lineage baseline. Beyond Proteina Mixta we introduce Proteina-Nanocuerpo as a separate baseline as Proteina-Complexa cannot handle joint framework and target conditioning. We adapt and finetune Proteina-Complexa directly with pseudo-VHH conditioning while retaining its pretrained trunk and original frozen autoencoder. Compatible projection extensions introduce the new conditioning features while preserving pretrained features. Please see Appendix D for details.

Proteina Mixta Nanobody Training. The multimodal autoencoder is trained on AFDB monomers, then fine-tuned on individual PDB and SAbDab2 chains sampled with equal probability. We freeze its 250k-step EMA checkpoint. Because this autoencoder defines the generator's latent operating space, no pretrained trunk transfers to it, and we train the trunk from scratch. Later stages warm-start from Mixta checkpoints that share the same frozen representation. To reduce the per-target generative variance, we use four of seven selected targets for weight arithmetic inspired from recent works in LLM post training to balance success on multiple targets. See Appendix E for training specifics.

4 Experiments

4.1 Monomer Design

To establish the validity of the mixed multimodal latent design, we evaluate whether Proteina Mixta can learn from multiple sequence–structure regimes while retaining strong unconditional monomer generation. We first compare the original La-Proteina autoencoder with the Mixta autoencoder, and then compare unconditional samples from the retrained monomer La-Proteina baseline (to match the training data) and Mixta trained with $\sim$415K AFDB structures (Lin et al., 2024). We generate 100 unconditional samples at length $\ell \in \{50, 100, 150, 200, 250\}$ and report co-designability, designability, mean pLDDT, and Foldseek (all-atom) / MMseqs2 (sequence) diversity, following Geffner et al. (2026). Details are included in Appendix B.2.

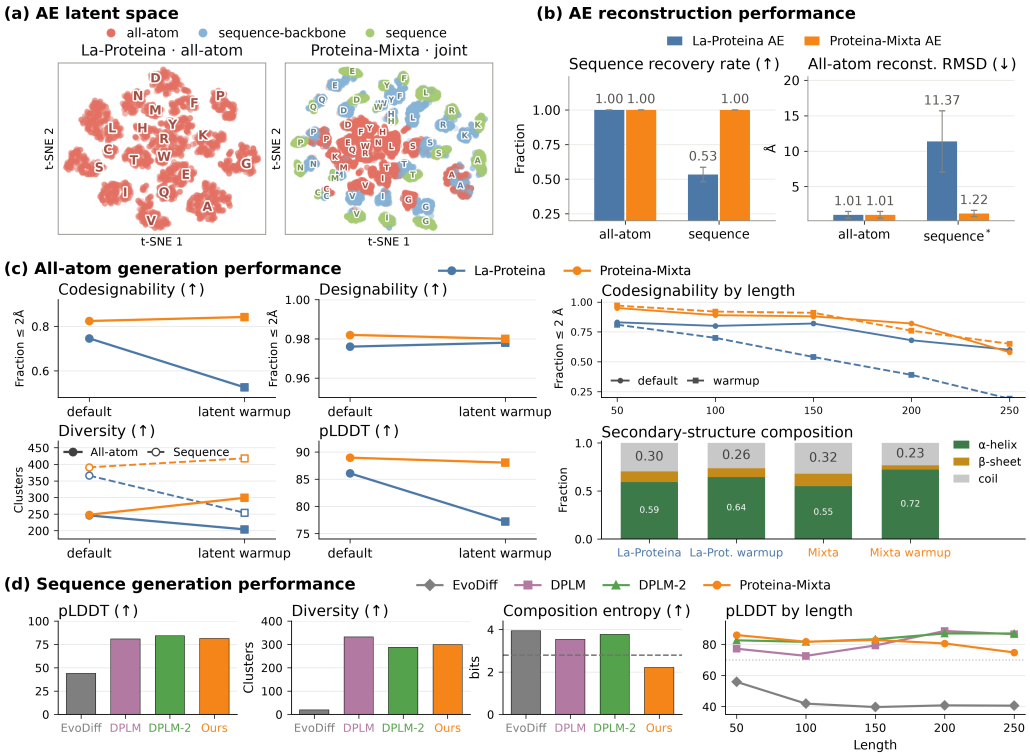


Figure 3: **Proteina Mixta monomer design.** **(a)** Autoencoder latent space t-SNE projections of latents encoded from sequence-only, backbone–sequence, and all-atom input regimes. La-Proteina only supports all-atom encoding. **(b)** Sequence recovery rate and all-atom reconstruction RMSD under all-atom and sequence-only encode–decode reconstruction. Note that decoder always takes C_α as additional input when doing all-atom reconstruction, even when encoder takes sequence* input. **(c)** All-atom monomer generation by the replicated La-Proteina baseline and Proteina Mixta, with default and latent warm-up sampling. **(d)** Baseline comparison of sequence monomer generation.

Mixta autoencoder couples sequence and structural information. Fig. 3a diagnoses how the two autoencoders behave across input regimes. The original La-Proteina autoencoder produces latent space with semantically meaningful clusters, while it only supports all-atom encoding. By contrast, the Mixta autoencoder is able to encode inputs of different regimes into a joint latent space with structured latent geometry. Both autoencoders retain high sequence recovery and all-atom reconstruction accuracy when all-atom information is given. However, under sequence-only encoding, La-Proteina's sequence recovery falls to 53.3% and its all-atom reconstruction RMSD rises to 11.4 Å, whereas Mixta preserves near-perfect sequence recovery and competitive all-atom RMSD. This suggests that packing objectives teach sequence latents to retain structural information, which is potentially beneficial for shared diffusion learning across latent regimes.

Mixta retains strong all-atom and sequence generation performance. Fig. 3c compares Proteina Mixta with the replicated La-Proteina baseline under matched unconditional sampling. With the default C_α early denoising schedule (Geffner et al., 2026), Mixta improves both co-designability, ProteinMPNN designability, and pLDDT. The improvement in co-designability is observed across the evaluated length range rather than being confined to short proteins. Meanwhile, its generated structures cover a comparable level of all-atom diversity and stronger sequence diversity, measured based on Foldseek- and MMseqs2-based clustering. Mixta also maintains a mixed secondary-structure distribution, showing that its gains do not arise from alpha helical collapse.

Fig. 3d evaluates sequence-only generation against EvoDiff (Alamdari et al., 2023), DPLM (Wang et al., 2024), and DPLM-2 (Wang et al., 2025b). Mixta is competitive in terms of both mean pLDDT and sequence diversity compared to DPLM and DPLM-2. Note that the monomer Proteina Mixta model is trained with only 415K structural data, while DPLM family inherits signals from 42 million sequences in UniRef50 (Suzek et al., 2015).

Table 1: **Nanobody design success.** Rates use 1,000 designs per target. Cells report unique success with no PSH, PPC, or PNC red flags (TNP-clean), followed by success before TNP screening in parentheses. Oracle-based methods use structure-predictor feedback during design. Times are per output without any acceleration efforts on NVIDIA A100 80GB GPU. Bold and underline indicate the best and second-best. The first four targets were used in checkpoint selection.

	Method	Sequence redesign	PD-L1	BHRF1	Insulin	TrkA	IL-3	CTLA-4	CEACAM6	Inference time (s)
Oracle free	BoltzGen	Boltz-IF	0.8% (0.8%)	1.3% (1.4%)	8.4% (8.7%)	0.5% (0.5%)	0.0% (0.0%)	0.1% (0.1%)	0.7% (0.8%)	22
	RFantibody	ProteinMPNN	0.6% (0.6%)	0.2% (0.2%)	6.5% (6.8%)	0.2% (0.2%)	0.1% (0.1%)	0.0% (0.0%)	0.1% (0.1%)	61
	Proteina-Nanocuerpo	None	**27.0%** (28.1%)	3.0% (7.4%)	3.3% (4.2%)	3.6% (4.8%)	0.2% (0.2%)	0.4% (0.7%)	0.6% (1.4%)	**4.4**
	Proteina Mixta	None	9.1% (35.8%)	**52.4%** (52.4%)	**28.7%** (35.6%)	**24.3%** (29.0%)	**0.4%** (0.4%)	**8.9%** (12.2%)	**3.0%** (3.4%)	21
Oracle based	Germinal	AbMPNN	4.8% (7.2%)	3.9% (5.4%)	14.5% (19.4%)	2.9% (3.8%)	1.9% (2.7%)	12.9% (14.9%)	3.7% (4.4%)	269
	BindCraft2	ProteinMPNN	1.0% (1.6%)	3.9% (6.1%)	12.0% (27.2%)	10.7% (22.3%)	12.9% (22.9%)	**28.9%** (55.4%)	3.7% (9.4%)	425
	ESMFold2	None	**23.9%** (60.4%)	9.8% (27.0%)	1.7% (46.1%)	4.7% (21.1%)	**18.2%** (49.5%)	12.2% (70.0%)	**50.3%** (74.0%)	373
	Proteina Mixta$_{\text{search}}$	None	17.7% (59.9%)	**61.7%** (61.9%)	**35.0%** (48.1%)	**30.9%** (34.2%)	10.0% (10.4%)	14.0% (22.7%)	22.8% (24.4%)	**54**

Latent-first denoising improves multi-regime generation. The latent warm-up schedule in Eq. (3) improves Mixta's quality–diversity trade-off, increasing co-designability and structural diversity while largely preserving pLDDT and ProteinMPNN designability. Applying the same schedule to La-Proteina instead reduces every metric except designability, indicating that latent warm-up is not a generic inference heuristic: its benefit is tied to the multi-regime autoencoder and diffusion training.

Framework-constrained nanobody generation. Continued training on structure-free OAS sequences, together with a new inference-time decoder guidance, enables framework-constrained VHH generation in both sequence-only and all-atom modes, even when framework conditioning is not explicitly learned. See Appendix C for details.

4.2 De Novo Nanobody Design

A controlled benchmark extending Germinal. We extend Germinal's framework-constrained design setting (Mille-Fragoso et al., 2026) to seven targets: PD-L1, BHRF1, insulin, TrkA, IL-3, CTLA-4, and CEACAM6. All methods receive the hNbBCII10 framework with HCDR1/HCDR2/HCDR3 lengths fixed to $11/8/18$[1] and are evaluated on 1,000 output designs per target and reported setting. We distinguish generation without structure-predictor feedback from predictor-guided optimization or search, following Proteina-Complexa (Didi et al., 2026). Sequence redesign and inference cost are reported separately: matched output counts do not imply matched computational budgets. No known binder for these targets uses this common *in vitro* validation framework (Bennett et al., 2026).

Structural evaluation. Protenix-v2 evaluates designs using independent binder and target templates, with binder side-chains and the binder–target rigid-body transformation withheld (Zhang et al., 2026). Extending Germinal-style confidence filtering, a success requires framework-sequence recovery and

$$\text{ipAE} \leq 6\,\text{Å}, \quad \text{pTM} \geq 0.83, \quad \text{ipTM} \geq 0.75, \quad \text{pDockQ2} \geq 0.45. \tag{6}$$

These confidence criteria assess predicted complex and interface quality.

Developability. We evaluate with the Therapeutic Nanobody Profiler (TNP Gordon et al. (2026)), a nanobody-specific in silico developability scorer. TNP folds structures from sequence using NanoBodyBuilder2 (Abanades et al., 2023) and compares six descriptors with a reference set of clinical-stage nanobodies. Red flags indicate values outside the biologically acceptable range. We report surface hydrophobicity (PSH), positive-charge patches (PPC), and negative-charge patches (PNC):

$$R_{\text{PSH}} = \{\text{PSH} < 73.40 \vee \text{PSH} > 155.47\}, \quad R_{\text{PPC}} = \{\text{PPC} > 1.18\}, \quad R_{\text{PNC}} = \{\text{PNC} > 1.88\}. \tag{7}$$

We define *TNP-clean success* as success with no red flag on these three descriptors. This computational screen complements structural confidence. It targets TNP experimental developability correlation. See Appendix F for metrics and validation controls.

Proteina Mixta improves the yield of TNP-clean designs. Table 1 reports TNP-clean success, with unfiltered TNP unique success in parentheses. Without sequence redesign, base Mixta leads the

[1] Germinal requires the reference nanobody scaffold's CDR lengths; we use these across methods for fairness.

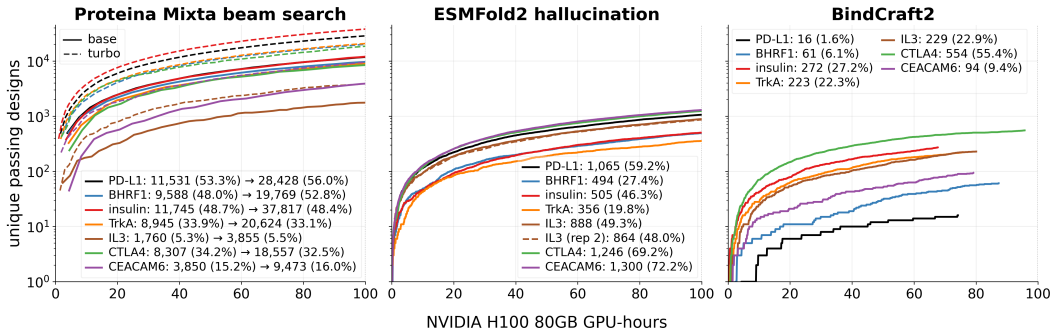


Figure 4: **Nanobody generation under compute budgets.** Cumulative unique passing designs versus generation GPU-hours for Proteina Mixta beam search, ESMFold2, and BindCraft2. Legends report final counts and passing percentages, with arrows indicating base-to-turbo results for Mixta.

oracle-free methods on six of seven targets after TNP screening: its clean rates on BHRF1, insulin, and TrkA are 52.4%, 28.7%, and 24.3%, versus 1.3%, 8.4%, and 0.5% for BoltzGen with Boltz-IF. PD-L1 is the exception, where Mixta's 35.8% unfiltered success falls to 9.1% after screening, below our own Proteina Nanocuerpo baseline's 27.0% clean rate. Structural success alone therefore obscures substantial target-dependent differences in predicted liabilities. Mixta's higher inference cost relative to Nanocuerpo comes from reintroducing the pair-bias attention trunk with triangular multiplication operations from Proteina(Geffner et al., 2025). See Appendix G for details.

Search increases yield further. Following Didi et al. (2026) we run Proteina Mixta with test time scaling (beam search) to improve both unfiltered and TNP-clean success over base Mixta on all seven targets. Table 1 reports TNP-clean success among the first 1,000 generated designs. Among oracle-based methods, Mixta ranks first on BHRF1, insulin, and TrkA; second on PD-L1, CTLA-4, and CEACAM6; and third on IL-3. BindCraft2 leads on CTLA-4, while ESMFold2 leads on PD-L1, IL-3, and CEACAM6. However, ESMFold2's insulin success drops from 46.1% before TNP screening to 1.7% afterward, highlighting the distinction between structural confidence and predicted surface liabilities. We present the full test-time scaling curves for the raw unique success rate under matched compute in Fig. 4, with further details in Appendix G. Proteina Mixta obtains an order of magnitude more successful designs than recent hallucination-based methods[2], and scales further in "turbo" mode (see Appendix G.3 (Anthropic, 2026)).

4.3 Sequence-Only Co-Training Improves Diversity and Sequence Quality

Mixta enables direct co-training on antibody repertoire sequences without refolding or inferred structural labels. We replace 10% of pseudo-VHH Teddymer training examples with liability-filtered OAS heavy-chain sequences (Kim et al., 2026) and compare against a structural control initialized from the same checkpoint. Both models generate in all-atom mode, testing whether sequence-only supervision improves joint sequence–structure outputs.

Metrics and comparisons. We compare matched training checkpoints across PD-L1, BHRF1, insulin, and TrkA, generating 1,000 designs per setting. Computational success is the fraction passing all confidence gates (Sec. 4.2). CDR3 sequences are grouped at 80% identity, and diversity is the effective number of families, $1/\sum_i p_i^2$, where p_i is each family's design fraction. The liability rate counts designs carrying at least one evaluated CDR motif, a potential N-linked glycosylation site or a specified W/G/Y/S repeat, distinct from TNP's structure-based descriptors.

Sequence supervision transfers to all-atom generation. Table 2 shows OAS co-training increases effective CDR3 diversity by 26% and reduces sequence-liability rates, while AbLang2 pseudo-perplexity is essentially unchanged. Because OAS sequences were filtered using the same liability rules we evaluate against, this reduction shows that selected sequence properties transfer through co-training to the all-atom structure generator.

[2]BindCraft2 is concurrent work with code released on 9/20/2026 and we plot first 1000 samples.

Table 2: **Effect of replacing 10% of structural training examples with liability-filtered OAS sequences.** Models share initialization, training, and inference settings.

	In silico success rate (%) ↑	AbLang2 pseudo-perplexity ↓	Effective CDR3 families ↑	CDR liability rate (%) ↓
Base	15.77	3.87	242.6	17.65
Base + OAS	10.84	3.87	304.9	15.30
Relative change	−31%	0%	+26%	−13%

Repertoire supervision introduces an interface trade-off. Aggregate interface success decreases from 15.77% to 10.84%, consistent with a mismatch between the added supervision and target-conditioned VHH design: OAS examples lack antigen context, are not restricted to nanobodies, and include conventional heavy chains that require a light-chain partner. Natural repertoire sequences therefore are not necessarily suitable standalone binders against these benchmark targets.

Our structure-prediction criteria reject over half of the known PDB VHH complexes in our reference set (Appendix F.2), so confidence-based filtering is imperfect even for experimentally observed interactions. We therefore read in silico success alongside open-source measures of sequence diversity and developability, not as a definitive measure of binder quality: on that reading, sequence-only co-training improves CDR3 diversity and reduces sequence liabilities despite lower predicted interface success. The unchanged AbLang2 pseudo-perplexity likely reflects low-temperature sampling, standard in protein design, which does not draw from the true data distribution. Sequence-only data therefore offers a lever on design properties that structural data alone does not.

4.4 VHH Sequence-Only Generation

We compare Proteina Mixta in sequence-only mode with the *in vitro* validated ADAPT model (Yeh et al., 2026) and an antigen-free AbLang2 baseline (Olsen et al., 2024). Under an independent refolding oracle (Protenix-v2, App. F.2), Mixta and ADAPT each produce exactly one PD-L1 success among 1,000 designs (Fig. 5), matching the antigen-free baseline, with no successes on the remaining targets. Near-identical failure across otherwise very different generators underscores the difficulty of paratope design from sequence alone, though ADAPT's reported *in vitro* binding results caution against reading sparse computational (un)successes as evidence of (absent) binding, given how current folding models struggle on known nanobodies (Park et al., 2026).

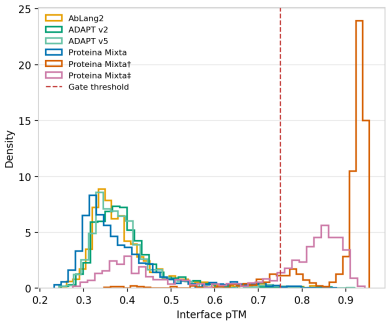


Figure 5: Sequence-only PD-L1 interface pTM. †/‡: RL-finetuned

To test whether this ceiling is specific to the base model, we post-train Mixta's sequence-only mode with online reinforcement learning (RL), rewarding ESMFold2-fast confidence. Under Protenix-v2 oracle, RL fine-tuning yields large PD-L1 gains compared to the base model (Appendix Fig. 13): given a structural training signal, sequence-only generation is not inherently capped at antigen-free performance. Neither checkpoint transfers the gain to the four untrained targets. Mixta's shared sequence–structure framework thus brings advances in sequence-only modeling, RL included, directly into joint generation. See Appendix H for training details, diversity analysis, and all target results.

5 Conclusion

We introduced Proteina Mixta, a multimodal latent flow-matching model that unifies sequence-only generation and all-atom co-design in one shared residue-level representation, reaching state-of-the-art monomer co-design and competitive sequence generation. Pseudo-VHH supervision extends Mixta to nanobody design without native CDR annotations, and co-training on sequence-only OAS repertoires increases CDR3 diversity while reducing sequence liabilities. Across seven targets, Mixta outperforms every oracle-free prior baseline without sequence redesign, and inference-time search increases yield further. Performance remains target-dependent, and computational success still requires experimental validation. These results point to multimodal latent modeling as a way to combine broad sequence supervision with precise structural control.

AI USE STATEMENT

We used generative AI tools to assist with implementing methods, writing and executing code, creating and modifying scientific figures, interpreting results, and drafting and revising manuscript text. AI tools also assisted with literature searches, reference formatting, and proposing hyperparameters. The authors validated AI-assisted implementations for correctness and made the final methodological and model-selection decisions. All AI-assisted material was reviewed by the authors, who take responsibility for the final content of this work, including its text, claims, code, figures, and other artifacts.

REFERENCES

Brennan Abanades, Wing Ki Wong, Fergus Boyles, Guy Georges, Alexander Bujotzek, and Charlotte M. Deane. ImmuneBuilder: Deep-learning models for predicting the structures of immune proteins. *Communications Biology*, 6(1):575, 2023. doi: 10.1038/s42003-023-04927-7.

Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex X. Lu, Nicolo Fusi, Ava P. Amini, and Kevin K. Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, 2023. doi: 10.1101/2023.09.11.556673. URL https://www.biorxiv.org/content/10.1101/2023.09.11.556673v1. Version 1. Version 2 (2024) adds three authors.

Anthropic. Uplifting biomolecular modeling: Inference optimization kits. GitHub software repository, 2026. URL https://github.com/anthropics/uplifting-biomolecular-modeling. Protenix v2 optimization kit. Revision f4f62fa6592a (full commit: https://github.com/anthropics/uplifting-biomolecular-modeling/tree/f4f62fa6592ae4938d49b1757bea0cfeff9f468e). Accessed September 24, 2026.

