Omniverso: partially latent flow matching for de novo antibody and nanobody design

Model, data, training and evaluation design, with the code of the base model

Danil Kotelnikov

12 October 2026

This paper specifies the design of the system and supplies the code of its base model. The Results, the Discussion and the Conclusions follow the first trained model. The Supplementary Information holds the mathematics, the data tables, the evaluation protocol, the code and the computational budget.

Abstract

Background. Antibodies and single-domain nanobodies bind a chosen epitope through a few hypervariable loops. Generative models for protein structure now propose binders on a computer. The published antibody generators differ in conditioning, in how they treat the framework, in training data and in evaluation, and their reported numbers do not come from one shared benchmark.

Objective. Omniverso is a system for the de novo design of VHH and Fv binders against a chosen antigen and epitope. This paper specifies the model, data, training and evaluation design of its base model and of the full system, and supplies the code of the base model, so that the work can be repeated from the text and the code package.

Methods. The base model is a partially latent flow-matching model configured for training from random initialisation on an unconditional task, and no released weights are loaded. An autoencoder with 255,873,676 parameters maps each residue to C-alpha coordinates and an 8-number latent. A denoiser with 153,102,091 parameters learns the velocity field of that state on independent coordinate and latent clocks. Both networks are configured for training on two A100 cards, and the number of updates follows from the measured update time. Trunk arithmetic is BF16, and coordinates, losses, optimiser state and averaged weights stay FP32. The full system adds antigen conditioning, an interface-prediction layer, steering, maturation, judging, a self-proteome specificity panel and fixed-size portfolio selection, specified as a design. Evaluation uses leakage-controlled splits with a temporal holdout and compares with other groups through their published numbers only. The panels are not matched.

Results.

Conclusions.

Keywords. antibody design; nanobody; flow matching; latent diffusion; specificity; in silico evaluation; reproducibility

Contents

1 Introduction 7

1.1 Antibodies, nanobodies and the design problem 7

1.2 Computational antibody design before generative models 10

1.2.1 Structure prediction 12

1.3 Generative models of protein structure 13

1.3.1 Why partially latent flow matching suits antibody-antigen design 16

1.4 Published antibody and nanobody generators 18

1.4.1 Protocols behind the reported rates 23

1.5 Evaluation, specificity and developability 25

1.5.1 Specificity, developability and immunogenicity 27

1.6 Gaps, aims and scope 29

2 Materials and Methods 30

2.1 System overview 30

2.2 State and geometry 33

2.2.1 Rigid motions and coordinate gauges 36

2.3 Immunoglobulin contract 38

2.3.1 Numbering, regions and the CDR3 prior 39

2.3.2 Scaffold profiles, cysteines and pairing 41

2.4 Data 42

2.4.1 Sources and licences 42

2.4.2 Curation, quality control and deduplication 44

2.4.3 Leakage axes, temporal holdout and split freezing 46

2.4.4 Shard format and sampling 48

2.5 Generative model 49

2.5.1 Probability paths and the training target 50

2.5.2 Block loss and independent block clocks 52

2.5.3 Autoencoder and decoder factorisation 54

2.5.4 Architecture and parameter counts 55

2.6 Training of the base model 58
2.7 Antigen conditioning and Interface-JEPA 62
2.7.1 Interface-JEPA 66
2.8 Guidance and sequential Monte Carlo 68
2.8.1 Sequential Monte Carlo 71
2.9 Maturation and editing 74
2.10 Judging and specificity 78
2.10.1 The signed panel and pair-specific losses 81
2.10.2 Thresholds and conformal risk control 83
2.11 Liability, immunogenicity and glycan channels 84
2.12 Decisions under ambiguity and portfolio selection 89
2.12.1 Fixed-size portfolio 91
2.13 Evaluation design 93
2.13.1 Reporting for the base model 94
2.13.2 Comparison with published numbers 95
2.13.3 Splits and statistical rules for the full system 96
2.13.4 Hypotheses, ablations and validation levels 98
2.14 Computing, software and availability 100
2.14.1 Scope, ethics and licence 101
3 Results 102
4 Discussion 102
5 Conclusions 102
Declarations 102
References 102

List of figures

Figure 1. One deposited complex sets the design problem, and the base model, with its training code implemented, sits between a campaign and a designed run. 31

Figure 2. Three nested residue sets are marked on one deposited nanobody. 34

Figure 3. Training regresses the velocity heads on a straight path with two block clocks, and a sampler walks one curve through the square of times. 50

Figure 4. The base model is an autoencoder of 12 plus 12 blocks and a denoiser of 14 blocks built from one sequence block. 55

Figure 5. Linked entries are grouped, cut at one date, and passed or failed by the upper bound of the failure rate. 93

List of tables

Table 1. Training scale stated by the papers behind the generative models in this section. ... 18
Table 2. Published antibody and nanobody generators, as the primary texts report them (Germinal: version 3). ... 20
Table 3. IMGT framework (FR) and complementarity-determining (CDR) regions used by the region masks. ... 40
Table 4. Sources of the base-model corpus. ... 43
Table 5. Leakage axes: clustering rule and threshold. ... 46
Table 6. Number of trainable parameters of the base-model networks. ... 57
Table 7. Training settings shared by both stages. ... 59
Table 8. Batch composition by length bucket. ... 60
Table 9. One distributed update of either stage, in order. ... 60
Table 10. Design values of Interface-JEPA: widths in numbers of channels, loss weights and variance floor dimensionless, vector counts in numbers of independent unit-variance Gaussian vectors, ceilings dimensionless. ... 68
Table 11. Tiers of the physics cascade. ... 81
Table 12. Reference bands of the Therapeutic Antibody Profiler for five metrics, derived from 242 clinical-stage therapeutics [15]. ... 88
Table 13. Contract of the three modules that act on a finished candidate. ... 89
Table 14. Panels of published works that the design cites: panel size (complexes, pairs or targets) and calendar date rule, as reported by the sources. ... 95
Table 15. Hypotheses H1 to H6 with their comparators and refuting conditions. ... 98

1 Introduction

1.1 Antibodies, nanobodies and the design problem

Antibodies are the largest class of biotherapeutics [1]. Each one binds its target through a few loops on its variable domains. A design that aims at a chosen patch of a chosen antigen has to produce those loops, the scaffold that holds them, and a molecule that a cell can make and a patient can tolerate.

Variable-domain architecture. A conventional antibody variable domain folds from antiparallel β-strands. They form four framework regions (FR1 to FR4) and carry three hypervariable loops, the complementarity-determining regions CDR1 to CDR3 [2]. A heavy-chain domain (VH) and a light-chain domain (VL) pair to form the variable fragment (Fv), so its binding site has six loops [3]. The loops are the main route to the antigen but not the only one. Loop definitions from Kabat, Chothia or the international ImMunoGeneTics information system (IMGT) may miss about 20 percent of the antigen-binding residues, and framework residues contact the antigen in many complexes [3]. In the IMGT analysis, 15 of 19 canonical loop shapes of CDR1 and CDR2 correspond to a single loop length. The same analysis finds no such correspondence for CDR3, whose length, sequence and flexibility vary too much [2]. A generator therefore cannot draw CDR3 from a short list of known shapes.

Camelids also make antibodies that consist of heavy chains only [4]. Their antigen-binding domain, the VHH, works without a partner chain [5]. A VHH has the fold of a VH with three CDRs and a mass of about 12 to 15 kDa [6, 7]. Its CDR3 averages 18 residues, against 12 in mouse and 14 in human antibodies. Some VHH add a disulfide between CDR3 and CDR1 (camels) or CDR2 (llamas) [6].

In a conventional antibody, one face of the heavy-chain domain meets VL. Four FR2 positions (IMGT 42, 49, 50 and 52) lie on that face and usually carry Val 42, Gly 49, Leu 50 and Trp 52 in VH. In heavy-chain antibodies, where the face is exposed, they carry smaller or more hydrophilic residues, typically Phe or Tyr 42, Glu 49, Arg 50 and Gly 52 [2, 6]. IMGT uses the residues at these four positions to tell a VHH from a VH. The word nanobody is also used for constructs with more than one variable domain [2]; this manuscript uses VHH for the single domain. A VHH paratope draws on the framework more than a conventional one. Gordon et al. analysed 345 single-domain complexes and report that IMGT positions 66, 52, 55 and 42 lay in the interaction-defined paratope in 50.4, 31.6, 27.2 and 24.1 percent of the structures [8].

Numbering the domain. Insertions and deletions make a raw sequence index unreliable for comparing loops of different length. The same index can fall in a framework strand in one antibody and at a loop tip in another. The IMGT scheme gives corresponding positions the same label in heavy, light and VHH domains [9]. It divides the domain into four framework regions and three CDRs [2].

FR1 spans positions 1 to 26 and holds Cys 23. CDR1 spans 27 to 38, with gaps at the apex when it is shorter than 12 positions. FR2 spans 39 to 55 and holds Trp 41 and the four VHH hallmark positions. CDR2 spans 56 to 65, with gaps at the apex when it is shorter than 10. FR3 spans 66 to 104 and holds a hydrophobic residue at 89 and Cys 104. CDR3 spans 105 to 117. FR4 spans 118 to 128, and Phe or Trp 118 starts the J-segment motif F/W-G-X-G. The Methods give the region limits as the masks use them (Section 2.3.1).

A CDR3 longer than 13 residues gains positions between 111 and 112, added in the order 112.1, 111.1, 112.2, 111.2 [2]. Along the chain these read 111, 111.1, 111.2, 112.2, 112.1, 112, so sorting the labels as decimal numbers would scramble the sequence. The tool ANARCI assigns IMGT, Kabat, Chothia, Enhanced Chothia or AHo numbers by aligning a sequence to hidden Markov models of germline domains [10].

Formats. A design can take one of three formats. The format fixes how many variable domains the design must produce, whether a partner chain constrains them, and what a later measurement refers to. A VHH is one heavy-chain variable domain, so the design supplies three CDRs on a framework suited to a single domain. An Fv is a non-covalent VH/VL pair. It needs six CDRs, two frameworks and a consistent pairing. A single-chain variable fragment (scFv) joins VH and VL in one chain with a linker, and adds the linker length and the domain order to the design. The full system starts with VHH and adds Fv and scFv after their acceptance gates (Section 2.3).

Isolated VH or VL fragments are reported to be sticky and poorly soluble, with reduced affinity, although some keep their specificity. In an Fv, FR2 residues influence the VH/VL contact and so the relative orientation of the CDRs, which makes the pairing part of the design [3]. An scFv joins the domains with a flexible glycine- and serine-rich linker that has to span about 3.5 nm, and either domain can come first [11]. Antigen-binding fragments (Fab) and immunoglobulin G (IgG) add constant domains. In some systems these change the affinity or specificity of an unchanged variable domain [3]. A result for one construct needs justification before it is applied to another.

Epitope and paratope. The paratope is the binding site on the antibody. The epitope is the binding site on the antigen [3]. A complex structure supports a contact-based definition: an antigen residue belongs to the epitope when any of its heavy atoms lies within a cut-off distance of a heavy atom of the antibody. The set grows with the cut-off and depends on the conformation and assembly of the antigen. The design uses a cut-off of 4.5 Å.

An epitope is defined relative to the antibody that recognises it. The amino-acid composition of epitopes is nearly the same as that of other surface residues, and the epitope depends to a great extent on the antibody [3]. Registering an epitope states the surface patch a design should contact. It does not show that a candidate recognises that patch.

Contact is a weaker statement than energy. Binding free energy can concentrate in a few interface residues, the hot-spot hypothesis [3]. Competition between two antibodies does not prove a shared epitope, because antibodies on closely adjacent or minimally overlapping epitopes can displace one another, reversibly and in a dose-dependent way [12]. Antigens may also carry mobile glycans that shield potentially large fractions of the protein surface [13]. An epitope read from a bare protein structure can be partly covered on the native molecule.

What a binder design must satisfy. A binder proposed for a registered antigen and epitope has to meet five requirements together. The full system assesses them through separate computational checks.

- **Sequence acceptance.** Complete numbering, region assignments that fit the format, the two conserved cysteines (23 and 104) and no unpaired cysteine. These rules screen sequences. They do not show that the domain folds or forms a disulfide bond.
- **Target contact.** A predicted pose that contacts the registered epitope and is accessible on the native antigen. Affinity and kinetics need measurement.
- **Specificity.** Low or no binding to a named off-target panel under stated conditions. Polyspecificity is a recognised obstacle to developing antibodies found by in vitro display [14].
- **Developability.** Raybould et al. describe five computed risk metrics: total CDR length, surface hydrophobicity, positive and negative charge in the CDRs, and asymmetry of heavy- and light-chain charge [15]. Asparagine (Asn) deamidation and aspartate (Asp) isomerisation in CDRs depend on local flexibility and structure [16]. A new N-glycosylation sequon, the motif that permits attachment of a glycan, can alter antigen binding [17].

- **Immunogenicity risk.** A repertoire-based humanness score, whose association with clinical immunogenicity its authors reported for their data [18], and a predictor of antigen presentation by major histocompatibility complex (MHC) class II molecules [19]. Both are proxies.

The published developability thresholds derive from clinical-stage antibodies, and one metric compares heavy- and light-chain charge [15]. Single-domain formats have no light chain, so applying the guideline to them needs separate justification.

Why design is attempted and what a computation can show. Discovery by immunisation or by in vitro display selects binders from an animal repertoire or a library. Display allows defined selection conditions and returns the sequence at once [20]. Design can start from the target instead. The epitope is stated first, and liabilities are screened before any molecule is made.

Computed results differ in what they can support. Some are exact checks of a rule, such as a cysteine count, or algorithmic assignments, such as a numbering. Others are model outputs checked against held-out reference data, which needs trustworthy labels and a stated split, or predicted structures with a confidence score. None of these results shows that a designed molecule binds, separates its target from other proteins or can be expressed. Those statements need assays on the exact construct. Confidence scores are easy to over-read: independent benchmarks report that the interface predicted TM-score of structure predictors often fails to separate cognate from shuffled antibody-antigen pairs (Section 1.2.1).

1.2 Computational antibody design before generative models

Computational antibody design has long worked with two unequal data sets. The Observed Antibody Space database holds more than one billion antibody sequences from more than 80 studies [21]. The structural database SAbDab2 holds 22,550 antibody structures from 11,664 PDB entries [22, 23]. Both counts are from October 2026. The ratio is more than 40,000 sequences for each structure. The method families below use these two data sets in different ways. For each one we ask what it gives a designer of binders to a chosen epitope, and where it stops.

Physics-based docking and Rosetta design. Docking places a modelled antibody on an antigen and ranks the poses by an energy or a statistical potential. The docking program PIPER, behind the ClusPro server, was much less accurate for antibody-antigen pairs than for other complexes, and an asymmetric potential derived from antibody interfaces improved it [24]. Ambrosetti et al. tested four docking suites on 16 antibody-antigen complexes. HADDOCK,

which takes restraints from the CDRs and from the epitope when known, gave the best success rate and model quality, with and without epitope information [25]. SnugDock lets the six CDR loops and the VH/VL orientation move during docking. On 15 test complexes it gave four medium and seven acceptable lowest-interface-energy predictions on the CAPRI rating [26].

Design adds sequence change to the search. Lippow and colleagues designed single mutations from electrostatic terms and combined them. They report a tenfold gain for cetuximab, to 52 pM, and a 140-fold gain for the model antibody D44.1, to 30 pM [27]. RosettaAntibodyDesign (RAbD) grafts CDRs from canonical structural clusters, designs their sequences from cluster profiles and samples backbones with a flexible-backbone protocol. It starts from an experimental or modelled antibody-antigen structure. On 60 native complexes its design risk ratios for the CDRs other than H3 lay between 2.4 and 4.0, and replacing single CDRs raised the affinity of two test complexes by 10 to 50 fold. The authors name sampling and design of the H3 loop as a major challenge for true de novo design [28].

These methods start from an antibody that is already known to bind, or from restraints that place the epitope. They locate or refine a binder. Proposing a binder from scratch for an epitope chosen in advance lies outside what these studies report.

Antibody language models. Antibody language models learn the distribution of natural sequences. AntiBERTy and IgLM were each trained on 558 million sequences [29, 30], and AbLang on sequences from the Observed Antibody Space database [31]. IgLM re-designs a span of the sequence, such as one CDR loop, using the residues on both sides. Its infilled CDR libraries had improved in silico developability profiles [30]. Sapiens, a language model trained on the same database, was benchmarked on 177 antibodies and gave humanisation results that the authors rate as comparable to those of human experts [18]. General protein language models also help: with no antigen or structure input, they suggested mutations that raised the affinity of four mature antibodies by up to sevenfold and of three unmatured antibodies by up to 160-fold, with 20 or fewer variants screened per antibody [32].

The limit follows from the input. A sequence model has no antigen, so it cannot choose an epitope. Its likelihood scores resemblance to natural sequences, and any link to binding has to be measured. A large part of each variable domain stays germline, which pulls models towards germline residues, yet mutations away from germline are often vital for specific binding. AbLang-2 was built to reduce this bias [33].

1.2.1 Structure prediction

Two kinds of predictor matter. Antibody-specific models predict the Fv from sequence. DeepAb [34], IgFold [35] and ABodyBuilder2 [36] are examples. IgFold predicts a structure in under 25 s and was run on 1.4 million paired sequences [35]. On 34 recently solved antibodies, the mean CDR-H3 RMSD was 2.81 Å for ABodyBuilder2, 2.90 Å for AlphaFold-Multimer and 3.28 to 3.46 Å for four other tools. CDR H3 was the least accurate loop for every tool in that comparison [36].

General complex predictors read the antibody and antigen sequences together. AlphaFold 2 reached accuracy competitive with experiment for most CASP14 targets [37], and AlphaFold-Multimer extended it to complexes [38]. An independent benchmark found a success rate of 11% for AlphaFold-Multimer on recently released antibody-antigen structures [39]. AlphaFold 3 reports substantially higher antibody-antigen accuracy than AlphaFold-Multimer v2.3 [40]. Boltz-1 and Chai-1 are models of the same class with released weights, and Boltz-2 followed [41–43]. In an independent test with one seed, AlphaFold 3 reached high-accuracy docking for 10.2% of antibodies and 13.3% of nanobodies, and Boltz-1 and Chai-1 for 5% or less [44].

Two further findings matter for design, and the section on evaluation gives the numbers (Section 1.5). First, a correct model is often among the samples but not at the top [45]. Second, confidence does not follow binding. The interface predicted TM-score (ipTM) of current predictors ranks cognate nanobody-antigen pairs only weakly above shuffled non-cognate pairs [46]. A design pipeline that filters candidates by these scores inherits both weaknesses: the best model is hard to pick, and cognate pairs are hard to separate from non-cognate ones.

Inverse folding. Inverse-folding models predict a sequence for a given backbone. ProteinMPNN recovered 52.4% of native residues on native backbones, against 32.9% for Rosetta, and rescued designs that had failed with Rosetta or AlphaFold [47]. ESM-IF1 recovered 51% on structurally held-out backbones after its training data were extended with AlphaFold2 predictions for 12 million sequences [48]. AntiFold fine-tunes ESM-IF1 on solved and predicted antibody structures. It outperforms existing tools on sequence recovery across the CDRs, and in zero-shot use its predictions correlate more strongly with antibody-antigen affinity [49]. An inverse-folding model needs a backbone, so another method must supply the backbone and its placement on the epitope. Recovery of native residues scores agreement with the natural sequence of a solved structure. Binding to a new antigen is a separate question.

Chained pipelines. The families can be chained. One published pipeline combined Ig-VAE scaffolds, ProteinMPNN and RFantibody for sequence design, tFold-Ab and tFold-Ag for structure prediction, four docking programmes, molecular dynamics and humanisation tools [50]. It was semi-automated and set cut-offs on ipTM, predicted aligned error (pAE) and RMSD. It reports computed binding free energies for three humanised nanobody candidates and no assay, and its authors name experimental validation as the next step.

Each family answers part of the design question. Docking and Rosetta design work around a binder that exists. Language models know no antigen. Structure predictors model complexes, and their scores are weak judges of binding. Inverse folding needs a backbone. The studies cited here use different antibodies, antigens, cut-off dates and success definitions, so their numbers cannot be ranked against one another. Several come from preprints without peer review. Generative structure models, covered next, attempt to combine these pieces in one model.

1.3 Generative models of protein structure

A generative model of protein structure learns the distribution of known structures and draws new samples from it. Two training recipes dominate the field: denoising diffusion and flow matching. Both turn random noise into a structure through many small steps, and they differ in what the network is asked to predict. The families below lead to La-Proteina and Proteina-Complexa, the two published models on which Omniverso builds. These papers supply architecture, dimensions and a reference for training scale.

Diffusion and score models. A diffusion model adds Gaussian noise to a training example in many small steps and trains a network to undo them [51]. In the continuous-time view, the network learns the score, the direction in which the density of noisy data rises fastest. Sampling follows a reverse-time stochastic differential equation, or an equivalent ordinary one, from noise back to data [52]. RFdiffusion brought this recipe to protein backbones. It fine-tuned the structure predictor RoseTTAFold to denoise residue frames and left sequence design to a second network, ProteinMPNN. Its authors tested binders to five targets in the laboratory [53]. Chroma used a diffusion process matched to polymer statistics and samples structure and sequence together; its authors tested 310 designed proteins experimentally [54]. AlphaFold 3 places a diffusion-based architecture inside a predictor of complexes [40].

Flow matching. Flow matching trains the same kind of network on a simpler target. Draw a noise sample $\bar{q}_0$ and a data sample $\bar{q}_1$, join them by a straight line, and ask the network to predict the velocity along the line. Here $\bar{q}$ is the state of a design made dimensionless as the

Methods describe, and $u \in [0,1]$ is the position on the line. The network output is v_θ, and the expectation $\mathbb{E}$ runs over pairs and positions:

$$\bar{q}_u = (1-u)\,\bar{q}_0 + u\,\bar{q}_1, \qquad \mathcal{L} = \mathbb{E} \parallel v_\theta(\bar{q}_u, u) - (\bar{q}_1 - \bar{q}_0) \parallel_2^2.$$

The loss is the squared distance between the network output and the velocity of the line, which is $\bar{q}_1 - \bar{q}_0$. Averaged over many pairs, the network learns the field that carries the noise distribution onto the data distribution. Generation integrates the learned field from $u = 0$ to $u = 1$ with an ordinary differential equation solver [55–57].

Gaussian paths include the diffusion paths as special cases [55]. For a straight path from independent Gaussian noise, the score can be read off the velocity v, where p_u is the density of states at position u:

$$\nabla_{\bar{q}} \log p_u(\bar{q}) = \frac{u\,v(\bar{q}, u) - \bar{q}}{1-u}.$$

A trained flow can therefore also run as a stochastic sampler [58, 59]. Pairing noise and data samples by minibatch optimal transport shortens the lines [60]. Short lines allow few solver steps. FrameFlow used five times fewer steps than the diffusion model it was adapted from [61].

Models with explicit rotations. A backbone can be described by one rigid frame per residue: a position and a rotation. FrameDiff diffuses these frames on SE(3), the group of rotations and translations of three-dimensional space, so that the model does not depend on how a protein is oriented [62]. Riemannian flow matching carries the straight-line recipe over to curved spaces such as the rotation group [63]. FrameFlow and FoldFlow apply it to frames [61, 64]. MultiFlow adds a discrete flow over the twenty amino-acid types, a continuous-time Markov chain, and generates sequence and backbone together; it leaves side chains out [65]. FoldFlow-2 conditions on sequence through a frozen protein language model and reports a scaffolding task around the CDR loops of VHH nanobodies [66].

Proteina takes the opposite route. It models C-alpha positions only and uses a transformer with no built-in rotation symmetry. Symmetry is taught by centring each protein and rotating it at random during training. The network keeps a table of features for every residue pair, the pair representation, which biases attention. Triangle layers that update this table are optional. Proteina trains models of 200 M and 400 M parameters on up to 20,874,485 predicted structures [58].

Latent models. A latent model runs the generative process in the compressed space of an autoencoder, as latent diffusion does for images [67]. For all-atom proteins the compression answers two difficulties. The number of atoms per residue depends on the residue type, so an atom array that follows the sequence has no fixed shape. The sequence is also categorical, while the flow above acts on continuous variables. A fixed-width continuous latent per residue removes both [68]. PLAID runs diffusion in a shared sequence-structure latent space taken from a pretrained predictor. Its diffusion model is trained on sequences only, so it can use far more data than structure databases hold [69]. LSD generates a coarse latent that decodes to a contact map, then diffuses coordinates conditioned on that map [70]. La-Proteina differs from both: it moves an explicit C-alpha trace and a residue latent together in one flow.

La-Proteina. La-Proteina keeps the C-alpha coordinates $x \in \mathbb{R}^{n_e \times 3}$ explicit and gives each residue a latent $z \in \mathbb{R}^{n_e \times 8}$ of eight continuous numbers, for n_e transported residues. Together they form 11 numbers per residue. A variational autoencoder learns the latent. Its encoder reads the sequence and all atoms. Its decoder reads x and z and returns residue-type probabilities and Atom37 coordinates, the 37 atom slots per residue, of which the decoded residue type selects the valid ones [68].

A denoiser then learns a flow over (x, z) with the autoencoder frozen. Each block has its own interpolation time, t_x and t_z, drawn independently, and its own velocity head. Both heads see both times. Subscript 0 marks noise and subscript 1 data:

$$x_{t_x} = (1 - t_x)\, x_0 + t_x\, x_1, \qquad z_{t_z} = (1 - t_z)\, z_0 + t_z\, z_1,$$

$$\mathcal{L} = \mathbb{E}[\| v_\theta^x - (x_1 - x_0) \|_2^2 + \| v_\theta^z - (z_1 - z_0) \|_2^2].$$

Each block follows its own straight line, and the loss adds the two squared errors. The Methods divide each term by the size of its block. Two clocks let sampling move the backbone ahead of the latent. In the authors' ablations on monomers, sampling the backbone faster than the latent was needed for good all-atom co-designability, and a latent that also encoded the C-alpha trace gave worse results.

The paper gives the encoder and decoder 12 transformer blocks and about 130 M parameters each, and the denoiser 14 blocks and about 160 M. The main models use no triangle layers. The training data are cluster representatives of the AlphaFold Database, about 350 thousand to 550 thousand structures of up to 512 residues. A long-chain model draws on 46,942,694 structures grouped into 4,035,594 clusters. The paper reports unconditional monomers of up to 800 residues and atomistic motif scaffolding, and no antibody data.

Proteina-Complexa. Proteina-Complexa extends La-Proteina to binder design [71]. The target enters the denoiser as extra tokens: Atom37 coordinates, residue identity and a flag that marks the hotspot residues near which the binder should form. These tokens are joined to the binder tokens, and the pair features cover binder and target together. The autoencoder is unchanged and encodes the binder alone, so one autoencoder serves every target. Training also adds a random global translation to the binder, with a standard deviation of 0.2 nm, so that the model must learn where the binder sits relative to the target.

Training runs in stages. The autoencoder learns on predicted monomers and then on PDB chains. The denoiser learns first on predicted monomers and then on Teddymer, a set of synthetic dimers made by splitting multidomain AlphaFold Database proteins (10 M dimers in 3.5 M clusters), mixed with PDB multimers at a Teddymer to PDB ratio of 8 to 2. At generation the model samples with reduced noise and adds test-time search: best-of-N, beam search, Feynman-Kac steering and Monte Carlo tree search, scored by the confidence of a structure predictor. The paper evaluates protein targets, small-molecule targets and enzyme design by in silico criteria only and tests no binder in the laboratory. It names antibodies only as a possible modality for future work.

1.3.1 Why partially latent flow matching suits antibody-antigen design

This is an argument from the published constructions. Five properties of the construction match the problem.

First, an antibody-antigen interface is made of side chains. Which residue sits at a CDR position, and how its side chain points, decide the contacts. A backbone-only generator leaves sequence to a second network. The partially latent state yields sequence and all-atom coordinates from one flow [68].

Second, the state has a fixed size. Eight latent numbers plus the three C-alpha numbers give 11 numbers per residue whatever the sequence, and every number is continuous. The main generator needs no categorical flow.

Third, the C-alpha trace stays explicit. Proteina-Complexa shows how target tokens, hotspot flags and translation noise place a binder against a target, and an antigen can enter the denoiser the same way.

Fourth, the autoencoder does not depend on the antigen. In Proteina-Complexa only the denoiser sees the target, so one autoencoder serves every antigen.

Fifth, the networks avoid triangle layers. The all-atom baseline P(all-atom), which uses them, ran out of 140 GB of GPU memory beyond 500 residues [68]. An antibody-antigen complex spans several hundred residues, so memory sets a limit.

Neither La-Proteina nor Proteina-Complexa reports an antibody or nanobody result, and neither handles IMGT numbering, framework and CDR roles or heavy-light pairing. Omniverso therefore applies a published construction to a new domain. Antibody generators built on related constructions are surveyed in the next section.

Published training compute. The reports state hardware, optimiser steps and batch size but no floating-point operation count for training. **Table 1** lists what each paper states.

System and stage	Trainable parameters	Training set	Steps, GPUs, batch per GPU	Samples seen (millions)
FrameFlow [61]	17.4 M	3,938 SCOPe examples below 128 residues	steps n/r; 1 day on 2 A100 GPUs	n/r
MultiFlow [65]	n/r	18,684 PDB proteins of 60 to 384 residues	200 epochs; 3 days on 4 A6000	n/r
FoldFlow-2 [66]	21 M (frozen ESM2-650M excluded)	PDB plus filtered synthetic structures	500 k steps; 4 days on 2 A100-40GB (192 GPU-hours)	n/r
RFdiffusion, as listed in [66]	n/r	PDB	25 k steps; 64 + 8 GPUs; 28 + 3 days, with RoseTTAFold initialisation	n/r
Proteina M_FS, 5 triangle layers [58]	200 M	588,318 AFDB clusters, 32 to 256 residues	200 k × 128 × 4	102.4
Proteina M_FS, no triangle layers	200 M	same	360 k × 96 × 10	345.6
Proteina M_21M, 4 triangle layers	400 M	20,874,485 AFDB structures, up to 256 residues	180 k × 128 × 4	92.2
La-Proteina autoencoder, 3 stages [68]	about 130 M each, encoder and decoder	AFDB clusters, up to 256, 512, 896 residues	500 k × 16; 140 k × 32; 180 k × 32; batch n/r	n/r
La-Proteina denoiser, 3 models	about 160 M	AFDB clusters up to 512 residues; long-chain pool of 46,942,694	390 k × 48 (no triangle layers); 120 k × 96 (triangle layers); 140 k × 128 (long chains); batch n/r	n/r
Proteina-Complexa autoencoder, AFDB stage [71]	256 M (encoder plus decoder)	AFDB monomers, up to 256 residues	500 k × 16 × 14	112.0
Proteina-Complexa autoencoder, PDB stage	256 M	110,976 PDB chains	40 k × 16 × 12	7.7

System and stage	Trainable parameters	Training set	Steps, GPUs, batch per GPU	Samples seen (millions)
Proteina-Complexa denoiser, pretraining	159 M	AFDB monomers, up to 256 residues	540 k × 32 × 12	207.4
Proteina-Complexa denoiser, fine-tuning	159 M	Teddymer plus PDB multimers, 8 to 2	290 k × 96 × 6	167.0

Table 1. Training scale stated by the papers behind the generative models in this section.

Samples seen are steps × GPUs × batch per GPU, in millions, assuming one microbatch per update. The Proteina-Complexa and La-Proteina runs used A100-80GB cards; the card model of the Proteina runs is not stated. n/r, not reported in the passages read. AFDB, AlphaFold Database.

The SE(3) backbone models trained in days on two to four cards, FoldFlow-2 with a language model pretrained elsewhere. The denoiser of Proteina-Complexa trained in two stages on 32 and then 96 A100-80GB cards and saw 374.4 M samples in total. Test-time search adds inference budgets: Proteina-Complexa uses 16 and 32 GPU-hours per target in its scaling tests and up to 1,100 GPU-hours in one case study. The RFdiffusion row gives the figure listed by the FoldFlow-2 authors.

The base model of this work has 255,873,676 autoencoder parameters and 153,102,091 denoiser parameters (Section 2.5.4). Its budget rule is given in Section 2.6. A sample count measures data seen, and compute also depends on sample length, so the comparison marks scale only.

1.4 Published antibody and nanobody generators

Since 2022, several groups have published generators that propose antibodies or nanobodies on a computer, and most report laboratory tests. We read the primary paper, preprint or technical report of each system in **Table 2** in October 2026. The numbers below are those the authors report; none was re-measured. Where versions of a paper differ, the version is named. Several sources are preprints or company reports without peer review. Supplementary Figure S8 places the systems by family, by what each report describes and by the compute that is reported.

System	Year	Conditioning	Framework	Training data	Laboratory test	Compute
DiffAb	2022	Antigen and bound-antibody framework; one CDR at a time	Fixed, from the native complex	34,781 pseudo-CDR complexes added (v1); size otherwise n/r	None	One A100; hours n/r
Absci 2023	2023	HER2, epitope, trastuzumab framework and CDR lengths	Fixed	n/r	About 10^6 display variants; 421 SPR-confirmed binders; one target	n/r
AbDiffuser	2023	None; HER2 task on binder-labelled CDR-H3 mutants	Generated	n/r; 169 M parameters	16 variants of trastuzumab; 37.5% bound HER2	One A100 80 GB; hours n/r
IgDiff	2024	None, or all CDRs on a fixed framework	Generated	n/r	28 sequences expressed; no binding assay	144 A10G GPU-hours (fine-tuning)
IgGM	2024	Antigen, framework sequence, optional epitope	Sequence fixed	Structures to 2022-12-31	7 of 60 scFv bound PD-L1 (2025 preprint)	1,920 A100 GPU-hours (two stages)
RFantibody	2024	Target, hotspots, framework, CDR lengths	Fixed	Antibody complexes; size n/r	Six targets; binders to four epitopes; cryo-EM	n/r
Germinal	2025	Predicted antigen structure, epitope, framework	Fixed by sequence bias	No network trained	Four targets; BLI binders; one cryo-EM	200 to 500 H100 GPU-hours per 200 to 400 designs
mBER	2025	Target, one hotspot, partly masked VHH	Fixed (IGHV3-23 based)	No network trained	145 targets by pooled phage display	n/r
BoltzGen	2025	Target, scaffold, optional binding site	Fixed (four scaffolds)	PDB to 2023-06-01	Nine targets (v1); nanomolar binders for six	n/r
Chai-2	2025	Target, up to four epitope residues	Fixed (five VHH, five VH-VL)	Structures before 2021-09-30	52 targets; 15.5% of tested designs bound; five cryo-EM structures	n/r
JAM-2†	2025	Target; epitope in the second set	CDRs plus a few FR2 and FR3 residues	n/r	16 targets; mean success 39% (VHH-Fc), 18% (antibody)	n/r
Latent-X2	2025	Target, one to four hotspots, framework, CDR lengths	Fixed (five VHH, five Fv)	n/r	18 targets; validated binders for nine	n/r

System	Year	Conditioning	Framework	Training data	Laboratory test	Compute
Origin-1	2026	Antigen, epitope, framework, CDR lengths	Fixed (four therapeutic)	Structures before 2024-01-01	Ten targets; hits for four; cryo-EM	Search up to about 1,005 H200 GPU-hours per target (v2)
tFold system	2025	Antigen, epitope, framework	Fixed germline	Structures to 2021-12-31	Four targets; pooled phage libraries	n/r
GeoFlow-V3	2025	Antigen, up to 10 epitope residues	Fixed (h-NbBcII10 FGLA)	n/r	Five targets; up to 50 designs per campaign	n/r
GaluxDesign	2025	Target, two to six epitope residues	n/r	n/r	Six targets, eight epitopes; ELISA and BLI	n/r
UniMoMo	2025	Binding-site residues; optional framework	Fixed	9,473 complexes	None	n/r
AgForce	2026	Antigen, epitope, framework	Fixed	2,338 complexes (CHIMERA-Bench split)	None	About 2 H100 GPU-hours
Ophiuchus-Ab	2026	None	Fixed for infilling, generated for humanisation	2,511,612 OAS pairs	None	8 A100 GPUs, 100 k steps; hours n/r

Table 2. Published antibody and nanobody generators, as the primary texts report them (Germinal: version 3). Compute is training unless stated, in GPU-hours where cards and time are given. n/r, not reported. CDR, complementarity-determining region; FR, framework region; Fv, variable fragment; scFv, single-chain Fv; SPR, surface plasmon resonance; BLI, biolayer interferometry; OAS, Observed Antibody Space. †, technical report without a registry identifier.

CDRs on a given complex. The first generators work on a complex or scaffold that already exists. DiffAb diffuses amino-acid types, positions and orientations to redesign one CDR at a time, given the antigen and the framework of the antibody bound to it. Its final test set holds 19 complexes from five CDR-H3 clusters, and it reports no laboratory test [72]. AbDiffuser and IgDiff generate the paired variable region, framework included, and take no antigen as input [73, 74]. Both tested few variants. For AbDiffuser, 37.5% of 16 submitted variants of trastuzumab bound HER2. For IgDiff, 28 sequences were expressed and no binding was measured. Absci's zero-shot study redesigned the heavy-chain CDRs of trastuzumab against HER2 with the epitope and framework given. The top 1,000 designs bound at 10.6% for CDR-H3 alone and 1.8% for all three heavy CDRs (version 4) [75].

IgGM generates CDR sequences and the structure of the whole variable region from the antigen structure, the framework sequence and an optional epitope [76]. Its benchmark uses 60

antibodies and 27 nanobodies released in the second half of 2023, after removal of sequences similar to the training set. A design counts as a success when DockQ exceeds 0.23, and the success rates are 0.433 for antibodies and 0.415 for nanobodies. A 2025 preprint adds one laboratory test: seven of 60 scFv designs against the PD-1 site of PD-L1 bound [77].

Epitope-conditioned design with public models. RFantibody fine-tunes RFdiffusion on antibody complexes. The user supplies the target structure, hotspot residues, a framework and the CDR lengths. ProteinMPNN designs the CDR sequences, and a fine-tuned RoseTTAFold2 filters the designs [78, 79]. Six targets were designed and binders to four epitopes were confirmed. Yeast display screened about 9,000 designs per target, and single-concentration SPR tested 95 per target. Success rates were 0 to 2% per target and the best affinities ran from 78 nM to 5.5 μM. A cryo-EM structure of one influenza haemagglutinin VHH matched its design at 3.0 Å. The authors classify the structure of one RBD VHH as a design failure. Later versions add scFvs, and the working scFvs came from heavy and light chains recombined across designs.

Germinal and mBER train no network. They optimise CDR sequences on a given framework by gradient descent through AlphaFold-Multimer. Germinal adds an antibody language-model term and a sequence-redesign step [80, 81]. It tested 43 to 101 nanobodies for each of four targets chosen by the authors. In version 3, BLI confirmed 7 of 25 PD-L1 designs taken to BLI, 2 of 11 for IL3, 4 of 11 for IL20 and 5 of 20 for BHRF1.

mBER designed 1,153,241 VHH sequences against 436 human cell-surface proteins and measured binding on 145 targets by pooled phage display [82]. On-design hit rates were significantly raised for 65 of the 145 targets (45%), and the median hit rate per design was about 0.4% among the 78 targets with any on-design binding. The paper reports no affinity and no structure.

BoltzGen is a binder-design model built on the Boltz-2 architecture. For nanobodies it fixes one of four default scaffolds and replaces the three CDRs with loops of random length [83]. In version 1, up to 15 nanobodies were tested for each of nine targets that the authors call novel, and nanomolar binders were found for six of the nine. Version 2 separates a screening hit from a confirmed binder and adds a tenth target.

Company reports. Chai Discovery, Nabla Bio, Latent Labs and Absci report antibody design. The first three offer access only through partnerships or a platform. Chai-2 designs VHH and scFv from a target structure and up to four epitope residues, and designs all CDR residues and their lengths on a scaffold chosen by the user [84]. On 52 targets, with up to 20 designs tested

per target and format, 15.5% of tested designs bound by BLI (20.0% VHH, 13.7% scFv). At least one binder was found for 26 targets. A second report reformats scFv hits as 88 IgGs, compares their developability with clinical antibodies, and shows five Fab-antigen cryo-EM structures at 2.9 to 3.9 Å in version 2 [85].

JAM-2† is a technical report of Nabla Bio, a PDF on the company website (`nabla-public.s3.us-east-1.amazonaws.com/2025_Nabla_JAM2.pdf`). It has no DOI, arXiv number or PubMed entry, is named here by its location and is not in the reference list. It tests VHH-Fc and full-length antibodies. On 16 targets with 45 designs per format, the report gives a mean per-target success of 39% for VHH-Fc and 18% for antibody. A second set of 10 targets with 20 chosen epitopes each was screened by yeast display, with about 600 VHH designs per epitope. For the GPCRs CXCR4 and CXCR7, 86 of 736 and 28 of 728 designs bound. The first JAM paper tested pooled libraries of about 10,000 designs per target, with hit rates of 0.03 to 0.19% [86].

Latent-X2 designs VHH and scFv from the target structure, one to four hotspot residues and one of five VHH or five Fv frameworks [87]. On 18 targets, 4 to 24 designs per modality and target gave validated binders for nine. The preprint ties "validated" to a reproducible five-point BLI or SPR fit. It also reports immunogenicity assays on one target, with four VHH and ten donors.

Origin-1 combines an all-atom diffusion model fine-tuned from Boltz-1, a sequence-design model and a co-folding score. It designs six CDRs on four therapeutic frameworks and tests full-length antibodies [88]. Its ten targets have no complex structure in the PDB and at most 60% identity to any protein that has one. With at most 95 antibodies tested per target, hits were found for four targets, one to four each. Cryo-EM at 3.0 to 3.1 Å matched two designs (version 1). The authors state that their hit rates are lower than others have recently reported.

Other groups report related systems. The tFold system of Wu and colleagues selected scFv from pooled phage libraries of 10,000 designs for each of four targets [89]. GeoFlow-V3 and GaluxDesign report five and six targets, with up to 50 designs per campaign or epitope [90, 91]. Ma and colleagues describe an adaptive deep-learning method for pocket-targeted CDR design [92]. TFDesign-sdAb adds Protein A binding to existing single-domain antibodies by mutating framework residues; it modifies a known antibody and proposes no binder from scratch [93]. UniMoMo, AgForce and Ophiuchus-Ab report computational benchmarks only, and the last has no antigen input [94–96].

Framework handling. The systems differ most in how they treat the framework. Of the 19 systems in **Table 2**, twelve keep it fixed outright, and two more, IgGM and Germinal, hold it by its sequence or by a bias. Several draw it from a short list: four scaffolds for BoltzGen and Origin-1, five VHH and five Fv frameworks for Latent-X2, and five VHH and five VH-VL frameworks for Chai-2. Germinal's bias still permits mutation. IgGM takes the framework sequence from the user and generates its structure. JAM-2† generates the CDRs and a few adjacent framework 2 and 3 residues that contact the antigen; the first JAM paper does not state how the framework is chosen. AbDiffuser and IgDiff generate the framework but take no antigen. Ophiuchus-Ab fixes the framework for infilling and generates it for humanisation, and GaluxDesign does not report framework handling.

The primary texts report no epitope-conditioned generation of the whole framework, and none compares fixed and generated frameworks at matched compute. A fixed framework lets the authors pick one with known properties; Chai-2 used VH3-23 and VH3-66 for their thermostability. Whether a fixed scaffold limits the reachable epitopes is not reported.

Validation and specificity. Binding is the usual readout, by BLI, SPR, yeast or phage display, or ELISA. Structures of designs are reported for RFantibody, Germinal (one scFv at 3.9 Å), Chai-2 (five complexes) and Origin-1 (two in version 1, three in version 2). No primary text reports a self-proteome panel, and the specificity tests are narrow. Chai-2 tested six antigens, and one of 23 designs bound above background. BoltzGen tested human serum albumin, JAM-2† ovalbumin, and Origin-1 two non-antigen proteins together with cross-design checks. The pooled mBER screen gives the widest view: among the 331,578 designs whose targets were screened, it found 2,207 on-design hits and 11,040 off-design hits. Chai-2, JAM-2†, Latent-X2 and Origin-1 report developability panels on expression yield, size-exclusion profile, hydrophobicity and polyspecificity. Compute is rarely stated, and GPU-hours do not transfer between card types; **Table 2** lists what each paper gives.

1.4.1 Protocols behind the reported rates

Six features of the study protocols determine the size of a reported rate.

- **Targets.** Each study chose its own, and overlap is rare. The examples span one-target studies and the pooled screen of mBER on 145 targets. Target counts must be read with the assay format and the candidate-selection procedure. JAM-2† tests 16 targets that overlap the Chai-2 study, in other formats. Epitopes were chosen by the authors, often from known complexes.

- **Novelty and cut-off.** Chai-2 excludes antigens above 70% identity over 80% coverage to antibody antigens released before 30 September 2021. BoltzGen excludes proteins above 30% identity in any bound PDB structure and trained on structures up to 1 June 2023. Origin-1 allows at most 60% identity to any protein with a complex and trained on structures before 1 January 2024. The tFold system trained on structures to 31 December 2021, and IgGM to 31 December 2022. The main texts of RFantibody, Germinal, mBER and Latent-X2 state no such criterion.
- **Designs tested.** Counts run from 4 to 24 per target and modality (Latent-X2) through 15 (BoltzGen) and up to 20 (Chai-2) to 43 to 101 (Germinal), and pooled libraries test 10,000 or more. A rate on individually tested designs and a rate on a pool are on different scales.
- **Selection before testing.** Tested designs are the best-ranked by structure-predictor confidence, so a rate depends on the filter and on the pool it drew from. The pass rate before filtering is rarely given.
- **Success definition.** Thresholds include a BLI response above 0.1 nm at one concentration (Chai-2 at 5 or 10 μM, JAM-2† at 500 nM), a reproducible five-point kinetic fit (Latent-X2) and an ELISA signal three times the negative control (tFold system). Origin-1 requires SPR above 10 RU at 2 μM with no off-target binding, and mBER uses pooled enrichment tested by Fisher's exact test. Formats also differ: monovalent VHH, dimeric VHH-Fc, scFv and IgG.
- **Level of the rate.** A rate per design, per epitope or per target answers different questions. At least one binder was found for 26 of 52 targets (Chai-2), 9 of 18 (Latent-X2), 6 of 9 (BoltzGen) and 4 of 10 (Origin-1).

Reported numbers also change between versions. Germinal's BHRF1 count fell from 11 of 20 in version 1 to 5 of 20 in version 3. BoltzGen redefined its hit between versions. The cryo-EM resolutions in the second Chai-2 report changed from 3.5 to 5.7 Å in version 1 to 2.9 to 3.9 Å in version 2. A rate is therefore quoted with its version.

Independent comparisons. The AIntibody challenge is a blinded, prospective comparison. It used a single antigen, the SARS-CoV-2 receptor-binding domain, asked for CDR redesign on fixed frameworks, and collected 511 sequences from 29 organisations [97]. Its authors report that design outside the library was highly variable and that many submissions did not outperform standard selections. The main text of that paper names none of the generators in **Table 2**. CHIMERA-Bench retrains eleven earlier CDR generators on shared epitope-group,

antigen-fold and temporal splits, and could not evaluate several methods under leakage control, RFantibody among them [98]. No shared benchmark connects the epitope-conditioned systems.

Weights and licences. Code and weights are released for DiffAb (Apache-2.0), IgGM, RFantibody, BoltzGen and mBER (MIT). Germinal's code is under Apache-2.0 but needs PyRosetta and IgLM, which carry non-commercial terms. The tFold code is under a non-commercial licence. Chai-2, JAM, JAM-2† and Latent-X2 are available only through partners or a platform. The Origin-1 repository holds designs and data; model weights were not found. The base model described here loads no released weights.

1.5 Evaluation, specificity and developability

A generated antibody is typically judged in three steps. A structure-prediction model scores the designed sequence against the antigen, property predictors screen the survivors, and a small set goes to the laboratory. The first step does most of the filtering, and the scoring model is a trained network with its own error rate. One benchmark paper describes the common workflow as generating thousands to millions of candidates and filtering them on internal confidence scores such as pLDDT, ipTM and PAE [46]. Such filters help: using AlphaFold2 or RoseTTAFold to assess whether a designed sequence folds and binds as designed raised de novo binder success nearly 10-fold [99].

Leakage and temporal holdouts. A result is only as informative as the separation between the test set and everything the model has seen. One analysis reports major leakage under common splits by sequence or metadata similarity and recommends splitting on the three-dimensional similarity of interfaces [100]. A model can also meet the test set in pretraining: for protein melting-point prediction, a split that accounts for the pretraining corpus changed the measured performance consistently against a naive split [101]. For antibody design, CHIMERA-Bench states that methods are evaluated on different SAbDab snapshots, non-overlapping test sets and incompatible metrics, and provides splits for unseen epitopes, unseen antigen folds and later-deposited targets [98]. A temporal holdout works only when its date belongs to the right model. One benchmark of 110 antibody-antigen complexes was cut at 30 September 2021, the training cut-off its authors state for all four methods tested [45]. Another used public entries deposited after May 2024 plus confidential structures [102]. A date alone does not exclude an old antibody redeposited with a new antigen variant, so identity and structure checks accompany it.

This study uses four leakage axes. Antigen sequence identity is limited to 40% at 80% coverage and antigen structure to a TM-score of 0.50 at 80% coverage. Epitope overlap is limited to a

Jaccard index of 0.30 on residues within 4.5 Å of the partner, and CDR-H3 identity to 80% at 80% coverage. Records linked on any axis go to the same side of the split, and a temporal cut-off of 1 January 2025 by PDB deposition date sends records deposited on or after it to the holdout (Section 2.4.3). These checks are a floor. They miss shared interface geometry, overlap with a foundation model's pretraining corpus, and group-level links such as lineages, assay plates and campaigns. A claim that a model generalises beyond its training records needs a prospective cohort collected after the model is frozen.

Co-folding judges and their calibration. A co-folding judge predicts the antibody-antigen complex and returns a confidence score. Three questions decide whether it can be used: does it find a correct pose, can it pick that pose among its own samples, and does its confidence separate binders from non-binders?

Pose accuracy for antibodies is low. FoldBench reports antibody-antigen failure rates above 50% for current methods [103]. On a benchmark of structures deposited after May 2024, the success rates of seven predictors rarely exceed 25% [102].

Picking the pose is the second weakness. For AlphaFold 3 on 110 antibody-antigen complexes unseen in training, the mean DockQ of the best of 200 models rose above 0.5, while the model ranked first by the judge's own confidence reached about 0.37. The authors trace the gap to errors in the predicted aligned error [45]. SNAC-DB finds the same pattern for Boltz-2 on 101 nanobody complexes. Across 1 to 1,000 samples per target, oracle selection improves from 11.9% to 50.5% success, while confidence ranking stays between 10.9% and 14.9% [102]. More samples raise the best pose the judge could return and leave the pose it does return largely unchanged.

Separating binders from non-binders is the third. Smorodina and colleagues paired 106 cognate nanobody-antigen complexes with 11,342 shuffled pairings. The average precision with which ipTM ranked cognate above shuffled pairs was 0.187 for AlphaFold 3, 0.067 for Chai-1 and 0.026 for Boltz-2, against a random baseline near 0.011. More sampling improved the structures without improving the separation [46]. The authors assume the shuffled pairs do not bind and note that a small share may. The score also depends on the input construct. ipTM falls when disordered or accessory regions are included although the interface is unchanged, and an interface-restricted score, ipSAE, separates true from false complexes better than ipTM does [104]. Across 3,766 designed binders against 15 targets, ipSAE outperformed ipAE and ipTM, and performance varied by target [105]. On 4,184 antibody-antigen pairs, Liang and colleagues

separate within-antigen ranking from cross-antigen ranking and show that interface confidence scores lose calibration across distinct antigens [106].

Designs chosen by a model are not an independent sample of the data that model was fitted on, and finite-sample confidence sets exist for that dependence [107]. Calibration is therefore measured on the population being judged. A judge report states the model, version, seeds, construct, score and threshold, with its error rate on decoys and on designs. The share of all designs that the judge admits and an external assay then fails is the quantity of interest (Section 2.10).

1.5.1 Specificity, developability and immunogenicity

Specificity. Specificity is relative to a panel. A candidate is specific if it binds the target and does not bind a named set of other proteins under a stated assay. Panels hold orthologs from the pharmacology species, paralogs, abundant circulating proteins and, for a systemic drug, membrane and secreted human proteins. A panel samples the proteome, so a clean result holds for its members only. Polyreactivity belongs to the candidate alone. The polyspecificity-reagent assay is a flow-cytometric readout compatible with yeast display, and it correlates with cross-interaction chromatography, a solubility surrogate, and with a baculovirus-particle ELISA, a surrogate for clearance [14]. For nanobodies the reagent was a detergent-solubilised insect-cell membrane preparation, and a sequence model trained on the readout reached an AUC above 0.8; over 90% of its predicted substitutions reduced polyreactivity in three scaffolds [108].

Pair-specific off-target binding belongs to a candidate and one named protein. A clean polyreactivity score does not show that a given paralog is unbound. A pair-specific panel does not detect general stickiness. A report needs both.

Computed panels are younger. SurfDiff scores target surface patches against related proteins before any binder exists, which ranks epitopes by selectivity without scoring any candidate [109]. Odin-Multi, for protein binders, treats cross-reactivity and specificity as explicit design objectives [110]. Co-folding confidence is a weak pair-specific oracle, as the shuffled-pair results above show, and a pair oracle needs matched assay labels for candidate-antigen pairs and a split by antigen family. The full system reports panel coverage beside every risk estimate (Section 2.10.1).

Developability. Developability is a vector of measurements: expression, monomeric purity, thermal stability, self-association, polyreactivity, viscosity, chemical stability and immunogenicity risk. Reference distributions come from clinical-stage antibodies. Jain and

colleagues expressed 137 clinical-stage antibodies as IgG1 and measured them in a dozen assays to set empirical boundaries [111]. The Therapeutic Antibody Profiler derives five computed guideline metrics from clinical-stage therapeutics, one of which is the asymmetry between heavy- and light-chain surface charge [15]. A single-domain VHH has no light chain, so that metric has no counterpart, and IgG1 boundaries do not carry over to other formats. A sequence motif only nominates a chemical liability: for Asn deamidation and Asp isomerisation in CDRs, conformational flexibility, the size of the next residue and secondary structure characterise the hotspots [16]. FLAb2 compiles public therapeutic-property datasets for expression, thermostability, aggregation, polyreactivity and pharmacokinetics [112]. In the blinded 2025 Ginkgo Antibody Developability Competition, sequence and structure predictors ranked hydrophobicity and polyspecificity better than affinity or self-association [113].

Immunogenicity. The humanness score OASis was reported to correlate with clinical immunogenicity [18]. NetMHCIIpan predicts peptide binding to MHC class II [19], yet an industry commentary names the missing correlation between predicted epitope content and anti-drug-antibody outcome as a clear limit on its use [114]. Scans cover the final assembled construct, because a junction creates peptides that neither domain contains.

Glycans. Glycans act on both sides of the interface. On the antigen, mobile glycans can shield a large share of the protein surface [13], so an epitope that looks accessible on a bare structure may be blocked on the glycoprotein. On the binder, an N-linked glycan in the variable domain can raise or lower affinity depending on its position [115] and adds chemical heterogeneity [116]. A designed binder with an Asn-X-Ser/Thr sequon (X is any residue except Pro) therefore needs a sequon-free sequence or measured occupancy before its contacts are read (Section 2.11).

What a benchmark reports. Two hit rates can be compared only if both state the targets, with their identifiers, deposit dates and similarity to the training sets of generator and judge. Both must also state the cut-off date, the model it is measured against, and the numbers of designs generated, scored and tested. They also state the judge (model, version, seeds, construct, score and threshold) and the success rule (assay, construct, concentration, threshold and replicate rule). Controls include shuffled and non-cognate decoys, an antigen swap and a reference binder. Specificity needs a panel manifest, the assay and the polyreactivity readout, and developability needs assays and references in the tested format. The statistics state the resampling unit, the interval and the number of targets. With 30 targets, the normal-approximation 95% margin near a 50% hit rate is about 18 percentage points. Any comparison

with a published number states which of these differences apply and that the panels are not matched (Section 2.13).

1.6 Gaps, aims and scope

Gaps in the published record. Generators of antibodies and nanobodies exist, and most report laboratory tests (Section 1.4). Among the systems we read, four features of the record bear on this study. The first is the framework. Twelve of the 19 systems in **Table 2** keep it fixed outright, and several draw it from a short list of scaffolds. IgGM takes the framework sequence from the user [76], TFDesign-sdAb mutates framework residues of a known antibody [93], and the two systems that generate the framework, AbDiffuser and IgDiff, take no antigen [73, 74]. The primary texts report no epitope-conditioned generation of the framework, the CDRs and their placement together, and no comparison of fixed and generated frameworks at matched compute.

The second is the generator. La-Proteina and Proteina-Complexa report no antibody result [68, 71] (Section 1.3.1), and UniMoMo, a latent diffusion model that includes antibodies among its benchmark domains, reports computational benchmarks only [94]. The third is specificity. No primary text reports a self-proteome panel, the specificity tests that exist are narrow, and co-folding confidence is a weak pair-specific oracle (Section 1.5.1). Odin-Multi treats cross-reactivity as a design objective and SurfDiff ranks epitopes by selectivity [109, 110].

The fourth is evaluation. Reported hit rates come from different targets, cut-off dates, numbers of designs, judges and success rules, and no shared benchmark connects the epitope-conditioned systems (Section 1.4.1). CHIMERA-Bench retrains earlier CDR generators on shared splits and covers CDR generators only [98]. Training compute is rarely stated, and Chai-2, JAM-2† and Latent-X2 are available only through partners or a platform.

Aims. This study has three aims. The first is to specify an unconditional generative base model, with its data format, autoencoder, denoiser, training procedure and compute rule, in enough detail to repeat the work, and to supply its code. The second is to specify the full system that would turn the base model into an antigen-conditioned design tool for VHH and Fv binders, with a free framework, typed evidence channels and a fixed-size selection. The third is to fix the evaluation design in advance, so that a later result is read against a protocol set beforehand: four leakage axes, a temporal holdout, a rule for comparison with published numbers, and the hypotheses H1 to H6 (Section 2.13.4).