Aureka AI OpenDDE project. Folding, reasoning, and scaling with open-source drug discovery engine, 2026. URL https://arxiv.org/abs/2607.03787.

Sankar Basu and Björn Wallner. DockQ: A quality measure for protein-protein docking models. *PLOS ONE*, 11(8):e0161879, 2016. doi: 10.1371/journal.pone.0161879.

Nathaniel R. Bennett, Joseph L. Watson, Robert J. Ragotte, Andrew J. Borst, DéJenaé L. See, Connor Weidle, Riti Biswas, Yutong Yu, Ellen L. Shrock, Russell Ault, Philip J. Y. Leung, Buwei Huang, Inna Goreshnik, John Tam, Kenneth D. Carr, Benedikt Singer, Cameron Criswell, Basile I. M. Wicky, Dionne Vafeados, Mariana Garcia Sanchez, Ho Min Kim, Susana Vázquez Torres, Sidney Chan, Shirley M. Sun, Timothy T. Spear, Yi Sun, Keelan O'Reilly, John M. Maris, Nikolaos G. Sgourakis, Roman A. Melnyk, Chang C. Liu, and David Baker. Atomically accurate de novo design of antibodies with RFdiffusion. *Nature*, 649(8095):183–193, 2026. doi: 10.1038/s41586-025-09721-5. URL https://doi.org/10.1038/s41586-025-09721-5.

Helen M. Berman, John D. Westbrook, Zukang Feng, Gary L. Gilliland, Talapady N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000. doi: 10.1093/nar/28.1.235.

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Z. Wu, Jin Sub Lee, Elise S. Bruguera, Jehan A. Keval, Mykhailo Kopylov, John E. Pak, Wesley Wu, Neil Thomas, Samson Mataraso, Alvin Hsu, Ashton C. Trotman-Grant, Kilian Fatras, Allan dos Santos Costa, Rohil Badkundri, Halil Akın, Deniz Oktay, Jonathan Deaton, Elizabeth Montabana, Hrishita Sitwala, Yue Yu, Marius Wiggert, Dylan Alexander Carlin, Anthony W. Goering, Tomasz Blazejewski, McCullen Sandora, Michael Hla, Tina Z. Jia, Leon H. Kloker, Nicholas J. Sofroniew, Masatoshi Uehara, Jassi Pannu, Sharrol Bachas, Daniel S. Liu, Tom Sercu, and Alexander Rives. Language modeling materializes a world model of protein biology. *bioRxiv*, 2026. doi: 10.64898/2026.06.03.729735.

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

Henriette L. Capel, Odysseas Vavourakis, Benjamin H. Williams, Christopher R. Taylor, and Charlotte M. Deane. SAbDab2: The structural antibody database in the age of machine learning. *bioRxiv*, 2026. doi: 10.64898/2026.06.16.732554. URL https://www.biorxiv.org/content/10.64898/2026.06.16.732554v1. Posted 20 June 2026.

Jiho Choi, Donggyun Kim, Chanhyuk Lee, and Seunghoon Hong. Revisiting weight averaging for model merging, 2024.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael M. Bronstein, Martin Steinegger, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Scaling atomistic protein binder design with generative pretraining and test-time compute. In *International Conference on Learning Representations (ICLR)*, 2026. Oral presentation.

Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Zídek, Russ Bates, Sam Blackwell, Jason Yim, Olaf Ronneberger, Sebastian Bodenstein, Michal Zielinski, Alex Bridgland, Anna Potapenko, Andrew Cowie, Kathryn Tunyasuvunakool, Rishub Jain, Ellen Clancy, Pushmeet Kohli, John Jumper, and Demis Hassabis. Protein complex prediction with AlphaFold-Multimer. *bioRxiv*, 2022. doi: 10.1101/2021.10.04.463034. URL https://doi.org/10.1101/2021.10.04.463034.

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Proteina: Scaling flow-based protein structure generative models. In *International Conference on Learning Representations (ICLR)*, 2025.

Tomas Geffner, Kieran Didi, Zhonglin Cao, Danny Reidenbach, Zuobai Zhang, Christian Dallago, Emine Kucukbenli, Karsten Kreis, and Arash Vahdat. La-Proteina: Atomistic protein generation via partially latent flow matching. In *International Conference on Learning Representations (ICLR)*, 2026.

Gemma L. Gordon, João Gervasio, Colby Souders, and Charlotte M. Deane. Characterising nanobody developability to improve therapeutic design using the therapeutic nanobody profiler. *Communications Biology*, 9:344, 2026. doi: 10.1038/s42003-026-09594-y.

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

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Antibody-antigen docking and design via hierarchical structure refinement. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10217–10227. PMLR, 2022. URL https://proceedings.mlr.press/v162/jin22a.html.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Jeonghyeon Kim, Nathaniel Blalock, Ameya Kulkarni, Kensuke Nakamura, and Philip A. Romero. Explicit representation of germline and non-germline residues improves antibody language modeling. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. doi: 10.64898/2026.05.06.723387. URL https://icml.cc/virtual/2026/poster/64954. Preprint: bioRxiv 2026.05.06.723387.

Xiangzhe Kong, Wenbing Huang, and Yang Liu. End-to-end full-atom antibody design. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17409–17429. PMLR, 2023. URL https://proceedings.mlr.press/v202/kong23c.html.

Yu-Ang Lee, Ching-Yun Ko, Tejaswini Pedapati, I-Hsin Chung, Mi-Yen Yeh, and Pin-Yu Chen. STAR: Spectral truncation and rescale for model merging. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 496–505, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-190-2. doi: 10.18653/v1/2025.naacl-short.42. URL https://aclanthology.org/2025.naacl-short.42/.

Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2. *arXiv preprint arXiv:2405.15489*, 2024.

Yeqing Lin, Minji Lee, Aakarsh Vermani, Ellena Jiang, Sebastiaan De Cooman, Matej Špeťko, and Mohammed AlQuraishi. Fast and ultra-capable protein design: Advancing the frontier through atomistic SE(3)-equivariance with Genie 3. *bioRxiv*, 2026. doi: 10.64898/2026.05.01.722168. URL https://www.biorxiv.org/content/10.64898/2026.05.01.722168v1.

Amy X. Lu, Jason Yang, Yueming Long, Theophile Lambert, Wilson Yan, Sarah A. Robinson, Kevin K. Yang, Vladimir Gligorijevic, Kyunghyun Cho, Richard Bonneau, Nathan C. Frey, Frances Arnold, and Pieter Abbeel. Controllable all-atom protein generation with latent diffusion. *bioRxiv*, 2026. doi: 10.1101/2024.12.02.626353. URL https://www.biorxiv.org/content/10.1101/2024.12.02.626353v3.

Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=jSorGn2Tjg.

Luis S. Mille-Fragoso, Claudia L. Driscoll, John N. Wang, Haoyu Dai, Talal Widatalla, Jim L. Zhang, Xiaowei Zhang, Bing Rao, Liang Feng, Brian L. Hie, and Xiaojing J. Gao. Efficient generation of epitope-targeted antibodies with Germinal. *Nature Biotechnology*, 2026. doi: 10.1038/s41587-026-03187-0. URL https://doi.org/10.1038/s41587-026-03187-0. Published online 23 June 2026.

Serge Muyldermans. Nanobodies: Natural single-domain antibodies. *Annual Review of Biochemistry*, 82:775–797, 2013. doi: 10.1146/annurev-biochem-063011-092449. URL https://doi.org/10.1146/annurev-biochem-063011-092449.

Tobias H. Olsen, Fergus Boyles, and Charlotte M. Deane. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1):141–146, 2022. doi: 10.1002/pro.4205. URL https://onlinelibrary.wiley.com/doi/10.1002/pro.4205.

Tobias H. Olsen, Iain H. Moal, and Charlotte M. Deane. Addressing the antibody germline bias and its effect on language models for improved antibody design. *Bioinformatics*, 40(11):btae618, 2024. doi: 10.1093/bioinformatics/btae618. URL https://academic.oup.com/bioinformatics/article/40/11/btae618/7845256.

Pacesa Lab. BindCraft2: Protein binder design software. GitHub repository, 2026. URL https://github.com/PacesaLab/BindCraft2. Accessed September 23, 2026.

Minjae Park, Roman Nett, Brian Petersen, and Arvind Sivasubramanian. Benchmarking antibody–antigen co-folding on human monomeric antigens. *bioRxiv*, 2026. doi: 10.64898/2026.08.20.745880. URL https://doi.org/10.64898/2026.08.20.745880.

Matthew I. J. Raybould, Claire Marks, Konrad Krawczyk, Bruck Taddese, Jaroslaw Nowak, Alan P. Lewis, Alexander Bujotzek, Jiye Shi, and Charlotte M. Deane. Five computational developability guidelines for therapeutic antibody profiling. *Proceedings of the National Academy of Sciences*, 116(10):4025–4030, 2019. doi: 10.1073/pnas.1810576116.

Constantin Schneider, Matthew I. J. Raybould, and Charlotte M. Deane. SAbDab in the age of biotherapeutics: Updates including SAbDab-nano, the nanobody structure tracker. *Nucleic Acids Research*, 50(D1):D1368–D1372, 2022. doi: 10.1093/nar/gkab1050. URL https://doi.org/10.1093/nar/gkab1050.

Richard W. Shuai, Jeffrey A. Ruffolo, and Jeffrey J. Gray. IgLM: Infilling language modeling for antibody sequence design. *Cell Systems*, 14(11):979–989.e4, 2023. doi: 10.1016/j.cels.2023.10.001. URL https://doi.org/10.1016/j.cels.2023.10.001.

Hannes Stark, Felix Faltings, MinGyu Choi, Yuxin Xie, Eunsu Hur, Timothy O'Donnell, Anton Bushuiev, Talip Uçar, Saro Passaro, Weian Mao, et al. BoltzGen: Toward universal binder design. *bioRxiv*, pp. 2025.11.20.689494, 2025. doi: 10.1101/2025.11.20.689494. URL https://www.biorxiv.org/content/10.1101/2025.11.20.689494v1.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017. doi: 10.1038/nbt.3988.

Baris E. Suzek, Yuqi Wang, Hongzhan Huang, Peter B. McGarvey, Cathy H. Wu, and the UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 03 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu739. URL https://doi.org/10.1093/bioinformatics/btu739.

Cheng Tan, Yijie Zhang, Zhangyang Gao, Yufei Huang, Haitao Lin, Lirong Wu, Fandi Wu, Mathieu Blanchette, and Stan Z. Li. dyAb: Flow matching for flexible antibody design with AlphaFold-driven pre-binding antigen. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 782–790, 2025. URL https://ojs.aaai.org/index.php/AAAI/article/view/32061.

Michel van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42(2):243–246, 2024. doi: 10.1038/s41587-023-01773-0.

Yannick Vogt, Mehdi Naouar, Maria Kalweit, Christoph Cornelius Miething, Justus Duyster, Joschka Boedecker, and Gabriel Kalweit. Betterbodies: Reinforcement learning guided diffusion for antibody sequence design. *arXiv preprint arXiv:2409.16298*, 2024a.

Yannick Vogt, Mehdi Naouar, Maria Kalweit, Christoph Cornelius Miething, Justus Duyster, Roland Mertelsmann, Gabriel Kalweit, and Joschka Boedecker. Stable online and offline reinforcement learning for antibody CDRH3 design. *arXiv preprint arXiv:2401.05341*, 2024b.

Ke Wang, Nikolaos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. LiNeS: Post-training layer scaling prevents forgetting and enhances model merging. In *International Conference on Learning Representations (ICLR)*, 2025a.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52309–52333. PMLR, 2024. URL https://proceedings.mlr.press/v235/wang24ct.html.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. DPLM-2: A multimodal diffusion protein language model. In *International Conference on Learning Representations (ICLR)*, 2025b. URL https://openreview.net/forum?id=5z9GjHgerY.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. doi: 10.1038/s41586-023-06415-8.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 2022.

Jiayang Wu, Xingyi Zhang, Xiangyu Dong, Kun Xie, Ziqi Liu, Wensheng Gan, Sibo Wang, and Le Song. Towards more accurate full-atom antibody co-design, 2025. URL https://arxiv.org/abs/2502.19391.

Sheng Xu, Qiantai Feng, Lifeng Qiao, Hao Wu, Tao Shen, Yu Cheng, Shuangjia Zheng, and Siqi Sun. Benchmarking all-atom biomolecular structure prediction with FoldBench. *Nature Communications*, 17(1):442, 2026. doi: 10.1038/s41467-025-67127-3. URL https://www.nature.com/articles/s41467-025-67127-3. Published online 4 December 2025.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Zhuo Yang, Jiaying He, Jiaqing Xie, Daolang Wang, Xipeng Qiu, Yuxin Wang, Tianfan Fu, and Beilun Wang. ABOPD: Antibody CDR design via on-policy distillation. *arXiv preprint arXiv:2607.18835*, 2026.

Hugh Yeh, Nikša Praljak, Zhijie Chen, Thomas M. Walsh, Juan L. Mendoza, Jun Huang, and Andrew L. Ferguson. Antibody design with steerable discrete diffusion. In *ICLR 2026 Workshop on Integrating Generative and Experimental Platforms for Biomolecular Design*, 2026. URL https://openreview.net/forum?id=k3UWKDmtMV.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR, 2024.

Vinicius Zambaldi, David La, Alexander E. Chu, Harshnira Patani, Amy E. Danson, Tristan O. C. Kwan, Thomas Frerix, Rosalia G. Schneider, David Saxton, Ashok Thillaisundaram, Zachary Wu, Isabel Moraes, Oskar Lange, Eliseo Papa, Gabriella Stanton, Victor Martin, Sukhdeep Singh, Lai H. Wong, Russ Bates, Simon A. Kohl, Josh Abramson, Andrew W. Senior, Yilmaz Alguel, Mary Y. Wu, Irene M. Aspalter, Katie Bentley, David L. V. Bauer, Peter Cherepanov, Demis Hassabis, Pushmeet Kohli, Rob Fergus, and Jue Wang. De novo design of high-affinity protein binders with AlphaProteo. *arXiv preprint arXiv:2409.08022*, 2024. doi: 10.48550/arXiv.2409.08022. URL https://arxiv.org/abs/2409.08022.

Yuxuan Zhang, Chengyue Gong, Jinyuan Sun, Jiaqi Guan, Milong Ren, Song Xue, Hanyu Zhang, Wenzhi Ma, Zhenyu Liu, Xinshi Chen, and Wenzhi Xiao. Protenix-v2: Broadening the reach of structure prediction and biomolecular design. *bioRxiv*, 2026. doi: 10.64898/2026.04.10.717613. URL https://www.biorxiv.org/content/10.64898/2026.04.10.717613v1.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. Model extrapolation expedites alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1025–1041, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.51.

Xiangxin Zhou, Dongyu Xue, Ruizhe Chen, Zaixiang Zheng, Liang Wang, and Quanquan Gu. Antigen-specific antibody design via direct energy-based preference optimization. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/daef77101ba5711084a57442c8cf2709-Abstract-Conference.html.

Wensi Zhu, Aditi Shenoy, Petras Kundrotas, and Arne Elofsson. Evaluation of AlphaFold-Multimer prediction on multi-chain protein complexes. *Bioinformatics*, 39(7):btad424, 2023. doi: 10.1093/bioinformatics/btad424. URL https://academic.oup.com/bioinformatics/article/39/7/btad424/7219714.

Appendix Contents

A Multimodal Autoencoder Training and Ablations **18**

A.1 Autoencoder Architecture and Training Setup 18

A.2 Alternative Autoencoders 18

A.3 Latent Space Qualitative Analysis 19

B Monomer Flow-Matching Training and Evaluation **20**

B.1 Multimodal Diffusion Training 20

B.2 Monomer Evaluation Protocol and Metrics 21

B.3 All-atom Generation against Baseline 21

B.4 All-atom Generation Comparison across Autoencoders 22

B.5 Interaction with Latent Warm-up Denoising 22

C Monomer Extension to Framework-constrained Nanobody Generation **23**

C.1 Inference-Time Guidance through the Frozen Decoder 24

C.2 Guidance Protocols 24

C.3 Nanobody Generation Across Modes from OAS Adaptation 25

D Proteina Nanocuerpo Training details **26**

D.1 Initialization and Weight Splicing 26

D.2 Conditioning Features 26

D.3 Pseudo-VHH Training Data 26

D.4 Conditioning Dropouts 28

D.5 Optimization 28

D.6 A CDR-focused redefinition of hotspots. 28

E Proteina Mixta Nanobody training and architecture details **29**

E.1 Architecture 29

E.2 Training Details 29

E.3 Reducing per-target stochasticity with weight arithmetic 30

F Nanobody Metrics **34**

F.1 Target Selection 34

F.2 Structure Prediction Metrics 34

F.3 Developability Profiling 40

G Generation Hyperparameters and Sampling Details **42**

G.1 Nanobody Sampling Hyperparameters 42

G.2 Sensitivity to Inference Settings 44

G.3 Turbo Mode: Accelerating Structure Prediction Drives Design Throughput 45

H Online Sequence-Only Reinforcement Learning **45**

H.1 Task, objective and reward 45

H.2 Sequence-only RL improves PD-L1 success 46

H.3 Structure-mode vs. sequence-only composition on PD-L1 46

H.4 Configuration ablations and a yield–diversity trade-off 47

H.5 Guardrails and future work 47

H.6 Limitations 48

H.7 Confidence, composition and diversity by target 49

I Baseline Implementations **54**

J Limitations and Future Work **56**

A Multimodal Autoencoder Training and Ablations

The main Proteina Mixta model uses the proposed autoencoder described below. We additionally evaluate two objective variants, *Simple-masked* and *Packing*, to examine how the reconstruction objective affects multi-regime latent organization and downstream generative performance. All three autoencoders share the same architecture, training data, and encoder curriculum. We refer to the proposed autoencoder as *Full*.

A.1 Autoencoder Architecture and Training Setup

Architecture. Mixta autoencoder share the same architectural setup as La-Proteina autoencoder (Geffner et al., 2026). It is a residue-level variational autoencoder with an 8-dimensional latent per residue. The encoder and decoder each contain 12 transformer layers with width 768 and 12 attention heads. The encoder can receive amino-acid identity, all-atom coordinates, backbone and side-chain angles, and pairwise backbone geometry. The decoder predicts amino-acid logits and, when given C_α coordinates, all-atom coordinates. The encoder outputs a diagonal Gaussian posterior. A sample is drawn by reparameterization, $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\varepsilon} \odot \exp(\log \boldsymbol{\sigma})$, and then passed through layer normalization.

Encoder regimes. Encoder context is sampled per example. At steady state, the encoder receives sequence-only input with probability 0.50, backbone–sequence input with probability 0.10, and all-atom input with probability 0.40. These frequencies arise from independent masks that drop all structural features with probability 0.50 and drop non-backbone structure with probability 0.20 when coordinates remain visible. Both masking probabilities ramp linearly from zero to their target values during the first 5k updates. Amino-acid identity is never dropped.

Independent decoder masking. The *Full* autoencoder independently samples whether the decoder receives C_α input. With probability 0.50, the decoder receives only the latent and is optimized for sequence reconstruction. Otherwise, it receives both the latent $\mathbf{z}$ and C_α and is optimized for sequence and all-atom reconstruction. Crossing the three encoder regimes with the two decoder regimes gives the six training configurations in Table 3. Structural loss is active exactly when the decoder receives C_α. Thus, every encoder regime is trained both for direct sequence decoding and for side-chain packing around a supplied backbone.

Optimization. The training loss is the batch mean of three terms. Sequence cross-entropy and all-atom coordinate L2 are each averaged over valid residues and weighted by 1. The coordinate term sums squared error over atoms and does not divide by the atom count. It is masked whenever the decoder does not receive C_α. The KL term is the closed-form KL of that diagonal Gaussian to $\mathcal{N}(0, I)$, summed over latent dimensions and averaged over valid residues. Its weight ramps linearly from 0 to 10^{-5} over the first 10k updates and then stays at 10^{-5}.

A.2 Alternative Autoencoders

To understand the effect of the reconstruction objective, we evaluate two autoencoder alternatives with the same 8-dimensional latent, transformer architecture, training data, and encoder-regime frequencies. We train each variant until its validation reconstruction losses converge. The proposed Full autoencoder also achieves low all-atom reconstruction error under sequence-only encoding (i.e., side-chain packing task), as shown in Figure 3b.

Simple-masked autoencoder. The simple-masked variant serves as a pure compression baseline. In each regime, the encoder compresses the observed sequence and structural information into an 8-dimensional per-residue latent, and the decoder reconstructs only the modalities available in that regime. With sequence-only input, the latent is decoded back to sequence (Table 3 objective #1). With backbone–sequence input, the decoder also receives the backbone, but side-chain reconstruction remains disabled. Only all-atom input is paired with all-atom reconstruction (Table 3 objective #6). Thus, this objective never asks the model to predict information withheld from both encoder and decoder.

Table 3: **Training objectives of the Full autoencoder.** Encoder and decoder contexts are sampled independently. "Full prob." gives the sampling probability of each resulting configuration.

#	Encoder input	Decoder input	Reconstruction target	Full prob.
1	Sequence	Latent $\mathbf{z}_{\text{seq}}$	Sequence	0.25
2	Sequence	Latent $\mathbf{z}_{\text{seq}}$, backbone	Sequence, all-atom	0.25
3	Sequence, backbone	Latent $\mathbf{z}_{\text{bb}}$	Sequence	0.05
4	Sequence, backbone	Latent $\mathbf{z}_{\text{bb}}$, backbone	Sequence, all-atom	0.05
5	Sequence, all-atom	Latent $\mathbf{z}_{\text{aa}}$	Sequence	0.20
6	Sequence, all-atom	Latent $\mathbf{z}_{\text{aa}}$, backbone	Sequence, all-atom	0.20

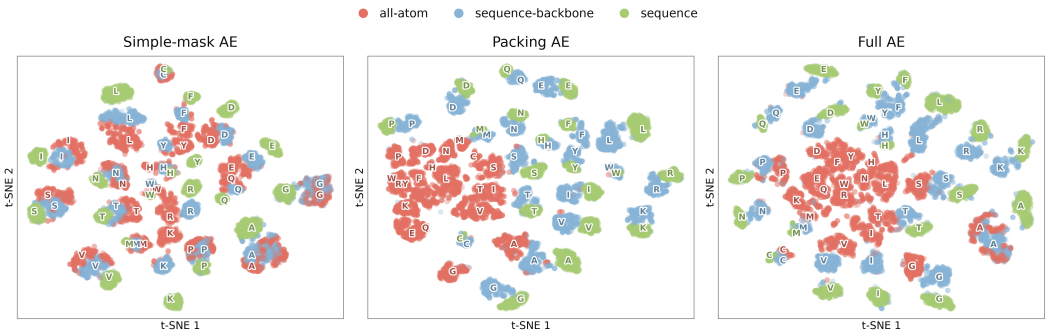


Figure 6: **Joint multi-regime t-SNE across autoencoder objectives.** Each panel shows one autoencoder (Simple-masked, Packing, or Full). Points are residue latents colored by encoder regime. For each amino-acid type and regime, the letter marks the highest-density location (the local mode) of that residue's points in the embedding.

Packing autoencoder. The packing variant makes decoder context independent of the encoder mask: the decoder always receives the ground-truth C_α trace together with the latent, and all-atom reconstruction remains active in every encoder regime (objectives #2, #4, and #6 in Table 3). Consequently, sequence-only and backbone–sequence latents must retain enough structural information for the decoder to recover side-chain geometry around the supplied backbone (i.e., side-chain packing).

A.3 Latent Space Qualitative Analysis

We sample $10{,}000$ AFDB structures and encode every valid residue with each autoencoder under sequence-only, backbone–sequence, and all-atom regimes. This produces three latent variables for every residue. We compute the joint t-SNE embedding across regimes for each autoencoder. For visualization, we select the same $4{,}000$ residue identities and retain the corresponding t-SNE projected variables from each regime, giving $4{,}000$ points per regime and $12{,}000$ points in each panel. We use the same random seed and t-SNE hyperparameters in all three panels.

Comparison across autoencoder objectives. As shown in Figure 6, simple-masked latent space is organized primarily into amino-acid clusters, with substantial overlap among the three regime colors within many clusters. The packing autoencoder shows a different pattern. All-atom latents form a broad central region, whereas sequence-only and backbone–sequence latents form more discrete amino-acid clusters around it. The Full autoencoder retains a structured central all-atom region while also organizing sequence-only and backbone–sequence latents into amino-acid-specific clusters. The similar organization of the packing and Full latent spaces is consistent with their shared side-chain packing objectives exerting a strong constraint on the representation. Packing is more demanding than direct compression because a reduced-context latent must support recovery of missing side-chain geometry around a supplied backbone. It therefore shapes the latent organization more strongly than the sequence reconstruction objective.

We nevertheless use the Full autoencoder because it also trains the decoder to recover sequence directly from a latent without backbone input. This latent-only decoding pathway is required for

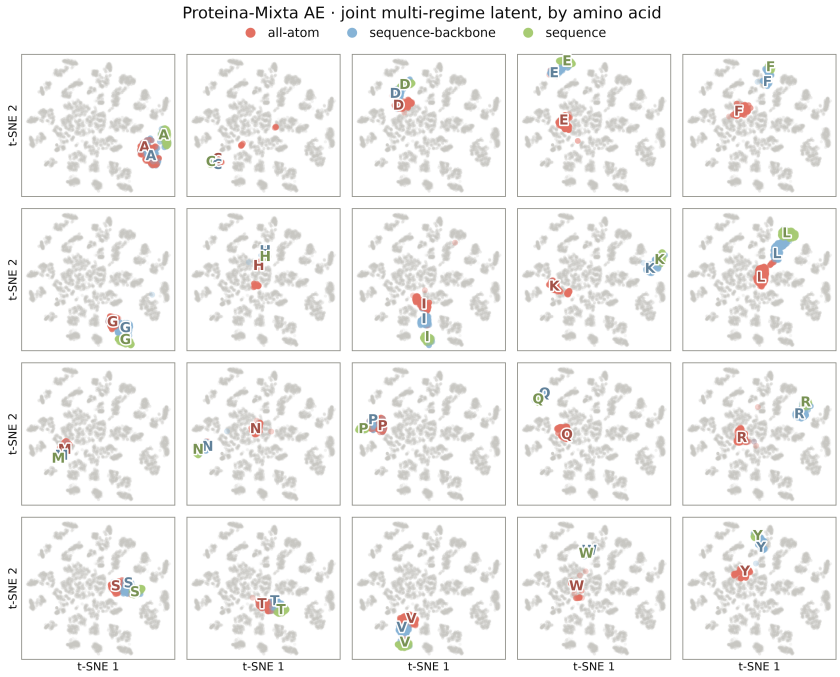


Figure 7: **Full autoencoder latents stratified by amino-acid type.** Each panel highlights one residue type in the shared t-SNE embedding, colored by encoder regime (all-atom, backbone–sequence, or sequence-only). Gray points show all other residues.

sequence-only generation and is absent from the packing autoencoder. The t-SNE comparison describes how the objectives organize latent space, but does not establish which representation is most useful for generation. We test that question by training matched diffusion models on each latent space in Appendix B.4.

Amino-acid stratification of the Full autoencoder. Figure 7 separates the embedding of the Full autoencoder by amino-acid type. In each panel, one residue type is colored by encoder regime and all other residues are shown in gray. The regime separation is most visible for several large, charged, polar, or aromatic residues, including E, F, K, Q, R, W, and Y. These residues have side chains whose preferred conformations depend strongly on local packing and interactions that are available to the all-atom encoder but not uniquely determined by sequence or backbone alone. By contrast, smaller residues such as A and G, as well as S and T, show closer visual overlap across regimes. How far the three regimes separate therefore depends on the residue, such that missing side-chain coordinates matter more when the side chain is large or chemically specific.

B Monomer Flow-Matching Training and Evaluation

We train one unconditional monomer flow model for each autoencoder variant in Appendix A. The diffusion architecture, data, optimization, masking curriculum, and evaluation are matched, while only the frozen autoencoder that defines the latent target changes. This design attributes differences in generation behavior to the representation learned by each autoencoder rather than to a different diffusion recipe. We label the proposed autoencoder *Full*, consistent with Table 3.

B.1 Multimodal Diffusion Training

Data and product space. Training uses the 415k AFDB subset, restricted to lengths at most 256 and filtered following the data pipeline in Lin et al. (2024). Each frozen autoencoder maps the available input regime to an 8-dimensional latent per residue. The flow model operates on the product

of explicit C_α coordinates $\mathbf{x}_{C_\alpha} \in \mathbb{R}^{L\times 3}$ and residue latents $\mathbf{z} \in \mathbb{R}^{L\times 8}$. The three regimes are

$$(\mathbf{x}_{C_\alpha}, \mathbf{z}_r) = \begin{cases} (\text{MASK}, \mathbf{z}_{\text{seq}}), & r = \text{seq}, \\ (\mathbf{x}_{C_\alpha}, \mathbf{z}_{\text{bb}}), & r = \text{bb}, \\ (\mathbf{x}_{C_\alpha}, \mathbf{z}_{\text{aa}}), & r = \text{aa}. \end{cases} \tag{8}$$

At steady state, all-atom, backbone–sequence, and sequence-only examples are sampled with probabilities 0.40, 0.10, and 0.50. The curriculum ramps linearly from all-atom training to this mixture over the first 50k updates. For sequence-only examples, every coordinate-derived features is zeroed and the C_α flow loss is disabled. The latent flow loss remains active, so sequence-only examples update the same network parameters used for joint generation.

Architecture. The shared flow network is a 14-layer transformer with width 768, 12 attention heads, and a 256-dimensional pair representation. It predicts velocity fields for both C_α coordinates and local latents and uses self-conditioning. The trunk does not use a learned modality embedding. The regime is supplied by which features are present. On sequence-only examples, every C_α token feature, pair-distance feature, and coordinate time embedding is zeroed. Coordinate and latent times are embedded separately, each with dimension 256, and enter the conditioning stream.

Optimization. Each flow loss is the squared error of the predicted clean sample, summed over channels and divided by the number of valid residues, then multiplied by $1/((1-t)^2+10^{-5})$. The two modality losses are added with equal weight after a batch mean. Sequence-only examples contribute zero to the coordinate term. Coordinate and latent times are sampled independently, so the two channels are not forced to share a noise level. Each time is drawn from a uniform–Beta mixture: with probability 0.02 it is uniform on $[0, 1]$, and otherwise it is drawn from a Beta distribution. Backbone time uses $\text{Beta}(1.9, 1.0)$, which places more mass near $t = 1$. Latent time uses $\text{Beta}(1.0, 1.5)$, which places more mass near $t = 0$. All three models are trained from scratch. The reported checkpoints are chosen by the codesignability–diversity trade-off: 110k updates for Proteina Mixta (Full autoencoder) and the diffusion with simple-masked autoencoder, and 140k updates for the diffusion with packing autoencoder. Training uses learning rate 10^{-4}, bfloat16 mixed precision, and EMA decay 0.999.

B.2 Monomer Evaluation Protocol and Metrics

Unless noted otherwise, we generate 100 unconditional samples at each length $L \in \{50, 100, 150, 200, 250\}$, for 500 designs per model and sampler.

Co-designability. Co-designability evaluates consistency of the model's jointly generated sequence and structure. We refold the generated sequence with ESMFold and compare the prediction with the generated structure. For the all-atom monomer comparison, a sample passes when all-atom self-consistency RMSD is at most 2 Å. We report the passing fraction.

ProteinMPNN designability. To isolate backbone quality from the generated sequence, ProteinMPNN redesigns each generated backbone and ESMFold refolds the redesigned sequence. A structure is designable when the minimum C_α self-consistency RMSD among the sampled redesigns is at most 2 Å. This metric asks whether the backbone supports at least one independently designed sequence, whereas co-designability evaluates the sequence emitted by Mixta itself.

Confidence and diversity. We report mean ESMFold pLDDT as a continuous confidence measure. Diversity is computed on the codesignable subset: Foldseek clusters quantify structural diversity and MMseqs2 clusters quantify sequence diversity (van Kempen et al., 2024; Steinegger & Söding, 2017). Because the number of passing designs changes across methods, cluster count and co-designability must be interpreted together. A high cluster fraction over very few passing samples is not evidence of broad useful coverage. We additionally report the secondary-structure composition of all generated structures.

B.3 All-atom Generation against Baseline

Table 4 reports the overall metrics plotted in Figure 3c. Each arm generates 100 samples at each length $L \in \{50, 100, 150, 200, 250\}$ (500 designs total). Diversity columns are Foldseek and MMseqs2

Table 4: **All-atom monomer generation and autoencoder diffusion ablations.** The first four rows reproduce the results in Figure 3c, while the last two report unconditional diffusion performance with alternative frozen autoencoders. Codesign. = all-atom co-designability; Design. = ProteinMPNN designability; Foldseek = all-atom structural diversity (cluster count among codesignable designs); MMseqs = sequence diversity (cluster count among codesignable designs). Arrows indicate whether higher (↑) is better. Best value in each metric column is bolded (secondary-structure composition excluded). We note that the architecture and data used to train all models in this table are identical.

Method	Codesign. ↑	Design. ↑	Foldseek ↑	MMseqs ↑	pLDDT ↑	α/β/coil
Main-figure generation results						
La-Proteina	0.746	0.976	246	366	86.1	0.59/0.11/0.30
La-Proteina latent-warmup	0.526	0.978	204	254	77.2	0.64/0.09/0.26
Proteina Mixta	0.824	**0.982**	248	391	**88.9**	0.55/0.13/0.32
Proteina Mixta latent-warmup	**0.842**	0.980	**299**	**418**	88.0	0.72/0.05/0.23
Autoencoder diffusion ablations						
Mixta (simple-masked)	0.730	0.964	240	279	85.4	0.61/0.11/0.28
Mixta (packing)	0.788	0.934	248	328	87.5	0.62/0.10/0.28

cluster counts on the codesignable subset. Secondary structure is the mean α/β/coil fraction over all generated structures.

B.4 All-atom Generation Comparison across Autoencoders

Table 4 also reports the unconditional diffusion ablation for the two alternative autoencoders. Each alternative's diffusion checkpoint is selected based on the best codesignability–diversity trade-off: 110k updates for simple-masked and 140k updates for packing. The selected simple-masked model reaches 0.730 co-designability and 0.964 designability, while the selected packing model reaches 0.788 and 0.934, respectively. Their Foldseek/MMseqs2 cluster counts are $240/279$ and $248/328$. Even though that the simple-masked latent space in Figure 6 keeps each amino acid tightly clustered with the three regimes lying closely nearby, that organization does not imply strong diffusion generation. On the other hand, Proteina Mixta trained on the Full autoencoder reaches the best generative quality with comparable generative diversity.

B.5 Interaction with Latent Warm-up Denoising

Figure 8 illustrates the schedule used for the latent warm-up denoising in Table 4. The sampler takes 400 steps, and the first 40 are warm-up. During warm-up, every C_α feature is masked and the backbone time is held at $t_x = 0$, so the network generates the sequence latent alone. Latent time moves linearly from $t_z = 0$ to $t_{\text{warm}} = 0.3$, with latent noise scale 0.05. The remaining 360 steps are joint. Latent time continues from 0.3 to 1 on a power ($p = 2$) schedule, and backbone time starts at 0 and follows a log schedule. In this joint stage the backbone and latent noise scales are 0.10 and 0.05. The 40 warm-up steps are taken from the same 400-step budget, so the scheduled and unscheduled samplers use the same number of network evaluations.

Latent warm-up fails for La-Proteina because the warm-up stage is effectively sequence-only latent generation: all coordinate features are masked while the latent is denoised. La-Proteina was not trained on sequence-only diffusion and therefore cannot generate reasonable warm-up latents. Proteina Mixta, by contrast, is trained with sequence-only diffusion. However, it is still interesting that latent warm-up supports the subsequent structure generation. At the handoff, C_α denoising begins while conditioned on a sequence latent that has been only partially denoised. This conditional C_α denoising setting is out of Mixta's training distribution. Its success suggests that, at this early denoising stage, the sequence latent trajectory is potentially aligned with the all-atom latent trajectory to support the transition to joint generation. Characterizing this cross-regime trajectory alignment, and explicitly training models to strengthen it, are promising directions for future work.

Despite its success in monomer generation, we do not use this denoising schedule for nanobody design. As we see in Section 4.4 generating nanobodies with sequence only even with a dedicated wetlab validated model is quite difficult. Therefore driving the structure generation with the sequence

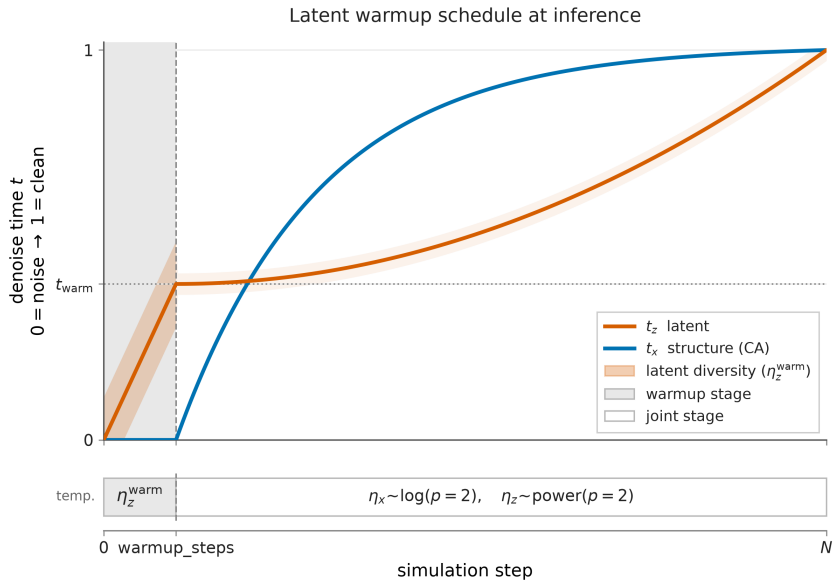


Figure 8: **Latent warm-up denoising schedule used in monomer generation.** Denoising time runs from 0 (noise) to 1 (clean) over N steps. In the gray warm-up stage, C_α features are masked, backbone time t_x stays at 0, and latent time t_z advances linearly to t_{warm}. In the joint stage, t_z continues from t_{warm} on a power ($p = 2$) schedule and t_x starts at 0 on a log schedule. The shaded band represent the latent noise scale during and after warm-up.

generation likely encounters similar difficulty. We leave this to future work to see how with the ability to operate in both the sequence and structure space can better biological interfaces be designed.

C Monomer Extension to Framework-constrained Nanobody Generation

Before moving to the full antibody-specific training and target-conditioned design experiments in Section 4.2, we present a second proof of concept for Proteina Mixta's multi-regime learning. The monomer experiments in Figure 3 establish that one generative model can operate across sequence-only and all-atom regimes. Here, we ask whether additional structure-free antibody sequences can be incorporated during continued domain adaptation and improve framework-constrained nanobody generation in both sequence-only and all-atom modes.

We denote the monomer model from Figure 3 by $\text{Mixta}_{\text{monomer}}$. Starting from this model, we train for another 48k steps on a stochastic 50:50 mixture of OAS heavy-chain sequences following (Kim et al., 2026) and the 415k AFDB subset. We denote the resulting model by $\text{Mixta}_{\text{monomer+OAS}}$. OAS examples contain no coordinates and therefore enter through the sequence-only regime. Within the structural half of the mixture, all-atom, backbone–sequence, and sequence-only regimes are sampled in proportions $0.90/0.05/0.05$. The resulting effective batch proportions are therefore approximately $0.45/0.025/0.525$. This monomer adaptation experiment is separate from the target-conditioned OAS co-training experiment in Section 4.3.

Although neither checkpoint is trained with explicit antibody framework conditioning, we can still steer generation toward a desired framework at inference time by backpropagating through the frozen decoder. We explore this capability in both sequence-only and joint all-atom generation. Specifically, during sampling, we guide the 91 framework positions toward the corresponding residues in hNbBCII10, without specifying the identities of any CDR residues. Both checkpoints remain frozen and are evaluated using the same guidance protocol within each generation mode. We generate 100 length-128 VHHs for each checkpoint and generation mode.

C.1 Inference-Time Guidance through the Frozen Decoder

Guidance before each integration step. At the current state and time, the flow model predicts a velocity for each active channel. Sequence-only generation predicts the latent velocity $\mathbf{v}_\theta^z(\mathbf{z}_t, t)$. All-atom generation also predicts the coordinate velocity $\mathbf{v}_\theta^x(\mathbf{x}_t, t)$. Before moving from t to $t + \Delta t$, we convert these velocities into the model's predicted clean sample

$$\widehat{\mathbf{z}}_1 = \mathbf{z}_t + (1 - t_z)\mathbf{v}_\theta^z, \qquad \widehat{\mathbf{x}}_1 = \mathbf{x}_t + (1 - t_x)\mathbf{v}_\theta^x, \tag{9}$$

The frozen decoder is applied to this predicted clean sample. We then backpropagate a framework loss through the decoder, use the resulting gradient to guide the predicted velocities by an additional velocity integration step.

Sequence-only guidance. The frozen autoencoder decoder maps $\widehat{\mathbf{z}}_1$ to amino-acid logits $\boldsymbol{\ell} \in \mathbb{R}^{L\times 20}$. Given framework mask $m_i \in \{0, 1\}$ and target framework residue y_i, we minimize

$$\mathcal{L}_{\mathrm{FW}} = \frac{1}{\sum_i m_i} \sum_{i:m_i=1} -\log \operatorname{softmax}(\boldsymbol{\ell}_i)_{y_i}. \tag{10}$$

This loss specifies only framework residues with no CDR target. We backpropagate Eq. (10) into $\widehat{\mathbf{z}}_1$ and rescale the gradient to be a guidance of the latent velocity:

$$\Delta\mathbf{v}^z = -\rho(t)\,\mathrm{RMS}(\mathbf{v}_\theta^z)\frac{\nabla_{\widehat{\mathbf{z}}_1}\mathcal{L}_{\mathrm{FW}}}{\mathrm{RMS}(\nabla_{\widehat{\mathbf{z}}_1}\mathcal{L}_{\mathrm{FW}})}, \quad \rho(t) = w(t)\min\left(s\sqrt{\mathcal{L}_{\mathrm{FW}}}, \rho_{\max}\right). \tag{11}$$

Here $w(t)$ turns the guidance on over a time schedule, s sets its strength, and $\rho_{\max}$ limits its size relative to the model velocity. The integrator uses the corrected velocity

$$\mathbf{v}^z = \mathbf{v}_\theta^z + \Delta\mathbf{v}^z, \qquad \mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \Delta t\,\mathbf{v}^z. \tag{12}$$

We do not zero this gradient at CDR positions. The loss is evaluated only on framework residues, but attention in the decoder lets it affect the latents of CDR residues as well. Self-conditioning continues to use the unguided clean prediction in Eq. (9).