Scope. This paper specifies a system and releases the code of its base model. The base model is an autoencoder and a flow-matching denoiser configured for training from random initialisation on an unconditional task (Section 2.1). It is implemented, and its code is supplied. Antigen-conditioned generation, guidance, maturation, judging, liability channels and portfolio selection are specified as the design of the full system. The work is computational and contains no laboratory step. The binding, specificity, safety and accuracy of a designed molecule are outside its scope. Other systems enter through their published numbers only, and a comparison states the protocol differences between the studies (Section 2.13.2). The Supplementary Information holds the mathematical derivations, the data tables, the evaluation protocol and the explanation of the code.

2 Materials and Methods

2.1 System overview

Omniverso is a computational system for the de novo design of antibody and nanobody binders against a registered antigen and epitope. The design connects molecular generation, decoding into sequences and structures, evaluation in typed channels and the selection of a fixed-size portfolio. It ties every output to the data, the model state, the numerical settings and the evaluation procedure that produced it. All outputs are computational predictions, and the work contains no wet-lab step.

Scope. This paper specifies the full system and releases the code of its base model. The base model (cycle 1) is an autoencoder and a flow-matching denoiser, configured for training from random initialisation on an unconditional task. It takes no antigen, no hotspot and no format token, and loads no released weight. The full system adds antigen conditioning, guidance, maturation, judging, liability channels and portfolio selection. These parts are specified as a design, and the code package covers the base model. The binding, specificity, safety and accuracy of a designed molecule or of the model are outside the scope of the manuscript.

Figure 1 draws the design problem on one deposited complex (panel a), the two binder formats (b), and the campaign, the base model and a designed run (c).

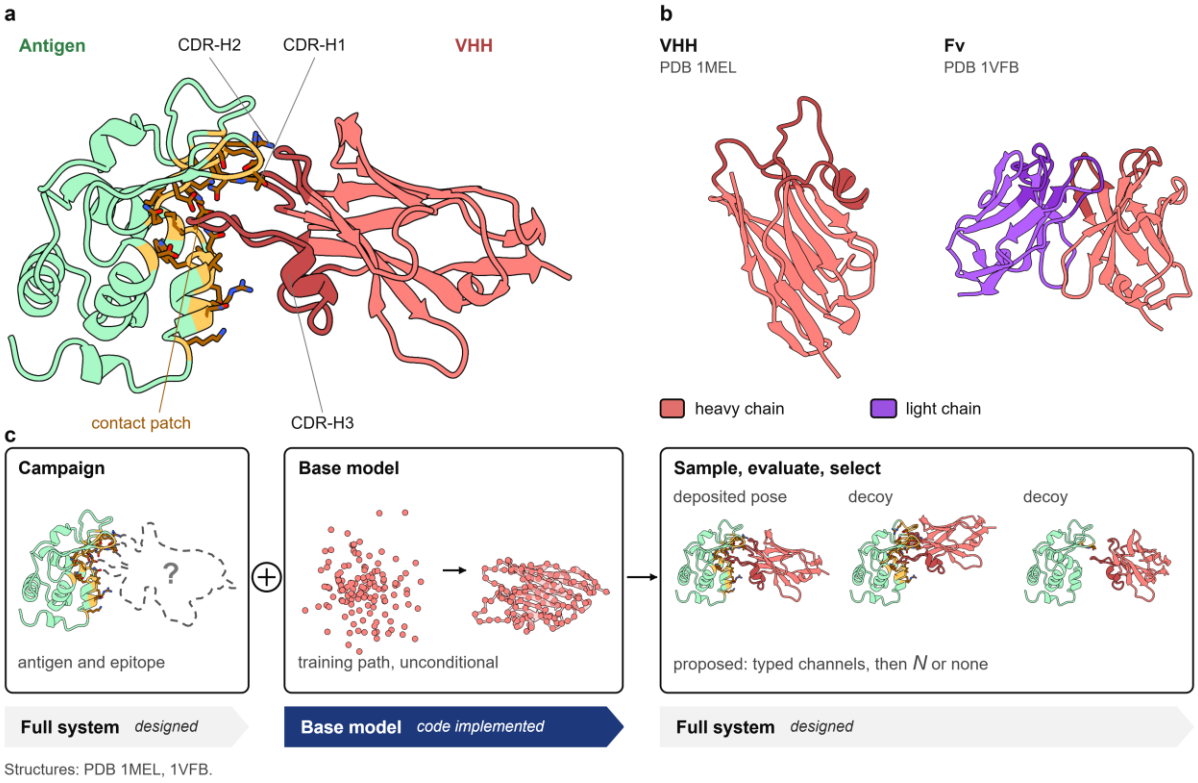

Figure 1. One deposited complex sets the design problem, and the base model, with its training code implemented, sits between a campaign and a designed run.

a, PDB 1MEL, the camel VHH cAb-Lys3 (chain A) bound to hen egg-white lysozyme (chain L). Each CDR loop carries the darker tone of its chain. The contact patch is every lysozyme residue with an atom within 4.5 Å of the VHH, 24 residues; it is a contact set of the deposited complex and not an epitope annotation. **b,** The two binder formats. The VHH of PDB 1MEL has one variable domain, and the Fv of PDB 1VFB has a heavy and a light chain. The two entries are different antibodies, and they name the formats only. **c,** A campaign names an antigen and an epitope, here stood for by the contact patch of a, and the binder to be designed is a dashed silhouette. The base model takes no antigen, no hotspot and no format token. Its frames are the C-alpha atoms of PDB 1MEL chain A on the straight training path $x_t = (1 - t)\, x_0 + t\, x_1$ at $t = 0$ and $t = 1$, with x_0 drawn with standard normal entries in nanometres, without the shared translation that the base law adds. A designed run then samples, evaluates and selects; the three poses are the deposited pose and two rigid-body decoys of it, and none of them is scored. Typed channels answer one question each, and the selector returns exactly N designs or a typed refusal. Panels a and b show deposited structures. The noise frames, the dashed silhouette and the two decoys are chosen illustrations. The cards and bands marked designed are specified as the design of the full system. CDR, complementarity-determining region; Fv, variable fragment; PDB, Protein Data Bank; VHH, variable domain of a heavy-chain-only antibody.

Base model. The autoencoder maps each residue to a deterministic latent of 8 numbers. Its encoder reads 169 numbers per residue: centred atom37 coordinates (111), an atom mask (37) and a residue category (21). Its decoder reads the C-alpha coordinates (3) and the latent (8), and returns atom37 coordinates and residue logits. The shards store coordinates in ångströms. The code centres each complex and multiplies the coordinates by 0.1, so the networks, the noise and the coordinate losses work in nanometres (Section 2.2). Encoder and decoder each have 12 blocks of width 768, and the autoencoder has 255,873,676 parameters (encoder 127,949,832, decoder 127,923,844).

The denoiser learns a velocity field on the flow state, which holds 11 numbers per residue: the C-alpha coordinates (3) and the latent (8). It has 14 blocks of width 768 with 12 heads, a pair stream of width 256 and no triangle layers. With a pair update in every layer it has 153,102,091 parameters. The reference denoiser of the same family has about 159 M parameters [71], so the count here is 3.7 percent lower. The construction follows partially latent flow matching [68]: coordinates and latent are transported from noise to data on independent times. The base model is a design of the same kind as the published networks; a comparison of its architecture and quality with theirs is outside the scope of this paper.

Training has two stages. The encoder and decoder train first. The encoder is then frozen, and the denoiser trains on its latents (Section 2.6). Training losses describe training behaviour and are not measures of model quality (Section 2.13).

Full system. The full system is designed to turn the base model into a campaign tool. A campaign defines the antigen, the epitope, the antibody format, the movable and sequence-design regions, a specificity panel and the size N of the final portfolio. The generator draws many molecular proposals, each as one numerical state. Independent channels then answer different questions about each decoded proposal. The selector returns a set of exactly N candidates, or the typed outcome `INFEASIBLE_PORTFOLIO` when no set meets the constraints.

The base model is specified in Section 2.2 to Section 2.6. The parts that only the full system has are the immunoglobulin contract (Section 2.3), antigen conditioning and Interface-JEPA (Section 2.7), guidance and sequential Monte Carlo (Section 2.8), maturation (Section 2.9), the judge and cross-reactivity instrument (Section 2.10), the liability, presentation and glycan channels (Section 2.11), the decision layer (Section 2.12) and the evaluation design (Section 2.13).

Design inputs. Six design inputs act through generation, steering and selection: the antigen, its topology, its physicochemistry, its surroundings, the shape of the paratope and the regime of the self-proteome panel. The first five enter the generator, and the self-proteome panel acts through steering and selection (Section 2.7).

Typed outputs. Each biological question has its own output type. The eight types are the pose score, the epitope metrics, the pair-binding probability, the affinity value, a generic polyspecificity readout, a named off-target result, a chemical liability and a predicted MHC-II presentation load. Each answers one question about one required object, for example one

antibody-antigen pose or one named antibody-antigen pair. Types are validated before ranking or aggregation, and no report converts one type into another.

Evidence channels. The design condition (what the campaign asked the generator to satisfy), the model prediction (what a computational judge reported) and the experimental observation (what a physical measurement recorded) stay distinct. This work has no experimental observation, and a report shows a missing measurement as missing.

Formats. VHH de novo is the first format admitted. Paired Fv de novo is added when its gates pass, and editing is inactive in version 1. A component returns a typed failure when its inputs are absent and never substitutes another dataset, model, numbering scheme, threshold or unit.

2.2 State and geometry

A molecule enters the model as a list of residues, and each residue carries a fixed set of numbers. The base model reads one protein complex at a time, with no antigen, no hotspot and no format token. **Figure 2** draws the three nested residue sets on one deposited nanobody: on the fold (panel a), on the sequence beside what the entry observes (b), and as a key with the counts (c).

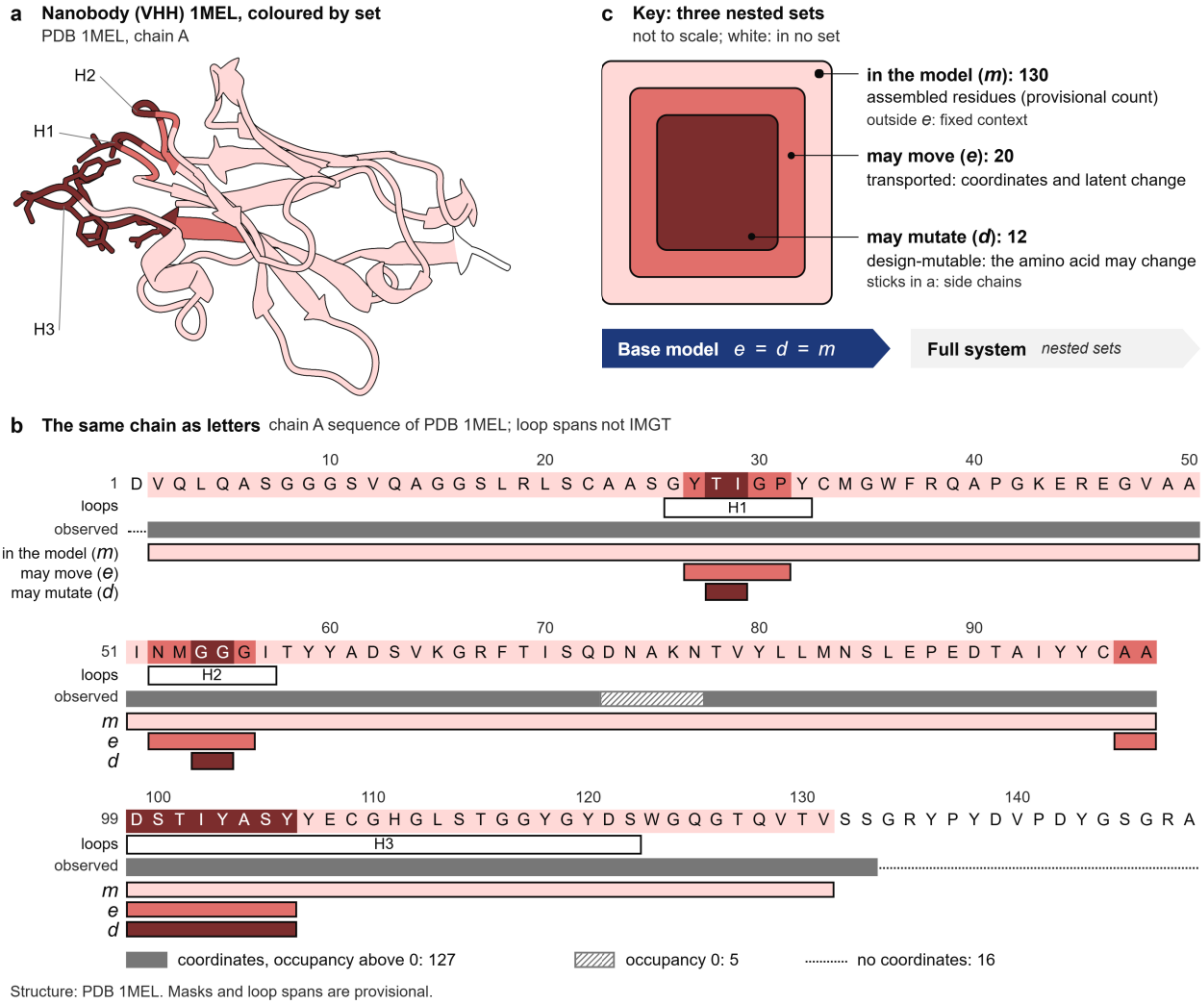

Figure 2. Three nested residue sets are marked on one deposited nanobody.

a, The fold of PDB 1MEL chain A, coloured by set: in the model m (assembled residues) pale, may move e (transported residues) rose, may mutate d (design-mutable residues) dark rose. Sticks are the side chains of the d residues. Positions without coordinates are not drawn, and the two positions with coordinates outside m are white. **b,** The same chain as letters, the 148 positions of the entity sequence in three rows. The loop track gives the chosen spans H1 26 to 32, H2 52 to 57 and H3 99 to 122 by chain-A sequence index, which are wider than the IMGT windows of the immunoglobulin contract. The observed track separates three states of the deposited entry: 127 positions carry coordinates with occupancy above 0, 5 positions (73 to 77) carry coordinates at occupancy 0, and 16 positions (1 and 134 to 148) carry none. The lower three tracks are m, e and d. **c,** The key, with the counts 130, 20 and 12 and one definition each. The transported state of a residue is its C-alpha position and its 8-number latent. The boxes are not drawn to the ratio of the counts. The implemented base model sets $e = d = m$ for every complex, and nested sets belong to the specified full system. The structure, the sequence and the three observation states are read from PDB 1MEL. The sets m, e and d and the loop spans are chosen for this example; the counts 130, 20 and 12 are example values, so 130 is not the length of the deposited chain. IMGT, ImMunoGeneTics numbering; PDB, Protein Data Bank; VHH, variable domain of a heavy-chain-only antibody.

Residues and atoms. Let $L_{\max}$ be the padded number of residue rows in one batch entry. Rows beyond the real length hold zeros and are excluded by the masks below. Each residue has a category s coded 0 to 20, a block of atom coordinates and a latent vector $z \in \mathbb{R}^8$. Atoms follow the Atom37 layout of La-Proteina [68]. Each residue owns 37 atom slots, and a slot holds three coordinates and a presence flag M_{atom}. The C-alpha atom is slot 1 and is present in every residue. An absent atom has coordinates of zero after centring.

The coordinate block has $37 \times 3 = 111$ numbers per residue, and the category is a one-hot vector of width 21. Coordinates are stored in ångströms in the shards and used in nanometres inside the model. The chain index and the residue index identify each residue uniquely. The encoder computes the latent.

Encoder, decoder and flow state. The encoder reads, for each residue, the 111 centred coordinates in nanometres, the 37 presence flags and the 21-wide category vector, which is 169 numbers. It is deterministic: one input gives one latent $z_i \in \mathbb{R}^8$ per residue, computed in FP32. The decoder reads the C-alpha position in nanometres and the latent, 3 + 8 numbers, and writes 37 × 3 coordinates in nanometres and 21 category logits. The flow transports only the C-alpha position and the latent of each residue,

$$q_{\mathrm{raw},i} = (x_{\mathrm{C}\alpha,i},\, z_i), \qquad x_{\mathrm{C}\alpha,i} \in \mathbb{R}^3, \quad z_i \in \mathbb{R}^8,$$

which is 11 numbers per residue. The decoder is trained to rebuild the other atoms and the category from these 11 numbers. The latent channels are learned and carry no assigned chemical meaning. The denoiser also receives the chain index and the residue index of every row. It encodes the index with sinusoids and adds a pair feature, the signed index difference clipped to −128 through 128, with one extra category for residues on different chains.

Masks and the state schema. Three nested masks select residues:

$$d_{\mathrm{res}} \preccurlyeq e_{\mathrm{res}} \preccurlyeq m_{\mathrm{res}}, \qquad d_{\mathrm{res}},\, e_{\mathrm{res}},\, m_{\mathrm{res}} \in \{0,1\}^{L_{\max}}.$$

The symbol $\preccurlyeq$ compares two masks entry by entry, and the sums of the masks are n_d, n_e and n_m. The assembled-residue mask m_{res} marks the residues of the assembled molecule, and rows outside it are padding. The transported mask e_{res} marks the residues of the transported state, and only these move. The design mask d_{res} marks the residues whose category may change.

A residue in e_{res} and outside d_{res} moves while its category stays clamped. A residue in m_{res} and outside e_{res} is context: it is stored, carries zero velocity and zero loss, and adds no dimension to the state. Such a residue reaches the model through the condition c. **Figure 2**b draws the three masks on one VHH sequence with $n_m = 130$, $n_e = 20$ and $n_d = 12$. The rules that derive the masks from IMGT regions are in Section 2.3.1.

The base model sets $e_{\mathrm{res}} = d_{\mathrm{res}} = m_{\mathrm{res}}$ for every complex. Every real residue moves and every category is decoded, so the code stores only m_{res}, as a padding mask. Clamped framework residues need $e_{\mathrm{res}} \subsetneq m_{\mathrm{res}}$, which the full system introduces.

The random state is packed from the rows of e_{res} alone. The padded arrays are storage. Packing gathers the selected rows into one vector, and scattering returns the updated rows to their places beside the fixed rows. Fixed and padding rows therefore never become random dimensions. The padded length, the three masks and a coordinate profile g (next subsection) form the state schema

$$\mathcal{S}_{\mathrm{state}} = (L_{\max},\, m_{\mathrm{res}},\, e_{\mathrm{res}},\, d_{\mathrm{res}},\, g).$$

The schema, the generation condition c and the packing order together fix the probability space. Conditioning variables stay outside the transported density.

2.2.1 Rigid motions and coordinate gauges

A rigid motion applies one rotation $R \in \mathrm{SO}(3)$ and one translation $t \in \mathbb{R}^3$ to every atom, $x_i \mapsto Rx_i + t$. The set of such motions is the group $\mathrm{SE}(3)$. Reflections are excluded because a mirror image turns L-amino acids into D-amino acids. A quantity is invariant when a rigid motion leaves it unchanged, as it does an interatomic distance. A quantity is equivariant when it moves with the molecule, as a displacement vector does. Physical laws give no preference to a laboratory frame, so a gauge, a rule that fixes the frame, gives one physical object one numerical form.

Two situations need different gauges. For an isolated molecule, global translation is redundant and the gauge removes it. When an antigen is present, the position of the antibody relative to the antigen is the quantity being designed, and the gauge keeps it. Three coordinate profiles result. In the centred frame of the base model, positions are expressed relative to the atom centroid, the coordinate dimension is $d_x = 3n$ for a complex of n residues and the state dimension is $D = 11n$. The unanchored monomer profile and the context-anchored profile are used by the full system, and n_e denotes the number of transported residues.

The unanchored monomer profile applies when $e_{\mathrm{res}} = m_{\mathrm{res}}$ and the solve has no antigen, fixed framework or external frame. It removes the centroid of the transported C-alpha positions:

$$y_x = (B_e^{\top} \otimes I_3)\,\mathrm{vec}(x_{C\alpha,e}), \qquad d_x = 3n_e - 3.$$

Here $x_{C\alpha,e}$ holds the n_e transported C-alpha positions and vec stacks them into one column. The matrix B_e has n_e rows and $n_e - 1$ orthonormal columns that span the residue patterns summing to zero. I_3 is the 3×3 identity, and the Kronecker product $\otimes$ applies the same centring to the x, y and z components. A registered Helmert or QR construction with fixed column signs makes B_e deterministic. In a one-dimensional example, three residues at 1, 4 and

7 Å have a centroid of 4 Å and centred positions of −3, 0 and 3 Å. Three numbers carry two degrees of freedom.

The context-anchored profile serves every solve with an antigen or a fixed framework, and every solve with $e_{\mathrm{res}} \subsetneq m_{\mathrm{res}}$. A registered context frame supplies an origin $a_c \in \mathbb{R}^3$ and a rotation $R_c \in \mathrm{SO}(3)$. Each transported position becomes

$$y_{x,i} = R_c^{\top}(x_i - a_c), \qquad d_x = 3n_e.$$

The origin is subtracted first, and the result is rotated into the axes of the frame. The relative placement of antibody and antigen stays in the state. If antibody and context move together, a_c and R_c move with them and $y_{x,i}$ is unchanged (proof in Supplementary Section S1). The frame identifier and the rule for degenerate epitope geometry are recorded with the campaign.

The base model uses neither chart. The shard reader returns coordinates as stored, in ångströms, and the model code centres every complex whenever a batch is used. The reference point is the mean position of all present atoms of the complex. Centring runs in FP32 in ångströms, and the code then multiplies the coordinates by 0.1, so the model works in nanometres. The atom centroid differs slightly from the C-alpha centroid, so the C-alpha positions have close to, but generally not exactly, zero mean.

The coordinate noise of the base model is a Gaussian with a standard deviation of 1 nm per coordinate, plus one translation per complex with a standard deviation of 0.2 nm (2 Å) per axis. The shared translation follows the training recipe of Proteina-Complexa, which gives it in nanometres. That recipe uses it to place a binder at an interface and describes it as irrelevant for monomer generation [71]. No likelihood is computed in the base model, so the full-dimensional support that the reduction by three secures for densities does not arise there.

The base-model networks have no symmetry guarantee. The denoiser projects raw coordinates into its feature stream and builds its pair features from distances, which are invariant. The architecture is not equivariant by construction, and the code applies no rotation augmentation. The full system requires the velocity field to satisfy $f_u(g \cdot x,\, g \cdot c) = \rho_g f_u(x, c)$, where ρ_g rotates the velocity of every residue by R and $u \in [0,1]$ is the master clock of the flow. With an invariant base law, such a field gives a flow that commutes with every rigid motion, so moving the antigen moves the generated antibody with it (proof in Supplementary Section S1). Equivariance is a consistency condition on the model and is not evidence of binding.

Scaling and the dimension of the state. Coordinates arrive in ångströms and latents have learned scales. The full system converts both to dimensionless numbers, so that one loss can weigh them:

$$\bar{y}_x = \frac{y_x}{\ell_0}, \qquad \bar{z}_{ir} = \frac{z_{ir} - \mu_{z,r}}{\sigma_{z,r}}.$$

Coordinates are divided by a registered length $\ell_0 > 0$ in ångströms. For channel r of residue i, the latent has the training-set mean $\mu_{z,r}$ subtracted and is divided by the training-set standard deviation $\sigma_{z,r}$. With $\ell_0 = 3$ Å, a coordinate of 6 Å becomes 2. A latent value of 5 with mean 3 and standard deviation 1 also becomes 2. The packed dimensionless state is $\bar{q} = (\bar{y}_x, \bar{z}_e) \in \mathbb{R}^D$. Its default base law is a standard Gaussian in each block, $\bar{y}_{x,0} \sim \mathcal{N}(0, I_{d_x})$ and $\mathrm{vec}_e(\bar{z}_0) \sim \mathcal{N}(0, I_{8n_e})$.

The base model applies one fixed length scale. The factor of 0.1 from ångströms to nanometres equals $\ell_0 = 10$ Å, so a coordinate of 6 Å enters the model as 0.6 nm. The latent is not standardised, which equals $\mu_z = 0$ and $\sigma_z = 1$, and the output of the encoder enters the flow unchanged. A penalty of weight 0.01 on the mean squared latent controls its scale. Each profile of the full system registers its own ℓ_0.

The state dimension adds the coordinate block to the latent block, $D = d_x + 8n_e$, so

$$D = \begin{cases} 11n_e - 3, & \text{unanchored monomer,} \\ 11n_e, & \text{context anchored or base-model frame.} \end{cases}$$

For $n_e = 130$ the latent block has $8 \times 130 = 1{,}040$ numbers. The unanchored coordinate block has $3 \times 129 = 387$, so $D = 1{,}427$. The anchored block has 390, so $D = 1{,}430$. For $n_e = 20$ the anchored state has $D = 220$. Rows in m_{res} outside e_{res} add nothing. The base model pads complexes to 300, 400 and 500 residues, so a full bucket holds 3,300, 4,400 and 5,500 flow numbers per complex. These figures give storage size and not state dimension, because padded rows are masked. Whether 8 latent numbers per residue keep enough side-chain and category information is a property of the trained autoencoder and is assessed with the reconstruction diagnostics of Section 2.13.

2.3 Immunoglobulin contract

The immunoglobulin contract is the set of rules that the full system applies to every antibody or nanobody that it reads or generates. The base model trains on an unconditional task and does not number, mask or admit residues by these rules.

2.3.1 Numbering, regions and the CDR3 prior

Numbering record. Every variable domain keeps three coordinate systems. The raw index is the zero-based offset of a residue in the submitted sequence. The tool position is the pair (base number, insertion letter) that the numbering tool returns. The IMGT label is the canonical site name, with insertions written as decimal ranks. The raw index shifts with every insertion, whereas an IMGT label names the same site in every domain. The tool output is kept, and normalisation appends a field and the converter version.

Numbering tools. The numbering of record comes from ANARCII, version 2.0.8 [117]. ANARCI [10] with HMMER runs as an independent cross-check and keeps its own record. We set the independent ANARCI/HMMER cross-check threshold to a bit score of 100. The cross-check does not replace the numbering of record, and a difference between the two position maps is stored as a discrepancy. The IMGT scheme [9] annotates 128 positions [10]. Curation rejects and counts every record that cannot be numbered (Section 2.4.2). The supplied support code (Supplementary Section S4) calls ANARCI with the package defaults to read CDR ranges, and it applies neither ANARCII nor the bit-score rule.

Insertions. The third CDR holds 13 positions, 105 to 117, in the basic scheme. A longer loop receives extra positions between 111 and 112, created in the order 112.1, 111.1, 112.2, 111.2 and so on. The numbering output writes them as letters, such as (112, A). The converter reads a letter as a spreadsheet-style rank (A is 1, Z is 26, AA is 27), so (112, A) becomes 112.1 and (111, B) becomes 111.2. The allocation rank, the order of creation, is

$$\mathrm{alloc}(112.r) = 2r - 1, \qquad \mathrm{alloc}(111.r) = 2r.$$

Four extra residues read 111, 111.1, 111.2, 112.2, 112.1, 112 from the N terminus. Creation order differs from chain order, so records are sorted by chain order; a lexical sort would reverse the right-hand labels. An insertion syntax outside the registered roster returns `numbering_unavailable`, and a round-trip test requires that decoding an encoded position returns the position.

Numbering audit. Gate F-IMGT audits the numbering caller on at least 600 independently annotated solved domains (VH, VL, VHH and documented insertion cases). A domain passes when its position map, region map and conserved anchors all agree with the reference. Acceptance needs a one-sided 95 percent exact-binomial lower limit of 0.995 on the pass fraction, no disagreement outside the documented conventions, and no mask index outside the numbered domain. With 600 domains the lower limit is $0.05^{1/600} = 0.99502$ when all pass

and 0.9921 with one failure, so a single failure fails the gate at that size. The thresholds are engineering settings.

Reference identity. For a protein sequence, IgBLAST [118] aligns the V region to translated germline references. The record stores the best reference and the aligned identity, which is the fraction of aligned V-region positions where the two residues match, under the declared gap policy. The result is typed `nearest_translated_V_reference`. An edit toward that reference is named `aa_reference_reversion`. The name `germline_allele_reversion` is reserved for assignments resolved from nucleotide sequence.

Regions and masks. Masks are built from IMGT labels and chain identifiers, never from raw indices (**Table 3**) [9]. Conserved anchors sit at fixed positions, among them cysteine 23, tryptophan 41, cysteine 104, and phenylalanine or tryptophan 118.

Region	IMGT positions	Number of positions
FR1	1 to 26	26
CDR1	27 to 38	12
FR2	39 to 55	17
CDR2	56 to 65	10
FR3	66 to 104	39
CDR3	105 to 117, plus insertions	13 plus insertions
FR4	118 to 128	11

Table 3. IMGT framework (FR) and complementarity-determining (CDR) regions used by the region masks. Positions are IMGT numbers; the counts sum to 128 before insertions.

The labels select residues into the three nested masks $d_{\mathrm{res}} \preccurlyeq e_{\mathrm{res}} \preccurlyeq m_{\mathrm{res}}$ for design-mutable, transported and assembled residues (**Figure 2**, panel b; Section 2.2). When a profile names a position that a sequence lacks, the profile returns `edit_profile_unavailable` for that sequence. A record with a missing CDR residue is rejected at curation, so no mask is built from an incomplete loop.

CDR3 length prior. The prior for format p, chain c and analysis stratum r is a smoothed count of CDR3 lengths in an admitted repertoire:

$$P(\ell \mid p, c, r) = \frac{n_{pcr\ell} + \alpha_0}{\sum_{j \in \mathcal{L}_{pcr}} n_{pcrj} + \alpha_0 |\mathcal{L}_{pcr}|}.$$

Count the clonotypes of length ℓ, add a pseudo-count $\alpha_0 > 0$ to every permitted length so that none has probability zero, and divide by the total. Here $n_{pcr\ell}$ is the clonotype-weighted count, so read depth does not inflate it, and $\mathcal{L}_{pcr}$ is the set of observed, policy-approved lengths. The

repertoire manifest records organism, tissue, disease status, filters, deduplication, sequencing technology, chain type and split. The prior applies when its repertoire files and count table exist. A hand-written range is typed `engineering_length_bound`. As an illustration, counts of 20, 50 and 30 at lengths 12, 13 and 14 with $\alpha_0 = 1$ give $(21,51,31)/103 \approx (0.204,0.495,0.301)$.

2.3.2 Scaffold profiles, cysteines and pairing

Scaffold profile. A scaffold profile is a versioned record tied to one construct. It names the format (VHH, VH, VL or paired Fv), the source construct, the residues allowed at each required IMGT position, the protected positions, the supported CDR3 lengths, the disulfide topology, the free-cysteine mode, the framework reference set and the assembly requirements. The label VHH selects a format, and the registered profile supplies the constraints. A VHH profile may protect positions that faced the light chain in a conventional VH, such as framework-2 positions 42, 49, 50 and 52 [6], or specify a disulfide between loops. A profile needs its own scaffold experiments and construct definition, and no universal site map is inferred. Gate F-VHH enumerates the whole exported panel against the profile and needs a conformance fraction of 1.000.

Cysteine table. Every decoded molecule carries a residue-level cysteine table. For each cysteine it records the chain and IMGT position, the role (canonical disulfide, profile disulfide, conjugation handle, unresolved or free), the partner position, the sulphur geometry when a structure exists, the source (parent, generated or edit) and a status. The canonical disulfide joins cysteine 23 in framework 1 to cysteine 104 in framework 3 [9]. A profile declares one of two free-cysteine modes. In `FREE_CYS_PROHIBITED`, every cysteine outside the profile fails admission. In `CONJUGATION_HANDLE`, named positions are reserved for a stated chemistry and every other unpaired cysteine fails; this mode needs assays of accessibility, scrambling, occupancy, heterogeneity and retained function on the assembled construct. The design uses the first mode.

VH/VL pairing. Paired formats add the chain identity, the provenance of the pair, the interface pose and a pairing-likeness score. ImmunoMatch [119] supplies $S_{\text{pair}}(\text{VH}, \text{VL}) \in \mathbb{R}$, typed `pairing_likeness`. It estimates how closely a heavy and light chain resemble the paired chains used in training. Physical association, expression-level pairing and binding keep separate endpoints. ImmunoMatch was trained on paired human repertoires and reports an AUC-ROC of 0.75 on a withheld set and 0.66 on three unseen donors. Paired-Fv admission also needs an interface pose check, a clash and buried-polar analysis and, where relevant, experimental evidence of chain pairing or product purity. A single-chain construct also needs a linker profile. Gate F-PAIR

tests cognate-versus-shuffled discrimination and the agreement between pairing-likeness and interface evidence.

Admission predicate. The predicate combines five checks for candidate X under profile p, which fixes the format:

$$\mathrm{IgPass}(X;p) = N(X) \wedge R(X;p) \wedge L(X;p) \wedge C(X;p) \wedge A(X;p).$$

Here N is numbering completeness, R reference and scaffold compliance, L CDR3 length support, C cysteine topology and A the assembled-format checks. These letters are local to the predicate. A candidate is admitted only when every required numbering, scaffold, length, cysteine and assembly check is available and passes. A check that the profile excludes is left out of the conjunction, and a descriptive result is labelled as such. A missing check is never read as a pass or a fail.

Each candidate emits its numbering record, reference identity, profile and prior identifiers, cysteine table and the result of every check. A canonical-class field for CDR loops is optional and cannot gate, train or reject a design.

VHH de novo is the first format admitted. Paired Fv de novo is added when its gates pass, and editing is inactive in version 1. Until a registered profile exists, the design screens sequences before scoring with an interim filter. It needs bonded cysteines at IMGT 23 and 104 (sulphur distance 1.9 to 2.2 Å in the decoded structure), no other free cysteine, no N-X-S/T sequon (X other than proline) inside CDR1 to CDR3, and a CDR3 of 5 to 25 residues. A registered profile replaces this filter, and the repertoire count table replaces the CDR3 bound.

2.4 Data

The base model is trained on protein structures. Source bytes arrive in quarantine. A source enters curation only after the admission check D0 passes. Curation then removes duplicate structures, numbers the antibody domains, applies structural quality control, annotates contacts, assigns each record to four leakage groups, applies the temporal holdout, freezes the splits and packs shards.

2.4.1 Sources and licences

Each dataset has one of six roles. Pretraining sets are unsupervised corpora. Supervised sets serve later tasks. Calibration sets fix a threshold that a gate reads. Benchmarks are held out from training by construction. Panels supply off-target or self background. References are term lists,

annotation tables and contracts. Every training role serves a model trained from scratch, and no released weights are loaded.

The sources of the base-model corpus are structures: experimental ones from the Protein Data Bank (PDB) [120], AlphaFold models [121], and synthetic dimers built from them. The Teddymer dimers come from splitting AlphaFold DB monomers at the domain boundaries of the TED annotation [122] and clustering the pairs with Foldseek [71]. **Table 4** lists the sources.

Source	Content and count	Reported licence
PDB	Dated mmCIF snapshot of 2026-05-28; first biological assembly of each entry	CC0 1.0
AlphaFold DB, release v6 of 2025-09-15	241,070,489 structures	CC BY 4.0
AlphaFold DB, Foldseek-clustered subset	About 3 M cluster representatives; about 550 k after the pLDDT, length and coil filters; about 350 k after the β-sheet filter	CC BY 4.0
TED domain assignments	Domain boundaries of AlphaFold DB models	CC BY 4.0
Teddymer synthetic dimers	3,556,223 clusters before the quality filter; 510,454 representatives after interface length above 10, interface pAE below 10 and interface pLDDT above 70 [71]	CC BY-NC 4.0
PDB multimers, filtered	45,856 structures after all filters [71]	CC0 1.0

Table 4. Sources of the base-model corpus.

Counts are numbers of structures, clusters or representatives as reported by each source or by [71]. The licence is the one stated by the source.

Base-model corpus. The corpus is drawn from the sources of **Table 4** through a filter ladder that the corpus manifest records. The temporal cut-off T^* applies to PDB entries by deposition date, and AlphaFold models and TED assignments carry no PDB deposition date. The code package also contains a generator of synthetic complexes that passes through the same reader and carries no biological content. Each complex is a seeded random walk of C-alpha positions with a step of 0.7 Å per axis. Atoms scatter 0.15 Å about the C-alpha, 8 percent of the other atom slots are masked, and categories are drawn at random from 0 to 20.

The antibody-specific datasets of the full system are SAbDab [22], SNAC-DB [102], Chimera-Bench [98], FoldBench [103], the Observed Antibody Space [21], IMGT/GENE-DB, CoV-AbDab, Thera-SAbDab, PLAbDab, SKEMPI 2.0, the polyspecificity reagent corpus [108], Absolut! [123] and the Antibody-Nanobody Design Dataset (ANDD) [124]. Supplementary Section S2 gives their counts, versions, roles and licence classes, together with the cut-off dates of published systems.

Licences were read from the release page of each source on 5 and 6 October 2026. Terms attach to the dataset, so an article under CC BY does not make its dataset CC BY. The AGPL-3.0 licence of the code grants nothing on a dataset. Sources under CC0 1.0 or CC BY 4.0 are admitted with attribution. They are the PDB, AlphaFold DB, TED, SAbDab, OAS, IMGT/GENE-DB, UniProt and its reference proteomes, Human Protein Atlas v25.1, SKEMPI 2.0, CoV-AbDab and the polyspecificity reagent corpus. OAS gathers data from many studies, and the terms of each study apply.

Four classes are excluded from the training lineage of weights that may be released. Non-commercial terms apply to Teddymer, Absolut! data, ASD v2, AbAgym, INDI2, PairedAbNGS, AbNatiV and NanoMelt. No-derivative or share-alike terms apply to SURFY, UniCarbKB, OMA, OpenCell images and the Adaptyv competition data. SNAC-DB reserves training to its authors. Custom or paid terms apply to GDPa1, PDBbind, NetMHCIIpan 4.3k and the IgFold OAS models. Sources with no stated licence serve as references or evaluation material until written terms exist. They are AB-Bind, Mason 2021, the Absolut! repository, TEMPRO, the Jain supplement, PLAbDab, PLAbDab-nano, Thera-SAbDab, cAb-Rep and FoldBench (whose article carries CC BY-NC-ND terms).

Training data for weights that may be released must lie under CC BY 4.0, CC0, MIT, Apache 2.0 or public-domain terms. Before a source is used, its licence is reviewed for the object and for the action (training, evaluation or redistribution). The review ends in one of three outcomes (allow, restrict or blocked) and is valid only with the digest of the governing terms. Every weight file carries a lineage record with its datasets, their licences and their file hashes.

2.4.2 Curation, quality control and deduplication

For each dataset, record its release, retrieval date, licence terms, inclusion criteria, deduplication rule and retained counts by split. Distinguish counts reported by the source from counts obtained after acquisition and filtering. For deduplication, the input count equals the retained count plus the number of duplicate exclusions.

D0 halts curation of the source on a checksum or schema failure, on a licence decision that does not permit the action, on a count that does not conserve, on a substituted snapshot and on a derivation that disagrees with its filter. Changed upstream bytes produce a new digest and stop D0.

Quality control decides whether a record may enter a contact annotation, a mask or a pretraining example. The rules are fixed before use and bound to a content digest. Records are kept at a

resolution of 4.0 Å or better when they serve as supervised or benchmark rows or enter the contact annotation of PDB records. Geometry used only for pretraining is recorded separately. The R-free value is recorded, and no rule uses it as a gate. Atoms enter the 4.5 Å contact set only above an occupancy of 0.50, in all sources. Among alternate locations the one with the highest occupancy is kept.

Missing density that touches the interface rejects a supervised or benchmark row. In pretraining-only rows, where no interface is read, the gap is modelled. A missing CDR residue rejects the record in every source. All antibody records are numbered with the IMGT scheme, and a numbering failure rejects the record, is counted and keeps the invocation.

The numbering of record comes from ANARCII, version 2.0.8 [117]. ANARCI [10] with HMMER runs as an independent cross-check with its own record. A record whose cross-check alignment has a bit score below 100 is rejected and counted, and a difference between the two position maps is stored as a discrepancy. Section 2.3 describes the numbering record. Each rejection raises a manifest counter and creates a discrepancy record, and the accepted count enters the conservation check.

Contact annotation applies a 4.5 Å atom-to-atom rule between antibody and antigen atoms whose occupancy passes the gate. It emits a contact-set bitmap per antigen and a rectangular antibody-to-epitope contact matrix. The bitmap seeds the A3 leakage axis below, and the matrix seeds the contact loss of the full system.

Cross-source deduplication. Several structural sources are views of the PDB and may share depositions: SAbDab, SAbDab-nano, SNAC-DB, AbDb, AACDB, Chimera-Bench, FoldBench, the Smorodina panel and the Hitawala–Gray benchmark. Deduplication runs before clustering, so that one deposition receives one set of group identifiers. A structure-derived record is reduced to the key

$$k = \mathrm{SHA256}(pdb \parallel asm \parallel Ab \parallel Ag),$$

where pdb is the entry identifier, asm the biological-assembly identifier, and Ab and Ag the sorted author chain identifiers of the antibody and antigen chains. The key is computed on the biological assembly of the admitted PDB snapshot and never on the identifier of a source. Chain identifiers are normalised to the snapshot, so a renaming by a curator does not change the key. A record without a structure is keyed on the SHA-256 of its numbered VH and VL sequence pair.

Two orders resolve a shared key. Annotation precedence decides which source supplies the record: SNAC-DB, then SAbDab and SAbDab-nano, then AbDb, then AACDB, then the raw PDB snapshot. SNAC-DB leads because its assembly-based curation resolves multi-chain epitopes. It reports multi-chain antigens in 33.5 percent of its complexes against 15.3 percent in its SAbDab comparison [102]. Its records serve for training only where its terms permit that use.

Partition precedence decides which split a shared key lands in, and it overrides annotation precedence. The order is benchmark and temporal holdout first, then calibration, then training. A key claimed by any admitted benchmark is removed from every training and pretraining manifest and from the calibration designs. Each realised manifest lists, for every other source, the shared keys before precedence and those removed by it, with the query that produced them. D0 requires zero benchmark overlap with training after precedence.

2.4.3 Leakage axes, temporal holdout and split freezing

Leakage means that information about a test record reaches training by a route that the split does not cut. The design uses four axes: antigen sequence (A1), antigen structure (A2), the set of antigen residues in contact with the antibody (A3) and the CDR-H3 sequence (A4). Whole-Fv identity is left out because conserved frameworks dominate that comparison while the antigen-contacting loops may still differ. **Table 5** gives the rule and threshold of each axis.

Axis	Object	Rule	Threshold
A1	Antigen sequence	MMseqs2 clustering [125]	Identity at least 40 percent, coverage at least 80 percent
A2	Antigen structure	Foldseek clustering [126]	TM-score at least 0.50, coverage at least 80 percent
A3	Epitope residue set	Jaccard index over mapped antigen residues within 4.5 Å of the antibody, single-linkage	Jaccard index at least 0.30
A4	CDR-H3	MMseqs2 clustering over IMGT positions 105 to 117, single-linkage	Identity at least 80 percent, coverage at least 80 percent

Table 5. Leakage axes: clustering rule and threshold.
Identity and coverage are in percent; the Jaccard index and the TM-score are dimensionless.

A3 uses the Jaccard index of epitope residue sets after mapping them into a shared antigen residue coordinate system, with contacts defined at 4.5 Å, a threshold of 0.30 and single-linkage clustering. The Jaccard index of two epitopes is the size of the intersection of their residue sets divided by the size of the union. An A3 comparison is defined only when that mapping specifies antigen equivalence, reference alignment, insertion handling, unmapped residues and multichain correspondence. A split cannot be declared A3-disjoint where a required comparison

is undefined. The value 0.30 is a setting chosen for discontinuous epitopes. Chimera-Bench groups only identical residue sets, which is a Jaccard index of 1 [98]. The clustering commands for A1, A2 and A4 are:

```
# A1, antigen sequence
mmseqs easy-cluster antigens.fasta clust/ag_seq tmp/ag_seq \
  --min-seq-id 0.40 -c 0.80 --cov-mode 0 --cluster-mode 0
# A2, antigen structure
foldseek easy-cluster antigen_pdbs/ clust/ag_str tmp/ag_str \
  --alignment-type 1 --tmscore-threshold 0.50 -c 0.80 -e 0.001
# A4, CDR-H3 extracted at IMGT 105-117
mmseqs easy-cluster cdrh3.fasta clust/h3 tmp/h3 \
  --min-seq-id 0.80 -c 0.80 --cov-mode 1 --cluster-mode 1
```

Coverage mode 0 measures coverage against the longer sequence of a pair, and mode 1 against the target. Cluster mode 1 joins sequences through transitive links, which is single-linkage clustering, whereas mode 0 is a greedy set cover.

Each curated record carries four labels, $(g_{A1}, g_{A2}, g_{A3}, g_{A4})$. Two records are joined if they share a label on any axis, and the split operates over the connected components of that union graph. A component is therefore never divided between test and training, and one very large component would reduce the material available for holding out.

A committed split must satisfy a disjointness predicate between a test partition T and the union C of training, pretraining and calibration partitions. Four conditions each hold under a deterministic pass over the frozen split. Every record of T was deposited on or after the cut-off T^*. No pair across T and C exceeds the A1 identity threshold on antigen sequence or the A4 threshold on CDR-H3. No pair exceeds the A2 TM-score threshold. No pair with a defined A3 comparison exceeds the A3 Jaccard threshold, and no required A3 comparison is undefined.

The conditions are conjunctive because each alone misses a route. Identity misses remote structural homologues. Structure misses high-identity sequences with unsolved structures. A date misses re-deposition of older entities.

Three further channels pass all four conditions. Splits by sequence or metadata similarity let pairs with similar interface structure cross the split [100]. A test record can sit in the pretraining corpus of an encoder that the model uses [101]. Lineages, epitope families, assay plates, campaigns and retrieval databases couple records in groups, and leakage-reduced splitting can be posed as an optimisation problem [127]. The design therefore adds interface-structure clustering and a record of pretraining provenance for every checkpoint. The base model loads no outside weights, so the second channel does not arise there. A claim that a model generalises beyond its training records needs a prospective cohort collected after the model is frozen. Metadata checks are a necessary floor and give no guarantee.

Temporal holdout. The corpus-wide cut-off is $T^= $ 2025-01-01 by PDB initial deposition date. A record deposited on or after T^* belongs to the temporal holdout. It is excluded from every training, pretraining and calibration manifest, whichever source supplied it. Each record stores a temporal label and a component label, and a record in the holdout removes its whole A1 to A4 component from training. This is the protocol of the full system. The support code of the base model applies the temporal rule and two local checks, whole-chain identity and similarity of C-alpha distance histograms, and drops the leaking test record instead (Supplementary Section S4). The date is later than the last deposit in the released training list of Proteina-Complexa, 2024-08-02 [71], so it controls overlap with the disclosed training set of that system and supports no other claim. SNAC-DB builds its public benchmark from PDB deposits of the preceding year, up to 31 March 2025, and lists the training cut-off dates of the models it evaluates [102]. Chimera-Bench and FoldBench keep their own date rules.

Split freezing. The frozen split file stores, for each record, its split, its temporal label and its four component labels. It also stores six named evaluation splits, each with its parent source and its disjointness condition: `pairing-eval-donor-disjoint` (gate F-PAIR), `tts-holdout-12ag` (F-TTS), `jepa-consequence-stream` (F-J7), `chi-ref-holdout` (F-CHI), `imgt-heldout-independent` (F-IMGT) and `glycan-contact-heldout` (F-GLY). The pairing split also carries donor identifiers, so that no donor stands on both sides. The split file is committed and hashed before training, and its digest is recorded with every run, so the exact membership of each split can be reproduced and audited.

2.4.4 Shard format and sampling

The packer of the full system writes a deterministic sample index with one row per sample. The row holds the sample identifier, the source-record digest, the split, the length bucket, the MSA-depth bucket, the format and coordinate profiles, the digest of the state schema, the shard identifier, the member offset and the payload hash. The payload stores full padded arrays and the three masks of Section 2.2. The loader assigns samples disjointly to ranks and workers, keeps a bounded shuffle buffer with a recorded generator state, conserves sample counts at epoch boundaries and restarts from a saved sampler state.

A base-model shard is an uncompressed NPZ archive of ragged complexes concatenated along the residue axis. Let N be the total number of residues in the shard and C the number of complexes. The arrays are `atom37` (float32, $N \times 37 \times 3$, in Å), `atom_mask` (bool, $N \times 37$), `aatype` (int8, N, categories 0 to 20), `chain` (int16, N), `residx` (int32, N), `offsets` (int64, $C + 1$, the row range of each complex) and `complex_id` (Unicode, C). The C-alpha sits at slot 1 of the 37 atom

slots. A JSON manifest of schema 1, in ångström units, lists the file name and the SHA-256 of every shard. The reader converts nothing: shards and manifests stay in ångströms, and the model multiplies the centred coordinates by 0.1 at its entry (Section 2.2). The format costs 488 bytes per residue (444 for coordinates, 37 for the mask and 7 for the three index fields), so ten million residues take 4.88 GB.

The reader refuses a manifest if a shard is missing, lies outside the manifest directory or has a different SHA-256, or if any array has the wrong type or shape. It also refuses non-finite coordinates, categories outside 0 to 20, offsets that do not start at zero or do not increase strictly, a residue without a C-alpha, a repeated pair of chain and residue index within a complex, and a repeated complex identifier. It reads every shard into memory once per rank, so a random draw never rereads a file, and the host memory needed is the shard bytes multiplied by the number of ranks plus the model and optimiser state (Supplementary Section S5).

Complexes fall into three disjoint length buckets: up to 300, 301 to 400 and 401 to 500 residues. Each bucket is padded to its upper limit, and masks exclude the padding. The global batch is 48 complexes in every update. The product of microbatch, accumulation steps and ranks gives 48 in each bucket: $8 \times 3 \times 2$ for 300, $6 \times 4 \times 2$ for 400 and $4 \times 6 \times 2$ for 500. In each update the sampler draws a bucket uniformly at random, unless a smoke schedule forces one, and then draws 48 complexes from it with replacement. The draw is seeded by the run identifier and the update index through the first 32 bits of a SHA-256 hash, so a resumed run repeats the same draws. Both ranks use one bucket per update and take separate slices of the 48 draws. A complex longer than 500 residues falls in no bucket and is never drawn, because the reference configuration applies no crop. Section 2.6 describes the procedure that consumes these batches.

2.5 Generative model

The generative model works in two steps. A velocity network moves random numbers towards the numbers that describe a real structure, and a decoder turns the result into amino acids and atom positions. Each residue carries 11 numbers. Three place its C-alpha atom, and eight are a learned latent vector from which the decoder recovers the residue type and the remaining atoms. Moving only the C-alpha atoms and the latent, instead of every atom, is the partially latent design of La-Proteina [68].

The denoiser takes its depth, width and head count from the Proteina-Complexa denoiser [71]. Both networks of the base model are implemented for training from random initialisation. The sampler, the master clock that drives it and the density formula belong to the full system, and

the base model draws no sample. Panels a to c of **Figure 3** show the path of the flow, the two block clocks and the two uses of the field. **Figure 4** shows the networks and the split of a residue into explicit and latent parts.

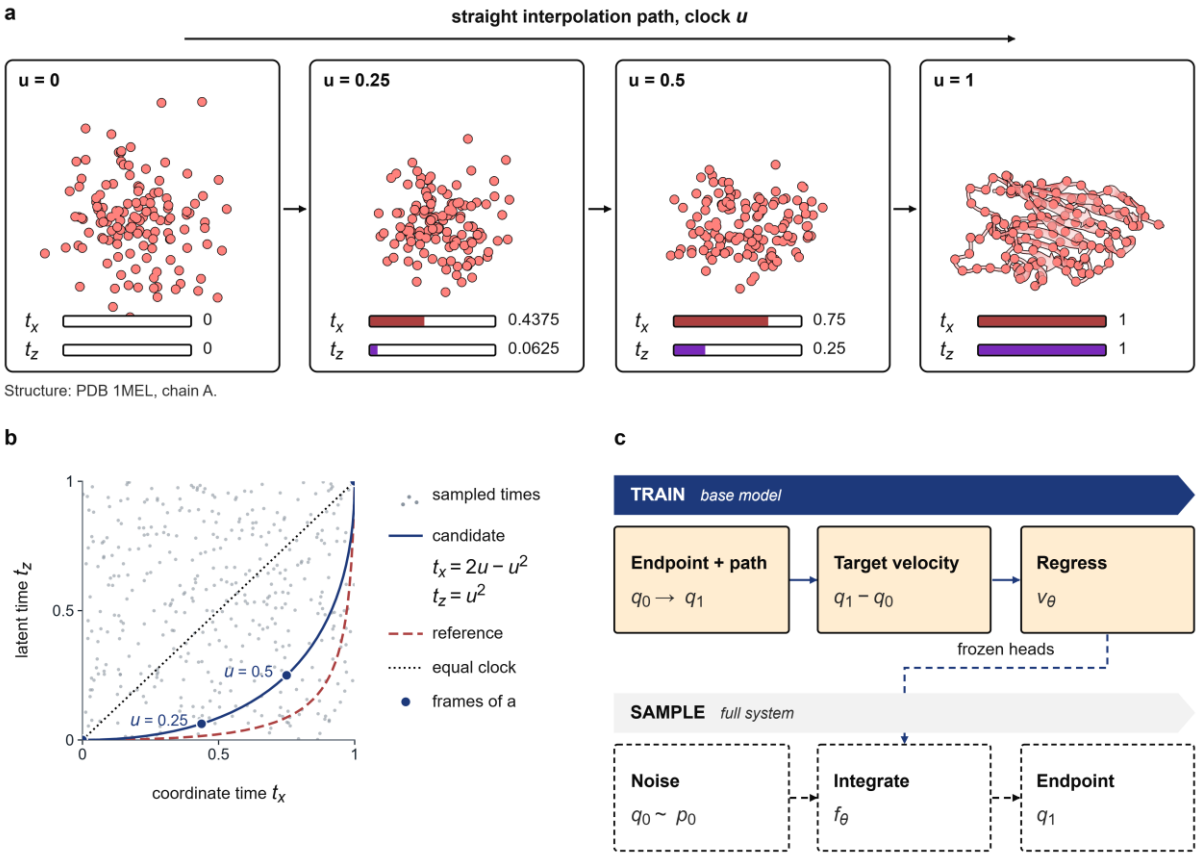

Figure 3. Training regresses the velocity heads on a straight path with two block clocks, and a sampler walks one curve through the square of times.

a, The C-alpha atoms of PDB 1MEL chain A on the straight interpolation $x_t = (1-t)\,x_0 + t\,x_1$, with x_0 drawn with standard normal entries in nanometres (seed 7), without the shared translation that the base law adds, at the master-clock values $u = 0$, 0.25, 0.5 and 1. The bars give the two block times of the candidate pair, $t_x = 2u - u^2$ (0, 0.4375, 0.75, 1) and $t_z = u^2$ (0, 0.0625, 0.25, 1). The latent block carries no picture. The straight interpolation is the path that training uses, and the pairing of the two block times with one clock u belongs to the sampler's candidate, because training draws the two times independently (b). **b,** Four hundred training times, drawn uniformly on the unit square of t_x and t_z. Three sampling curves cross the square: the candidate, the reference $t_x = (1 - 10^{-2u})/(1 - 10^{-2})$ at $t_z = u^2$, and the equal clock $t_x = t_z = u$. The two marked points are the intermediate frames of a, and at $u = 0.25$ the candidate has slopes 1.5 and 0.5. **c,** Training pairs an endpoint with a noise draw. The target is the end difference $q_1 - q_0$, and the velocity heads v_θ are regressed on it. Sampling integrates the master-clock field $f_\theta = (t_x' v_\theta^x,\ t_z' v_\theta^z)$ from q_0 to q_1 with the heads frozen. Panel a shows a deposited structure on a chosen interpolation, and the times of b are drawn from the training law. Sampling is specified as a design, and the base model draws no sample. PDB, Protein Data Bank.

2.5.1 Probability paths and the training target

Transport. Let q be the list of D numbers that describes one complex, and let $u \in [0,1]$ measure progress along a path. A probability path is a family of densities p_u over states. The density p_0 is a Gaussian that can be sampled, and p_1 is the density of encoded real structures. A velocity field f_u moves each state, so that a state follows the ordinary differential equation

$$\frac{\mathrm{d}q}{\mathrm{d}u} = f_u(q), \qquad q(0) = q_0 \sim p_0.$$

When the field carries p_0 to p_u, the densities obey the continuity equation, which states that probability is neither created nor destroyed:

$$\partial_u p_u + \nabla \cdot (p_u f_u) = 0.$$

Where the field makes neighbouring states converge, the density rises, and where it makes them separate, the density falls. This is the construction of a continuous normalising flow [128]. For $f_u(q) = -q$ and $q_0 = 2$ the solution is $q_u = 2e^{-u}$, so $q_1 = 2/e \approx 0.736$, nearby states end closer by the factor e^{-1}, and the density rises by the factor e (Supplementary Figure S2b).

Conditional paths. The intermediate densities p_u are unknown, so f_u cannot be fitted to them directly. Flow matching replaces this problem with an easier one [55]. For each data state q_1 one chooses a conditional path $p_u(q \mid q_1)$ whose velocity $b_u(q \mid q_1)$ has a closed form. Averaging over the data gives the marginal path $p_u(q) = \int p_u(q \mid q_1)\, p_{\mathrm{data}}(q_1)\, \mathrm{d}q_1$. The field that generates it is the mean of the conditional velocities over every data state that could have produced q at progress u. The network v_θ is trained to match the conditional velocity at a sampled state:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E} \parallel v_\theta(q_u, u) - b_u(q_u \mid q_1) \parallel^2.$$

This loss has the same gradient with respect to θ as the loss on the marginal field, under the regularity conditions of the source [55]; the proof is in Supplementary Section S1. Training therefore solves no ODE. It draws a data state, a base state and a progress value, builds the intermediate state and regresses a velocity.