All-atom guidance. In joint generation, the decoder takes both $(\widehat{\mathbf{x}}_1, \widehat{\mathbf{z}}_1)$. We backpropagate the framework cross-entropy through both channels and also apply a framework-only structural loss to $\widehat{\mathbf{x}}_1$,

$$\mathcal{L}_{\mathrm{struct}} = \tfrac{1}{2}\mathcal{L}_{\mathrm{pair}} + \tfrac{1}{2}\mathcal{L}_{\mathrm{Kabsch}}, \tag{13}$$

where $\mathcal{L}_{\mathrm{pair}}$ compares framework C_α pairwise distances and $\mathcal{L}_{\mathrm{Kabsch}}$ compares framework coordinates after an optimal rotation. The latent and coordinate velocities are then corrected separately. The latent correction follows the normalized framework-cross-entropy gradient. The coordinate correction combines the normalized sequence and structural gradients, removes their center of mass, and is capped relative to the coordinate-velocity RMS. Latent gradients remain active at all residues, while coordinate gradients are applied only at framework positions. As in sequence-only guidance, each corrected velocity is the model velocity plus its guidance term,

$$\mathbf{v}^z = \mathbf{v}_\theta^z + \Delta\mathbf{v}^z, \qquad \mathbf{v}^x = \mathbf{v}_\theta^x + \Delta\mathbf{v}^x, \tag{14}$$

and the sampler integrates

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \Delta t\,\mathbf{v}^z, \qquad \mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \Delta t\,\mathbf{v}^x. \tag{15}$$

C.2 Guidance Protocols

For sequence-only generation, we use a uniform latent time grid with 400 steps and latent noise scale $\eta = 0.4$. The guidance in Eq. (11) uses strength $s = 6$ and maximum velocity ratio $\rho_{\max} = 8$. Its weight ramps from zero at $t = 0.05$ to full strength at $t = 0.50$ and remains active thereafter. These settings were selected empirically by hyperparameter tuning to reach high mean framework identity while leaving CDR positions unconstrained.

For all-atom generation, we use a uniform latent time grid and a log-spaced C_α time grid, again with 400 steps. Latent framework-sequence guidance uses the same strength 6 and maximum velocity ratio 8. For the coordinate correction, the normalized sequence and structural directions have strengths 4

Table 5: **Sequence-only framework-constrained generation.** We generate 100 samples per model. Framework identity is the mean fraction of matching framework residues. AbLang2 whole-VHH and CDR pseudo-perplexities are evaluated over all samples.

Model	FW identity ↑	AbLang2 PPL ↓	AbLang2 CDR PPL ↓
$\text{Mixta}_{\text{monomer}}$	99.38%	4.50	36.28
$\text{Mixta}_{\text{monomer+OAS}}$	99.44%	4.41	34.04

Table 6: **All-atom framework-constrained generation.** We generate 100 samples per model under strong framework guidance. Framework identity is the mean fraction of matching framework residues; RMSD is framework C_α Kabsch distance to the reference. AbLang2 whole-VHH and CDR pseudo-perplexities are evaluated over all samples.

Model	FW identity ↑	Reference RMSD ↓	AbLang2 PPL ↓	AbLang2 CDR PPL ↓
$\text{Mixta}_{\text{monomer}}$	99.97%	0.123 Å	3.88	22.13
$\text{Mixta}_{\text{monomer+OAS}}$	99.85%	0.106 Å	3.78	19.86

and 8, respectively, and their sum has maximum velocity ratio 8. Coordinate guidance stays at full strength throughout sampling. Latent sequence guidance is off before $t = 0.15$, increases from zero to full strength linearly from $t = 0.15$ to $t = 0.35$, remains at full strength until $t = 0.90$, and returns to zero by $t = 0.995$. Sampling uses cold noise scales $\eta_{\text{bb}} = 0.20$ and $\eta_z = 0.05$. These settings were also selected empirically by hyperparameter tuning.

C.3 Nanobody Generation Across Modes from OAS Adaptation

We compare $\text{Mixta}_{\text{monomer}}$ with $\text{Mixta}_{\text{monomer+OAS}}$ after 48k additional updates on a mixture that includes OAS sequences. We observe that this short continuation already improves framework-constrained nanobody generation in both sequence-only and all-atom modes. Although the checkpoints are not strictly matched for the number of training iterations, we consider the comparison a reasonably valid proof of concept: AFDB contains little antibody-repertoire supervision, so further training on AFDB alone would not be expected to show the antibody-generation behavior examined here.

Evaluation metrics. For each generated VHH, framework identity is the percentage of the 91 framework positions whose amino-acid identities match hNbBCII10. For all-atom generation, we additionally compute the framework C_α RMSD after Kabsch alignment of the generated framework to hNbBCII10. AbLang2 (Olsen et al., 2024) PPL is the per-sequence pseudo-perplexity obtained by averaging position-wise pseudo-negative-log-likelihood over the complete VHH and then exponentiating. AbLang2 CDR PPL is calculated in the same way, except that the average includes only CDR positions while AbLang2 still receives the complete VHH as context. Lower RMSD and perplexity are better, whereas higher framework identity is better. Every reported value is the mean over all 100 generated VHHs.

Improved Nanobody Sequence Generation. $\text{Mixta}_{\text{monomer+OAS}}$ reaches a slightly lower whole-VHH AbLang2 PPL (4.41 versus 4.50) and a lower CDR PPL (34.04 versus 36.28), while framework identity stays high. It shows that the additional sequence-only adaptation from OAS data already improve agreement with the antibody language model in sequence-only generation.

Improved Structure-guided Nanobody Sequence Design. We report the framework-constrained all-atom runs as structure-guided nanobody sequence design. We do not treat the sampled coordinates as a reliable nanobody structure for two reasons. First, even though the 48k continuation introduce some antibody signals from OAS, the structural training set still contains few nanobody structures. Second, the generated VHHs are unbound. Without an antigen, the CDR loops tend to be more flexible, meaning that a single generated structure can be a weak description of the nanobody, and nanobody structure prediction can be a even more challenging and less well-formulated task. We therefore only evaluate the sequence quality.

Under the same framework guidance, $\text{Mixta}_{\text{monomer+OAS}}$ lowers whole-VHH AbLang2 PPL from 3.88 to 3.78 and CDR PPL from 22.13 to 19.86, while framework identity stays near-perfect. These sequences also agree more closely with AbLang2 than the sequence-only samples above. Guiding generation with both the framework sequence and backbone therefore produces nanobody sequences that match the antibody language model more closely than sequence-only generation.

D PROTEINA NANOCUERPO TRAINING DETAILS

Proteina Nanocuerpo is a nanobody-specialized flow model obtained by fine-tuning the general protein-complex model, *Complexa*, on pseudo-VHH interfaces. It reuses Complexa's frozen autoencoder and trunk weights and adds antibody-specific conditioning features, a nanobody-shaped data pipeline, and a set of conditioning dropouts that make every context channel independently droppable at inference. The released checkpoint is taken after 470k fine-tuning updates and is referred to throughout as *bias-fix 470k*.

D.1 INITIALIZATION AND WEIGHT SPLICING

Frozen autoencoder. The latent target is defined by the Complexa autoencoder, which is held frozen throughout. It maps each residue to an 8-dimensional local latent, and the flow model operates on the product of explicit C_α coordinates $\mathbf{x} \in \mathbb{R}^{L\times 3}$ and residue latents $\mathbf{z} \in \mathbb{R}^{L\times 8}$, exactly as in Section B. Because the autoencoder is shared with Complexa and never updated, the latent space is identical to the one the initialization was trained in, so fine-tuning does not have to re-learn the representation.

Trunk initialization. The flow trunk is initialized from the Complexa complex checkpoint rather than from scratch. The architecture is unchanged at 14 transformer layers, width 768, 12 attention heads, and a 256-dimensional pair representation ($\approx$ 160M parameters).

Splicing new features into a pretrained encoder. Adding conditioning channels changes the input width of the feature-embedding matrices, so the pretrained weights no longer align by position. We handle this by *order-preserving zero-padding*: every Complexa feature keeps its original index in the concatenated input vector and loads its pretrained columns, and all new antibody features are appended strictly after them, with their columns initialized to zero. A new channel therefore contributes nothing at step 0 and the spliced model is numerically identical to Complexa on the first update, which makes the fine-tune a continuation rather than a restart. The one exception is the cross-sequence pair factory (`concat_pair_factory`), which is excluded from checkpoint loading and randomly initialized.

D.2 CONDITIONING FEATURES

Table 7 lists the conditioning channels. Complexa contributes the noisy state, self-conditioning, optional ground-truth coordinate and residue-type channels, epitope hotspot masks, and relative position encodings. The additions fall into three groups: an explicit *framework* scaffold representation (absolute and center-of-mass-relative atom37 coordinates, sequence, atom-presence mask, backbone torsions and side-chain χ angles), secondary structure and chain-break labels, and a four-label per-residue *CDR identity* embedding that distinguishes framework from CDR1, CDR2 and CDR3. The CDR identity feature is the channel that makes the loop regions addressable at inference and is the main functional difference from Complexa's generic binder conditioning.

Antigen context is supplied as additional tokens appended to the sequence representation, together with antigen×antigen and binder×antigen pair distances. Framework and motif concatenation paths are disabled, so the framework enters only through the per-residue channels above.

D.3 PSEUDO-VHH TRAINING DATA

Source. Training uses a single data source, the TED dimer interface set (*Teddymer*), sampled with probability 1.0. There is no true-antibody (SAbDab) leg: every training example is a generic

Table 7: **Proteina Nanocuerpo conditioning features.** Dimensions are per residue for sequence features, per residue pair for pair features, and per antigen residue for antigen-context features. *Inherited* channels load their pretrained columns from the Complexa checkpoint at their original indices; *added* channels are appended after them and zero-initialized.

Group	Feature	Dim.	Source
Residue (sequence) features — total 827			
State	`xt_bb_ca`	3	inherited
	`xt_local_latents`	8	inherited
Self-cond.	`x_sc_bb_ca`	3	inherited
	`x_sc_local_latents`	8	inherited
Optional GT	`optional_ca_coors_nm`	3	inherited
	`optional_res_type`	20	inherited
Epitope	`hotspot_mask_seq`	1	inherited
Position	`res_seq_pdb_idx`	256	inherited
Chain	`chain_break_per_res`	1	added
Framework	`framework_abs_coords`	148	added
	`framework_rel_coords`	148	added
	`framework_seq`	20	added
	`framework_mask_seq`	37	added
	`framework_torsion_angles`	63	added
	`framework_sc_angles`	88	added
Structure	`sse_type`	4	added
CDR identity	`cdr_type_seq_feat`	16	added
Pair features — total 303			
Position	`rel_seq_sep`	127	inherited
State	`xt_bb_ca_pair_dists`	30	inherited
Self-cond.	`x_sc_bb_ca_pair_dists`	30	inherited
Optional GT	`optional_ca_pair_dist`	30	inherited
Chain	`chain_idx_pair`	1	inherited
Epitope	`hotspot_mask_pair`	1	inherited
Framework	`x_framework_pair_dists`	84	added
Antigen context tokens — total 548			
Geometry	`target_abs_coords`	148	inherited
	`target_rel_coords`	148	inherited
Sequence	`target_seq`	20	inherited
	`target_mask_seq`	37	inherited
Torsions	`target_sc_angles`	88	inherited
	`target_torsion_angles`	63	inherited
Epitope	`target_hotspot_mask`	1	inherited
Structure	`target_sse_type`	4	added
Biochemistry	`target_interface_feats`	39	added

protein–protein interface converted into a nanobody-shaped problem. Because the model never sees a real antibody during this stage, the binary true-antibody indicator feature is left disabled.

Pseudo-VHH conversion. Every sample passes through `PseudoVHHTransform`, which reinterprets the binder chain of a generic interface as a VHH: it selects three loop regions to act as CDRs and labels the remainder as framework. Regions are placed subject to a minimum inter-CDR gap of 5 residues and minimum FR1 and FR4 lengths of 5 and 3 residues, using an interface cutoff of 10 Å. CDR indices are assigned by *interface contact count* rather than by sequence order (`pseudo_cdr_ordering = interaction`), so the loop making the most antigen contacts is labelled CDR3, matching the dominant role CDR3 plays in real nanobodies. Epitope hotspots are then sampled with CDR3-weighted probability rather than uniformly, which concentrates the hotspot signal on the epitope sub-region that the designed CDR3 will actually engage.

Cropping and framing. Complexes are cropped to 500 tokens around the interface using an 8 Å C_α contact criterion, with binders capped at 250 residues, targets at a minimum of 50 residues,

Table 8: **Conditioning dropout rates.** Each channel is dropped independently per example. Structure-level dropouts are applied by the training loop; label-level dropouts are applied inside the data transforms.

Channel	What is removed	Rate
Antigen (CFG)	all antigen tokens and pair features	0.50
Framework	framework coordinates and sequence	0.50
Framework structure	framework coordinates and atom37 mask, sequence kept	0.50
CDR SSE labels	secondary-structure labels on CDRs	0.50
Target SSE labels	secondary-structure labels on the antigen	0.50
Framework SSE labels	secondary-structure labels on framework	0.00
Contacts	binder–antigen contact map	0.50
Hotspots	epitope hotspot annotation	0.50

and a 15 Å spatial crop radius around the target. Framework coordinates are extracted, centered and randomly rotated. The complex is centered on the target or the framework, and on a coin flip ($p = 0.5$) on the epitope hotspots instead, so the model is not able to infer the binding site from the global frame alone.

D.4 Conditioning Dropouts

Nanobody design at inference specifies arbitrary subsets of the context: an antigen with or without hotspots, a framework with or without its structure, CDR lengths with or without secondary structure. To make every such subset in-distribution, each conditioning channel is dropped independently during training (Table 8). Two of these deserve comment. *Framework structure dropout* removes the framework coordinates and the atom37 presence mask while retaining the framework sequence and IMGT numbering, which is what enables sequence-only framework specification at inference. *Target dropout* removes the antigen entirely and is the classifier-free guidance channel that the antigen guidance weight acts on at sampling time.

Dropped channels are removed from the batch rather than zeroed, so the corresponding features take their documented absent-path defaults instead of being read as a degenerate value, and framework structure dropout additionally clears the framework-active flag so that zeroed coordinates are never interpreted as a framework collapsed onto the origin. Self-conditioning is computed *before* the dropouts are applied, matching the ordering the checkpoint was trained with; the self-conditioning input can therefore still carry a framework or antigen that the dropouts subsequently remove.

D.5 Optimization

The model is trained on 8 nodes of 8 GPUs at local batch size 6, in fp32, with Adam at learning rate 10^{-4}, self-conditioning enabled at $p = 0.5$ drawn independently per rank, no recycling, and EMA decay 0.999. Coordinate times are drawn from a mixture of a uniform and a Beta distribution ($p_1 = 1.9$, $p_2 = 1.0$, mixture weight 0.02) and latent times from a uniform distribution; the two are sampled independently, so coordinates and latents are not forced to share a noise level. Validation uses 400 flow steps. We evaluate the EMA checkpoint at 470k updates.

D.6 A CDR-focused redefinition of hotspots.

Complexa treats epitope hotspots as a generic binding-site hint: for each complex it draws a single fraction $p \sim \mathcal{U}(0, 0.2)$ and selects that fraction of the antigen's interface residues *uniformly at random*, then zeroes the whole mask with probability 0.5. Every interface residue is therefore equally likely to be revealed, which is the right prior when the binder is an arbitrary protein but the wrong one for a nanobody: a VHH does not engage its epitope uniformly, it engages it predominantly through CDR3. We replace the uniform draw with a *per-residue weighted Bernoulli* conditioned on which CDR does the binding.

E Proteina Mixta Nanobody training and architecture details

E.1 Architecture

Autoencoder adaptation. We retain the multimodal autoencoder architecture introduced above, which maps sequence-only, backbone–sequence, and all-atom inputs into a shared residue-level latent space. Before training the nanobody generator, we fine-tune this autoencoder on individual chains sampled equally from PDB and SAbDab2 (Capel et al., 2026). This mixture combines general protein structures with antibody-domain examples while preserving the same multimodal reconstruction formulation. We select the EMA checkpoint at 250k fine-tuning updates and freeze it throughout generator training. Thus, antibody structures are used for representation learning, whereas framework and CDR conditioning in the structural generation task is learned from pseudo-VHH annotations.

Generator architecture. The generator follows the partially latent formulation of La-Proteina (Geffner et al., 2026), jointly modeling explicit C_α coordinates and continuous residue latents with a transformer and pair representations. Unlike the fine-tuned autoencoder, the generator is initialized from scratch. Its 14-layer trunk includes triangular multiplicative updates after every second layer, giving seven pair-update blocks over the joint binder–target representation. These updates follow the architectural approach of Proteina and La-Proteina (Geffner et al., 2025; 2026) and allow pair features to incorporate information through intermediate residues. Restoring these updates was associated with faster convergence in our training experiments, at the cost of increased memory use and inference time. Because the configuration without pair updates used a larger batch, this observation is not an isolated architecture ablation. We retain the same antibody specific feature additions used for Proteina Nanocuerpo discussed in Appendix D.

E.2 Training Details

Training data. Generator training combines AFDB monomers from the Genie3 data preparation (Lin et al., 2026), restricted to at most 256 residues; synthetic Teddymer interfaces introduced by Proteina-Complexa (Didi et al., 2026); and experimental protein–protein interfaces from the PDB (Berman et al., 2000), following the Complexa data pipeline (Didi et al., 2026). Both interface sources receive synthetic framework and CDR annotations through the pseudo-VHH transformation. These annotations define the conditional design task without requiring naturally occurring antibody–antigen pairs. We never train on real CDR annotations other than on the OAS data ablation. The initial mixture was motivated by Proteina-Nanocuerpo, which trained on Teddymer with 50% target dropout. We hypothesized that allocating half the mixture to monomer generation and half to binder training with the same dropout would yield an effective 75% target-free share. We therefore initialized Mixta with 75% monomers and allocated the remaining 25% primarily to Teddymer (20% Teddymer, 5% PDB interfaces), reflecting its central role in Nanocuerpo training.

Training phases and mixture comparisons. Table 9 summarizes the configured sampling probabilities. Initial training emphasizes monomers, with 75% AFDB, 20% Teddymer, and 5% PDB interfaces. After 230k updates, we initialize an interface-focused phase from the EMA weights and reduce monomer sampling to 5%. Two alternative branches start from the same checkpoint: one transfers the PDB probability to Teddymer, and the other transfers the Teddymer probability to PDB. These branches compare the two interface sources while retaining the same monomer probability.

The final adaptation phase starts after a further 170k updates of the mixed-interface branch, corresponding to 400k inherited generator updates. It assigns all nominal structural sampling probability to Teddymer, removing both AFDB monomers and PDB interfaces from the scheduled mixture. We also halve the learning rate and introduce sequence-only framework conditioning in this phase. Increasing emphasis on Teddymer and subsequently removing the other structural components proved useful for target-conditioned generation in the explored recipes. Since the final transition changes both data and optimization settings, its benefit cannot be attributed to the data mixture alone.

Table 9: **Generator training phases and alternatives.** Entries are configured sampling probabilities (%). The three interface-focused variants share the same 230k initialization. Teddymer adaptation and OAS co-training both start from the mixed-interface branch after 170k additional updates.

Phase	AFDB	Teddymer	PDB	OAS	Learning rate
Initial training	75	20	5	0	10^{-4}
Mixed interfaces	5	76	19	0	10^{-4}
Without PDB interfaces	5	95	0	0	10^{-4}
Without Teddymer	5	0	95	0	10^{-4}
Teddymer only	0	100	0	0	5×10^{-5}
OAS co-training	0	90	0	10	5×10^{-5}

Modality and conditioning sampling. For structural examples, binder inputs use all-atom, backbone–sequence, and sequence-only regimes with probabilities 0.90, 0.01, and 0.09, respectively; antigen structure remains observed. Framework conditioning is dropped with probability 0.5. Additional dropout on CDR labels, secondary structure, dataset identity, and epitope conditioning supports generation with partially specified inputs.

Optimization and sequence-only extension. Training uses 32 NVIDIA H100 GPUs, with a batch of six examples per GPU. We use Muon for attention and transition matrices, with AdamW-matched update scaling, weight decay 0.1 for Muon parameters, and gradient clipping at 1.0. Learning rates are constant within phases, without optimizer warmup. Phase transitions load EMA weights and restart optimizer state; reported phase-local steps therefore exclude inherited training.

For the sequence-only ablation, we replace 10% of Teddymer sampling with liability-filtered OAS heavy-chain sequences. This branch and its structural-only control share the same initialization and optimization settings. OAS examples are encoded directly from sequence, without refolding or coordinate supervision, while the shared generator retains structure-conditioned generation. The frozen autoencoder is identical across all branches, allowing their generator checkpoints to be compared and combined by the weight-arithmetic procedure described in Section E.3.

E.3 Reducing per-target stochasticity with weight arithmetic

Our approach extends the checkpoint averaging already used in Proteina-Complexa, where model weights are averaged over the final 10,000 training steps. Although averaging reduces variation across training updates, we observe substantial target-dependent fluctuations: an update that improves one antigen can degrade another. We therefore extend temporal averaging to validation-guided combinations of compatible checkpoints, seeking a more balanced model across development targets. The resulting weights define a single generator, with no ensemble evaluation or additional inference cost.