A Gaussian conditional path mixes the data state with noise $\zeta \sim \mathcal{N}(0, I)$ as $q_u = \alpha(u)\, q_1 + \sigma(u)\, \zeta$, with $\alpha(0) = 0$, $\sigma(0) = 1$, $\alpha(1) = 1$ and $\sigma(1) = 0$. Differentiating at fixed q_1 and ζ gives $b_u = \alpha'(u)\, q_1 + \sigma'(u)\, \zeta$. For $\alpha(u) = u$, $\sigma(u) = 1 - u$, $q_1 = 2$, $\zeta = -1$ and $u = 0.5$ the state is 0.5 and the velocity is 3.

The base model uses the simplest such path. It pairs a base state q_0 with a data state q_1 and draws the line between them:

$$q_u = (1 - u)\, q_0 + u\, q_1, \qquad b_u(q_u \mid q_0, q_1) = q_1 - q_0.$$

The same interpolation applies to the correlated Gaussian base law defined below. The target is constant along each line, and the learned field is the average of this target over all pairs whose lines cross at a point. For $q_0 = (-1,2)$, $q_1 = (3,0)$ and $u = 0.25$ the state is $(0,1.5)$ and the

target is $(4, -2)$. A network output of $(3.5, -1)$ would give a squared error of 1.25. Panel a of **Figure 3** shows the straight path on one deposited chain. Each data state is paired with an independent noise draw. Minibatch optimal-transport pairing would choose shorter pairs [60], and the design does not use it.

Flow state and base law. For one complex of n residues the state is $q = (x, z)$. The base model transports every present residue, so $n_e = n_m = n$ in the notation of Section 2.2. The array $x \in \mathbb{R}^{n\times 3}$ holds C-alpha coordinates in nanometres, centred on the centroid of all present atoms of the complex. The shards store angstroms, so the code centres in angstroms and then multiplies by 0.1 (Section 2.2). The array $z \in \mathbb{R}^{n\times 8}$ holds the latent that the frozen encoder returns for each residue, and the latent is used as the encoder returns it, with no standardisation. The state has $D = 11n$ numbers, so a complex of 130 residues has $D = 1{,}430$. The chart that removes the three global translations has $D = 11n - 3$ and is not used.

The base law is Gaussian, with independent blocks. The coordinate noise is $x_0 = \varepsilon + \mathbf{1} \otimes d$, where ε has independent standard normal entries, so its standard deviation is 1 nm per coordinate, and d is one draw per complex from $\mathcal{N}(0, (0.2\,\mathrm{nm})^2 I_3)$, added to every residue. The coordinate block therefore has covariance $(I_n + 0.04\,\mathbf{1}\mathbf{1}^\top) \otimes I_3$ in nm². The shared translation of 0.2 nm, which is 2 Å, is the value of the Proteina-Complexa binder model, which adds it so that the network reasons about global position and gives it in nanometres [71]. The latent noise z_0 has independent standard normal entries. The velocity targets are $x_1 - x_0$ and $z_1 - z_0$, in nanometres for the coordinate block.

The denoiser is a transformer with no built-in rotational symmetry. The base model has no antigen frame and uses no rotation augmentation, and the field is learned in the centred frame of the input.

2.5.2 Block loss and independent block clocks

Block-normalised loss. The two blocks differ in size, with $3n$ coordinate numbers and $8n$ latent numbers. A plain sum of squares would let the wider block dominate. The loss divides each block's squared error by the block size, so each block contributes its mean squared error:

$$\mathcal{L}_{\text{block}} = \frac{\lambda_x}{d_x} \parallel r_x \parallel^2 + \frac{\lambda_z}{d_z} \parallel r_z \parallel^2.$$

Here r_x and r_z are the differences between predicted and target velocity in the coordinate and latent blocks, $d_x = 3n$ and $d_z = 8n$ are the block sizes, and $\lambda_x = \lambda_z = 1$. With $d_x = 60$, $\parallel$

$r_x \|^2 = 30$, $d_z = 160$ and $\| r_z \|^2 = 80$ the loss is $30/60 + 80/160 = 0.5 + 0.5 = 1$. In the code, the coordinate term of a complex is the masked sum of squared residuals (in nm²) divided by the number of present residues and by the block width 3. The latent term is the same sum over the latent block, divided by the number of present residues and by the block width 8. Padding is excluded. The update loss is the sum over complexes divided by the global batch of 48.

Independent block times. Coordinates and latent can move at different speeds, so each block has its own time. Training draws the coordinate time t_x and the latent time t_z independently and uniformly on $[0,1]$, once per complex. The published models draw them from mixtures of a uniform and a beta law that favour $t_x > t_z$: $0.02\,\mathrm{U}(0,1) + 0.98\,\mathrm{Beta}(1.9,1)$ for t_x and $0.02\,\mathrm{U}(0,1) + 0.98\,\mathrm{Beta}(1,1.5)$ for t_z [68, 71]. The base model does not use the mixture.

The interpolated state is $q_t = (x_{t_x}, z_{t_z})$ with $x_{t_x} = (1 - t_x)\,x_0 + t_x\,x_1$ and $z_{t_z} = (1 - t_z)\,z_0 + t_z\,z_1$. The network receives both times and returns a coordinate velocity v_θ^x and a latent velocity v_θ^z. Their targets are the end differences. The network therefore learns a field over the whole unit square of times (panel b of **Figure 3**).

Master clock. A sampler needs one curve through the square and one clock. Let $t_x(u)$ and $t_z(u)$ be non-decreasing, absolutely continuous schedules that equal 0 at $u = 0$ and 1 at $u = 1$. The heads give velocity per unit of their own time. The chain rule converts it to velocity per unit of the master clock u, and it is applied once:

$$\frac{\mathrm{d}q}{\mathrm{d}u} = f_\theta(q,u) = \begin{bmatrix} t_x'(u)\, v_\theta^x(q, t_x(u), t_z(u)) \\ t_z'(u)\, v_\theta^z(q, t_x(u), t_z(u)) \end{bmatrix}.$$

Training never applies these factors. For $t_x = 2u - u^2$ and $t_z = u^2$ at $u = 0.25$ the times are 0.4375 and 0.0625, and the slopes are 1.5 and 0.5. Head outputs of 4 and 2 then give master-clock velocities of 6 and 1.

The sampler of the full system compares three curves. The reference curve follows the exponential and quadratic schedules of the published sampler [68]: $t_x(u) = (1 - 10^{-2u})/(1 - 10^{-2})$ with $t_z(u) = u^2$. The equal clock sets $t_x = t_z = u$. The geometry-leading candidate is the pair in the worked example. Panel b of **Figure 3** draws all three on the square, and panel c contrasts training, which has both ends of the path, with sampling, which has only noise and the frozen field.

2.5.3 Autoencoder and decoder factorisation

The flow transports only the C-alpha coordinates and the latent, so a decoder must supply the rest. Let s be the residue sequence, $M_{\text{atom}}(s)$ the atom37 mask that marks which of the 37 atom slots exist for those residues, and y the coordinates of the non-C-alpha atoms. Let s_d be the designable residues and s^c the clamped ones copied from the context. The joint density follows the order in which the pieces are needed:

$$p(q, s, M_{\text{atom}}, y) = p_\theta(q)\ p_\phi(s_d \mid x, z)\ \delta_{s^c}\ \delta_{M_{\text{atom}}(s)}\ p_\phi(y \mid s, M_{\text{atom}}, x, z).$$

Read from left to right, the flow gives the state, and the decoder gives a probability for each designable residue. Clamped residues are copied. The atom mask is a fixed function of the sequence, so it is a point mass. The other atoms follow, given the sequence, the mask and the state. A serine and a tryptophan have different atoms, which fixes the order. The sequence factor uses counting measure and the coordinate factor uses Lebesgue measure on the present atoms. The base model has no context, so every residue is designable and the clamped factor is empty.

The base model implements the encoder and the decoder as two deterministic networks that train together (panel a of **Figure 4**). The encoder reads the 169 numbers of each residue packet (111 centred coordinates in nanometres, 37 mask entries and 21 category entries; Section 2.2) and returns the latent. The decoder reads the centred C-alpha position and the latent, 11 numbers in all. It returns 37 by 3 coordinates in nanometres, with the C-alpha slot included, and 21 residue logits. Neither network sees the chain index. The loss of one complex with n residues and N_a present atoms is

$$\mathcal{L}_{\text{AE}} = \frac{1}{3N_a} \sum_{(i,a)} \| \hat{y}_{ia} - y_{ia} \|^2 + \frac{1}{n} \sum_i \text{CE}\,(s_i, \ell_i) + 0.01\ \frac{1}{8n} \sum_i \| z_i \|^2.$$

The first term is the mean squared error over present atoms, which corresponds to a Gaussian density of fixed variance. It is in nm² per coordinate, and the reconstruction report converts its mean to a per-coordinate root-mean-square error in angstroms (Section 2.13.1). The second is the cross-entropy of the residue category, with logits ℓ_i. The third replaces the Kullback-Leibler term of a variational autoencoder. It keeps the latent from growing without bound, and its weight is 0.01. La-Proteina uses a Gaussian posterior with a Kullback-Leibler weight of 10^{-4} [68], which the base model does not.

Training uses the atoms that are present in the data record. The mask of a decoded sequence is needed only for generation. The encoder is frozen before the denoiser trains, and the denoiser stage reads the latent that the frozen encoder gives for each structure (Section 2.6).

2.5.4 Architecture and parameter counts

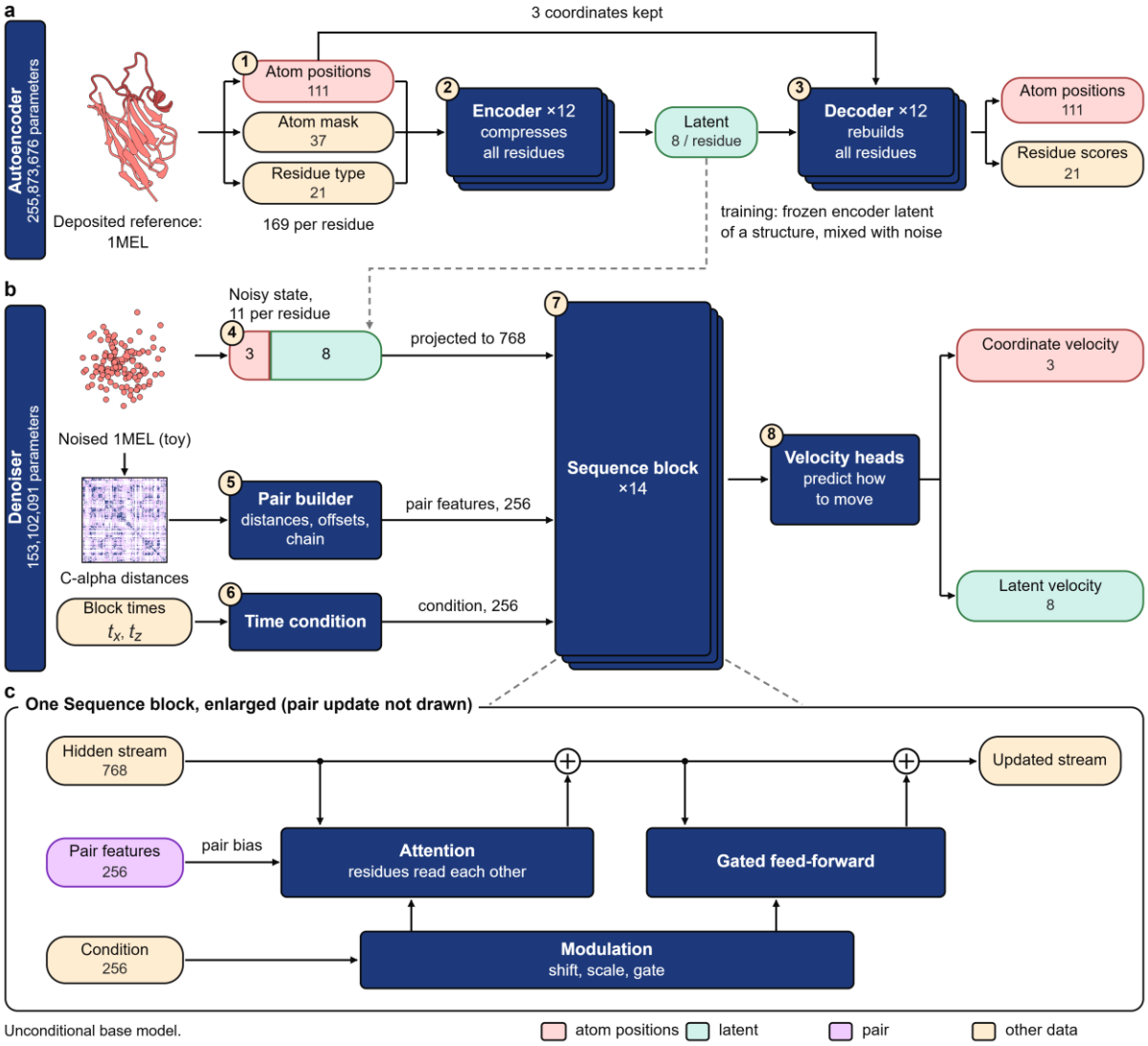

Figure 4. The base model is an autoencoder of 12 plus 12 blocks and a denoiser of 14 blocks built from one sequence block.

a, The autoencoder, 255,873,676 parameters. Each residue enters as 169 numbers: 111 atom positions, a 37-entry atom mask and 21 residue-type entries. Twelve encoder blocks compress them to a latent of 8 numbers per residue, and twelve decoder blocks rebuild the residue. The decoder reads that latent together with the centred C-alpha position, the 3 coordinates carried over the encoder on the bypass, and returns 111 atom positions and 21 residue scores. The encoder and the decoder read all residues in context, so the latent of one residue depends on the whole complex. The picture is the deposited VHH of PDB 1MEL chain A and stands at the input as a reference. **b,** The denoiser, 153,102,091 parameters. The noisy state holds 11 numbers per residue, 3 coordinates and 8 latent, and is projected to width 768; the picture shows the 3 coordinate numbers only. A pair builder sums a binned C-alpha distance, the separation along the chain and a same-chain flag into 256 pair features, and the tile is the C-alpha distance map of the frame above it. The two block times give a condition of width 256. Fourteen sequence blocks feed two velocity heads, which return 3 coordinate numbers and 8 latent numbers per residue. In training the latent part of the state is the latent of a structure from the frozen encoder, mixed with noise as $(1 - t_z)\, z_0 + t_z\, z_1$, where z_0 is the noise and z_1 is the encoder latent. **c,** One sequence block, enlarged. A hidden stream of width 768 passes through two residual additions. Attention reads the stream and takes the pair features as a bias, a gated feed-forward layer follows, and the condition drives both halves through a shift, a scale and a gate. The pair update is not drawn: in the reference configuration each of the 14 denoiser blocks updates the pair features before attention, and the encoder and the decoder use the block without the pair path. Both pictures are PDB 1MEL, the deposited chain in a and a frame of the toy interpolation along the straight training path at $t_x = 0.4375$ in b. The parameter counts are calculated from the module shapes of the specified networks, which are built without weights. The base model

is unconditional and takes no antigen. PDB, Protein Data Bank; VHH, variable domain of a heavy-chain-only antibody.

Block. The encoder, the decoder and the denoiser are stacks of one block (**Figure 4**a to c). The encoder and decoder have 12 blocks and the denoiser has 14. Every block has width 768 and 12 attention heads of width 64, and has a modulation layer, a self-attention layer and a gated feed-forward layer. A condition vector of width 256 passes through one linear layer to six vectors of width 768: a shift, a scale and a gate for the attention half, and the same three for the feed-forward half. The layout is the adaptive layer norm of diffusion transformers [129], with a sigmoid gate. In the denoiser the condition is the time embedding. The encoder and decoder have neither time nor antigen, so their condition is a learned constant vector of width 256, initialised at zero.

Feed-forward layer and masking. The feed-forward layer widens 768 to 6144, splits into a value and a gate of 3072 and returns to 768. The value half is multiplied by the sigmoid-linear unit (SiLU) of the gate half, in the pattern of gated linear units [130], and the hidden width is not reduced to equalise parameters. No network has dropout or triangle layers. Keys of padded residues are masked, and padded positions are zeroed after each residual update.

Inputs and outputs. Each network projects its per-residue input to width 768 and adds a sinusoidal encoding of the residue index (256 to 768). The encoder reads 169 numbers and returns the latent through a map from 768 to 8. The decoder reads 11 numbers, the centred C-alpha position and the latent, and returns 37 by 3 coordinates through a map from 768 to 111 and 21 logits. The denoiser reads the 11 numbers of the noisy state and returns the velocities through two linear heads, 768 to 3 and 768 to 8. Each time is multiplied by 1,000 and passes through a sinusoidal encoding of width 256. The two encodings are joined to width 512 and mapped by linear layers 512 to 256 to 256 to give the condition.

Pair representation. Only the denoiser has one. Its attention logits receive a pair bias, one linear map of the pair representation to 12 heads. In the encoder and decoder the bias is zero. The pair representation is the sum of three terms.

Pair features. Each C-alpha distance of the noisy coordinates is computed in nanometres and assigned to the nearest of 30 bins whose centres are defined at 1, 2, up to 30 Å, and mapped from 30 to 256 numbers. The code converts the 29 midpoints between the centres to nanometres once, so the bins keep their definition in angstroms, and a distance equal to a midpoint falls in the lower bin. The signed index difference, clipped to the range from -128 to 128, selects one of 258 embedding rows of width 128, and a linear map takes it to 256. Pairs in different chains

take the separate category 257. A two-row embedding of width 256 marks whether two residues share a chain. The published configuration also bins the distances of a predicted clean structure [71]. The base model leaves them out because they need a second network pass.

Pair update. A residual MLP with a layer norm, 256 to 512 to 256, updates the pair representation before the attention of its layer. The default updates it in all 14 layers. Variants update it in every second layer or never (**Table 6**), and with no update the pair features still bias attention in every layer. The use of pair-biased attention follows La-Proteina [68].

Precision. The trunk runs in BF16, and coordinates, losses, the optimiser and the averaged weights stay in FP32 (Section 2.6).

Parameter counts. Table 6 gives parameter counts obtained by building each module on the meta device, which allocates shapes and no storage, and summing the tensors. A block without a pair path holds 10,634,496 parameters. The pair bias adds 3,072 and each pair update adds 263,424. The feed-forward layers hold about two thirds of each network, attention about a fifth and modulation about a tenth. **Table 6** also names the layers that carry a pair update in each variant. The per-tensor table is in Supplementary Section S5.

Module	Pair updates	Parameters
Encoder	none	127,949,832
Decoder	none	127,923,844
Autoencoder (encoder plus decoder)	none	255,873,676
Denoiser, every layer (default)	all 14 layers	153,102,091
Denoiser, every second layer	layers 2, 4, 6, 8, 10, 12, 14	151,258,123
Denoiser, no pair update	none	149,414,155
Decoder plus denoiser (default)	all 14 layers	281,025,935

Table 6. Number of trainable parameters of the base-model networks.

The last row is the pair that a sampler loads, the decoder plus the denoiser. Updating the pair representation in every second layer removes 1,843,968 parameters from the default denoiser, and omitting the update removes 3,687,936.

The Proteina-Complexa denoiser has 159 M parameters and its autoencoder 256 M [71]. The denoiser here has 5,897,909 parameters, or 3.7 percent, fewer than 159 M, and the autoencoder is within 0.1 percent of 256 M. The remaining gap comes from details that the published tables do not give, such as the width of the pair MLP and the module list. La-Proteina reports about 160 M for its denoiser in its text [68]. A pair update adds few parameters but acts on a tensor of size L by L by 256, so its cost in time and memory cannot be read from the count (Section 2.6).

Flow density. For a continuously differentiable velocity field whose flow is a diffeomorphism, the log density of a state follows from the change of variables. Along a trajectory the log density changes at a rate equal to minus the divergence of the field, and integrating from the base state to the endpoint gives

$$\log p_1(q_1) = \log p_0(q_0) - \int_0^1 \operatorname{div} f_\theta(q_u, u)\, \mathrm{d}u.$$

The divergence is the trace of the Jacobian of the field. With the master clock it equals $t_x'(u)\,\mathrm{tr}(\partial v_\theta^x / \partial x) + t_z'(u)\,\mathrm{tr}(\partial v_\theta^z / \partial z)$, because the coupling terms between coordinates and latent lie off the diagonal. A Hutchinson estimator replaces the trace by the mean of $\xi^\top J \xi$ over random probe vectors ξ whose covariance is the identity [131]. A density in angstroms differs from one in standardised units by the determinant of the change of units, so a report states which one it gives. The base-model denoiser assigns distances to hard bins, so its field is not continuously differentiable in the coordinates, and the base model computes no likelihood. The derivations of the density formula, the clock rule and the change of units are in Supplementary Section S1.

2.6 Training of the base model

The base model is specified for training from random initialisation in two stages, one after the other. No released weight is loaded. In stage `ae`, the encoder and the decoder train together to turn each residue of a complex into eight latent numbers and to decode them into atoms. The encoder is then frozen. In stage `denoiser`, the velocity network learns the field of the flow over C-alpha coordinates and latents (Section 2.5) from the latents that the frozen encoder returns in evaluation mode. The decoder takes no part in the second stage, and no gradient reaches the encoder. Both stages read the same shards (Section 2.4.4). The model is unconditional: it has no antigen, hotspot or format input. **Table 8** gives the length buckets with their microbatch, accumulation steps and ranks, and **Table 9** gives one distributed update with the skip rule and the moving average.

Units and losses. Unit conversion follows Section 2.2. The autoencoder uses the objective of Section 2.5.3 and receives the true centred C-alpha positions. The denoiser uses the noise and interpolation of Section 2.5.1 and the block-normalised loss of Section 2.5.2, with independent uniform coordinate and latent times. Padding is excluded from both objectives. Reconstruction reporting follows Section 2.13.1.

Optimiser and schedule. Both stages use AdamW [132] with the settings of **Table 7**. With zero weight decay AdamW reduces to Adam. The learning rate is 10^{-4} after a linear warm-up: the update that follows k successful updates uses $10^{-4}\min\{1, (k+1)/500\}$, so the first update uses 2×10^{-7} and the 500th reaches 10^{-4}. No decay follows. Skipped updates do not advance the schedule. Gradients are clipped to a global 2-norm of 1.

Setting	Value
Optimiser	AdamW, betas 0.9 and 0.999, epsilon 10^{-8}, weight decay 0
Learning rate	10^{-4} after 500 warm-up updates, then constant
Gradient clipping	global 2-norm 1
Moving average	decay 0.999, first copy after step 1,000
Global batch	48 complexes per update, on two ranks
Loss weights	coordinate 1, latent 1, latent penalty 0.01
Block times	t_x, t_z independent, uniform on [0,1]
Checkpoint interval	1,000 updates

Table 7. Training settings shared by both stages.
The learning rate applies to each optimiser update. Loss weights and block times are dimensionless.

Moving average. A moving average of the parameters is kept for later use. Its decay is 0.999, so it averages over about 1,000 updates. The copy is taken at the first successful update whose step count exceeds 1,000, so that the average starts from weights that have left the warm-up. After the copy, each successful update sets the average to 0.999 times itself plus 0.001 times the current parameters. Only parameters are averaged, and the average is stored in every checkpoint.

Length buckets and batch composition. Complexes fall into three disjoint length buckets, up to 300, 301 to 400 and 401 to 500 residues. Each bucket is padded to its upper limit and masked (Section 2.4.4). Every update uses 48 complexes from one bucket. Two ranks share the work, each with a microbatch m and a accumulation steps, so that $2ma = 48$ (**Table 8**). The code derives a as $48/(\text{ranks} \times m)$, so a single process doubles it and still uses 48 complexes. A bucket is drawn uniformly at random for each update. All ranks obtain the same bucket and the same 48 indices, drawn with replacement inside the bucket, and each rank takes its own slices.

Bucket (residues)	Padded length	Microbatch per rank	Accumulation steps	Ranks	Complexes per update
up to 300	300	8	3	2	48
301 to 400	400	6	4	2	48
401 to 500	500	4	6	2	48

Table 8. Batch composition by length bucket.
Lengths are in residues; the microbatch is counted in complexes per rank and per accumulation step.

Distributed update. The loss of a microstep is the sum of the per-complex losses divided by 48. Summed over microsteps and ranks, it is the mean over the 48 complexes of the update, whatever the bucket. The data-parallel wrapper of PyTorch averages gradients over ranks by default [133], which would halve them under this scale. A communication hook therefore replaces the average by an FP32 sum and refuses a buffer of any other type. Only the last microstep exchanges gradients. The earlier ones run without synchronisation and accumulate locally.

Non-finite gradients. After the last microstep the gradient is clipped, and every rank checks that its norm and every entry are finite. The ranks share one flag, the minimum over ranks. If it is false, gradients are cleared, and the weights, optimiser state, moving average, schedule and data cursor stay as they were. The random state saved at the start of the update is restored, so the retry uses the same batch and the same noise. Three consecutive skips stop the run. **Table 9** states one update in order.

Step	Rule
1	Draw one length bucket and 48 complex indices, the same on both ranks, and save the random state
2	For each microstep, the loss is the sum of the per-complex losses divided by 48; only the last microstep exchanges gradients, and the earlier ones accumulate without synchronisation
3	Clip the gradient to a global 2-norm of 1
4	Each rank sets a flag from the finiteness of the norm and of every gradient entry; the ranks share the minimum of the flags
5	If the flag is false, clear the gradients, restore the saved random state, advance no counter and add one to the skip count; three consecutive skips stop the run
6	Otherwise take the optimiser step, add one to the step count, advance the warm-up schedule and reset the skip count
7	If the step count exceeds 1,000, copy the parameters at the first such update, and afterwards set the average to 0.999 times itself plus 0.001 times the parameters
8	Write a checkpoint every 1,000 successful updates and at the last update

Table 9. One distributed update of either stage, in order.
A skipped update advances neither the step count, the schedule, the moving average nor the data cursor. The rules are those of the training code and of the cycle-1 parameter lock.

Mixed precision. Trunk matrix products and attention run in BF16 under autocast. BF16 has the exponent range of FP32 [134]. The specified configuration uses BF16 trunk arithmetic without loss scaling and retains the stated optimiser settings. Attention softmax is accumulated in FP32 inside the kernel, and the attention bias is computed in FP32. BF16 keeps 8 significant bits, so a coordinate of 3 nm would be rounded on a grid of 0.016 nm, or 0.16 Å, and for this

reason no coordinate enters BF16. Parameters, accumulated gradients, optimiser state, the moving average and both residual streams stay in FP32. So do layer normalisation, centring, unit conversion, noise, interpolation, distances, the output heads, the losses, the latent penalty, clipping and the gradient exchange between ranks. TF32 is off.

Activation checkpointing. Activation checkpointing recomputes the intermediate tensors of each layer in the backward pass, at the cost of one extra forward pass [135, 136]. It is applied to every layer of the trunks.

Latents and seeds. The frozen encoder supplies the latent of each complex from the canonical input, with no rotation and no crop before encoding, because the encoder reads the whole complex. A cache of these latents may replace the live encoder pass, and it is valid only under that condition. Every random draw derives from a SHA-256 hash of the run identifier, the rank and the update index (Section 2.14).

Run record, checkpoints and resume. Each base-model training run records its data manifest and configuration and writes checkpoints and a training log. A checkpoint is written after a completed optimiser, moving-average and scheduler step, every 1,000 updates and at the last update. It holds the model, optimiser state, moving average, scheduler, step, data cursor and the random state of every rank, and the model file is renamed last, as the commit marker. Resume restores all of these and requires the same stage, configuration, device type and number of ranks.

Number of updates. The number of successful updates of stage s follows from the measured update time. For an allocated stage time T_s it is

$$N_s = \left\lfloor \frac{T_s}{\bar{t}_s} \right\rfloor, \qquad \bar{t}_s = \frac{1}{3}\sum_b t_{s,b} + \frac{c_s}{I}.$$

Here $t_{s,b}$ is the measured time of one complete update of stage s in length bucket b on the two cards, the sum runs over the three buckets, c_s is the time of one checkpoint write and $I = 1{,}000$ is the checkpoint interval. Compute each bucket mean from complete-update wall times after subtracting the corresponding `checkpoint_ms`, converting milliseconds to seconds. The supplied timing-summary means include checkpoint writes and must not be used directly. The term c_s/I then adds the amortised checkpoint cost once. N_s is calculated before launch and supplied to the training script through the option `--updates`. A skipped update does not advance the count. We estimate the matrix-product work of the checkpointed denoiser blocks as four times their forward work: a forward pass, a backward pass of twice that cost and one recomputed forward pass. Including the modulation layers, this is about 3.0, 5.3 and 8.1 TFLOP for one complex at

lengths 300, 400 and 500, and it excludes the forward pass of the frozen encoder (derivation in Supplementary Section S5). The pair update takes 88 to 92 percent of the forward arithmetic and only 2.4 percent of the parameters, so the measured time, and not the parameter count, sets the number of updates.

2.7 Antigen conditioning and Interface-JEPA

The full system adds an antigen channel and a predictive critic, Interface-JEPA, to the base model. The base model takes no antigen, hotspot or format token as input. Where a value comes from a published configuration it serves as a reference value for a model trained from scratch, and no released weight or code is loaded. Supplementary Figure S7 shows the contacts of one deposited complex under the shared contact rule (panel a) and the training scheme of Interface-JEPA (b).

Conditioning contract. The antigen channel tells the generator where an antibody may touch the antigen and which chemistry the touching residues offer. Let A be the antigen structure, E the requested epitope (a set of antigen residues), F the fixed antibody context and X the generated antibody state. Let C_{pred} be a predicted soft contact plan, with weights in [0,1] between movable antibody residues and epitope residues (loop residues with a fixed framework, the whole variable domain with a free one). Let $\chi_E(A)$ be the chemical features of the epitope residues. The model generates through the plan, in the context-anchored coordinate profile of Section 2.2. Only the movable antibody residues are transported, and the rest of the antibody and the antigen are conditioning data:

$$p_\theta(X \mid A, E, F) = \int p_\theta\big(X \mid C_{\mathrm{pred}},\, C_{\mathrm{pred}} \odot \chi_E(A),\, F\big)\, p_\theta(C_{\mathrm{pred}} \mid A, E, F)\, \mathrm{d}C_{\mathrm{pred}}.$$

The symbol $\odot$ stands for a plan-gated message, $g_i = \sum_{j \in E} C_{\mathrm{pred},ij}\ \chi_j(A)$ for antibody position i. A contact that the plan proposes passes the charge, polarity and hydrogen-bonding geometry of the epitope residue to the decoder. A global antigen embedding without this gate is excluded, because the identity of the target could then bypass the plan.

Four antigen schemas are specified: protein epitope, peptide epitope, glycoprotein epitope (usable after gate F-GLY passes) and small molecule (which needs an atom-and-bond schema and a ligand-bearing validation set). An epitope record carries one provenance type: experimental contact, experimental mapping, computational pose, predicted patch or user-declared patch. The types stay separate in training and evaluation. A report finds that baseline CDR generators return nearly identical CDRs whatever the target [95], so gate F-CB compares

generation under the true antigen with generation under a decoy, with the antigen removed and with a renumbered copy. A separate complex predictor scores each output against the true antigen.

Residue encoder and cross-attention. Each epitope residue becomes one token c_j that a rigid motion of the antigen leaves unchanged. The token omits deposited residue numbers, which shift under harmless renumbering and can encode target identity. It is built from amino-acid identity, chain and signed within-chain offset. For each of the atom slots N, C, O and C-beta it adds a presence mask, a radial-basis encoding of the distance to C-alpha (in units of the coordinate scale ℓ_0) and the unit direction to C-alpha in the local residue frame. A masked sum over the slots and a linear map give c_j. The local frame uses N, C-alpha and C, with fallbacks at peptide termini and for glycine, and flags for the fallbacks are embedded in the token.

Antibody residues read the tokens by cross-attention in an epitope-centred frame whose origin is the mean C-alpha position of E. Let $\bar{x}_i$ and r_j be the scaled coordinates of antibody residue i and epitope residue j, q_i a query, k_j a key, d_h the head width and ϵ_C a small constant. The weight is

$$\alpha_{ij} = \mathrm{softmax}_{j\in E}\left(q_i^{\top}k_j/\sqrt{d_h} + b_{\mathrm{geo}}(\left\|\bar{x}_i - r_j\right\|_2) + \log(C_{\mathrm{pred},ij} + \epsilon_C)\right).$$

The first term compares content, the second is a bias that depends on distance, and the third lets the plan raise or lower a pair. Pair distances are unchanged by a joint rotation and translation, and gate F-CB checks this together with invariance to renumbering. The geometric bias is ablated by master-clock interval (Section 2.5), because it may carry little signal near noise.

Contact-plan objective. A solver may refine the learned output C_{pred}, starting from $\Gamma^{(0)}$ and returning an accepted plan $\Gamma^{\star}$. The plan $\Gamma^{\star}$ minimises an unbalanced transport objective. Let μ_R and μ_E be normalised exposure weights on movable antibody positions and epitope residues, and let $C = C^{\mathrm{geom}} + \beta_{\mathrm{chem}}C^{\mathrm{chem}}$ be the dimensionless ground cost, with a geometric and a chemical term. Then

$$\Gamma^{\star} = \underset{\Gamma\geq 0}{\mathrm{argmin}}\ \langle\Gamma, C\rangle + \rho_{\mathrm{OT}}\mathrm{KL}(\Gamma\mathbf{1} \mid \mu_R) + \rho_{\mathrm{OT}}\mathrm{KL}(\Gamma^{\top}\mathbf{1} \mid \mu_E) + \varepsilon_{\mathrm{OT}}\mathrm{KL}(\Gamma \mid \mu_R \otimes \mu_E).$$

The first term charges transport cost. The two middle terms charge mass created or destroyed at the margins, so part of a declared patch may stay unmatched. The last term smooths the plan and makes the minimiser unique. KL is the generalised divergence for positive vectors of

unequal mass. The solver works in the log domain [137, 138]. For this unbalanced family, Sinkhorn divergences remove the entropic bias when two measures are compared [139].

For $\varepsilon_{\mathrm{OT}} > 0$, $\rho_{\mathrm{OT}} > 0$, finite C and strictly positive marginals, the minimiser exists and is unique. The smoothing changes the optimal value by at most $\varepsilon_{\mathrm{OT}}\mathrm{KL}(\Gamma_0^\star \mid \mu_R \otimes \mu_E)$, where $\Gamma_0^\star$ minimises the objective at $\varepsilon_{\mathrm{OT}} = 0$ (Theorem 4.2 and Proposition 4.5 of Supplementary Section S1).

Fidelity controls. Two training controls address a shortcut in which the generator copies framework context and ignores the antigen. Framework dropout zeroes the input features of a framework residue with probability p_{drop}, except at positions that the format profile protects. For a VHH, the protected set may include the FR2 positions that formerly met the light chain. Framework masks use the retained ANARCII numbering map described in Section 2.3.1. The probability is chosen by a sweep that includes 0.3.

Same-antigen patch contrast adds a retrieval loss. It asks the model to tell the requested patch from non-overlapping patches of matched area and accessibility on the same antigen:

$$\mathcal{L}_{\mathrm{patch}} = -\log \frac{\exp(r^\top e_+/\tau)}{\sum_{k=1}^{K} \exp\,(r^\top e_k/\tau)}.$$

Here r summarises the continuous decoder logits, e_k embeds patch k among K patches that include the requested patch e_+, and τ is a temperature. Gate F-CYC reports retrieval accuracy by antigen-fold cluster, patch area, accessibility, CDR-H3 length and format, and the loss is active once the gate accepts.

The six design inputs. Six design inputs act through generation, steering and selection: the antigen, its topology, its physicochemistry, its surroundings, the shape of the paratope and the regime of the self-proteome panel. The first four enter the generator as one antigen description. The antigen is given as atom37 coordinates, amino-acid identity and a hotspot indicator, as in the published recipe [71]. Topology enters through pair features and the frame features of the residue token, and physicochemistry through amino-acid identity, atom37 features and the plan-gated chemistry. The surroundings are the antigen residues inside the crop, and hotspot tokens mark the epitope. The crop holds at most 500 residues in total and at least 50 antigen residues, with candidates within 15 Å of the antibody taken in contiguous stretches [71]. For each example, hotspots are drawn uniformly from 3 to 12 interface residues (heavy-atom contact within 4.5 Å), or all of them with probability 0.2.

The paratope shape is set by the fixed framework clamped as context, or by a free framework, together with a CDR length prior. The prior comes from the empirical nanobody distribution in SAbDab by IMGT numbering, from the 5th to the 95th percentile with add-one smoothing [22]. The self-proteome regime is the declared panel. It acts through selection and through negative potentials in steering, and the generator has no tensor for it. Global translation noise with a spread of 0.2 nm is added to the antibody coordinates in training [71]. This equals the spread of the shared translation in the base-model noise (0.2 nm, 2 Å). A format token selects the format profile. VHH de novo is the first admitted format, and the paired Fv format follows once its gates pass.

Epitope registration. Registration measures whether a decoded pose occupies the requested patch. The contacted antigen residues $\mathcal{C}$ are those with a heavy atom within $d_c = 4.5$ Å of a heavy atom of the full antibody variable domain R_V, framework included. Precision, coverage and their harmonic mean are

$$P_E = \frac{|\mathcal{C} \cap E|}{|\mathcal{C}|}, \qquad R_E = \frac{|\mathcal{C} \cap E|}{|E|}, \qquad \mathrm{EpiF1} = \frac{2P_E R_E}{P_E + R_E},$$

and each is zero when its denominator is zero. The scope is the full domain because framework residues often contact antigen in single-domain antibodies. In 345 non-redundant single-domain complexes, IMGT position 66 lies in 50.4 percent of interaction-defined paratopes, and positions 52, 55 and 42 in 31.6, 27.2 and 24.1 percent [8]. A loop-only footprint can therefore hide a miss. Take an epitope of ten residues that loop contacts cover completely, with framework contacts on ten further antigen residues outside it. The loop-only footprint scores $P_E = R_E = \mathrm{EpiF1} = 1$, and the full-domain footprint scores $P_E = 0.5$ and $\mathrm{EpiF1} = 2/3$. The loop-only value is a diagnostic and is not the admission criterion.

Training labels ($Y_{ij} = 1$ when antibody residue i and antigen residue j lie within d_c), the contact target of Interface-JEPA and gate F-EPI share one residue scope, one atom mask and one cutoff. The metrics describe a structure proposal and become native-pose metrics when E and the reference geometry come from an experimental complex. Gate F-EPI reports each provenance stratum separately.

Glycan-aware accessibility. A glycan can hide an epitope. A declared set $\mathcal{H}$ of glycosylation scenarios fixes the protein conformer, occupied sites, glycoforms, glycan library and a seed. The accessibility $\sigma_j^{(h)}$ of epitope residue j in scenario h is the fraction of grafted glycan

conformers through which a probe still reaches the residue. Grafting follows GlycoSHIELD [13]. The envelope over scenarios is $\underline{\sigma}_j = \min_{h\in\mathcal{H}} \sigma_j^{(h)}$ and $\overline{\sigma}_j = \max_{h\in\mathcal{H}} \sigma_j^{(h)}$.

The antigen token carries a central value, the interval width and a support mask, and the gate reads the lower envelope over required residues. Scenarios on both sides of a floor leave the accessibility unresolved. Until gate F-GLY accepts, the feature is descriptive. The scenarios cover antigen glycosylation only, so a binder with an N-glycan sequon is unresolved for glycan effects (Section 2.11).

2.7.1 Interface-JEPA

Interface-JEPA (joint-embedding predictive architecture) is a predictive representation layer for antibody-antigen interfaces. It reconstructs masked interface targets in latent space and exposes four typed heads. It acts as critic and conditioner and generates no molecules. Following I-JEPA [140], an online context encoder and a predictor see the visible context. A target encoder, an exponential moving average of the online encoder, sees the masked native target and passes no gradient. A structure-aware joint-embedding predictive architecture has been combined with diffusion to reconstruct antibody CDR loops [141].

Targets. The contact target holds, for antibody residue i and antigen residue j, the contact indicator y_{ij}, the clipped minimum distance over ℓ_0, a distance-bin vector, orientation features and the exposure of each residue. Here ℓ_0 is the length scale of the full system, which differs from the base-model factor of 0.1 nm per ångström. The set V_B holds the antibody residues of the labelled complex, taken over the full variable domain R_V and not the loops alone, and V_A holds its antigen residues. The target has shape $|V_B| \times |V_A| \times (4 + B_d + B_o)$ for B_d distance bins and B_o orientation channels. A complex of 120 antibody and 180 antigen residues gives 21,600 pairs. The complex target adds both residue tensors and the relative pose after antigen-frame anchoring.

Objective. The target parameters $\bar{\theta}$ follow the online parameters θ by $\bar{\theta}_{k+1} = \mu_k \bar{\theta}_k + (1 - \mu_k)\theta_k$, where k counts successful optimiser updates. A step skipped for a non-finite gradient advances neither k nor $\bar{\theta}$, and all ranks share that decision. The target starts as a copy of the online network, projector included. The momentum μ_k rises linearly from 0.996 to 1.0 over the training steps, the schedule reported for I-JEPA [140]. For masked units $r \in \mathcal{M}$ with predictions $\hat{h}_r$ and target latents h_r, the main loss is

$$\mathcal{L}_{\text{pred}} = \frac{1}{|\mathcal{M}|} \sum_{r \in \mathcal{M}} \| \operatorname{norm}(\hat{h}_r) - \operatorname{stopgrad}(\operatorname{norm}(h_r)) \|_2^2,$$

where $\operatorname{norm}(x) = x/\max(\|x\|_2, \epsilon_n)$ with $\epsilon_n = 10^{-6}$. Anchor terms keep the information that the heads need: a clipped binary cross-entropy on contacts, a Huber loss on distances and a residue term, with weights λ_c, λ_d and λ_r. The clip uses $\epsilon_b = 10^{-6}$, because the excess-risk bound below needs a bounded loss. The full objective is $\mathcal{L}_{\text{JEPA}} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{anchor}} + \lambda_v \mathcal{L}_{\text{var}} + \lambda_{\text{cov}} \mathcal{L}_{\text{cov}}$, with the last two terms defined below.

Heads. The four heads are contact, complex, comparative maturation and generic polyspecificity. The contact head predicts the contact indicator, distance bins and orientation, and is calibrated pairwise by antigen cluster and label provenance. The complex head scores a proposed complex and is judged by ranking within an antigen and by cognate discrimination. The comparative maturation head orders two variants from one assay context by $\Pr(a \succ b) = \sigma(r_\eta(a) - r_\eta(b))$, with logistic σ and scalar score r_η [142]. The generic polyspecificity head predicts the assay response to a polyspecificity reagent, and an isotonic regression fitted after freezing calibrates it. Binding probability and affinity belong to separate assay heads and are not outputs of this layer. Each head has its own target schema, calibration cohort and gate.

Data are split into P_{hist} for training and selection, P_{cal} for calibration, and P_{gen}, an untouched cohort for a frozen generator. The calibration population is divided into a map-fitting subset and a risk-calibration subset (Section 2.10.2). A head is admitted when its gates hold. Gates F-J1a, F-J1b and F-J1c compare one cognate antigen with ten nested decoys on at least 200 independent designs, and all three must hold.

Anti-collapse terms. A latent predictor can collapse to a constant. Two terms act on a batch of B raw projection outputs $z \in \mathbb{R}^d$ in FP32, before normalisation, because the trace of normalised vectors is at most $B/(B-1)$ and cannot reach the covariance margin. The variance term $\mathcal{L}_{\text{var}}$ is a hinge that pushes the standard deviation of each dimension, computed with a small constant δ under the square root, to at least τ. The covariance term $\mathcal{L}_{\text{cov}}$ penalises the squared off-diagonal entries of the batch covariance. Both follow VICReg [143], whose target standard deviation is 1.

Two properties are stated for these terms. First (Theorem 6.1 of Supplementary Section S1), if $\mathcal{L}_{\text{var}} \le \eta$, $\mathcal{L}_{\text{cov}} \le \kappa$ and $B \ge d + 1$, the smallest eigenvalue of the batch covariance $\hat{\Sigma}$ satisfies $\lambda_{\min}(\hat{\Sigma}) \ge (\tau - d\eta)_+^2 - \delta - \sqrt{(d-1)\kappa}$, where $(x)_+ = \max(x, 0)$. The floor constrains the minimum per-dimension variance, so gate F-J3 reads the minimum and not the median. Second

(Theorem 6.3), for a loss bounded by M, the optimal risk of a predictor of a label Y that sees only the representation h of an input X exceeds the optimal risk given X by at most $M\sqrt{2\,I(X;Y \mid h)}$, with I the conditional mutual information. A small prediction loss does not bound this information (Example 6.4), so the anchor terms provide the empirical handle on it.

The design values are in **Table 10**. The weights follow the VICReg defaults of 25, 25 and 1 once the prediction weight is set to 1. The vector counts equal $1 + (d-1)^2/0.80982$ at $\tau = 1$, $\delta = 10^{-4}$ and margin 0.1. At $d = 32$ the gate requires $(1-32\eta)_+^2 - 10^{-4} - \sqrt{31\kappa} \geq 0.1$. The two ceilings in **Table 10** are one-at-a-time limits, each with the other penalty at zero. A pair of values is checked against the joint requirement. Gate F-J3 evaluates the bounds on a fixed random projection of the head input to 32 dimensions, on at least 200 untouched complexes, with one vector per interface residue, equal weight per complex and a cluster bootstrap over complexes and antigen families.

Item	Value
Representation width read by the heads	256
Width d of the regularised projection	32
Variance floor τ; constant δ	1; 10^{-4}
Weights of $\mathcal{L}_{\mathrm{var}}$ and $\mathcal{L}_{\mathrm{cov}}$ relative to $\mathcal{L}_{\mathrm{pred}}$	1 and 0.04
Anchor weights $\lambda_c, \lambda_d, \lambda_r$	each from 0.25, 0.5, 1, 2
Momentum schedule	0.996 to 1.0, linear
Vectors needed at $d = 16,32,64,256$	279; 1,188; 4,903; 80,297
One-at-a-time ceilings at $d = 32$, margin 0.1	$\eta^{\max} = (1-\sqrt{0.1001})/32 = 0.02136$; $\kappa^{\max} = 0.8999^2/31 = 0.02612$

Table 10. Design values of Interface-JEPA: widths in numbers of channels, loss weights and variance floor dimensionless, vector counts in numbers of independent unit-variance Gaussian vectors, ceilings dimensionless. The anchor weights are chosen on P_{hist}.

2.8 Guidance and sequential Monte Carlo

The full system steers the sampler towards a declared utility in two ways. Guidance moves each trajectory along a preconditioned utility direction, and sequential Monte Carlo (SMC) weights and resamples a population of trajectories. The base model includes neither guidance nor a generation sampler. Supplementary Figure S3 shows a guided run in five moves (panel a), a two-objective example (b), one step judged by its effect on the built molecule (c) and the weights with one resampling draw (d).

Stochastic interpolants and the score. Steering needs a stochastic sampler and a score. A stochastic interpolant adds a Gaussian bridge term to a deterministic path from a base state $\bar{q}_0$ to a data state $\bar{q}_1$:

$$\bar{q}_u = \mathcal{I}(u, \bar{q}_0, \bar{q}_1) + \gamma(u)\,\zeta, \qquad \zeta \sim \mathcal{N}(0, I_D).$$

Here $\mathcal{I}$ is the deterministic interpolation, $\gamma(u)$ is the noise width, zero at both ends and positive between them, and D is the dimension of the packed state of Section 2.2. The score of the density p_u is the average hidden noise divided by the width, with its sign reversed:

$$\nabla_q \log p_u(q) = -\frac{1}{\gamma(u)}\,\mathbb{E}[\zeta \mid \bar{q}_u = q], \qquad 0 < u < 1.$$

For any diffusion coefficient $\epsilon(u) \geq 0$, adding $\epsilon\,\nabla \log p_u$ to the velocity and injecting noise $\sqrt{2\epsilon}\,\mathrm{d}W_u$ keeps the same one-time marginals [59]. Velocity and score must belong to the same interpolant. Under independent pairing of base and data states the score follows from the velocity heads, so no score head is trained. The published La-Proteina sampler integrates this equation by the Euler-Maruyama method in 400 steps, with a noise scale of 0.1 for both blocks [68]. The design takes these values as starting settings and registers one bridge-noise profile and one diffusion profile per block, because one scalar does not serve both clocks.

Clean-endpoint utilities. A property model needs a nearly finished molecule, and a state in mid-path is far from one. Every utility therefore acts on an estimate of the clean endpoint. For the linear block paths of the base model the estimate adds the rest of the velocity:

$$\hat{x}_1 = x_u + [1 - t_x(u)]\,v_\theta^x, \qquad \hat{z}_1 = z_u + [1 - t_z(u)]\,v_\theta^z.$$

Here x_u and z_u are the coordinate and latent blocks at master-clock progress u, $t_x(u)$ and $t_z(u)$ are the schedules of Section 2.5.2, and v_θ^x, v_θ^z are the velocity heads. A frozen decoder maps $\hat{q}_1 = (\hat{x}_1, \hat{z}_1)$ to soft decoded features $h_\phi(\hat{q}_1, c_{\text{gen}})$, and a utility head $\widetilde{U}_j$ scores them:

$$U_j = \widetilde{U}_j\big(h_\phi(\hat{q}_1, c_{\text{gen}}),\, c_{\text{guide}}\big).$$

The condition c_{gen} is the generation condition, and c_{guide} names the admitted heads, gains, negative-design panel members, feature weights and trust limits. Each head is trained on such estimates at the corruption times of the generator and takes the time as input, so its gradient exists at intermediate states. The gradient is taken with the denoiser output held fixed, so a step needs no backward pass through the denoiser.

Each head declares an endpoint type, for example a contact prediction, a pose score or a generic polyspecificity response. Scores combine only within one type, unit, assay mapping and target condition. A named off-target needs its own calibrated pair oracle, and without one the term is unavailable. Hard chemical, scaffold and assembly rules stay outside the optimiser as projections or rejections. The pose and polyspecificity heads of Interface-JEPA (Section 2.7.1) are candidate utilities. The guided field adds a preconditioned utility direction to the learned master-time field f_θ:

$$\frac{\mathrm{d}\bar{q}}{\mathrm{d}u} = f_\theta(\bar{q}, u, c_{\mathrm{gen}}) + \kappa(u; c_{\mathrm{guide}})\, G_\phi(\bar{q})^{-1}\, \nabla_{\bar{q}} U.$$

The gain $\kappa(u)$ and the utility U are dimensionless, larger U is preferred, and G_ϕ is the decoder metric defined next.

Decoder-feature metric and trust region. Equal latent steps can change the decoded molecule by unequal amounts, so the design measures a step by its effect on decoded features (Supplementary Figure S3c). A soft-token adapter $h_\phi\colon \mathbb{R}^D \to \mathbb{R}^{D_h}$ replaces each designable residue by the probability-weighted mixture of the 20 amino acids and runs the decoder once. For n_m assembled residues it returns $D_h = 131\, n_m$ features: per residue 3 for the C_α position, 108 for the other atom means and 20 for the residue probabilities. Let $J_h = \partial h_\phi / \partial \bar{q}$ be its Jacobian, W_h a symmetric non-negative feature weight matrix set by the steering task, and $\tau_{\mathrm{ridge}} > 0$ a ridge. The metric is

$$G_\phi = J_h^\top W_h J_h + \tau_{\mathrm{ridge}}\, I_D.$$

A smooth decoder induces the pullback $J_h^\top J_h$ as a Riemannian metric on the latent space [144]. The ridge gives $v^\top G_\phi v \geq \tau_{\mathrm{ridge}} \|v\|^2 > 0$, so G_ϕ is positive definite and the solve $G_\phi d = \nabla U$ is defined.

The cap bounds the guided increment in the local decoder-feature metric. For solver step h, gain κ and ascent direction $d = G_\phi^{-1} \nabla U$, the guided increment is $\Delta_{\mathrm{guide}} = h\kappa\alpha d$ with

$$\alpha = \min\left(1,\ \frac{\Delta_G}{h\kappa\sqrt{d^\top G_\phi d + \epsilon}}\right),$$

where ϵ is a small constant. The cap holds the metric norm of the increment at or below the trust radius Δ_G, so the increment stays small in that metric even when the inverse is ill-

conditioned. Acceptance additionally uses checks on the hard-decoded candidate, and a failed molecular check rejects the step.

Several objectives. A fixed weighted sum can push against a conflicting objective. The multiple-gradient descent algorithm (MGDA) takes the weights from the gradients themselves [145, 146]. Guided diffusion samplers have used the same Pareto idea for multi-objective generation [147]. Write each criterion as a scaled loss $\tilde{f}_i = (f_i - b_i)/r_i$ with registered baseline b_i and scale $r_i > 0$, where a utility enters as its negative. Let $g_i = \nabla \tilde{f}_i$ and $a_i = G_\phi^{-1} g_i$. The weights solve a small quadratic programme over the simplex Δ_{n_f} of n_f non-negative weights that sum to one:

$$\lambda^\star = \arg\min_{\lambda \in \Delta_{n_f}} \| \sum_i \lambda_i a_i \|_{G_\phi}^2, \qquad w = \sum_i \lambda_i^\star a_i.$$

This projects the origin onto the convex hull of the preconditioned gradients in the metric G_ϕ. For every i, $g_i^\top w \geq \|w\|_{G_\phi}^2$, so $-w$ lowers every scaled loss to first order when $w \neq 0$, and $w = 0$ marks a state that is Pareto-stationary to first order [145]. As a check of this property, $G = I$, $g_1 = (1,0)$ and $g_2 = (0,2)$ give the weights $(0.8,0.2)$ and $w = (0.8,0.4)$, and both inner products equal $\|w\|^2 = 0.8$ (Supplementary Figure S3b).

A finite step is accepted when every scaled loss falls by at least $c_\mathrm{A} h \|w\|_{G_\phi}^2$ for a constant $0 < c_\mathrm{A} < 1$ (Armijo condition) and $h\|w\|_{G_\phi} \leq \Delta_G$. Backtracking shortens h until both hold. The weights change with the state, so the combined field is a controlled vector field and not the gradient of one potential. Gate F-MN compares min-norm steering with a fixed-weight sum at matched seeds and compute before activation.

2.8.1 Sequential Monte Carlo

Guidance moves particles without correcting the law that results. SMC runs N_p trajectories together, weights each move by how far the proposal differs from the base process, and resamples when a few weights dominate [148]. The design uses SMC for the stochastic profile only, because two deterministic updates are point masses on different states and have no density ratio. The deterministic guided field (the controlled-ODE pushforward) has no correctness proposition.

Choose checkpoints $0 = u_0 < \cdots < u_M = 1$. Let K_j be the base transition kernel, R_j the proposal kernel and $\Psi_j \geq 0$ a potential that carries the preference. The target is the base path

law re-weighted by the product of the potentials, with normaliser Z_M. The twisted construction follows Wu and colleagues [149]. The incremental weight is

$$\omega_j(q_{j-1}, q_j) = \Psi_j(q_{j-1}, q_j)\, \frac{\mathrm{dK}_j(q_{j-1},\cdot)}{\mathrm{dR}_j(q_{j-1},\cdot)}(q_j).$$

The ratio exists when $\mathsf{K}_j(q,\cdot)$ is absolutely continuous with respect to $\mathsf{R}_j(q,\cdot)$. Every move the base kernel can make must be open to the proposal, because no finite weight restores missing support (Supplementary Figure S3a, move 2).

First configuration. The base kernel serves as proposal, so $\omega_j = \Psi_j$ for $j \geq 1$, and $\Psi_0 \equiv 1$. Rewards are evaluated at steps 100, 200, 300 and 400 of 400, so $M = 4$. Set $R_0 = 0$ and use $\Psi_j = \exp\{\beta(R_j - R_{j-1})\}$, where R_j is the interface confidence of the complex-prediction judge (Section 2.10) at check j, computed on the clean-endpoint estimate of the state, and β is an inverse temperature. The product telescopes to $\exp\{\beta(R_M - R_0)\} = \exp\{\beta R_M\}$, a tilt by the final reward, as in recent Feynman-Kac and particle-steered generative samplers [150–154] and in Proteina-Complexa, whose reward is an interface score from a structure predictor [71]. The settings are $N_p \in \{8,16,32\}$ and $\beta \in \{0.5,1,2\}$, chosen on a validation set before any evaluation set is read. Guided proposals can be added later, and the support condition is then checked for each configuration. The normaliser Z_M belongs to this fixed, discrete, reduced-noise model.

Weights and resampling. Weights are held as logarithms and normalised by a shifted log-sum-exp, which stays finite where the exponentials underflow. Missing support, all log weights equal to $-\infty$ and a NaN or $+\infty$ are failures, whereas finite log weights never abort a run. The effective sample size $\mathrm{ESS} = 1/\sum_k (\bar{w}^{(k)})^2$ lies between 1 and N_p (Supplementary Figure S3d). Resampling is stratified, runs when ESS falls below $N_p/2$ and is skipped at the last step. The ancestor indices, the weights before and after resampling, the ESS and the log normaliser are stored.

Estimator. The normaliser is estimated by the product, over the steps, of the mean incremental weight:

$$\hat{Z}_M = \prod_{j=1}^{M} \bar{\ell}_j\,, \qquad \bar{\ell}_j = \sum_{k=1}^{N_p} \widetilde{w}_{j-1}^{(k)}\; \omega_j\Big(q_{j-1}^{(a_k)}, q_j^{(k)}\Big).$$

Here $\widetilde{w}_{j-1}^{(k)}$ is the normalised weight with which particle k enters step j ($\widetilde{w}_{j-1}^{(k)} = 1/N_p$ after resampling and $\widetilde{w}_{j-1}^{(k)} = \bar{w}_{j-1}^{(k)}$ when resampling is skipped), and a_k is its ancestor. The first

particles come from the initial law with equal weights. The untwisted configuration has no step-0 factor, because $\Psi_0 \equiv 1$ and the initial proposal is the initial law. Just after resampling the weights are equal and $\bar{\ell}_j$ is the plain mean. When resampling is skipped, the carried weights are generally unequal, and replacing them by equal weights can bias the estimate. If some $\bar{\ell}_j = 0$, then $\hat{Z}_M = 0$. The normalised estimate of an expectation is $\hat{\eta}_M(\varphi) = \sum_k \bar{w}_M^{(k)} \varphi(q_M^{(k)})$.