Unlike standard EMA, whose coefficients follow a prescribed temporal decay, our procedure uses development-target performance to select the construction. We describe this as validation-guided weight arithmetic for improving cross-target balance. Checkpoint combinations, residual-strength multipliers, and the final construction are selected using PD-L1, BHRF1, insulin, and TrkA; the norm-calibration coefficients below are calculated from the weights. IL-3, CTLA-4, and CEACAM6 are excluded from this checkpoint selection. This exclusion concerns selection of the merged checkpoint only. It does not extend to target-specific sampler settings (Appendix G.1), which are selected per target for all seven antigens. The evaluated Proteina Mixta generator uses the final merged checkpoint $B_3^{\ddagger}$, defined in Equation 23, with fixed weights across targets and sampling configurations. We note we use PD-L1, insulin, BHRF1, and TrkA for variance reduction. CEACAM6, IL3, and CTLA-4 the more difficult targets we maintain separate.

Motivation and relation to prior work. Weight averaging can improve generalization without increasing inference cost, as demonstrated by stochastic weight averaging and model soups (Izmailov et al., 2018; Wortsman et al., 2022). Task arithmetic extends this approach by treating parameter differences as editable directions (Ilharco et al., 2023). We combine averaging, extrapolation, and selective residual updates to obtain a single Proteina Mixta checkpoint with more balanced

Table 10: **Target-specific experts versus a single merged checkpoint.** Unique success rates (%) from 1,000 designs per target and setting. A target-specific expert is the best-observed checkpoint for that target; its worst-other-target column reports its lowest success on the remaining targets. Each of the final two columns uses one checkpoint across targets, with best-observed per-target sampler settings; rates therefore differ from the shared-sampler evaluation in Table 11.

	Target-specific expert		One checkpoint across targets	
Target	Success	Worst other target	Unmixed	Selected merge
PD-L1	66.0	3.4	8.7	35.9
BHRF1	87.5	1.1	64.9	43.5
Insulin	86.0	0.9	36.6	35.6
TrkA	57.3	1.1	16.9	29.0
Worst target	—	—	8.7	**29.0**

performance across antigens. This is an application of existing weight-space ideas, with model-specific choices of coefficients, clipping, and parameter groups.

Individual checkpoints can achieve high success on one antigen while performing poorly on others (Table 10). Using best-observed per-target settings, the selected merged checkpoint has a worst-target unique success rate of 29.0%, compared with 8.7% for the reported unmixed checkpoint, while sacrificing some target-specific peaks. Unique success counts distinct full binder sequences passing the design gate with zero framework mutations, deduplicated within each arm and divided by its $1{,}000$ designs. Because Table 10 uses best-observed per-target sampler settings, its rates for $B_3^{\ddagger}$ differ from the shared-sampler rates reported in Table 11c. These results characterize cross-target balance; they do not establish a reduction in sampling variance across independent runs.

Constructing the base checkpoints. All models share a frozen autoencoder. Arithmetic is restricted to 751 diffusion parameter tensors; the 765 autoencoder tensors are copied unchanged. Let A_1, A_2, A_3 denote EMA checkpoints at training steps 320k, 340k, and 350k, with corresponding raw checkpoints R_1, R_2, R_3. Define

$$M = \frac{1}{3}\sum_{i=1}^{3} A_i, \qquad R = \frac{1}{3}\sum_{i=1}^{3} R_i,$$
$$L[k] = \min_i A_i[k], \quad U[k] = \max_i A_i[k], \tag{16}$$

Here k indexes named parameter tensors, and $X[k]$ denotes tensor k of checkpoint X. Minima and maxima are elementwise. In the implementation, M and R denote the saved means, including their original dtype rounding. Using the quantities of Equation 16, define the bounded reversal operator

$$\mathcal{E}_a[k] = \mathrm{cast}_k[\mathrm{clip}(M[k] + a(M[k] - R[k]),\, L[k],\, U[k])]\,, \tag{17}$$

where cast_k converts the result to the original dtype of tensor k, and $\mathrm{clip}(x, L, U)$ applies $\min(U, \max(L, x))$ coordinatewise. Arithmetic inside the cast is performed in float64. Equation 17 uses the extrapolation form of ExPO (Zheng et al., 2025), adapted here to EMA and raw checkpoint means with elementwise clipping. Module-dependent coefficients are related to parameter-group scaling in prior merging work (Yang et al., 2024; Wang et al., 2025a); our coefficient choices are specific to this model.

The first base, B_1, uses $B_1[k] = \mathcal{E}_{0.50}[k]$ for all diffusion tensors. Let $\mathcal{P}$ contain the 287 diffusion tensors whose names begin with `nn.pair_update_layers.`, and let $\mathcal{F}$ contain the remaining 464 diffusion tensors. The second base, B_2, applies a weaker reversal to the pair-update modules:

$$B_2[k] = \mathcal{E}_{\alpha_k^{(2)}}[k], \qquad \alpha_k^{(2)} = \begin{cases} 0.50, & k \in \mathcal{P}, \\ 0.625, & k \in \mathcal{F}. \end{cases} \tag{18}$$

The third base, B_3, increases only the pair-update coefficient of Equation 18 from 0.50 to 0.5625:

$$B_3[k] = \begin{cases} \mathcal{E}_{0.5625}[k], & k \in \mathcal{P}, \\ B_2[k], & \text{otherwise}. \end{cases} \tag{19}$$

Thus the effective coefficients of B_3 are 0.5625 for pair-update tensors and 0.625 for other diffusion tensors. Operationally, the latter tensors are copied byte-for-byte from saved B_2. Pair-update tensors are recomputed from M and R via Equation 17, rather than obtained by a further extrapolation from B_2. All three bases preserve the frozen autoencoder unchanged.

Table 11: **Weight-arithmetic ablations.** Unique success rates (%) from 1,000 designs per target and setting. Panel (a) reports best-observed results across available replicates and samplers; it is not a matched comparison with panels (b) or (c). Panel (b) uses a shared sampler for all variants. Panel (c) reevaluates B_2 and $B_3^{\ddagger}$ at a second shared sampler on one compute cluster. Bold identifies the best worst-target and mean rates within each panel.

Construction	Model	PD-L1	BHRF1	Insulin	TrkA	Worst	Mean
(a) Parent checkpoints and uniform mean; best-observed settings							
EMA, 320k	A_1	29.9	28.6	22.4	6.6	6.6	21.9
EMA, 340k	A_2	10.6	59.2	33.4	11.8	10.6	**28.8**
EMA, 350k	A_3	31.3	12.9	33.3	5.7	5.7	20.8
Uniform mean	M	36.0	17.7	39.0	14.3	**14.3**	26.8
(b) Fixed sampler: backbone temperature 0.10*, latent temperature* 0.02							
Global extrapolation	B_1	27.9	36.0	30.4	23.2	23.2	29.4
Module-scoped variant	B_2	24.9	31.7	29.9	23.7	23.7	27.5
Module-scoped base	B_3	26.5	31.1	30.7	23.6	23.6	28.0
Uniform residual, β	B_3'	27.6	32.6	30.4	23.8	23.8	28.6
Uniform residual, 4β	B_3''	30.1	33.4	28.2	24.5	**24.5**	29.1
Uniform residual, 8β	B_3'''	35.5	36.8	24.4	20.6	20.6	29.3
Split update, 4β	$B_3^{\dagger}$	33.0	36.3	30.6	23.9	23.9	30.9
Split update, 8β	$B_3^{\ddagger}$	34.4	38.2	30.9	24.4	24.4	**32.0**
(c) Reevaluation: backbone temperature 0.05*, latent temperature* 0.02							
Module-scoped variant	B_2	22.8	36.9	32.2	27.4	22.8	29.8
Selected merge	$B_3^{\ddagger}$	34.6	43.5	33.6	29.0	**29.0**	**35.2**

A locally centered donor residual. We obtain a residual from a second training branch with a shared 400k-step prefix and a different subsequent data mix. This prefix comprises 230k scratch-training steps followed by 170k finalmix steps, after which the teddy-only branch diverges. Let T_-, T_0, and T_+ denote donor EMA checkpoints at finalmix-local steps 460k, 470k, and 480k. We subtract their local mean:

$$M_D = \frac{T_- + T_0 + T_+}{3}, \qquad D = T_0 - M_D = \frac{2T_0 - T_- - T_+}{3}. \tag{20}$$

Equation 20 is related to the centered residuals used by CART (Choi et al., 2024), but uses temporal neighbors rather than task-specific models and does not apply rank truncation. In exact arithmetic, the subtraction cancels any component common to the three checkpoints and any exactly linear local trajectory across these equally spaced steps. The implementation subtracts the saved mean M_D, so this identity holds up to saved-mean rounding. The construction isolates a local deviation rather than transferring the donor's absolute weights.

Separating conditioning and matrix updates. Let $\mathcal{C}$ contain the 98 transition-conditioning tensors and $\mathcal{W}$ the 28 SwiGLU matrices, with exact membership specified below. These groups are disjoint. To test residual strength, we first apply the same coefficient to both groups:

$$B_3^{(s)}[k] = \begin{cases} \mathrm{cast}_k(B_3[k] + s\beta D[k]), & k \in \mathcal{C} \cup \mathcal{W}, \\ B_3[k], & \text{otherwise}, \end{cases} \qquad \beta \approx 0.0288. \tag{21}$$

The variants of Equation 21 with $s = 1, 4, 8$ are denoted by B_3', B_3'', and B_3''', respectively. Each is constructed directly from B_3; the primes do not represent cumulative updates. The calibration of β is given in Equation 25.

We then separate conditioning strength from the matrix update. The masked variant $B_3^\dagger$ copies the saved conditioning tensors of B_3'' and applies a connected-channel mask Q to the matrix residual, where $\odot$ denotes elementwise multiplication:

$$B_3^\dagger[k] = \begin{cases} B_3''[k], & k \in \mathcal{C}, \\ \mathrm{cast}_k(B_3[k] + c(Q[k] \odot D[k])), & k \in \mathcal{W}, \\ B_3[k], & \text{otherwise}, \end{cases} \tag{22}$$

where $c \approx 0.1633$ is calibrated in Equation 26 and Q is specified below. The selected checkpoint $B_3^\ddagger$ doubles only the saved conditioning displacement of Equation 22:

$$B_3^\ddagger[k] = \begin{cases} \mathrm{cast}_k(B_3[k] + 2(B_3''[k] - B_3[k])), & k \in \mathcal{C}, \\ B_3^\dagger[k], & \text{otherwise}. \end{cases} \tag{23}$$

Thus $B_3^\dagger$ and $B_3^\ddagger$ use conditioning strengths of 4β and 8β, respectively, up to intermediate rounding, while their masked matrices are identical. The remaining 625 diffusion tensors and all 765 autoencoder tensors are unchanged from B_3. No further normalization is applied after doubling the conditioning displacement.

These updates follow the parameter-difference formulation of task arithmetic (Ilharco et al., 2023). Selective updates also relate to TIES and DARE (Yadav et al., 2023; Yu et al., 2024), although our architectural channel mask differs from their respective trimming, sign-resolution, and random drop-and-rescale procedures. For fixed checkpoints, coefficients, and mask, the construction produces one merged model with no additional inference cost.

Choosing the residual scales. We calibrate residual coefficients by matching displacement norms, allowing comparisons of directions and supports at controlled update magnitude. Reference-norm calibration has precedent in LiNeS (Wang et al., 2025a), while STAR restores residual magnitude after spectral truncation using a nuclear-norm criterion (Lee et al., 2025). Our construction uses a global Euclidean norm over the selected parameter groups. The reference checkpoints and groups were experimental choices; β and c were calculated from these choices, while residual-strength multipliers and the final construction were selected empirically.

For a tensor group $\mathcal{G}$, define $\|X\|_{\mathcal{G}}^2 = \sum_{k \in \mathcal{G}} \|X[k]\|_F^2$, where $\|\cdot\|_F$ is the square root of the sum of squared tensor entries. Let $\mathcal{S} = \mathcal{C} \cup \mathcal{W}$. An earlier reference checkpoint H applies a weak donor mixture to these 126 transition tensors:

$$H[k] = \mathrm{cast}_k\left(\frac{255B_3[k] + M_D[k]}{256}\right), \qquad k \in \mathcal{S}, \tag{24}$$

with all other tensors copied from B_3, and M_D as defined in Equation 20. We match the initial centered-residual update to the reference displacement of Equation 24:

$$r_{\mathcal{S}} = \|H - B_3\|_{\mathcal{S}}, \qquad \beta = \min\left\{\frac{1}{8}, \frac{r_{\mathcal{S}}}{\|D\|_{\mathcal{S}}}\right\}. \tag{25}$$

The measured norms are $r_{\mathcal{S}} \approx 0.693092084221$ and $\|D\|_{\mathcal{S}} \approx 24.064456617649$, yielding $\beta \approx 0.0288$; the cap is inactive. This makes the first centered update approximately equal in magnitude to the earlier mixture while changing its direction. The reference H supplies only the radius, not an additional weight contribution. The subsequent factors $s \in \{1, 4, 8\}$ in Equation 21 control residual strength.

For the masked matrix update of Equation 22, we preserve the matrix displacement of B_3'':

$$r_{\mathcal{W}} = \|B_3'' - B_3\|_{\mathcal{W}}, \qquad c = \frac{r_{\mathcal{W}}}{\|Q \odot D\|_{\mathcal{W}}}. \tag{26}$$

Here $r_{\mathcal{W}} \approx 2.757607797344$ and $\|Q \odot D\|_{\mathcal{W}} \approx 16.887470682080$, yielding $c \approx 0.1633$. This is one global normalization over the 28 matrices, not separate normalization per block. Because the half-mask retains approximately half the squared matrix-residual norm, c is close to $\sqrt{2}(4\beta)$.

Effects of residual strength and scope. At the fixed sampler in Table 11b, module-scoped extrapolation alone provides little improvement in the worst-target rate over B_1. Adding the centered residual of Equation 20 at moderate strength improves balance: B_3'' reaches a worst-target rate of 24.5%. Increasing the coefficient uniformly to 8β improves PD-L1 and BHRF1 but lowers insulin and TrkA success, reducing the worst-target rate to 20.6%. Separating conditioning and matrix updates in $B_3^{\ddagger}$ (Equation 23) raises the mean to 32.0% while retaining a worst-target rate of 24.4%, close to the best observed value in this panel. Thus, the strongest uniform residual does not provide the best cross-target balance.

Reevaluation and interpretation. At the second shared sampler (Table 11c), $B_3^{\ddagger}$ improves all four target rates relative to B_2, increasing the worst-target rate from 22.8% to 29.0% and the mean from 29.8% to 35.2%. This comparison evaluates the combined construction, since the models differ in both base extrapolation (Equations 18 and 19) and residual updates (Equations 22 and 23). Best-observed results reflect selection over available checkpoints, samplers, and replicates, rather than an independent estimate after selection. Ordinary per-arm binomial confidence intervals would not account for this selection or measure variation across independent runs; deduplication also means unique-success counts are not automatically binomial. The comparisons are therefore descriptive. Together, these experiments support weight arithmetic as a practical means of improving cross-target balance within one checkpoint, without establishing statistical significance or generalization beyond the evaluated targets.

F Nanobody Metrics

F.1 Target Selection

Literature-based comparison set. The main benchmark comprises seven antigens: PD-L1, BHRF1, insulin, TrkA, IL-3, CTLA-4, and CEACAM6. Our selection emphasizes comparison on established design problems, following the literature-based benchmarking approach of RFdiffusion, AlphaProteo, Germinal, and Proteina-Complexa (Watson et al., 2023; Zambaldi et al., 2024; Mille-Fragoso et al., 2026; Didi et al., 2026). RFdiffusion evaluated binder design against five targets, including PD-L1 and TrkA. Proteina-Complexa subsequently assembled 22 target settings from AlphaProteo and BindCraft, retaining target crops and hotspot specifications from the source studies. These precedents motivate retaining shared antigens and explicit epitope specifications for comparisons across design methods.

We adapt the approach of Germinal to framework-constrained nanobody generation: all methods receive the same target-specific inputs, hNbBCII10 framework, prescribed CDR lengths, and sampling budget. Overlap in antigen identity does not imply identical design tasks across publications, since target crops, epitopes, binder formats, and evaluation protocols can differ. Our reported rates therefore compare methods within the present benchmark rather than directly against published success rates. We deliberately emphasize well-characterized, literature-standard antigens, where reference structures and established epitopes support interpretable evaluation. These common inputs control the comparison across methods; they do not establish equal training exposure or exclude antigen overlap with any model's pretraining data. The development targets are PD-L1, BHRF1, insulin, and TrkA; IL-3, CTLA-4, and CEACAM6 are separated from checkpoint selection. This split is distinct from a training-data holdout and from inference-setting selection.

PD-L1 uses antigen chain A only; the reference nanobody in chain B is excluded. Both insulin chains belong to the antigen. The supplied Germinal scaffold has 91 fixed framework residues and 37 designed CDR residues, for a total VHH length of 128.

F.2 Structure Prediction Metrics

Evaluation setting and relationship to Germinal. Our benchmark extends the framework-constrained design setting of Germinal (Mille-Fragoso et al., 2026). We use the hNbBCII10 framework, prescribe HCDR1/HCDR2/HCDR3 lengths of $11/8/18$, and evaluate $1{,}000$ generated designs

Table 12: **Shared nanobody target inputs.** Residue identifiers use original PDB chain labels and numbering, not sequence offsets.

Target	Crop	Hotspots
PD-L1	A1–115	A37, A39, A41, A96, A98
BHRF1	A2–158	A65, A74, A77, A82, A85, A93
Insulin	A1–21; B1–30	A2, A21, B12, B16
TrkA	X282–382	X294, X296, X333
IL-3	A1–112	A23, A25, A26, A31, A40, A104
CTLA-4	G2–119	G45, G47, G63, G103
CEACAM6	B2–108	B30, B42, B50, B90, B96, B100

Table 13: **Fixed framework segments in the Germinal scaffold.** IMGT intervals contain numbering gaps; lengths count actual residues. CDR1/2/3 occupy IMGT 27–38, 56–65, and 105–117 (with insertions), respectively, and have fixed lengths 11/8/18.

Region	IMGT	Length	Sequence
FR1	1–26	25	QVQLVESGGGLVQPGGSLRLSCAAS
FR2	39–55	17	LGWFRQAPGQGLEAVAA
FR3	66–104	38	YYADSVKGRFTISRDNSKNTLYLQMNSLRAEDTAVYYC
FR4	118–128	11	WGQGTLVTVSS

per target under common criteria. [3] We adopt the use of structure-prediction confidence to screen candidate complexes, but define a common evaluation gate rather than reproduce Germinal's complete experimental selection pipeline. In particular, the thresholds below, exact framework preservation, and sequence deduplication define our benchmark endpoint.

All methods are evaluated using Protenix-v2 (Zhang et al., 2026) under the same folding protocol. Its confidence outputs describe the predicted complex; they do not measure binding affinity or establish that an interaction occurs experimentally. We distinguish these reference-free confidence metrics from DockQ, which measures structural accuracy against an experimentally determined complex.

Oracle choice and search dependence. We use Protenix-v2 as our common evaluator based on its reported antibody–antigen prediction capability (Zhang et al., 2026). Following the predictor-guided inference-time search setting of Proteina-Complexa (Didi et al., 2026), Mixta search optimizes interface PAE from the predictor also used for final evaluation. This creates an advantage specific to this evaluation protocol: its reward directly optimizes one component of the success gate. A common evaluator does not remove this dependence, and an independent-oracle assessment remains necessary to test transfer.

Validation on experimentally determined VHH complexes: setup. We benchmark AF2, template-conditioned Protenix-v2, OpenDDE, ESMFold2, and ColabDesign on the same SAbDab VHH–protein-antigen complex set (Schneider et al., 2022). AF2 uses target templates and an initial guess, following Proteina-Complexa (Didi et al., 2026). Starting from 2,905 candidate pairs, we retain structures deposited after 30 September 2021, then filter for protein antigens, valid resolution metadata, and usable cleaned coordinates. This leaves 1,410 pairs, from which joint VHH–antigen sequence clustering selects 508 representatives spanning 440 PDB entries (459 single-chain and 49 multichain antigens).

Coverage varies slightly by predictor: AF2 and Protenix-v2 produce predictions for all 508 representatives; the OpenDDE comparison covers 504 of 508 after excluding failed folds; the ESMFold2, ESMFold2-Fast, and ColabDesign comparison covers 476 of 508, reflecting ESMFold2's lack of template support and ColabDesign's single-sample reference protocol.

[3] Germinal requires CDR lengths to match the supplied reference scaffold; we therefore match scaffolds and lengths across methods.