A twist $h_j(q)$ is the expected product of the remaining potentials from state q at checkpoint j, with $h_M \equiv 1$. Keep K_j as the reference kernel. For positive functions h_j define $\Psi_0^h = h_0$ and $\Psi_j^h(q, q') = \Psi_j(q, q')\, h_j(q')/h_{j-1}(q)$. Compute the weights as $\omega_0 = h_0\, \mathrm{d}p_0/\mathrm{d}\mathsf{R}_0$ and $\omega_j = \Psi_j^h\, \mathrm{d}\mathsf{K}_j/\mathrm{d}\mathsf{R}_j$, and include the initial factor in $\hat{Z}_M = \prod_{j=0}^{M} \bar{\ell}_j$. For exact backward functions choose the proposal $\mathsf{R}_j = \mathsf{K}_j^h = \mathsf{K}_j \Psi_j h_j / h_{j-1}$ and $\mathsf{R}_0 = p_0 h_0 / Z_M$. Then $\omega_0 = Z_M$ and $\omega_j = 1$ for $j \geq 1$, which gives zero variance (Theorem 4.10 of Supplementary Section S1). Approximate positive twists retain unbiasedness under the stated importance-weighting and resampling assumptions. The first configuration omits twisting.

The statements about this estimator are mathematical results on idealised or finite models, and Supplementary Section S1 gives the assumptions and proofs. Under the support condition, bounded incremental weights and offspring counts whose mean given the past is $N_p \bar{w}$, $\mathbb{E}[\hat{Z}_M] = Z_M$ for every $N_p \geq 1$ and every resampling schedule that depends only on the past, an ESS trigger included. With the exact twist and that initial proposal, $\hat{Z}_M = Z_M$ on every realisation, and with an approximate positive twist $\hat{Z}_M$ stays unbiased. The logarithm $\log \hat{Z}_M$ is biased low by Jensen's inequality, and $\hat{\eta}_M$ is a ratio that is biased in general at finite N_p. Consistency of the weighted particles is shown for resampling before every step by a multinomial, residual or stratified scheme [155]. It is not shown for systematic resampling on unordered particles, whose offspring counts are not negatively associated, or for an adaptive schedule. No usable a-priori constant for a bias of order $1/N_p$ is available, the controlled-ODE pushforward has no correctness proposition, and no single guarantee composes the SMC bias with head calibration and specificity control.

Replicates and diagnostics. Because no a-priori error constant is available, the design runs $R \geq 32$ independent replicates from independent seeds and reports their mean, the empirical standard error and a 95 percent percentile interval. The floor of 32 is a setting of the design. When the estimator is biased, a percentile interval over replicates does not certify coverage of the true value. ESS is a diagnostic that reads the weights only [156], and it equals N_p right after

resampling even when every particle descends from one ancestor, so the stored ancestor indices are inspected too.

2.9 Maturation and editing

Maturation edits one existing antibody or nanobody, the parent, within a declared mutation budget. The base model has no parent input, no editing mask and no maturation step. The full system adds them by partial noising: a generative model restarted from a noised copy of a sample and run on to the data end returns a variant of that sample, as SDEdit does for images [157]. The restart point lies on the master clock of the flow (Section 2.5). Corruption, guidance, solver, hard decoding, context restoration and rejection together form one proposal kernel K_{edit} (**Table 13**). Published methods steer discrete sequence models with a predictor or classifier [158–160]. The editor here also moves structure, and the design excludes fine-tuning.

Editing contract. Three nested masks decide what an edit may touch (Section 2.2): $d_{\mathrm{res}} \preccurlyeq e_{\mathrm{res}} \preccurlyeq m_{\mathrm{res}}$ for design-mutable, transported and assembled residues. Rows in m_{res} outside e_{res} are fixed context. Rows in e_{res} outside d_{res} move in space while their amino-acid identity stays clamped. The frozen encoder supplies the packed parent state $q^{\mathrm{par}} = (x_e^{\mathrm{par}}, z_e^{\mathrm{par}})$, the C-alpha positions and 8-number latents of the n_e transported residues. One edit draws

$$q' \sim K_{\mathrm{edit}}\big(\cdot \mid q^{\mathrm{par}}, m_{\mathrm{res}}, e_{\mathrm{res}}, d_{\mathrm{res}}, \mathcal{S}, c_{\mathrm{gen}}, u^{*}, c_{\mathrm{guide}}, \omega\big).$$

Here $\mathcal{S}_i$ is the set of amino acids allowed at position i, which holds only the parent residue where $d_{\mathrm{res},i} = 0$. The symbol c_{gen} is the antigen and format context, u^{*} the re-entry point, c_{guide} the guidance configuration and ω the noise and decoding seeds. Edits use the context-anchored profile (Section 2.2.1), so the flow state has $11n_e$ numbers and fixed rows add no dimension, loss or velocity. Positions follow IMGT labels (Section 2.3.1).

Four format profiles set the masks. `VHH_CDR_EDIT` designs selected VHH CDR positions, transports them with registered seam neighbours and fixes the remaining framework and scaffold, with a VHH scaffold and cysteine table as an additional gate. `FV_CDR_EDIT` does the same for selected VH and VL CDR positions, fixes the frameworks, the chain register and protected interface anchors, and adds the assembled VH/VL pairing and pose as a gate (Section 2.3.2). `FRAMEWORK_HUMANIZE` designs listed framework positions, transports them with local geometry neighbours, fixes the parental CDRs, and gates on humanness and retention of the parental binding mode. `LIABILITY_REPAIR` designs the residues of one liability window and gates on liability removal without introduced liabilities.

Partial corruption on the master clock. The flow moves a state from noise at master time $u = 0$ to data at $u = 1$. The schedules $t_x(u)$ and $t_z(u)$ give the share of the data endpoint that coordinates and latents have reached. An edit enters part-way. It picks one re-entry point $u^* \in (0,1)$, draws base noise $(x_e^0, z_e^0) \in \mathbb{R}^{11n_e}$ and mixes noise and parent on the linear path used in training:

$$x_{e,u^*} = (1 - t_x(u^*))\, x_e^0 + t_x(u^*)\, x_e^{\mathrm{par}}, \qquad z_{e,u^*} = (1 - t_z(u^*))\, z_e^0 + t_z(u^*)\, z_e^{\mathrm{par}}.$$

A larger u^* keeps more of the parent. One master clock keeps the edit on the sampling curve, while the two schedules still corrupt coordinates and latents to different degrees. With the schedules $t_x(u) = 2u - u^2$ and $t_z(u) = u^2$ and $u^* = 0.75$, the edit keeps $t_x = 0.9375$ of the parent coordinates and $t_z = 0.5625$ of the parent latents.

After each solver step the fixed context is written back. This inserts conditioning data and leaves the transported density unchanged. A hard decode then turns the category logits into one amino acid per residue inside $\mathcal{S}_i$, which restores parent identities outside d_{res}. Atom-mask, bond, chirality, assembly and profile checks decide acceptance. A seam check measures bond geometry, clashes and decoder residuals around every mask boundary, and a mask and profile pair that fails it is disabled.

Mutation-budget controller. A user asks for k mutations, but the editor controls u^*, so a controller learns the link. For each u^* on a grid it decodes R unguided proposals per calibration parent and averages the number of design positions that differ from the parent:

$$\hat{k}(u^*; L, p) = \frac{1}{R} \sum_{r=1}^{R} d_H \left(s^{(r)}|_{d_{\mathrm{res}}}, s^{\mathrm{par}}|_{d_{\mathrm{res}}} \right).$$

Here d_H is the Hamming distance, $s^{(r)}$ the r-th decoded sequence, L the editable length and p the format profile. For a requested expected budget k the controller interpolates this curve inside the calibration grid where it is monotone. An exact budget is a separate mode that accepts only decodes with $d_H = k$ and reports the rejection rate. If no u^* yields k, the controller reports the budget as unavailable, with the realised edit-count distribution. Gate F-M4 tests fidelity on held-out parents, by length and format stratum. It requires an upper confidence limit of the deviation within $\tau(k) = \max\{1, 0.2k\}$ mutations in every cell, and monotonicity and invertibility of the curve over each qualified length band. The protocol uses 300 parents per length decile and budgets $k = 1$ to 10.

Guided editing. Guidance follows the endpoint, metric and trust-region contracts of Section 2.8. Affinity change uses the thermodynamic sign $\Delta\Delta G = \Delta G_{\text{variant}} - \Delta G_{\text{parent}}$, so a negative value predicts stronger binding, and the affinity utility is $U_{\text{aff}} = -\widehat{\Delta\Delta G}$. The utility gradient, preconditioned by the decoder-feature metric G_ϕ, gives the direction $d = G_\phi^{-1}\nabla U_{\text{aff}}$ on the packed e_{res} axes, with coordinate part d^x and latent part d^z. The edit solver adds it to the schedule-scaled base field, with the arguments of the heads left out:

$$\frac{\mathrm{d}}{\mathrm{d}u}\begin{bmatrix} x_e \\ z_e \end{bmatrix} = \begin{bmatrix} t_x'(u)\, v_\theta^x \\ t_z'(u)\, v_\theta^z \end{bmatrix} + \begin{bmatrix} \kappa_x(u)\, I & 0 \\ 0 & \kappa_z(u)\, I \end{bmatrix}\begin{bmatrix} d^x \\ d^z \end{bmatrix}.$$

The heads give velocity per unit block time, and the schedule derivatives convert it to master time once. The gains $\kappa_x(u)$ and $\kappa_z(u)$ are stored with the edit profile. The trust cap of Section 2.8 limits the metric norm of each guided increment. A property model trained on finished molecules can be unreliable on partly corrupted states. A guidance-depth check therefore compares matched guided and unguided edits on a frozen grid of u^* values for soft-to-hard score agreement, enrichment by an independent ranker, validity and seam failures, pose retention and diversity. A schedule is active only where u clears every check.

Mutation supply. The default editor proposes amino-acid changes at d_{res} positions through the learned latent kernel. Real hypermutation rates depend on neighbouring nucleotides, which the S5F branch supplies when the parent has an observed nucleotide sequence, a species and chain class inside the model scope and a compatible lineage. It uses the 5-mer targeting and substitution models of Yaari et al. [161], whose stated scope is human heavy chains. A change is weighted by how often its site mutates, how often the base becomes the new base, and how much the model prefers the translated result. For nucleotide position i and substitution $b \to b'$,

$$Q_{\text{S5F}}(i, b \to b') \propto \mu_{\text{mut}}(c_5(i))\, \sigma(b \to b' \mid c_5(i)) \exp\{\beta_{\text{S}}\, U(\text{translate}(s_{\text{nt}}^{i \to b'}))\}.$$

Here $c_5(i)$ is the 5-mer around i, μ_{mut} the relative mutability of that context, σ the substitution probability given it, $s_{\text{nt}}^{i \to b'}$ the nucleotide sequence after the change, U a model utility and β_{S} its strength. Stop codons, frameshifts, forbidden cysteine changes and protected residues get zero weight. Gate F-M5 admits the branch once motif assets, source scope and a fixed-horizon comparison against uniform targeting are fixed, and targeting is uniform until then.

Holding the framework and revising it on purpose. Hard preservation covers chain register, identities outside d_{res}, coordinates outside e_{res}, disulfide topology, required scaffold positions, and atom and chemistry validity. A violation rejects the draw. Soft preservation adds a cost on

the distance from the parent in the decoder-feature metric, so a move that barely changes the decoded molecule costs little: $\delta_G(X, X^{\mathrm{par}}) = \sqrt{\Delta q^{\top} G_{\phi}(q^{\mathrm{par}}; c_{\mathrm{gen}})\, \Delta q}$ for the change Δq of the packed state.

Revising the framework is a separate act. With protein sequence only, IgBLAST returns the nearest translated germline V reference [118], and an edit toward it is named `aa_reference_reversion`. For a candidate reversion set R the editor chooses at most k_R reversions that raise humanness most while pose, affinity and liabilities stay acceptable:

$$\max_{S \subseteq R,\ |S| \le k_R} U_{\mathrm{human}}(X_S) \quad \text{subject to} \quad \mathrm{PosePass}(X_S) = 1,\ \widehat{\Delta\Delta G}(X_S) \le \tau_{\Delta\Delta G},\ \mathrm{LiabilityPass}(X_S) = 1,$$

where X_S is the candidate with the reversions in S, U_{human} a humanness score and $\tau_{\Delta\Delta G}$ an affinity-loss limit. Because mutation effects interact, the editor enumerates exactly for small $|R|$, otherwise uses a bounded beam, and evaluates the final subset jointly. Gate F-M6 uses at least 100 held-out parents under one declared affinity assay. It requires a lower parent-cluster-bootstrap limit of 0.05 on the median OASis humanness gain [18], a lower exact-binomial limit of 0.90 on the fraction with non-inferior measured affinity, and no newly introduced structure-confirmed liability.

Liability-aware editing. Every hard-decoded proposal is scanned by the liability channel (Section 2.11). Motifs inherited from the parent stay apart from motifs the edit adds:

$$\mathcal{L}_{\mathrm{new}}(X', X^{\mathrm{par}}) = \mathcal{L}(X') \backslash \mathcal{L}(X^{\mathrm{par}}),$$

where $\mathcal{L}(X)$ is the set of liability motifs in X. A sequence motif only nominates a site for a structural check, because motif matching over-predicts. In 37 therapeutic antibodies it gave false-positive rates of 0.31 for aspartate and 0.41 for asparagine sites, against 0.02 and 0.04 for a structure-based classifier [16]. Structure-confirmed liabilities drive rejection. Gate F-M7 takes 500 accepted variants and requires a one-sided binomial upper limit of at most 0.02 on the fraction that introduce a structure-confirmed CDR liability. It reports its rejection yield, so rejecting every variant cannot pass. Generic polyspecificity is measured again after every round, for example with a sequence-based nanobody predictor [108]. Pair-specific off-targets use their own panel (Section 2.10).

Rounds and acceptance. Rounds, mutation budgets, guidance schedules and judge budgets are fixed before the evaluation that decides release. Gate F-MAT compares matured candidates

with the parent and a declared baseline on seven co-primary endpoints, with the parent as statistical unit. They cover affinity change, pose and epitope retention, off-target margin, polyspecificity response, budget fidelity, liability and scaffold compliance, and diversity. AbBiBench supplies affinity-ranking tasks once its data, licence and splits are admitted [162]. Section 2.13 places these gates in the evaluation design.

2.10 Judging and specificity

The judging layer of the full system has two parts. The complex judge proposes and ranks antibody-antigen poses, which is a question of geometry. The specificity instrument measures cross-reactivity against a named panel of other proteins. **Table 13** gives the contract of the complex judge. The co-folding models and physics programs named below are published systems that the design calls as external tools. Their outputs sit in separate fields, and no field is converted into another.

Input and output. The judge reads a candidate, a protein antigen record, a declared epitope and an antigen-state profile. It returns an ensemble of proposed complexes, one selected pose and the registration of that pose against the epitope. The profiles cover a VHH or a paired Fv against one protein state, several registered antigen conformers with one report per state, and a peptide antigen. An antigen residue is identified by six items: structure checksum, model, chain, author residue number, insertion code and a normalised index. A chain letter and an integer are insufficient, because a revised structure or an insertion code can move the atoms that a number points to. The epitope uses the same identifier. A judge version binds the weights, the template cut-off, the alignment databases, the sampling settings, the seed algorithm and the numerical precision. The native complex of a test case is excluded from the antibody templates.

Seed ensembles and clusters. For candidate i and a roster of S registered seeds, the co-folder returns poses x_{ij}, $j = 1, \dots, S$. The number S is fixed for a profile, and a different S needs a new calibration. Seed replicates measure the variability of the algorithm for one pair. They carry no information on binding, which needs independent candidates with measured endpoints. In the benchmark reviewed in the Introduction, more samples per target improved structure and left the separation of cognate from shuffled pairs unchanged [46] (Section 1.5), so the design spends compute on seeds and negative controls. The comparator is $S = 15$, and qualification selects the smallest fixed S that meets the registered pose margin at matched compute.

Poses are clustered by interface RMSD after antigen alignment, with the algorithm, cut-off, linkage, atom mask and handling of missing residues fixed in advance. Write $C_{i1}, \dots, C_{im}$ for

the clusters of candidate i. The largest cluster $C_{ic_i^*}$ is selected, with ties broken by the best reranker score and then the smallest seed index. The pose is a seed majority when $|C_{ic_i^*}| > S/2$. Otherwise the pose is called multimodal, and the largest cluster supplies a descriptive representative that no downstream gate treats as a majority.

Reranking. A reranker r_η picks a pose inside the selected cluster. It trains on reference complexes, where each sampled pose has a geometric label Y such as DockQ, a continuous score from 0 to 1 that combines the fraction of native contacts with two RMSD terms [163]. Training pairs (u, v) come from one target and satisfy $Y_u > Y_v$. The loss rewards a higher score for the better pose of every pair:

$$\mathcal{L}_{\text{rank}} = -\sum_{t} \sum_{(u,v)\in\mathcal{O}_t} w_{uv} \log\sigma\big(r_\eta(x_u) - r_\eta(x_v)\big).$$

Here t runs over targets, $\mathcal{O}_t$ is the set of ordered pose pairs of target t, w_{uv} is a pair weight and σ is the logistic function. The reranker learns the order of seeds for one antibody-antigen pair. Ordering different candidates by binding is a separate task that needs assayed candidates under one protocol. The emitted pose score is a frozen monotone readout of the reranker and the ensemble features, with a model-specific scale. Search quality (does the ensemble hold a good pose) and ranking quality (does the rule pick it) are reported separately, because benchmarks show a gap between them. The internal confidence of a co-folder often fails to identify the best sampled model, and the reported success on recent antibody and VHH complexes is low [45, 102] (Section 1.5).

Epitope registration and state margin. Registration reuses P_E, R_E and EpiF1 of Section 2.7, computed on the selected pose. They describe placement against a declared patch. With a native complex the report adds interface and ligand RMSD, the fraction of native contacts and DockQ. For antigen conformers the judge runs the same seeds on each and reports a state margin, the pose score on the intended state minus the best score on any other state. A margin describes geometry. Selectivity between states needs assays on both.

Admission error and yield. A pose gate admits some candidates, and an external endpoint later succeeds or fails for each. Let $A_i = 1$ when candidate i passes the gate, $Y_i = 1$ when the registered endpoint succeeds and $B_i = 1 - Y_i$. Over n candidates the report gives the bad-admission incidence, the admission yield, the bad share among admitted candidates and the good yield:

$$\widehat{\mathrm{BAI}} = \frac{1}{n}\sum_i A_i\, B_i, \quad \hat{y}_{\mathrm{admit}} = \frac{1}{n}\sum_i A_i\,, \quad \hat{\pi}_{\mathrm{bad|admit}} = \frac{\sum_i A_i\, B_i}{\sum_i A_i}, \quad \hat{y}_{\mathrm{good}} = \frac{1}{n}\sum_i A_i\, Y_i.$$

The third quantity is undefined when nothing is admitted. The four are reported together, because the incidence alone reaches zero for a gate that admits nothing. The cohort is an untouched set from a frozen generator, steering policy, triage rule and judge profile, and the endpoint Y_i is an assay outcome. Acceptance limits for each judge component are listed in Supplementary Section S3.

Co-folders and physics tools. The judge draws on three published co-folders. AlphaFold 3 predicts the joint structure of proteins, nucleic acids, small molecules and ions with a diffusion-based network [40]. Boltz-2 is a structure and affinity model with open weights, and its affinity head is described for small-molecule binders, so the judge uses it for geometry only [43]. Chai-1 predicts complexes with or without sequence alignments [42]. Boltz-2 is the primary co-folder and Chai-1 adds a second model family. Antibody chains enter with empty alignments and the antigen reads a precomputed alignment. Each invocation draws one diffusion sample with 3 recycling steps and 200 sampling steps. Co-folders from different families are not independent judges, because their training sets overlap, so the design measures their dependence.

Physics tools run as a cascade on a shortlist, and each tier has an admitting rule and its own endpoint type (**Table 11**). Short molecular dynamics with OpenMM [164] tests whether the pose is retained. MM-GBSA gives a protocol-bound end-state energy rank [165], and Flex ddG ranks mutations from a backrub ensemble in Rosetta [166]. Well-tempered metadynamics [167] gives a one-dimensional free-energy profile along the interface distance, which is read as a rank and is not a binding free energy. None of the four outputs enters a field for pose, pair binding or affinity. With S seeds over M survivors the campaign needs $S \cdot M$ co-fold inferences, and a pre-triage by the complex head of Interface-JEPA (Section 2.7) could reduce this number.

Tier	Settings	Admitting rule
Co-fold ensemble	15 seeds	Every survivor
Reranked shortlist	Ensemble poses ranked by the reranker	Top decile of reranker score and cross-tool ipTM agreement within 0.15
Short MD	5 ns at 300 K, 4 fs step, first 1 ns discarded	At least 20 interface heavy atoms
MM-GBSA	Protocol-bound end-state energy rank	Mean interface RMSD below 3.0 Å and upper bound of the Theil-Sen slope at most 0.25 Å per ns
Flex ddG	Backrub ensemble	MM-GBSA rank in the top third and a mutation question from the maturation loop
Metadynamics	50 ns, well-tempered	Top 8 by rank

Table 11. Tiers of the physics cascade.

Distances are in angstrom and times in nanoseconds. The slope bound is a one-sided 95% moving-block bootstrap bound.

2.10.1 The signed panel and pair-specific losses

Panel. Specificity is relative to a panel, and the instrument reports risk for a named panel only. For a candidate X with intended antigen A^{+}, the signed panel is $\Pi = \Pi_{\mathrm{req}} \dot{\cup} \Pi_{\mathrm{forbid}}$, the disjoint union of required members, which the binder should bind (for example orthologs of the pharmacology species), and forbidden members, which it should not bind (for example human paralogs). The panel is built from the campaign and the target and never from the candidate. Its inputs are the indication, route, species, target family, reference proteome, tissue-exposure rules, isoform policy and available structures. The campaign profile sets the tiers: for systemic use the forbidden tiers hold accessible human paralogs and membrane, extracellular and secreted self proteins, and for a reagent antibody they hold proteins of the sample matrix and common homologs. Each member records its sign and tier with the rationale, its identifier, release, isoform and sequence checksum, the source of its structure, its exposure class, the rule that builds its surface patch, its reference mass ν_k^{panel} and, when it is sampled, its sampling probability q_k. Linear peptides, folded proteins and computed surface patches test different interaction modes and keep different tier labels. A peptide-negative result holds for the linear tier only.

Readouts. Three readouts are kept apart. The pair-binding probability $o_{\mathrm{pair}}(X, \pi)$ belongs to a candidate and one named panel member. The pose score belongs to a proposed pair and measures geometric plausibility. The generic polyspecificity $o_{\mathrm{PSR}}(X)$ belongs to the candidate alone: it is the response to a polyspecificity reagent, a detergent-solubilised insect-cell membrane preparation in the nanobody assay [14, 108]. It flags broadly sticky candidates and

cannot say whether a given paralog is bound. Co-folder confidence does not estimate a pair-binding probability. In the nanobody benchmark reviewed in the Introduction, ipTM separated cognate from shuffled pairs only weakly [46] (Section 1.5). An admissible pair oracle therefore consumes both X and π, predicts a named binding endpoint and is calibrated on held-out pair labels.

Losses. Let $p_k = o_{\text{pair}}(X, \pi_k) \in [0,1]$ be a calibrated pair-binding probability under one assay family. The forbidden-hit mass is the weight of forbidden members predicted to bind above a threshold λ_f. The required-miss mass is the weight of required members predicted to bind at or below a threshold λ_r:

$$\ell_f(X) = \sum_{k \in \Pi_{\text{forbid}}} w_k^f \, \mathbb{1}[p_k > \lambda_f], \qquad \ell_r(X) = \sum_{k \in \Pi_{\text{req}}} w_k^r \, \mathbb{1}[p_k \le \lambda_r].$$

The weights are non-negative and sum to one within each sign, and $\mathbb{1}[\cdot]$ is 1 when its condition holds and 0 otherwise. The hit mass falls as λ_f rises and the miss mass rises with λ_r, so the thresholds are separate. The two numbers keep off-target binding apart from missing intended breadth. When members are sampled with probability q_k to stand for a reference measure with masses ν_k^{panel}, each carries the importance ratio ν_k^{panel}/q_k, and the estimate is the ratio-weighted mean of the indicators. Support ($q_k > 0$ wherever $\nu_k^{\text{panel}} > 0$) is checked first. The report lists the normalised weights $\tilde{w}_k$ and the effective panel size $K_{\text{eff}} = 1/\sum_k \tilde{w}_k^2$, which falls from the number of members towards 1 as one member takes all the weight. Uncertainty comes from repeated, independently seeded panel draws and from resampling candidates, stored apart.

Coverage. Every panel risk is reported with the number of members per tier, the reference mass covered (defined only for a finite, normalised member frame) and the fraction of sequence clusters represented. It adds the fraction of members with compatible structural inputs, the fraction with a pair-specific assay or calibrated oracle, the reference mass whose exposure class is unresolved, and K_{eff}.

Oracles and delivered sets. Pose, pair-binding and generic polyspecificity oracles stay in separate fields. Two oracles are combined only when they measure the same endpoint under compatible conditions and a declared measurement model defines the combination. Agreement of two oracles is a diagnostic, since both can agree while ranking the measured outcomes in reverse. A combined risk summary counts as panel risk only after its frozen form has been checked against held-out, construct-matched measured outcomes.

Candidates chosen for a portfolio form a delivered set, which differs from the off-target panel. Epitope competition between binders is stored as a directed graph, and each edge records whether one binder blocks, displaces or sandwiches another under the protocol of the assay. Competition is not transitive. Antibodies at adjacent or slightly overlapping epitopes can displace one another [12], so A may compete with B and B with C while A and C do not compete. Epitope bins are therefore a derived view under a named rule, either identical blocking fingerprints or complete-linkage clustering at a fixed cut height. Section 2.12 describes the selection.

2.10.2 Thresholds and conformal risk control

Populations. The three populations P_{hist}, P_{cal} and P_{gen} are those of Section 2.7.1. Oracles train on P_{hist}. The calibration population P_{cal} is divided into a map-fitting subset and a risk-calibration subset. The score-calibration map is fitted on the map-fitting subset and then frozen, and thresholds are chosen on the risk-calibration subset, which is exchangeable with the candidates it serves. A map fitted on the same candidates would make the losses depend on one another and void the exchangeability that the guarantee below needs. The cohort P_{gen} comes from one frozen generator, guide, selector, panel profile and assay protocol, and it measures risk after the shift to generated designs. It keeps every selected candidate, including failures of synthesis, expression or assay, because removal by outcome would change the quantity measured. The deployment gap is the empirical risk on P_{gen} minus that on the risk-calibration subset at the chosen threshold, reported with a cluster-bootstrap interval.

Two-threshold grid. Thresholds are first chosen on the risk-calibration subset from finite grids. The grids, the ceilings α_f and α_r, a cluster bootstrap and a tie rule are fixed before calibration. A pair of thresholds is acceptable when the bootstrap upper endpoints of both mean losses lie at or below their ceilings. If none is, panel export is blocked. This rule carries no coverage guarantee, because the two losses move in opposite directions. The reference configuration therefore uses one alarm threshold on the whole panel, since a false-alarm rate of 10 percent per pair would raise an alarm on 88 percent of designs against 20 independent negatives if the alarms are independent.

Conformal risk control. A guarantee needs a different construction. Conformal risk control picks a threshold $\hat{\lambda}$ from n risk-calibration examples so that the expected loss on a new example is at most α [168]. The loss is $L_i(\lambda) = \mathbb{1}[X_i \in \widehat{\mathcal{D}}_\lambda]\, B_i$, with the failure indicator $B_i = 1 - Y_i$: candidate i is called designable at threshold λ and later found bad. The theorem needs four

conditions. The loss is right-continuous in λ, which an admission rule $s(X) > \lambda$ with a strict inequality secures. It is non-increasing in λ. It is at most B_L, with some $\lambda_{\max}$ at which it is at most α. And the $n + 1$ losses are exchangeable. The threshold is

$$\hat{\lambda} = \inf\left\{\lambda\colon\ \hat{R}_n(\lambda) \le \alpha - \frac{B_L - \alpha}{n}\right\}, \qquad \hat{R}_n(\lambda) = \frac{1}{n}\sum_{i=1}^{n} L_i\,(\lambda),$$

If the feasible set is empty, set $\hat{\lambda} = \lambda_{\max}$, where $\lambda_{\max}$ admits no candidate; otherwise use the displayed infimum. Then $\mathbb{E}[L_{n+1}(\hat{\lambda})] \le \alpha$. For $B_L = 1$ and $\alpha = 0.10$ the empirical inequality can be feasible only when $n \ge 9$. Independently tuned thresholds do not define the required nested scalar family. A fixed scalar score, such as the minimum of two scores oriented so that larger values favour admission, defines nested sets through $s(X) > \lambda$. Supplementary Section S1 gives the proof and the fallback rule (Theorem 5.6).