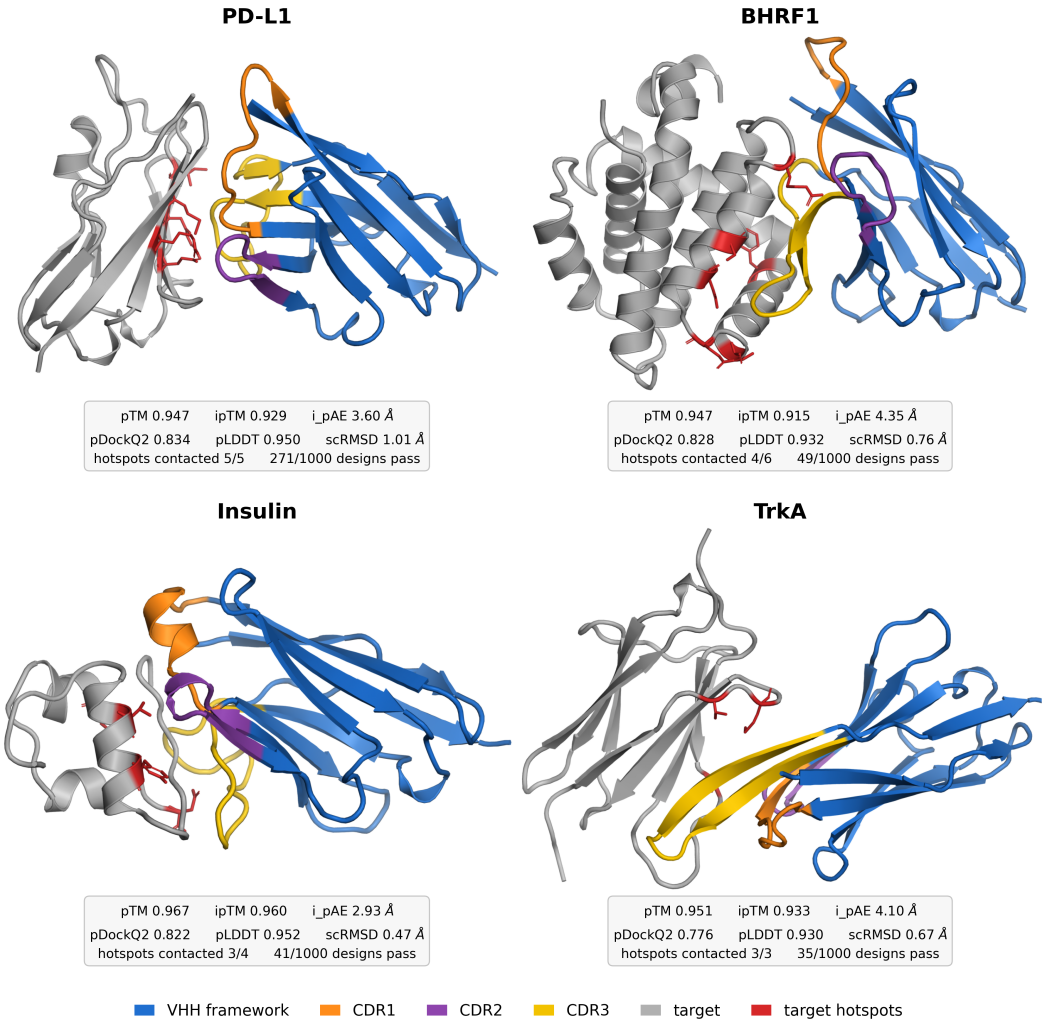


Figure 9: Representative *Proteina Mixta* designs for PD-L1, BHRF1, insulin, and TrkA. One design is shown for each target, with the HCDR regions highlighted. Structure-prediction confidence scores are reported in each panel.

Docking accuracy: AF2 versus Protenix-v2. Evaluating the ranked top-1 prediction from one seed against the experimental structure, Protenix-v2 achieves 64.8% acceptable-or-better docking accuracy (DockQ > 0.23), versus 34.1% for AF2, with high-quality predictions (DockQ > 0.8) at 47.6% versus 24.2% (Basu & Wallner, 2016).

Extended comparison across predictors. OpenDDE's two checkpoints and Protenix-v2 achieve comparable acceptable-or-better rates (65–68%), while ESMFold2, ESMFold2-Fast, and ColabDesign trail substantially (36–39%), driven primarily by a lower high-quality fraction rather than a lower acceptable-quality fraction.

Confidence-gate recovery. Separately, the four Germinal-style confidence gates accept $151/508$ (29.72%) Protenix-v2 predictions and $20/508$ (3.94%) AF2 predictions; all 20 AF2 passes also pass with Protenix-v2. Under the alternative AlphaProteo-style gate (interface PAE ≤ 7 Å, pLDDT ≥ 0.9, and binder C_α scRMSD < 1.5 Å), the corresponding counts are $169/508$ (33.27%) and $51/508$ (10.04%).

Table 14: **VHH–antigen prediction accuracy and confidence screening.** AF2 and Protenix-v2 evaluate all 508 SAbDab representatives. DockQ uses ranked top-1 predictions from a single seed; percentages are reported to one decimal place. Confidence-gate counts are separate endpoints and individual-gate counts overlap.

Endpoint	AF2	Protenix-v2
Reference-based structure accuracy		
DockQ > 0.23 (acceptable or better)	34.1%	64.8%
$0.49 <$ DockQ ≤ 0.8 (medium)	7.5%	13.6%
DockQ > 0.8 (high)	24.2%	47.6%
Confidence-gate acceptance (counts out of 508)		
Interface PAE ≤ 6 Å	20	151
pTM ≥ 0.83	121	291
ipTM ≥ 0.75	125	311
$\text{pDockQ}_2 \geq 0.45$	103	296
All four gates	20 (3.94%)	151 (29.72%)

Table 15: **DockQ success tiers across evaluated predictors, same SAbDab set.** OpenDDE rows cover 504/508 representatives; ESMFold2, ESMFold2-Fast, and ColabDesign rows cover 476/508 (see coverage note above). Ranked top-1 prediction from a single seed.

Method	High (> 0.8)	Medium (0.49–0.8)	Acceptable (0.23–0.49)	Acceptable or better
Protenix-v2	48%	13%	4%	65%
OpenDDE-abag	50%	13%	3%	66%
OpenDDE-v1	51%	15%	2%	68%
ESMFold2	12%	20%	7%	39%
ESMFold2-Fast	9%	21%	8%	38%
ColabDesign	25%	8%	3%	36%

Interpretation and caveats. These results support Protenix-v2's use in our benchmark through improved measured docking accuracy and greater recovery of known complexes under confidence filtering. Nevertheless, its strict confidence gates reject $357/508$ (70.28%) known complexes, so computational failure cannot establish absent binding. The deposition-date filter reduces temporal overlap but does not alone verify exclusion from each model's training data.

Template support and predictor comparison. ESMFold2 does not support structural templating and is therefore assessed without templates. In our comparison, its docking accuracy is of the same order as AF2, which uses target templates and an initial guess following Proteina-Complexa (Didi et al., 2026). This comparison reflects the supported inference protocols; it does not isolate the effect of templates from differences between models.

Confidence calibration: i_pTM versus DockQ. We additionally bucket predictions by agreement between predicted confidence and realized quality on the common comparison set above. Confidence uses i_pTM (high ≥ 0.6, medium 0.3–0.6, low ≤ 0.3); quality is DockQ> 0.23. *Reliable* predictions pair high confidence with good quality; *false positives* pair high confidence with poor quality; *false negatives* pair low confidence with good quality; *reject* pairs low confidence with poor quality; *medium confidence* predictions are reported separately rather than assigned to a quality-agreement bucket.

Accuracy, confidence calibration, and oracle cost. We additionally evaluate the antibody–antigen-specific checkpoint from OpenDDE (Aureka AI OpenDDE project, 2026). In our comparison, this checkpoint provides an approximately 1% relative improvement in DockQ success over Protenix-v2, but its confidence outputs are less well calibrated and inference cost is approximately twice as high. We therefore select Protenix-v2 for the combination of docking accuracy, confidence calibration, and inference cost among the evaluated backends. This is a choice for repeated computational screening, not a claim that Protenix-v2 maximizes docking accuracy alone.

Table 16: **Confidence-quality calibration buckets.** Percentages sum to 100 per row (rounding). Protenix-v2 has the lowest combined reject and false-positive rate among the evaluated predictors.

Method	Reliable	False positive	False negative	Reject	Medium conf.
Protenix-v2	61%	5%	1%	19%	14%
ESMFold2	29%	5%	3%	48%	16%
ESMFold2-Fast	25%	3%	3%	55%	14%
ColabDesign	29%	0%	3%	64%	4%

CDR-randomization control. Scrambling the 1000 generated CDR sequences yields zero successes under the common computational gate in the assessed negative-control population. This establishes rejection of these controls, but does not prove experimental specificity or exclude all predictor exploitation.

Local confidence: pLDDT. Predicted local distance difference test (pLDDT) estimates the accuracy of local atomic environments (Jumper et al., 2021). The underlying lDDT measure assesses preservation of local interatomic distances without requiring global structural alignment. For a confidence head predicting a distribution $q_i(b)$ over local-accuracy bins with centers $\ell_b \in [0, 1]$,

$$\mathrm{pLDDT}_i = 100 \sum_b q_i(b)\ell_b. \tag{27}$$

Higher values indicate greater local confidence. A well-folded nanobody can nevertheless have an incorrect orientation relative to its antigen, so high pLDDT alone does not establish interface accuracy. We do not impose a separate pLDDT threshold in the strict gate below; local confidence instead contributes to pDockQ2.

Relative-position confidence: PAE and interface PAE. Predicted aligned error (PAE) estimates positional error after alignment in a specified residue or token frame (Jumper et al., 2021). Let $P_{ij}(b)$ denote the predicted distribution over aligned-error bins with centers e_b in Å. Its expectation is

$$A_{ij} = \sum_b P_{ij}(b)e_b. \tag{28}$$

PAE is directional: exchanging the alignment frame and the evaluated position generally changes the score, so A_{ij} need not equal A_{ji}.

Let $\mathcal{B}$ and $\mathcal{T}$ contain the binder and target residues, respectively, with $\mathcal{T}$ including all retained antigen chains. Our mean interface PAE averages both binder–target blocks:

$$\mathrm{iPAE} = \frac{1}{2|\mathcal{B}||\mathcal{T}|} \sum_{i\in\mathcal{B}} \sum_{j\in\mathcal{T}} \left(A_{ij} + A_{ji}\right). \tag{29}$$

Here "interface" denotes interchain residue pairs, rather than only spatially contacting residues. Lower values indicate greater confidence in relative binder–target placement. This all-pairs average can also reflect uncertainty in distant antigen regions and therefore differs from contact-restricted PAE.

Global and interchain confidence: pTM and ipTM. Predicted TM-score (pTM) estimates global structural accuracy using the aligned-error distribution (Jumper et al., 2021). Interface predicted TM-score (ipTM) restricts the aggregation to residues on different chains (Evans et al., 2022). For N protein residues or tokens, define the expected TM contribution

$$K_{ij} = \sum_b P_{ij}(b) \frac{1}{1 + \left(e_b/d_0(N)\right)^2}, \qquad d_0(N) = 1.24\big(\max(N, 19) - 15\big)^{1/3} - 1.8. \tag{30}$$

For uniform residue weights, the standard protein formulation can be written as

$$\mathrm{pTM} = \max_i \frac{1}{N} \sum_{j=1}^{N} K_{ij}, \qquad \mathrm{ipTM} = \max_i \frac{\sum_j m_{ij} K_{ij}}{\sum_j m_{ij}}, \tag{31}$$

where $m_{ij} = 1$ when residues i and j belong to different chains and is zero otherwise. We use the scores returned by the predictor, including its token masks and weighting conventions.

Both scores increase with predicted accuracy, but their scope differs. pTM summarizes the full complex and can be dominated by a large, confidently folded antigen. ipTM emphasizes interchain placement, although for multichain antigens the full-complex score can also reflect antigen–antigen interfaces. Neither score is a calibrated probability of binding. Importantly, Equation 30 averages the TM kernel over the predicted error distribution; applying the kernel to mean PAE is not mathematically equivalent.

Reference-based interface accuracy: DockQ. DockQ combines native-contact recovery and structural alignment errors into a continuous docking-quality score (Basu & Wallner, 2016):

$$\mathrm{DockQ} = \frac{1}{3}\left[F_{\mathrm{nat}} + \frac{1}{1+\left(\mathrm{iRMSD}/1.5\,\text{Å}\right)^2} + \frac{1}{1+\left(\mathrm{LRMSD}/8.5\,\text{Å}\right)^2}\right]. \tag{32}$$

Here F_{nat} is the fraction of native interchain contacts recovered, iRMSD measures interface backbone deviation after interface superposition, and LRMSD measures ligand backbone deviation after superposing the receptor. For a nanobody–antigen complex, the nanobody serves as the ligand and the antigen as the receptor.

DockQ is a principal interface-accuracy endpoint in antibody–antigen structure-prediction benchmarks. For example, FoldBench evaluates antibody–antigen docking using DockQ and reports success at $\mathrm{DockQ} \geq 0.23$ (Xu et al., 2026). However, DockQ requires a reference complex. It cannot be measured for a new design without a corresponding experimental structure. Comparing a generated complex with its refolded prediction instead measures model agreement, not experimental accuracy.

Predicted interface quality: pDockQ2. We therefore include pDockQ2, a reference-free estimator fitted to chain-versus-rest DockQ scores (Zhu et al., 2023). It combines local interface confidence with a nonlinear penalty on interchain PAE. For chain c, let $\mathcal{I}_c$ contain directed contact pairs (i, j) with i on chain c, j on another chain, and a residue-representative distance of at most 10 Å. Define

$$u_c = \frac{1}{|\mathcal{I}_c|}\sum_{(i,j)\in\mathcal{I}_c}\frac{1}{1+\left(A_{ij}/10\,\text{Å}\right)^2}, \qquad h_c = \frac{1}{|\mathcal{I}_c|}\sum_{(i,j)\in\mathcal{I}_c}\mathrm{pLDDT}_i, \qquad x_c = u_c h_c. \tag{33}$$

The implemented contact-wise average weights a residue once for each contact it makes. The fitted mapping is

$$\mathrm{pDockQ2}_c = \frac{L}{1+\exp[-k(x_c - x_0)]} + b, \tag{34}$$

with

$$L = 1.31034849, \qquad x_0 = 84.7326239, \qquad k = 0.0747157696, \qquad b = 0.00501886443. \tag{35}$$

pLDDT is supplied on the 0–100 scale and PAE in Å. These units are essential to the fitted mapping. Unlike pTM, this construction applies the error kernel to the predicted mean PAE.

Our scorer adapts the pDockQ2 evaluation utilities used by Germinal (Mille-Fragoso et al., 2026), while the metric and fitted coefficients originate from Zhu et al. (2023). The inspected implementation uses C_β distances when both residues contain C_β, and otherwise uses C_α distances. It reports the arithmetic mean over the K chains:

$$\overline{\mathrm{pDockQ2}} = \frac{1}{K}\sum_{c=1}^{K}\mathrm{pDockQ2}_c. \tag{36}$$

For multichain antigens, this summary includes antigen–antigen contacts and is not a binder-only interface score. We retain the same aggregation across compared methods.

The DockQ connection motivates inclusion of pDockQ2, but does not make their numerical thresholds interchangeable. In particular, $\overline{\mathrm{pDockQ2}} \geq 0.45$ is our screening criterion, not proof of $\mathrm{DockQ} \geq 0.45$, nor a calibrated probability of experimental binding. Its original calibration on AlphaFold-Multimer predictions also need not transfer unchanged to Protenix-v2 nanobody designs.

Table 17: **TNP flag thresholds used in this work.** Amber intervals include their endpoints. Values outside the listed red and amber conditions are green. Thresholds are taken from the TNP `assign_flag` implementation.

Descriptor	Red condition	Amber intervals
PSH	< 73.40 or > 155.47	$[73.40, 79.59] \cup [126.83, 155.47]$
PPC	> 1.18	$[0.39, 1.18]$
PNC	> 1.88	$[1.47, 1.88]$
L	< 20 or > 39	$[20, 24] \cup [37, 39]$
L_3	< 5 or > 23	$[5, 8] \cup [21, 23]$
C	< 0.56 or > 1.61	$[0.56, 0.81] \cup [1.57, 1.61]$

Strict success and sequence uniqueness. For design d, define the structural-confidence gate

$$G(d) = \mathbf{1}\begin{bmatrix} \mathrm{iPAE}(d) \le 6\,\text{Å} \ \wedge\ \mathrm{pTM}(d) \ge 0.83 \\ \wedge\ \mathrm{ipTM}(d) \ge 0.75 \ \wedge\ \overline{\mathrm{pDockQ2}}(d) \ge 0.45 \end{bmatrix}. \tag{37}$$

These thresholds are fixed across the benchmark. The combination probes global topology, relative chain placement, and contact-local confidence, although the metrics are correlated because they share predictor outputs.

Let $\mathcal{F}$ be the supplied framework positions and a_i^{ref} their reference identities. The framework mutation count is

$$n_{\mathrm{FW}}(d) = \sum_{i \in \mathcal{F}} \mathbf{1}\big[a_i(d) \neq a_i^{\mathrm{ref}}\big]. \tag{38}$$

Strict success requires

$$S(d) = G(d)\,\mathbf{1}[n_{\mathrm{FW}}(d) = 0]. \tag{39}$$

For a target with sampling budget N, let $s(d)$ denote the complete binder sequence. The unique strict success rate is

$$r_{\mathrm{strict}} = \frac{|\{s(d) : d \in \{1, \ldots, N\},\ S(d) = 1\}|}{N}. \tag{40}$$

Identical successful sequences count once per target; the denominator remains the full sampling budget, rather than the number surviving intermediate filters.

Limits of confidence-based evaluation. Antibody–antigen structure prediction remains challenging, including on low-homology benchmarks (Xu et al., 2026). In our assessed reference set, the four confidence gates reject 357 of 508 known PDB VHH complexes (70.28%). Consequently, failure to pass is not evidence that a sequence cannot bind, and passing does not establish binding or affinity. We interpret these rates as performance under a shared computational protocol and evaluate sequence diversity and developability-related properties separately.

F.3 Developability Profiling

TNP evaluation. We assess predicted developability using the Therapeutic Nanobody Profiler (TNP) (Gordon et al., 2026), a nanobody-specific extension of the Therapeutic Antibody Profiler (TAP) (Raybould et al., 2019). TNP predicts each nanobody structure from sequence using NanoBodyBuilder2, part of ImmuneBuilder (Abanades et al., 2023), and evaluates six descriptors against clinical-stage nanobody reference distributions. This profiling uses the nanobody model independently of its predicted antigen-bound pose.

The descriptors are total CDR length (L), CDR3 length (L_3), CDR3 compactness (C), patches of surface hydrophobicity (PSH), patches of positive charge (PPC), and patches of negative charge (PNC). The surface descriptors characterize the CDR vicinity, rather than simply counting hydrophobic or charged amino acids in the sequence. Red flags indicate values outside the implemented clinical reference bounds; amber flags identify the designated reference tails. These are screening categories, not measured probabilities of development failure.

Surface liabilities and fixed CDR geometry. We emphasize PSH, PPC, and PNC because these descriptors discriminate among designs in the analyzed cohort. Their red-flag indicators are

$$\begin{aligned} R_{\mathrm{PSH}}(d) &= \mathbf{1}[\mathrm{PSH}(d) < 73.40 \ \vee \ \mathrm{PSH}(d) > 155.47]\,, \\ R_{\mathrm{PPC}}(d) &= \mathbf{1}[\mathrm{PPC}(d) > 1.18]\,, \\ R_{\mathrm{PNC}}(d) &= \mathbf{1}[\mathrm{PNC}(d) > 1.88]\,. \end{aligned} \tag{41}$$

These descriptors identify unusual surface properties associated with developability concerns; they do not directly measure aggregation, polyspecificity, or pharmacokinetics.

For our fixed $11/8/18$ CDR layout, $L = 37$ is always amber and $L_3 = 18$ is green. CDR3 compactness remains green throughout the analyzed search cohort, with values ranging from 1.07 to 1.51. Thus, requiring every descriptor to be green would exclude every design solely because of the prescribed total CDR length. The reported surface screen permits amber flags.

Defining TNP-clean success. Define the absence of surface red flags as

$$C_{\mathrm{TNP}}(d) = \prod_{m \in \{\mathrm{PSH,PPC,PNC}\}} [1 - R_m(d)]\,. \tag{42}$$

A TNP-clean success is a strict success with $C_{\mathrm{TNP}}(d) = 1$. Its unique success rate is

$$r_{\mathrm{clean}} = \frac{|\{s(d) : S(d) = 1,\ C_{\mathrm{TNP}}(d) = 1\}|}{N}. \tag{43}$$

This denominator is the full generation budget. It differs from the fraction of successful binders carrying a liability, whose denominator is the evaluated successful-binder cohort.

A binder carrying multiple red flags is counted once in the union of flagged designs. Descriptor-specific red fractions may therefore overlap and should not be added. Missing or failed TNP assessments do not establish that a design is clean and should be reported separately. TNP-clean success remains a computational screening endpoint, not evidence of experimental developability.

Positive-charge patches dominate the observed liabilities. We analyze an earlier search cohort of $2{,}371$ strict successes, distinct from the updated search configurations in Table 1. PPC flags 22.9% of this cohort, compared with 7.1% for PSH and 3.2% for PNC; these fractions overlap. Of 542 PPC-red designs, 450 (83.0%) target PD-L1, making this target the largest contributor to the observed positive-charge liability burden.

Hydrophobicity approaches the upper reference tail. All PSH-red designs exceed the upper threshold: the minimum observed PSH is 77.33, above the lower red cutoff. Median PSH is 126.06, close to the start of the upper amber interval at 126.83, and 41.8% of designs receive an amber PSH flag. Thus, elevated hydrophobicity is common even among designs that avoid a red flag. These findings identify surface charge, particularly on PD-L1, and hydrophobicity as priorities for improving candidate quality alongside structural confidence.

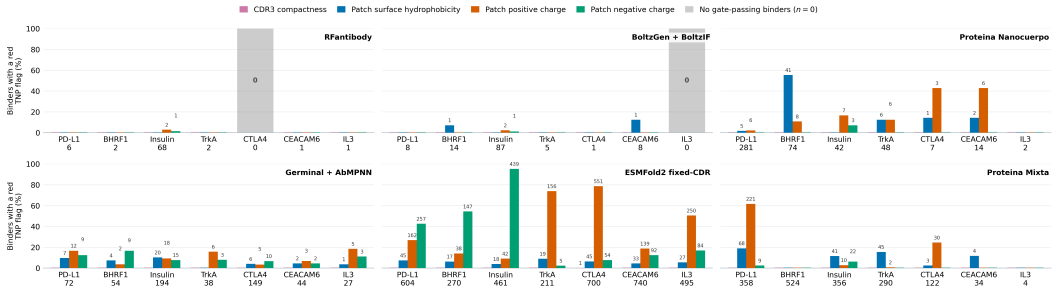


Figure 10: **Therapeutic red flags across design methods.** Percentage of gate-passing binders carrying at least one red TNP flag that measure developability issues, shown per target for six design methods. Bars are colored by flag category (CDR3 compactness, patch surface hydrophobicity, patch positive charge, patch negative charge); the number above each bar is the count of flagged binders, and the number below each target label is the total number of gate-passing binders (n). Gray panels denote targets for which a method produced no gate-passing binders ($n = 0$).

G Generation Hyperparameters and Sampling Details

G.1 Nanobody Sampling Hyperparameters

Shared model and sampler. Both Proteina Mixta variants use the same selected merged checkpoint, $B_3^{\ddagger}$, and frozen 250k-update EMA autoencoder. Generation jointly evolves backbone C_α coordinates and residue-level latents using 400 integration steps with self-conditioning enabled, classifier-free guidance weight 1.0, no fold-class conditioning, and no recycling. We use generation seed 5 and retain the fixed $11/8/18$ CDR lengths throughout the benchmark. Direct generation uses one complete denoising trajectory per design; the search variant branches and scores trajectories during denoising.

Backbone and latent temperatures. We tune backbone and latent stochasticity independently. The reported temperatures, τ_{bb} and τ_{lat}, correspond to the sampler's noise-variance multipliers rather than softmax temperatures. For modality $m \in \{\mathrm{bb}, \mathrm{lat}\}$, the stochastic increment takes the form

$$\Delta \mathbf{u}_m^{\mathrm{noise}} = \sqrt{2 g_m(t) \tau_m \Delta t}\, \boldsymbol{\epsilon}_m, \qquad \boldsymbol{\epsilon}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{44}$$

Thus, the injected standard deviation scales with $\sqrt{\tau_m}$. These parameters control noise injection and do not directly rescale the deterministic drift. The stochastic $\mathtt{sc}$ sampler uses configured score scaling 1.0 and endpoint ODE settings $t_{\mathrm{low}} = 0.02$ and $t_{\mathrm{high}} = 0.98$. These are taken directly from Didi et al. (2026).

Direct-generation settings. The non-search row evaluates $1{,}000$ designs per target using a fixed temperature pair selected for that target from a preceding sweep (Table 18). Generation uses 40 shards with batch size 16. These results therefore use target-specific sampler selection, rather than one universal temperature pair. Most selected settings reduce latent noise more strongly than backbone noise; none uses the default pair $(0.10, 0.10)$. We note time comparisons are normalized to measure time to generate single sample and are not biased by batching or parallelization efforts.

Table 18: **Target-specific sampling temperatures.** IL-3, CTLA-4, and CEACAM6 were held out from checkpoint selection, but participate in the subsequent inference grid searches. Each pair denotes $(\tau_{\mathrm{bb}}, \tau_{\mathrm{lat}})$. Direct generation uses one fixed pair per target. Search selects one listed pair per batch and retains it throughout that batch's trajectories.

Target	Direct generation	Search temperature pairs
PD-L1	$(0.05, 0.10)$	$(0.05, 0.10)$, $(0.05, 0.02)$, $(0.02, 0.10)$, $(0.10, 0.10)$
BHRF1	$(0.01, 0.02)$	$(0.02, 0.04)$, $(0.01, 0.02)$, $(0.05, 0.02)$, $(0.10, 0.10)$
Insulin	$(0.02, 0.02)$	$(0.02, 0.02)$, $(0.05, 0.02)$, $(0.02, 0.04)$, $(0.10, 0.10)$
TrkA	$(0.05, 0.02)$	$(0.05, 0.02)$, $(0.02, 0.02)$, $(0.05, 0.10)$, $(0.10, 0.10)$
IL-3	$(0.10, 0.02)$	$(0.10, 0.02)$, $(0.20, 0.02)$, $(0.05, 0.02)$, $(0.10, 0.10)$
CTLA-4	$(0.10, 0.01)$	$(0.10, 0.02)$, $(0.10, 0.01)$, $(0.15, 0.02)$, $(0.10, 0.10)$
CEACAM6	$(0.10, 0.02)$	$(0.20, 0.02)$

Beam-search procedure. We follow the predictor-guided search setting of Proteina-Complexa (Didi et al., 2026), using the evaluation predictor for the interface-PAE reward; this is not independent-oracle validation. The search row uses beam search with completed lookahead rollouts, following the inference-time search approach of Proteina-Complexa (Didi et al., 2026). For each origin, W active beams branch into B children each. Children advance to the next search checkpoint, then undergo a lookahead rollout to the end of denoising for decoding and reward evaluation. The W highest-reward candidates are selected within each origin, and their states at the search checkpoint seed the next round. Selection uses ordinary top-W ranking, with no sequence-duplicate penalty and no absolute reward cutoff.

We use three checkpoint schedules:

$$\begin{aligned} \mathcal{S}_3 &= (0, 134, 267, 400), \\ \mathcal{S}_8 &= (0, 50, 100, 150, 200, 250, 300, 350, 400), \\ \mathcal{S}_{8,\mathrm{late}} &= (0, 200, 225, 250, 275, 300, 325, 350, 400). \end{aligned} \tag{45}$$

The late schedule postpones the first selection until step 200, then concentrates subsequent selections in the latter half of denoising. Table 19 gives the target-specific beam configurations.

Table 19: **Beam-search configurations.** W is beam width, B is children per beam, and K is the number of branch-and-select rounds. Saved outputs include completed lookahead samples and final beam survivors, before deduplication. We found the new schedules to improve results for targets that were more difficult for the base model

Target	$W \times B$	Schedule	K	Origins	Saved outputs
PD-L1	4×4	$\mathcal{S}_3$	3	768	39,936
BHRF1	4×4	$\mathcal{S}_3$	3	768	39,936
Insulin	4×4	$\mathcal{S}_3$	3	768	39,936
TrkA	4×4	$\mathcal{S}_3$	3	1,152	59,904
IL-3	2×8	$\mathcal{S}_{8,\mathrm{late}}$	8	384	49,920
CTLA-4	4×4	$\mathcal{S}_8$	8	384	50,688
CEACAM6	4×4	$\mathcal{S}_{8,\mathrm{late}}$	8	384	50,688

Search reward. Candidates are scored using Protenix-v2 (Zhang et al., 2026), with fold seed 101, bf16 precision, and the cuEquivariance kernel. Reward-side folding uses neither MSAs nor templates. The sole active reward is negative mean interface PAE:

$$\mathcal{R}(d) = -\frac{\mathrm{iPAE}(d)}{31\,\text{Å}}, \tag{46}$$

where iPAE averages both binder–target cross-chain blocks and is not restricted to hotspot residues. All other reward components have zero weight. Consequently, pTM, ipTM, pDockQ2, and developability descriptors do not enter the selection objective, although confidence metrics remain correlated.

Reward folds are cached by binder sequence within each run. Repeated sequences reuse the earlier predicted structure and confidence values, rather than obtaining a new fold for each generated conformation. Reward-side predictions and downstream evaluation should therefore be distinguished, including their respective input protocols.

Temperature mixtures during search. For six targets, search uses four temperature pairs, including the default $(0.10, 0.10)$ control. CEACAM6 instead uses the single pair $(0.20, 0.02)$. A deterministic hash of mixture seed 0, shard identity, and the batch's first origin selects a pair by taking the hash modulo the number of available pairs. The choice consumes no denoising random numbers and remains fixed across every branch, checkpoint, and lookahead rollout in that batch. Both temperatures are recorded with each output. This provides a reproducible allocation across settings, but does not guarantee equal representation in a finite run or its reported prefix.

Target-specific modifications. The antigen is centered on its hotspot residues with probability 0.5, increased to 1.0 for CEACAM6 search. IL-3 uses beam width 2 with eight children per beam, retaining the same $WB = 16$ expansion as the 4×4 configurations. It also replaces the default backbone noise schedule $g_{\mathrm{bb}}(t) \simeq 1/t$ with $g_{\mathrm{bb}}(t) \simeq (1-t)/t$, with numerical stabilization at the endpoints. This further suppresses backbone noise late in denoising.

Execution and reported sample population. Search uses bf16 precision, `torch.compile`, 48 generation shards, and 48 evaluation shards. The batch size is eight origins, and the search chunk limit is 128, matching the expanded batch size $8WB$. All completed lookahead structures are saved alongside final survivors. For n_{orig} origins and K search rounds, the configured output count is

$$N_{\mathrm{saved}} = n_{\mathrm{orig}}(WBK + W), \qquad N_{\mathrm{survivors}} = n_{\mathrm{orig}}W. \tag{47}$$

Reported search success rates use the first 1,000 designs produced in generation order, not a reward-ranked subset, followed by the common strict gate and sequence deduplication. This population includes lookahead candidates, not only final beam survivors, and related candidates can share trajectory prefixes. The 1,000-design reporting denominator is therefore distinct from the total number of generated candidates, reward evaluations, or computational budget of the search run. The remaining saved outputs are retained for analysis and do not enter the reported 1,000-design success rates.

Table 20: **Nanobody design under shared and tuned inference settings.** Rates use 1,000 outputs per target, the hNbBCII10 framework, and CDR lengths $11/8/18$. Cells report unique strict TNP-clean success (no PSH/PPC/PNC red flags), with pre-TNP success in parentheses; dashes denote unavailable measurements. All Mixta rows use one checkpoint and frozen AE. Numeric subscripts specify shared backbone/latent temperatures. Shared search fixes a 4×4 beam and checkpoints $0/134/267/400$, retaining per-target temperature mixtures. Held-out refers to checkpoint selection, not inference tuning. Bold/underline mark the best/second-best TNP-clean rates within each block. Search uses the first 1,000 generated outputs, not reward-ranked selection (Appendix G).

			Development targets				Held out from checkpoint selection		
	Method	**Sequence redesign**	**PD-L1**	**BHRF1**	**Insulin**	**TrkA**	**IL-3**	**CTLA-4**	**CEACAM6**
Oracle free	BoltzGen	Boltz-IF	0.8% (0.8%)	1.3% (1.4%)	8.4% (8.7%)	0.5% (0.5%)	0.0% (0.0%)	0.1% (0.1%)	0.7% (0.8%)
	RFantibody	ProteinMPNN	0.6% (0.6%)	0.2% (0.2%)	6.5% (6.8%)	0.2% (0.2%)	0.1% (0.1%)	0.0% (0.0%)	0.1% (0.1%)
	Proteina-Nanocuerpo	None	**27.0%** (28.1%)	3.0% (7.4%)	3.3% (4.2%)	3.6% (4.8%)	0.2% (0.2%)	0.4% (0.7%)	0.6% (1.4%)
	Proteina Mixta$_{(0.05,\,0.02)}$	None	8.8% (34.6%)	43.5% (43.5%)	27.1% (33.6%)	24.3% (29.0%)	0.2% (0.2%)	7.8% (10.7%)	2.6% (2.9%)
	Proteina Mixta$_{(0.10,\,0.02)}$	None	8.7% (34.1%)	38.1% (38.1%)	24.9% (30.9%)	19.9% (23.7%)	0.4% (0.4%)	8.7% (11.9%)	3.0% (3.4%)
	Proteina Mixta$_{(0.05,\,0.10)}$	None	9.1% (35.8%)	38.1% (38.1%)	24.7% (30.6%)	17.6% (21.0%)	0.2% (0.2%)	7.4% (10.2%)	1.9% (2.2%)
	Proteina Mixta$_{\text{tuned}}$	None	9.1% (35.8%)	**52.4%** (52.4%)	**28.7%** (35.6%)	**24.3%** (29.0%)	**0.4%** (0.4%)	**8.9%** (12.2%)	**3.0%** (3.4%)
Oracle based	Germinal	AbMPNN	4.8% (7.2%)	3.9% (5.4%)	14.5% (19.4%)	2.9% (3.8%)	1.9% (2.7%)	12.9% (14.9%)	3.7% (4.4%)
	BindCraft2	ProteinMPNN	1.0% (1.6%)	3.9% (6.1%)	12.0% (27.2%)	10.7% (22.3%)	12.9% (22.9%)	**28.9%** (55.4%)	3.7% (9.4%)
	ESMFold2	None	**23.9%** (60.4%)	9.8% (27.0%)	1.7% (46.1%)	4.7% (21.1%)	**18.2%** (49.5%)	12.2% (70.0%)	**50.3%** (74.0%)
	Proteina Mixta$_{\text{search, shared}}$	None	17.7% (59.9%)	61.7% (61.9%)	35.0% (48.1%)	30.9% (34.2%)	2.8% (2.9%)	10.1% (16.3%)	9.4% (10.1%)
	Proteina Mixta$_{\text{search, tuned}}$	None	17.7% (59.9%)	**61.7%** (61.9%)	**35.0%** (48.1%)	**30.9%** (34.2%)	10.0% (10.4%)	14.0% (22.7%)	22.8% (24.4%)

Table 21: **Temperature mixtures for shared beam search.** Pairs denote backbone and latent noise scales. All targets use the same 4×4 beam and three-segment selection schedule. One pair is selected per batch and retained throughout its trajectories. The default $(0.10, 0.10)$ pair is included in every mixture.

Target	Pair 1	Pair 2	Pair 3	Default control
PD-L1	$(0.05, 0.10)$	$(0.05, 0.02)$	$(0.02, 0.10)$	$(0.10, 0.10)$
BHRF1	$(0.02, 0.04)$	$(0.01, 0.02)$	$(0.05, 0.02)$	$(0.10, 0.10)$
Insulin	$(0.02, 0.02)$	$(0.05, 0.02)$	$(0.02, 0.04)$	$(0.10, 0.10)$
TrkA	$(0.05, 0.02)$	$(0.02, 0.02)$	$(0.05, 0.10)$	$(0.10, 0.10)$
IL-3	$(0.10, 0.20)$	$(0.10, 0.02)$	$(0.20, 0.20)$	$(0.10, 0.10)$
CTLA-4	$(0.10, 0.02)$	$(0.10, 0.01)$	$(0.15, 0.02)$	$(0.10, 0.10)$
CEACAM6	$(0.10, 0.02)$	$(0.15, 0.02)$	$(0.05, 0.02)$	$(0.10, 0.10)$

G.2 Sensitivity to Inference Settings

One checkpoint supports direct generation and search. All Proteina Mixta variants use the same generator checkpoint and frozen autoencoder, without target-specific training or checkpoint selection. For direct generation, the best evaluated shared temperature pair, $(0.05, 0.02)$, produces $1{,}545$ unique strict successes across $7{,}000$ designs, retaining 91.5% of the $1{,}688$ successes obtained with per-target temperature selection. These comparisons concern success before TNP screening, since TNP-clean rates are unavailable for the shared-temperature populations.

Shared search configuration. We additionally evaluate a common beam-search configuration across all seven targets: beam width $W = 4$, four children per beam, and selection checkpoints at steps $0/134/267/400$. Batch size is eight, the maximum expanded batch size is 128, and the temperature-mixture seed is zero. All targets use the same negative interface-PAE reward. Each target nevertheless retains its own four-pair temperature mixture (Table 21). Thus, this comparison removes target-specific choices of beam geometry and selection schedule, but does not impose a shared temperature distribution.

Effect of target-specific search settings. Shared search produces $2{,}334$ unique strict successes, compared with $2{,}616$ for target-specific search, retaining 89.2% of the tuned yield. The first four targets use the same configurations in both rows and consequently have identical results. The additional 282 successes arise on IL-3, CTLA-4, and CEACAM6, whose rates increase from 2.9%, 16.3%, and 10.1% to 10.4%, 22.7%, and 24.4%, respectively. These gains reflect the combined changes in inference settings rather than beam geometry alone. In particular, tuned IL-3 search changes the temperature mixture, beam shape, selection schedule, and backbone noise taper, while

tuned CEACAM6 search also changes temperature and hotspot centering. The shared-search result therefore demonstrates strong performance with a common beam configuration, while leaving temperature selection target-specific.

Rationale for target-specific sampling temperatures. The balance between exploration and structural consistency can depend on the target epitope. Surface curvature, accessibility, and the distribution of hydrophobic and polar contacts constrain the binder orientations and CDR conformations compatible with an interface. Prior binder design studies illustrate this dependence; for example, TrkA binders engage a curved surface of its ligand-binding domain through complementary hydrophobic and polar interactions (Cao et al., 2022; Watson et al., 2023). With a fixed VHH framework and CDR lengths, different epitopes may therefore benefit from different amounts of backbone and latent stochasticity. Lower noise can favor consistency within promising configurations, whereas higher noise can broaden the conformations and sequence-derived representations explored. We consequently tune these two noise scales independently while retaining the same model weights. This is a mechanistic motivation, rather than a demonstrated explanation of each selected temperature: we have not established a quantitative relationship between epitope accessibility and optimal noise scale, and greater target difficulty does not necessarily imply that higher temperatures are preferable. Table 20 quantifies sensitivity to these choices. Shared search fixes its structure, but retains target-specific temperature mixtures; it does not remove all inference tuning.

G.3 Turbo Mode: Accelerating Structure Prediction Drives Design Throughput

Turbo accelerates the folding oracle. Turbo uses the same Mixta checkpoint, search policy, and compiled bf16 generation path, with an accelerated implementation of the Protenix-v2 reward oracle (Anthropic, 2026). The optimization kit's `fast` mode combines fused kernels, CUDA-graph capture, removal of redundant computation, and reduced precision. On IL-3, it produces $2.11\times$ more designs per GPU-hour at a similar unique-success rate per design ($1.05\times$), giving approximately $2.2\times$ more unique successes per GPU-hour.

H Online Sequence-Only Reinforcement Learning

Because Proteina Mixta learns a single shared latent space over sequence and structure, the same online RL machinery we use to post-train structure-conditioned generation applies directly to sequence-only generation, without any architectural change. This appendix reports the result of applying it: online RL substantially improves the PD-L1 success rate of Mixta's sequence-only mode under our independent structural oracle, from 0.1% at the untrained starting checkpoint to 83.0% and 42.1% for two guard ablations. We treat this as initial evidence that Mixta's joint sequence–structure framework is a viable substrate for sequence-only RL, and scope this appendix accordingly: training details, the achieved PD-L1 result, and the configuration ablations that produced it. Extending training beyond PD-L1 to the other four targets, and a full mechanistic study of the reward's robustness to composition-driven shortcuts, are directions we leave to future work and only summarize briefly below.

H.1 Task, objective and reward

Task. Mixta generates a VHH binder against PD-L1 (germinal nanobody framework (Mille-Fragoso et al., 2026), hotspot residues A37/A39/A41/A96/A98, fixed CDR lengths 11/8/18) in sequence-only mode: binder coordinates are withheld and only the amino-acid sequence is emitted, while the antigen retains full all-atom structure.

Related work. Prior work has applied reinforcement learning to antibody CDR sequence design using online/offline value-based RL over binding-affinity predictors (Vogt et al., 2024b), RL-guided latent diffusion (Vogt et al., 2024a), and on-policy distillation against native structural geometry (Yang et al., 2026). These methods train a dedicated sequence-only generator; here, RL instead post-trains one operating mode of a single joint sequence–structure model, which is what allows the same online RL setup used elsewhere in this work to transfer to sequence-only generation with no separate training pipeline.

Online RL objective. We use Advantage-Weighted Matching (AWM) (Xue et al., 2026), which reweights the flow-matching loss by a *group-relative* advantage under a clipped, off-policy importance ratio against a frozen (EMA) old policy. For a group $\mathcal{G}$ of G proposals sharing a rollout context, with per-design reward r_i,

$$A_i = r_i - \frac{1}{G}\sum_{j\in\mathcal{G}} r_j, \tag{48}$$

$$\rho_i(\theta) = \exp\big(\mathcal{L}_{\text{old}}(x_i) - \mathcal{L}_\theta(x_i)\big), \tag{49}$$

$$\mathcal{L}_{\text{AWM}}(\theta) = -\,\mathbb{E}_{i\in\mathcal{G}}\Big[\,\text{clip}\,\big(\rho_i(\theta),\,0,\,c\big)\,A_i\Big], \tag{50}$$

where $\mathcal{L}_\theta$ is the per-sample flow-matching loss under the current policy and c is a hard off-policy ratio clip.

Reward. The training reward scores a refolded binder–antigen complex (folding backend: ESMFold2-Fast) against three threshold-violation terms plus a joint-pass bonus:

$$\text{viol}_{i\text{PAE}} = \max\Big(0,\ \tfrac{i\text{PAE}-7\,\text{Å}}{7\,\text{Å}}\Big), \tag{51}$$

$$\text{viol}_{i\text{pTM}} = \max\Big(0,\ \tfrac{0.75-i\text{pTM}}{0.25}\Big), \tag{52}$$

$$\text{viol}_{\text{pDockQ2}} = \max\Big(0,\ \tfrac{0.45-\text{pDockQ2}}{0.55}\Big), \tag{53}$$

$$r(x) = \tfrac{1}{3}\sum_m \big(1-\text{viol}_m(x)\big)\ +\ 0.5\cdot\mathbb{1}\big[\text{viol}_m(x)=0\ \forall m\big]. \tag{54}$$

This training reward is distinct from, and never used as, the independent evaluation oracle: every pass rate reported anywhere in this work (main text and this appendix) is computed post hoc by an independent Protenix-based refold (five poses, 10 cycles, 200 diffusion steps, no MSA, no binder template/coordinates), applying the stricter evaluation gate over pTM, interface pTM and pDockQ2 (Zhu et al., 2023),

$$\text{gate}(x) = \mathbb{1}\big[i\text{PAE}\le 6\,\text{Å}\big]\wedge\mathbb{1}[\text{pTM}\ge 0.83]\wedge\mathbb{1}[i\text{pTM}\ge 0.75]\wedge\mathbb{1}[\text{pDockQ2}\ge 0.45]\,. \tag{55}$$

We report only this independent gate throughout, never the training-time reward, so that the numbers below reflect an oracle the policy was not directly optimized against.

H.2 Sequence-only RL improves PD-L1 success

Two checkpoints, both trained online on PD-L1 only under Eq. (50)–equation 54 from the same sequence-only starting point, reach substantially higher PD-L1 gate-pass rates than every sequence-only baseline evaluated in the main text (Tab. 22): from 0.1% at the untrained starting checkpoint – indistinguishable from AbLang2 (Olsen et al., 2024) and both ADAPT baselines – to 83.0% (Proteina Mixta†) and 42.2% (Proteina Mixta‡). Both checkpoints share optimizer, distributed-training and EMA settings (AdamW, lr 3×10^{-5}, 512 global proposals/update, gradient-norm clip 0.5, EMA cap 0.8) and were checkpointed at less than the full 300-update training budget available, so neither represents a converged endpoint; PD-L1 was the only training target, and neither checkpoint transfers this gain to the four held-out targets (Tab. 22), which we attribute to single-target training rather than to a limitation of sequence-only RL in general.

H.3 Structure-mode vs. sequence-only composition on PD-L1

To contextualize the composition effects discussed above, we compare the three PD-L1 sequence-only cohorts (Proteina Mixta, Proteina Mixta†, Proteina Mixta‡) against a structure-mode Mixta design set on the same target, using the same unique-CDR-sequence composition and per-position entropy measures as Sec. H.7 (Figs. 11 and 12).

Comparing structure-mode designs against the three sequence-only cohorts reveals a clear compositional split rather than a shared attractor. The sequence-only cohorts are consistently enriched for charged residues in CDR3 – the base model's top residue is arginine (28%, K+R$=0.31$), and even after RL the K+R fraction stays non-trivial (0.067–0.19) with net negative charge (D+E) driven

Table 22: Independent-oracle gate-pass rate (Eq. (55), all four legs jointly) out of 1,000 designs per method/target. PD-L1 is the only target used for RL training; the remaining four are held-out.

Method	PD-L1	Insulin	TrkA	BHRF1	IL-3
AbLang2 (Olsen et al., 2024)	0/1000 (0%)	1/1000 (0.1%)	0/1000 (0%)	0/1000 (0%)	0/1000 (0%)
ADAPT (Yeh et al., 2026) v2	1/1000 (0.1%)	0/1000 (0%)	0/1000 (0%)	0/1000 (0%)	0/1000 (0%)
ADAPT (Yeh et al., 2026) v5	1/1000 (0.1%)	0/1000 (0%)	0/1000 (0%)	0/1000 (0%)	0/1000 (0%)
Proteina Mixta	1/1000 (0.1%)	2/1000 (0.2%)	0/1000 (0%)	0/1000 (0%)	0/1000 (0%)
Proteina Mixta†	**830/1000 (83.0%)**	0/1000 (0%)	0/1000 (0%)	0/1000 (0%)	0/1000 (0%)
Proteina Mixta‡	**422/1000 (42.2%)**	0/1000 (0%)	0/1000 (0%)	0/1000 (0%)	0/1000 (0%)

Table 23: CDR3 charge composition (out of 1,000 designs) for the three PD-L1 sequence-only cohorts and the structure-mode design set.

Cohort	CDR3 top residue	CDR3 K+R	CDR3 D+E
Proteina Mixta (sequence-only)	R (28.0%)	0.311	0.160
Proteina Mixta (sequence-only)†	L (38.9%)	0.067	0.008
Proteina Mixta (sequence-only)‡	Y (13.1%)	0.189	0.075
Proteina Mixta (default all-atom)	G (16.2%)	0.065	0.051

toward zero, particularly for Mixta†. The structure-mode set looks categorically different: no CDR is charge-dominated, and every CDR's top residues are small, flexible or turn-forming (G/A/P/S for CDR1 and CDR3, K/G/S/I for CDR2), consistent with a model shaping the loop's backbone geometry rather than relying on electrostatics to score a confident interface. The two modes also fail in different places: CDR2 is the most diverse region for every sequence-only cohort (991–1000 unique CDR2s of 1000, mean entropy 2.36–3.22 bits) but collapses hardest for structure mode (220 unique CDR2s, mean entropy 1.39 bits – roughly half the sequence-only level), while CDR3 – the primary hotspot-contacting loop, and the region sequence-only RL collapses most (Mixta†: 615 unique, 0.696 bits) – stays comparatively diverse under structure mode (889 unique, 2.704 bits). In other words, sequence-only generation and structure-mode generation appear to hit different, non-overlapping failure/success modes: sequence-only optimization is vulnerable to charge-based reward hacking concentrated in CDR3, while structure-mode generation instead shows a composition and diversity bottleneck localized to CDR2, plausibly reflecting constraints imposed by backbone geometry that sequence-only generation never has to satisfy.

H.4 Configuration ablations and a yield–diversity trade-off

Proteina Mixta† (O12, EMA200) and Proteina Mixta‡ (O20, EMA250) ablate two candidate regularizers against composition collapse in the reward: a redundancy (group-collapse) penalty and a composition-reward term, summarized in Tab. 24. Comparing them surfaces a yield–diversity trade-off worth flagging for future guard design (Tab. 25): Proteina Mixta†'s higher pass rate concentrates on 24 CDR3 families under a greedy $\geq 80\%$-identity clustering, while Proteina Mixta‡'s lower pass rate is spread over 421 families, essentially one per pass. Because the two checkpoints differ in exactly these two hyperparameters – and were additionally evaluated on different hardware (Tab. 24) – we treat this as a single, informative ablation rather than a controlled study, and do not draw a stronger conclusion from it than that the redundancy penalty measurably affects solution diversity at a given training budget.

H.5 Guardrails and future work

A structure-confidence reward of the form in Eq. (54) is, in principle, satisfiable by amino-acid composition alone rather than by genuine paratope–epitope complementarity: a strongly charged loop can fold into a confident-looking interface against an oppositely charged surface under a structure predictor. A separate, controlled diagnostic sweep of early-stage, unguarded online sequence-only RL found exactly this failure mode – rising CDR net charge accompanying rising oracle confidence –

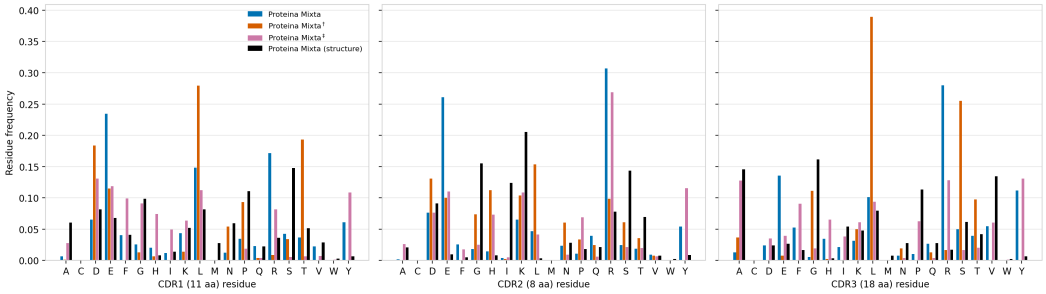


Figure 11: PD-L1 CDR1/CDR2/CDR3 amino-acid composition, computed over unique CDR sequences, for the three sequence-only Mixta cohorts and the structure-mode Mixta design set.

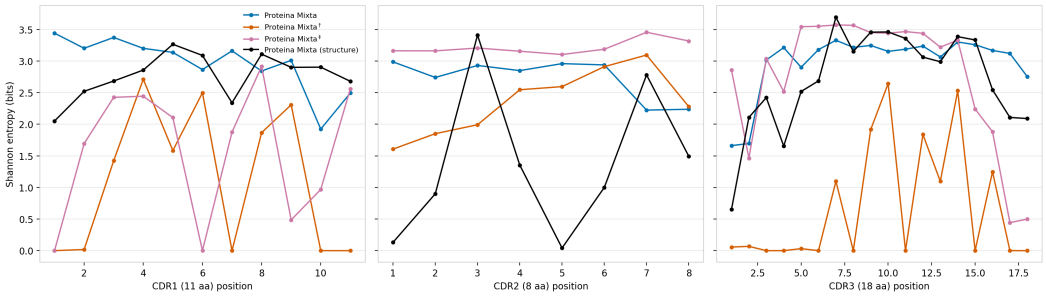


Figure 12: PD-L1 per-position Shannon entropy (bits) for CDR1/CDR2/CDR3, computed over unique CDR sequences, for the three sequence-only Mixta cohorts and the structure-mode Mixta design set.

motivating the two hyperparameters ablated in Sec. H.4 as regularizers against it:

$$c(x) = \frac{\big|\,|\{i : x_i \in \{K, R\}\}| - |\{i : x_i \in \{D, E\}\}|\,\big|}{L}, \tag{56}$$

$$m(x) = \max_{w \,\in\, \text{windows}_8(x)} \; \max_{a \,\in\, \mathcal{A}} f_a(w), \tag{57}$$

$$d(x, \mathcal{G}) = \frac{1}{|\mathcal{G}| - 1} \sum_{y \in \mathcal{G} \setminus \{x\}} \frac{1}{|V(\mathcal{G})|} \sum_{p \in V(\mathcal{G})} \mathbb{1}[x_p = y_p], \tag{58}$$

where Eq. (56)–equation 57 penalize net charge density and local compositional degeneracy in 8-residue windows (thresholds calibrated to be zero-penalty on 8,000 natural VHH sequences), and Eq. (58) penalizes similarity within a rollout group $\mathcal{G}$ over its within-group varying positions $V(\mathcal{G})$. The campaign checkpoints in Sec. H.2 already ablate Eq. (56)–equation 57 (the composition-reward weight) and a redundancy penalty consistent with Eq. (58); the family-diversity gap in Tab. 25 is consistent with, though not sufficient on its own to confirm, these regularizers trading off yield against solution diversity as intended. We leave a full mechanistic validation of this trade-off, and of whether it reflects composition-driven reward hacking specifically, to future work, together with two natural extensions: training beyond PD-L1 to the other four targets, and running these guards to convergence rather than at the partial training budgets used here.

H.6 Limitations

- **Checkpoint selection was exploratory.** The campaign checkpoints evaluated here (EMA steps 200 and 250 of a 300-step budget) were manually selected rather than the output of an independent sweep.
- **Training reward and evaluation oracle use different backends** (ESMFold2-Fast vs. the independent Protenix-based refold), by design. All pass rates in this appendix use only the independent oracle.

Table 24: Configuration differences between the two campaign checkpoints, both ablations of a shared anchor configuration.

Setting	Proteina Mixta† (O12, EMA200)	Proteina Mixta‡ (O20, EMA250)
Redundancy penalty weight	0 (disabled)	1 (enabled, threshold 0.18)
Composition-reward weight	1 (unchanged)	0 (disabled)
Reference/KL weight	0.02	0.1
Saved checkpoint	EMA step 200 / 300	EMA step 250 / 300

Table 25: PD-L1 pass rate alongside CDR3 family counts among passing designs under a greedy $\geq 80\%$-identity clustering. Family count is a diversity diagnostic, not part of the gate.

Method	PD-L1 gate-pass rate	CDR3 families among passes
Proteina Mixta	1/1000 (0.1%)	1
Proteina Mixta†	**830/1000 (83.0%)**	24
Proteina Mixta‡	422/1000 (42.2%)	**421**

- **No cross-target transfer.** Both campaign checkpoints were trained on PD-L1 only and show zero passes on all four held-out targets (Tab. 22); extending training to the other targets is future work, not a result reported here.

H.7 Confidence, composition and diversity by target

Figs. 13 to 17 show all three views – confidence-score distributions, CDR amino-acid composition and per-position diversity – for all six methods on each of the five evaluation targets. On PD-L1 (Fig. 13), the confidence distributions visibly shift for the two RL checkpoints toward the gate thresholds relative to every sequence-only baseline, consistent with Tab. 22; Proteina Mixta†'s reduced CDR3 positional entropy is consistent with, though not sufficient on its own to prove, the composition-collapse mechanism discussed in Sec. H.5. On the four held-out targets (Figs. 14 to 17), every method, including both RL checkpoints, fails Eq. (55) on every design, consistent with PD-L1-only training.



Figure 13: PD-L1: confidence-score distributions (top), CDR amino-acid composition (middle) and per-position diversity (bottom) for all six methods, computed over unique CDR sequences.



Figure 14: Insulin: confidence-score distributions (top), CDR amino-acid composition (middle) and per-position diversity (bottom) for all six methods.



Figure 15: TrkA: confidence-score distributions (top), CDR amino-acid composition (middle) and per-position diversity (bottom) for all six methods.



Figure 16: BHRF1: confidence-score distributions (top), CDR amino-acid composition (middle) and per-position diversity (bottom) for all six methods.



Figure 17: IL-3: confidence-score distributions (top), CDR amino-acid composition (middle) and per-position diversity (bottom) for all six methods.

I Baseline Implementations

We use the publicly released implementations described below, with the shared antigen crops, hotspots, hNbBCII10 framework, and CDR lengths $11/8/18$. We evaluate 1,000 designs per method and target through the same Protenix-v2 pipeline. Sequence-redesign stages are retained where specified; all reported successes must satisfy our common evaluation criteria, irrespective of a method's internal filters.

RFantibody. We use the code and model weights from the RFantibody repository (Bennett et al., 2026). Its antibody-adapted RFdiffusion model generates CDR backbones conditioned on the framework and antigen. We then use ProteinMPNN to design the CDR sequences, preserving framework residues. The resulting sequences are evaluated with Protenix-v2 rather than using RFantibody's native RoseTTAFold2 filtering results as the benchmark endpoint.

BoltzGen. We use the implementation from the BoltzGen repository (Stark et al., 2025). We retain the supplied framework and design the three CDRs under the shared length constraints. The main comparison uses Boltz-IF sequence redesign following generation. Direct design-stage outputs and inverse-folded outputs are treated as separate populations; the latter are evaluated using their redesigned sequences. Although the exported inverse-folding pipeline includes Boltz-2 refolding, our reported success metrics come from the common Protenix-v2 evaluation.

Germinal. We use the implementation from the Germinal repository (Mille-Fragoso et al., 2026), including its AbMPNN inverse-folding stage. For each attempted trajectory, we retain its best avail-

able output, apply AbMPNN sequence redesign, and evaluate the top-ranked redesigned candidate. We include all 1,000 attempted trajectories, including those stopped by Germinal's internal stage-wise filters. Approximately one-third of trajectories fail these internal filters in our runs; however, AbMPNN redesign can yield candidates from this subset that subsequently pass our Protenix-v2 criteria. We therefore do not assign failure solely from Germinal's internal acceptance decision or restrict evaluation to its internally accepted outputs. Each trajectory contributes at most one candidate, and the denominator remains 1,000. This procedure measures success after sequence redesign under our common evaluator.

BindCraft2. We use the implementation from the BindCraft2 repository (Pacesa Lab, 2026) in the VHH design setting. BindCraft2 optimizes sequences through AlphaFold2 and subsequently performs ProteinMPNN redesign. We evaluate the final redesigned sequences with Protenix-v2 under the same framework-recovery and structural-confidence criteria as the other methods. BindCraft2 and Germinal are classified as oracle-based because they query structure predictors during design; RFantibody and BoltzGen are classified as oracle-free generators, with downstream refolding used for evaluation.

ESMFold2. ESMFold2 has no released binder-design entry point, so we built one from the design tutorial in the ESM repository (`cookbook/tutorials/binder_design.py`), which implements Algorithm 11, "Gradient-Guided Binder Sequence Optimization," of Candido et al. (2026). The method is hallucination rather than generation: a soft binder sequence is optimized by gradient descent on per-position amino-acid logits, backpropagating through a differentiable ESMFold2 forward pass of the binder–antigen complex under distogram losses for intra-chain contact, inter-chain contact and globularity, regularized by ESMC pseudo-perplexity to keep designs inside the language model's sequence distribution. We keep the algorithm and its losses unchanged and adapt only the prompt: framework positions are pinned to the hNbBCII10 scaffold and receive no gradient, so only the three CDRs are designed, at the shared lengths $11/8/18$. Each design is an independent optimization from its own seed, and we evaluate $1{,}000$ designs per target with Protenix-v2 under the common success criteria rather than the tutorial's internal gates.

ADAPT. We include ADAPT as a sequence-based antibody design baseline. We asked the authors to generate samples using our specified hNbBCII10 framework and CDR lengths $11/8/18$. The ADAPT authors performed sample generation; we independently conducted all downstream analyses, including structure prediction, application of the common success criteria, and comparison with Proteina Mixta in sequence-only generation mode. We note that rather than the first 1000 sequences like the other methods we received the first 1000 unique sequences. We note that even with this the results for all sequence-only baselines are the same.,

Framework recovery and success accounting. Baseline outputs are checked against the supplied framework after sequence redesign. Our strict endpoint requires zero framework substitutions (all methods but Germinal guarentee this by design). Successful sequences are deduplicated within each method–target population, and success rates use the full sampling budget. Thus, internal filtering, inverse folding, and common downstream evaluation remain distinct stages of the comparison.

La-Proteina. For monomer co-design, we use the public La-Proteina implementation (Geffner et al., 2026) to train a baseline on the same data as Proteina Mixta. We generate 100 samples at each length in $\{50, 100, 150, 200, 250\}$ and apply the same evaluation protocol to both models. Co-design metrics use each model's generated sequences; ProteinMPNN-based designability is evaluated separately.

EvoDiff. For the sequence-generation comparison, we use the public code and published weights from the EvoDiff repository (Alamdari et al., 2023). We generate 100 samples at each length in $\{50, 100, 150, 200, 250\}$ and apply the same evaluation protocol used for Proteina Mixta.

DPLM & DPLM-2. We use the public code and published weights from the DPLM repository for both DPLM (Wang et al., 2024) and DPLM-2 (Wang et al., 2025b). Each model is evaluated on the same budget of 100 samples at each of the five lengths, under the same protocol as Proteina Mixta.

AbLang2. We use the public AbLang2 implementation (Olsen et al., 2024) as an antigen-blind sequence-generation baseline. We encode the hNbBCII10 scaffold with its three CDR regions masked, then sequentially sample masked residues conditioned on previously sampled positions to generate 1,000 sequences, preserving the framework and CDR lengths $11/8/18$. Because generation does not condition on the antigen, we reuse the same sequence population for every tested target. Each sequence is evaluated with each antigen through the common structure-prediction pipeline and success criteria.

Reproducibility Statement

Models and training. The multi-regime autoencoder architecture, objectives, and the six encoder/decoder training configurations are given in Appendix A, with the proposed setup in Appendix A.1, the configuration grid in Table 3, and the alternative objectives used for ablation in Appendix A.2. Monomer flow-matching training is described in Appendix B.1. The nanobody generator architecture, training data, phase mixtures (Table 9), modality and conditioning dropout, and optimization settings are in Appendix E. The Proteina-Nanocuerpo baseline is documented separately in Appendix D. Framework-guided generation through the frozen decoder, including the guidance objectives and selected settings, is in Appendix C and Appendix C.2.

The evaluated checkpoint. All reported Proteina Mixta nanobody results use a single checkpoint. Its construction, parameter groups, connected-channel mask and seed, calibration of the residual coefficients, and stored numerical values are specified in Appendix E.3, with ablations over the construction in Table 11. Checkpoint selection used PD-L1, BHRF1, insulin, and TrkA only.

Inference. Sampling hyperparameters, per-target temperatures (Table 18), beam-search configurations (Table 19), the search reward (Equation 46), and the definition of the reported sample population (Equation 47) are in Appendix G. Sensitivity to shared versus target-specific inference settings is reported in Appendix G.2.

Evaluation. Shared target crops, hotspots, and the fixed framework are listed in Appendix F.1 (Tables 12 and 13). Structure-prediction metrics, the strict success gate, and the oracle validation on experimentally determined VHH complexes are defined in Appendix F.2 (Tables 14, 15, and 16). Developability profiling and the TNP thresholds are in Appendix F.3 and Table 17. Monomer evaluation follows Appendix B.2. Baseline implementations, sequence redesign stages, and framework-recovery accounting are in Appendix I.

Sequence-only reinforcement learning. The RL task, objective, reward, and the independent evaluation gate are specified in Appendix H.1, with configuration differences between the reported checkpoints in Table 24 and the full analysis in Appendix H.

Release. All code, data, scripts, model weights, and inference accelerations will be released open source after the review period. Our training and inference code builds directly on the publicly available Proteina-Complexa implementation, `https://github.com/NVIDIA-BioNeMo/Proteina-Complexa`.

J Limitations and Future Work

Our antibody evaluation focuses on framework-constrained nanobodies, whose single-domain architecture simplifies experimental validation. This scope does not establish performance on paired-chain antibodies, single-chain variable fragments (scFvs), or other antibody formats, which introduce additional constraints on chain pairing and domain organization. Extending Mixta to these formats remains future work.

We use a controlled benchmark with common targets, frameworks, CDR lengths, and sampling budgets to isolate differences between methods. These restrictions limit conclusions about broader target coverage, alternative frameworks, and variable-length CDR design. Moreover, structure-prediction confidence and TNP flags are computational screening criteria, not direct measurements of

binding or developability. Our strict criteria reject more than half of the known PDB VHH complexes in the assessed reference set, illustrating that computational failure need not imply absent binding. Conversely, passing these criteria does not establish experimental activity. Broader benchmarks, calibration against experimental outcomes, and prospective measurements of binding, specificity, and developability are needed to assess practical design quality.