Limits of the guarantee. Two limits remain for a valid path. First, the guarantee controls the marginal rate of candidates that are called and bad. The share of bad candidates among those selected can differ from it by an arbitrary factor. A guarantee on that share needs Learn then Test [169] or conformal selection [170].

Second, a prospective campaign is a shifted population. Let Q be the campaign distribution and W a bound on the density ratio $\mathrm{d}Q/\mathrm{d}P$. The threshold chosen on the risk-calibration subset then has risk at most $\min(W\alpha, B_L)$ under Q. The bound is uninformative once $W \ge B_L/\alpha$, which is 10 at $B_L = 1$ and $\alpha = 0.10$. Weights for the design procedure can restore control in a weighted version of the rule, if they can be computed and the rule that labels a candidate bad is unchanged [107, 168]. The second condition can fail when the antigen changes, because a candidate is bad by its binding to the deployment antigen.

2.11 Liability, immunogenicity and glycan channels

The full system evaluates three typed channels on the final assembled construct: chemical-liability flags, a predicted presentation load on major histocompatibility complex class II (MHC-II), and glycan-occlusion scenarios. A sequence screen nominates sites, and a structural or measured endpoint decides what a campaign does with them. No channel output is a claim about stability, immunogenicity or binding, which would need a matched assay. Guidance (Section 2.8) and maturation (Section 2.9) read the scores.

Scanned sequence. The scanned sequence is the product as it would be made: VH and VL domains or the VHH, linkers, tags, junctions, cleavage remnants, any Fc sequence, and every

arm of a multispecific assembly. An isolated variable domain would miss peptides and motifs that form across a junction, so every window that crosses a component boundary is labelled.

Chemical liabilities. A sequence motif only nominates a site. Let $s_1 \ldots s_L$ be the final sequence and r a liability family. A deterministic scanner returns $m_r(i; s) \in \{0,1\}$ for the window anchored at position i, with value 1 when a susceptible context is present. A structural hotspot model returns $h_{\psi,r}(i; X, \kappa) \in [0,1]$, where X is the hard-decoded structure and κ holds the conditions: pH, temperature, buffer, stress duration, redox state and construct format. The structural liability score sums the product over positions:

$$L_r(X; \kappa) = \sum_{i=1}^{L} m_r\,(i; s)\, h_{\psi,r}(i; X, \kappa).$$

The hotspot term estimates whether the structure around a motif resembles that of known hotspots. A score is interpretable only under the conditions and calibration cohort of $h_{\psi,r}$. An exposed Asn in an NG window with $h_{\psi,r} = 0.8$ is therefore ranked for confirmation ahead of a buried one with $h_{\psi,r} = 0.1$ (illustrative values). Structure-aware evidence exists for Asn deamidation and Asp isomerisation. In a large uniform set of antibody variable-domain residues, degradation hotspots measured by mass spectrometry were characterised by conformational flexibility, the size of the following residue and secondary-structure parameters. A CDR predictor was derived from them [16]. Other families (Met and Trp oxidation, N-terminal cyclisation, fragmentation) give sequence flags only. A site keeps the label of a sequence motif or a structural estimate until a site-resolved forced-degradation assay on the product, with binding retention after stress, replaces it.

The scanner also reports every N-X-S/T sequon (X is any residue except Pro) in the binder. Its occupancy is unknown until a glycopeptide mass-spectrometry measurement on the same construct and host exists, and no occupancy probability is computed. Surface charge patches at a declared pH are developability descriptors, and a gate may use one only after calibration against a named assay. Steering uses a smooth expectation of each motif indicator, whereas admission uses the hard-decoded sequence.

Predicted MHC-II presentation load. The load summarises predicted peptide presentation by a declared panel of human leukocyte antigen (HLA) class-II molecules. It serves as a sequence-level design penalty and as an input to immunogenicity assessment. It does not measure an anti-drug antibody response, and an industry commentary names the missing correlation between

predicted T-cell epitope content and clinical anti-drug antibody outcome as a limit of such tools [114]. Humanness is a separate field.

The final sequence s is cut into overlapping windows of $L_p = 15$ residues, so the window set is $\mathcal{W}_{15}(s) = \{s_{i:i+14} : i = 1, \dots, |s| - 14\}$. NetMHCIIpan-4.0 supports variable peptide lengths and defaults to 15-mers for FASTA input, and it reports the binding core it identifies [19]. The design pins one release (4.3k), and every threshold names that release. The 4.3 update adds inverted peptide binders and covers HLA-DR, DP and DQ [171]. Strong-binder and weak-binder labels are release defaults with no clinical meaning: version 4.0 uses ranks below 2 and 10 percent [19], and the 4.3k service lists 1 and 5 percent.

The panel H lists complete molecule identifiers, with alpha and beta chains for HLA-DP and HLA-DQ. The weight π_h of molecule h comes with its source population, frequency data and normalisation. Let r_{wh} be the tool rank of window w for molecule h, φ a monotone transform of it oriented so that larger values mean more burden, and $\omega_{\text{region}}(w)$ the weight of the region that contains w. The load averages the burdens of each window with the panel weights, scales them by region weight and sums over windows:

$$L_{\text{MHCII}}(s; H) = \sum_{w \in \mathcal{W}_{L_p}(s)} \omega_{\text{region}}(w) \sum_{h \in H} \pi_h \, \varphi(r_{wh}).$$

A weight vector is a population weight when it has a population source and a design weight otherwise. Region weights separate CDR, framework, constant-region, linker, tag and junction windows, and a window that crosses two components takes the junction weight. Burdens 0.7 and 0.2 on two molecules weighted 0.6 and 0.4 give $0.6(0.7) + 0.4(0.2) = 0.50$ for one 15-mer before its region weight. The adapter also counts the deduplicated physical core and HLA events that the pinned release labels strong. A change in this load supports a lower predicted core load and nothing more.

Glycan occlusion. Antigen glycans are mobile and can shield large parts of a protein surface. GlycoSHIELD grafts sampled glycan conformers onto a static structure in minutes, and its authors apply it to N-cadherin, coronavirus spike proteins and gamma-aminobutyric acid receptors [13]. Let $\mathcal{H} = \{h_1, \dots, h_M\}$ be a finite set of glycosylation scenarios for antigen A. Each fixes the protein conformer, occupied sites, site-specific glycoforms, conformer-library version and sampling seed. For epitope residue j, the accessibility in scenario h is the fraction of N_h sampled conformers that leave the residue reachable to a probe:

$$\sigma_j^{(h)} = \frac{1}{N_h} \sum_{n=1}^{N_h} \mathbb{1}\,[\text{probe reaches residue } j \text{ in conformer } (h, n)].$$

Because N_h is finite, every value carries Monte Carlo uncertainty. Scenarios stay unweighted until independent occupancy and glycoform evidence supplies weights. The envelope of residue j runs from the minimum $\underline{\sigma}_j$ to the maximum $\overline{\sigma}_j$ over scenarios. For an epitope E with residue weights a_j, the aggregate is $S_E^{(h)} = \sum_{j \in E} a_j \, \sigma_j^{(h)} / \sum_{j \in E} a_j$, with lower envelope $\underline{S}_E = \min_h S_E^{(h)}$. Required residues E_{req} also receive $\underline{\sigma}_{\text{req}} = \min_{h,\, j \in E_{\text{req}}} \sigma_j^{(h)}$, which is not applicable for an empty set. The channel sends central accessibility, envelope width, sampling uncertainty and a support mask to antigen conditioning (Section 2.7).

Floors for $\underline{S}_E$ and $\underline{\sigma}_{\text{req}}$ are set before evaluation. An epitope is accessible when its lower envelope clears both floors and occluded when its upper envelope lies below a floor. It is unresolved when the envelope or the Monte Carlo interval crosses a floor, and out of scope when the site, glycoform, chemistry or antigen class lacks library support. These states describe accessibility in the registered ensemble and make no statement on measured binding.

The scenario set covers glycosylation of the antigen only. A candidate that carries a binder sequon is therefore unresolved for glycan effects, because its contact scenarios omit binder glycoforms. A variable-domain glycan can raise or lower affinity depending on its position [115] and adds chemical heterogeneity [116]. Admission needs a sequon-free sequence, a qualified aglycosylated format, or measured occupancy with matched binding data. The admission predicate of Section 2.3 already rejects sequons inside the CDRs.

Order of evaluation. The hard-decoded candidate passes six steps in a fixed order. The first is validity of sequence, atoms and disulfides (Section 2.3). The next three are the chemical motif and structure-context label with binder sequons, the MHC-II presentation load of the final construct, and glycan occlusion of the antigen epitope. The fifth is the pose, pair-binding, affinity and panel-risk endpoints (Section 2.10), and the last is experimental confirmation where the campaign profile specifies one. A missing required input leaves the result unavailable, and an interval that crosses a decision floor leaves it unresolved. Each channel acts on steering or admission only after a pre-specified comparison of channel-active against channel-inactive designs under the same conditioning, seeds and step budget (Supplementary Section S3). The liability comparison tests the structural proxy and measures no degradation.

Developability predictors. Developability is treated as a vector of axes, and a missing axis stays missing: it is filled with neither zero nor a mean. The axes come from published predictors: AbNatiV for VHH nativeness [172], NanoMelt [173] and TEMPRO [174] for the apparent melting temperature T_m, BioPhi with OASis for humanness [18], the sequence-liability scanner above, and the MHC-II load. Disagreement between the two T_m predictors is descriptive. Each predictor returns, for every registered metric, a value, units, an interval and a flag that states whether it is calibrated. A registry fixes the orientation of each metric (higher is better, lower is better or two-sided), its units and its valid range, and a value outside the range is an error. A predictor whose output is unavailable gives no admission decision.

Intervals follow an empirical rule on a named held-out set of natural antibodies. For a predictor $\hat{f}$ with residuals $\hat{f}(x_i) - y_i$ on n held-out items, the lower endpoint for a higher-is-better metric is the prediction minus the residual order statistic of rank $\lceil (n+1)(1-\alpha) \rceil$, with $\alpha = 0.10$. A lower-is-better metric uses the mirrored rule. The rank exists only if it does not exceed n, which at this α needs $n \geq 9$. Otherwise the predictor is flagged as uncalibrated, and a percentile filter serves as an engineering screen. These intervals apply to the stated reference population, and application to generated designs requires calibration on that population. A coverage statement needs a procedure whose assumptions hold for the population, such as conformal risk control [168].

Five metrics use clinical-stage reference bands. Raybould and colleagues modelled 242 clinical-stage therapeutics and derived guidelines for five metrics [15]. An amber flag marks the extremes of that distribution and a red flag a value not seen in it (**Table 12**). Percentiles from one assay, buffer and format are not universal limits, and a VHH has no light chain, so charge symmetry does not apply to it.

Metric (orientation)	Amber	Red
Total CDR length (two-sided)	54 to 60	above 60
Surface hydrophobicity patches, `psh` (two-sided)	83.84 to 100.71 and 156.20 to 173.85	below 83.84 or above 173.85
Positive-charge patches, `ppc` (lower is better)	1.25 to 3.16	above 3.16
Negative-charge patches, `pnc` (lower is better)	1.84 to 3.50	above 3.50
Fv charge symmetry, `sfvcsp` (higher is better)	-20.40 to -6.30	below -20.40

Table 12. Reference bands of the Therapeutic Antibody Profiler for five metrics, derived from 242 clinical-stage therapeutics [15].

Total CDR length is in residues, and the patch metrics and charge symmetry are in the units of the profiler.

A framework residue rule becomes a hard constraint only inside a hashed scaffold profile backed by a primary experiment (Section 2.3.2). Without an executable checker no universal residue set is substituted.

2.12 Decisions under ambiguity and portfolio selection

The selection layer reads scored candidates and returns a set of exactly N designs, or no set when none is feasible or the search budget ends. It first combines the scores of several plausible models into one cautious utility, then chooses the set. **Table 13** gives the inputs, the operations, the outputs and the status of the three modules that act on a finished candidate: maturation (Section 2.9), pose judging (Section 2.10) and this layer. Supplementary Figure S5 draws the layer: worlds give a cautious score, gated candidates enter an integer programme, and the solver returns a set or a refusal.

Module	Inputs	Operations	Outputs	Status
Maturation	One parent antibody or nanobody, the three nested masks, a mutation budget, and the packed parent state from the frozen encoder	Partial corruption to a re-entry point on the master clock, guided solve to the data end, hard decoding, restoration of the fixed context, acceptance or rejection	A variant of the parent with its realised mutation count, or a rejected draw	Proposed
Pose judging	A candidate, a protein antigen record, a declared epitope and an antigen-state profile	A co-folder run over a fixed roster of registered seeds, pose clustering, reranking inside the selected cluster, registration against the declared epitope	An ensemble of proposed complexes, one selected pose, the pose score, the epitope metrics P_E, R_E and EpiF1, and a state margin where conformers are declared	Proposed
Portfolio selection	A finite candidate pool with its gate results, one utility per plausible world with reference weights, a target specification with weights, the committed size N and the campaign constraints	The cautious utility over the worlds, the coverage objective, and an integer programme under the size, cluster and gate constraints	A set of exactly N designs with its realised coverage, every constraint value and the removal contribution of each member; otherwise a proved absence of a feasible set, or an exhausted budget with no proof	Proposed

Table 13. Contract of the three modules that act on a finished candidate.
Each row names what the module consumes, what it does and what it returns. All three modules belong to the full system and are specified as a design.

Model ambiguity. Calibrated predictors can disagree, and a design that ranks first under one checkpoint can rank tenth under another. The layer therefore scores each design in every one of a declared set of plausible worlds and aggregates the scores with caution. Let $\Omega = \{1, \dots, M\}$ index the worlds. A world is one surrogate checkpoint, assay mapping, panel perturbation or

antigen-family scenario. World j has a reference weight $\pi_j > 0$, with $\sum_j \pi_j = 1$. For candidate X and antigen A, a registered transform puts the raw score of world j on one dimensionless scale, the utility $U_j(X, A)$, where larger is better. The entropic pessimistic utility lets poor worlds count for more. It multiplies each reference weight by $e^{-\eta U_j}$, which grows as the utility falls, sums, and maps the result back to the utility scale:

$$U_{\text{ent}}(X, A) = -\frac{1}{\eta}\log\left(\sum_{j=1}^{M} \pi_j \; e^{-\eta U_j(X,A)}\right), \qquad \eta > 0.$$

The risk aversion η is dimensionless. As $\eta \to 0$ the value tends to the reference-weighted mean $\sum_j \pi_j U_j$, and as η grows it tends to $\min_j U_j$. An identity gives the meaning. Let ν be any weights over the worlds, and let $\mathrm{KL}(\nu \mid \pi) = \sum_j \nu_j \log(\nu_j/\pi_j)$ be the Kullback-Leibler divergence (KL) from the reference weights. Then

$$U_{\text{ent}} = \min_{\nu}\left\{\sum_j \nu_j U_j + 1/\eta \,\mathrm{KL}(\nu \mid \pi)\right\}, \qquad \nu_j^{\star} = \frac{\pi_j e^{-\eta U_j}}{\sum_k \pi_k e^{-\eta U_k}}.$$

An adversary shifts weight towards the poor worlds and pays a KL price, scaled by $1/\eta$, for leaving the reference. The minimiser is $\nu^{\star}$, and the identity follows from $\mathrm{KL}(\nu \mid \nu^{\star}) \geq 0$ (Supplementary Section S1). To check the identity by hand, take $\pi = (0.5, 0.5)$, $U = (0.9, 0.3)$ and $\eta = 2$. Then $U_{\text{ent}} \approx 0.515$ and $\nu^{\star} \approx (0.231, 0.769)$, against a plain mean of 0.6.

A hard KL ball caps how far the adversary may move: $U_{\text{KL-ball}}(\delta) = \min\{\sum_j \nu_j U_j : \mathrm{KL}(\nu \mid \pi) \leq \delta\}$. For $\delta > 0$ the point $\nu = \pi$ is strictly feasible, and convex duality gives

$$U_{\text{KL-ball}}(\delta) = \sup_{\lambda_{\text{DRO}} \geq 0}\left[-\lambda_{\text{DRO}} \log \sum_j \pi_j \, e^{-U_j/\lambda_{\text{DRO}}} - \lambda_{\text{DRO}}\, \delta\right].$$

The entropic form fixes $\lambda_{\text{DRO}} = 1/\eta$. The ball chooses λ_{DRO} and adds the radius term, so η and δ describe different families. Both forms reweight only the listed worlds, so a plausible world missing from the list stays out. The Wasserstein set moves probability mass between nearby world descriptions. For a nominal law $\hat{P}$ on a declared world space, a ground cost c_W and a radius ρ_W, $U_{\text{W}} = \inf\{\mathbb{E}_Q[U] : W_{c_W}(Q, \hat{P}) \leq \rho_W\}$. The ground cost decides which shifts count as near. The duality that makes the set tractable needs U to be lower semicontinuous, with integrability and growth conditions, and a finite reformulation needs more structure [175]. Guidance (Section 2.8) uses the entropic utility, and the Wasserstein value is evaluated offline.

Every campaign stores the world membership, the utility transforms, the reference weights and a leave-one-world-out sensitivity.

2.12.1 Fixed-size portfolio

Coverage objective. A campaign carries a fixed number of designs forward. The layer fixes an integer N before any candidate score is read, then chooses the best set of that size. The N highest scores can be near copies that share one epitope, and one poor epitope would then lose the whole set. Let $\mathcal{V}$ be a finite candidate pool. A target specification $\mathcal{T} = \{(A_j, E_j, w_j)\}_{j=1}^{n_T}$ lists antigens A_j, epitopes E_j and weights $w_j \geq 0$. Let $p_j(X) \in [0,1]$ be the binding-utility input of candidate X for target j, obtained from the utilities above by a registered map. The coverage of a set $\mathcal{B} \subseteq \mathcal{V}$ is

$$F_{\mathrm{cov}}(\mathcal{B}) = \sum_{j=1}^{n_T} w_j \log\left(1 + \sum_{X \in \mathcal{B}} p_j\,(X)\right).$$

For each target, the inputs of all chosen designs are added, one is added to the sum, and the logarithm is weighted. The logarithm flattens, so a design aimed at a target that is already covered adds little. Write $S_j(\mathcal{B}) = \sum_{X \in \mathcal{B}} p_j\,(X)$. Adding X raises the coverage by $\sum_j w_j \log\big(1 + p_j(X)/(1 + S_j(\mathcal{B}))\big)$, which cannot grow when $\mathcal{B}$ grows. Hence F_{cov} is monotone and submodular by direct calculation. As a check of diminishing returns, take two targets of weight 1 and $p(A) = (0.8,0.1)$, $p(B) = (0.2,0.7)$. Candidate B alone adds 0.713, and added to $\{A\}$ it adds $1.281 - 0.683 = 0.598$.

Selection problem. The selection solves

$$\max_{\mathcal{B} \subseteq \mathcal{V}} F_{\mathrm{cov}}(\mathcal{B}) \quad \text{subject to} \quad |\mathcal{B}| = N,\ R_{\mathrm{self}}(\mathcal{B}) \leq r_{\max},\ \sum_{i<k} \mathrm{collision}\,(X_i, X_k) \leq c_{\max},\ \mathcal{B} \in \mathcal{F}_{\mathrm{format}}.$$

R_{self} is a set-level self-risk. The collision sum counts each unordered pair once. The family $\mathcal{F}_{\mathrm{format}}$ holds the chain-pairing, valency, linker and assembly rules of the chosen format (Section 2.3). Every member also passes the per-candidate gates $\ell_f(X) \leq \alpha_f$ and $\ell_r(X) \leq \alpha_r$ on the forbidden-hit and required-miss masses (Section 2.10). The reference configuration uses one alarm threshold, because the coverage guarantee of conformal risk control holds for a scalar nested family of thresholds and a grid over two thresholds carries none [168]. A candidate with a missing evaluation is never counted as cleared. The reference configuration sets $N = 5$,

$c_{\max} = 2$ and $R_{\text{self}} \equiv 0$, and counts a collision when two designs lie in one CDR-H3 cluster at 80 percent identity. For $N = 5$ this allows at most two designs per cluster.

Integer programme. Each candidate i gets a binary variable z_i, with $z_i = 1$ when it is chosen. The reference configuration has one target, so F_{cov} is an increasing function of $\sum_i z_i\, p(X_i)$. The programme maximises that sum subject to $\sum_i z_i = N$, at most two chosen designs per CDR-H3 cluster, and $z_i = 0$ for any candidate that fails a gate. HiGHS [176] solves it through the `milp` routine of SciPy [177], with a time limit of 60 s, a relative gap of 10^{-4} and a pool limit of 1,000 candidates. With several targets, the concave logarithm enters through tangent lines: each target gets an auxiliary variable bounded above by tangents of $\log(1 + s)$ at fixed points. Tangents lie above the curve, so the solver bound stays an upper bound on the exact objective, and the incumbent is re-scored exactly. Greedy forward selection reaches at least $1 - (1 - 1/N)^N$ of the optimum under a size constraint alone, which is 0.672 at $N = 5$ and tends to 0.632 as N grows [178]. No such factor is claimed once self-risk, collision and format constraints apply, and the solver supplies a proved bound instead.

Outcomes. The solver returns one of four outcomes, and each is a statement about proof. It returns an incumbent set whose upper bound is proved equal to it, or an incumbent set with a proved upper bound and the relative gap. It returns no set when a solver certificate or presolve proves that no feasible set of size N exists, or when the budget ends with neither an incumbent nor a proof. A heuristic that finds no feasible set therefore cannot report infeasibility. The report of a returned set lists the committed N, the realised coverage, every constraint value and, for each member, the removal contribution $\Delta_X^{\text{remove}} = F_{\text{cov}}(\mathcal{B}) - F_{\text{cov}}(\mathcal{B}\backslash\{X\})$, with members in canonical identifier order. An empty acceptable threshold grid upstream blocks panel export before selection (Section 2.10).

Scope of the guarantees. The entropic identity, the dual of the KL ball and the submodularity of F_{cov} are proved (Supplementary Section S1). No guarantee covers a greedy rule under the added constraints. The Wasserstein duality holds for a given utility only after its conditions are checked. The marginal control of false-designable calls is a rate over candidates, and the share of failing designs among the N chosen ones can differ from it by an arbitrary factor. These guarantees do not compose into an end-to-end rate.

2.13 Evaluation design

The evaluation design fixes, before any outcome is read, how the data are split, how the system is compared with published work, how evidence is counted and which claims the evidence can carry. **Figure 5** shows three steps of it. Linked records are grouped and move together (panel a). The groups are cut at one date, set against the cut-offs that published work reports (b). A check is decided by the upper bound of the failure rate (c). The base model generates no structure, so its logged quantities describe the course of training and do not measure the quality of a design.

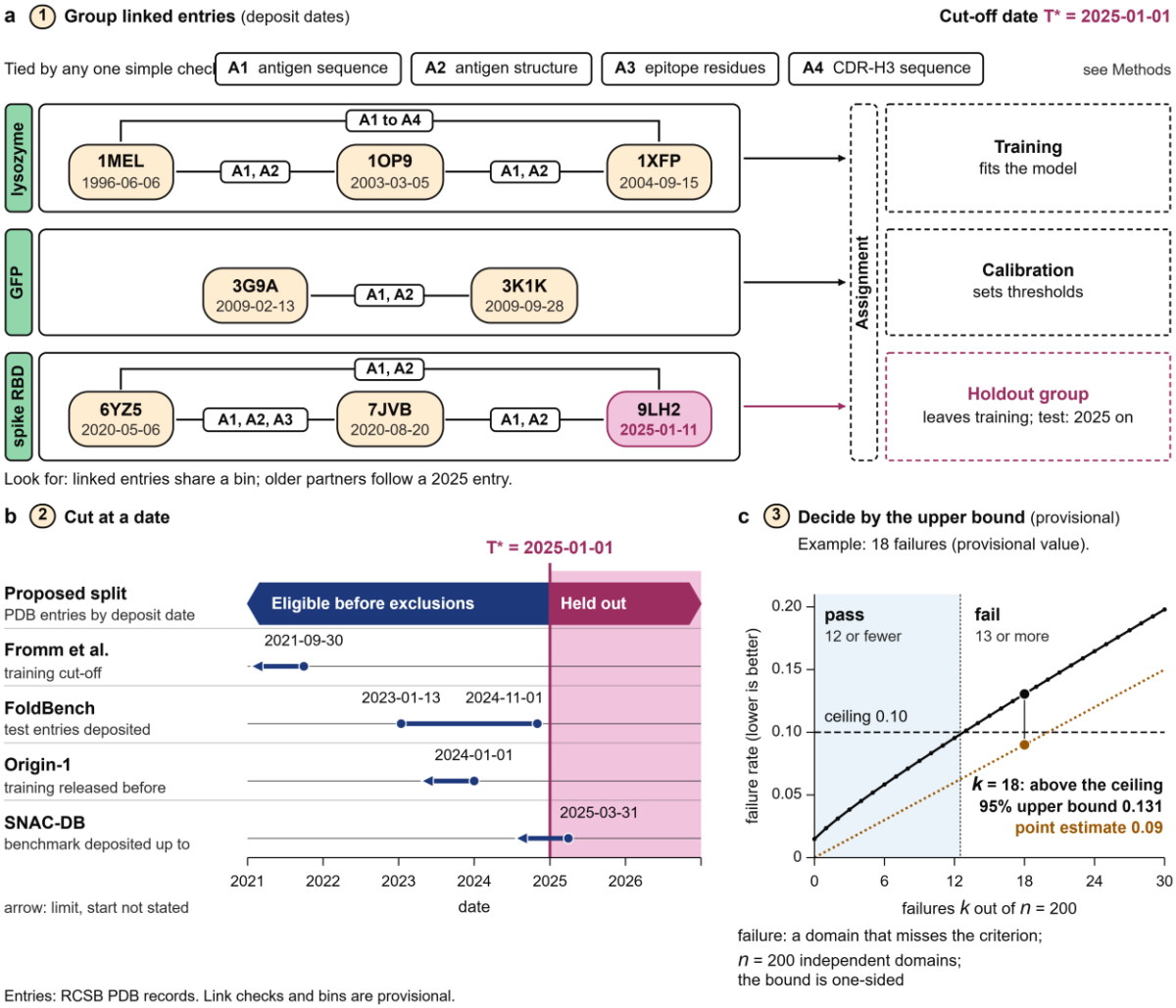

Figure 5. Linked entries are grouped, cut at one date, and passed or failed by the upper bound of the failure rate.

a, Eight nanobody-antigen entries of the Protein Data Bank, each with its identifier and initial deposit date, in three groups by antigen. Two entries are tied when any one of four links holds: A1 antigen sequence, A2 antigen structure, A3 epitope residues or A4 CDR-H3 sequence. The chip on a link lists the axes that hold. These links were computed for this figure with a local alignment, a template-modelling lower bound, an epitope Jaccard on 4.5 Å contacts and a regular expression for CDR-H3. A1 and the A4 link of 1MEL with 1XFP were reproduced with the sequence search of the RCSB. The figure uses these checks in place of the clustering tools of the Methods. A group moves to one side as a whole: 9LH2 was deposited on 2025-01-11, so its group leaves training together with its older partners. Only the 2025 entry is a test record, and its older partners stay outside the test set. **b,** The dates that four published sources report, set against T^* = 2025-01-01 by deposition date. The four are a training

cut-off (Fromm et al.), the window in which test entries were deposited (FoldBench), the date before which training entries were released (Origin-1) and the date up to which benchmark entries were deposited (SNAC-DB). An arrow marks a limit whose start the source does not state. Only the training date of each source is drawn; Origin-1 also reports a test window to 2025-07-10. The top bar covers entries of the Protein Data Bank by deposit date, and benchmark keys, whole components and failed checks remove further records. **c,** A worked example with n = 200 independent domains and a ceiling of 0.10, where a failure is a domain that misses the criterion. The one-sided 95% exact (Clopper-Pearson) upper bound clears the ceiling for 12 or fewer failures. Eighteen failures give a point estimate of 0.09 and an upper bound of 0.131 (Beta quantile 0.95, parameters 19 and 182), so the check fails. A failed check says that the data do not clear the ceiling, and says nothing about the true rate. The entries, their dates and the four published dates are retrieved records. The three groups are subsets of their components, and the real spike RBD component holds hundreds of entries. The assignment of the groups to training, calibration and the holdout group is the proposal drawn here, and k = 18 is a value chosen for the example. CDR-H3, third heavy-chain complementarity-determining region; GFP, green fluorescent protein; RBD, receptor-binding domain.

2.13.1 Reporting for the base model

The training script prints one line on rank 0 after every tenth successful update. The line holds the loss summed over the updates since the previous line, the terms of that loss, the global gradient norm and the mean wall time per update. In the autoencoder stage the terms are the masked coordinate error over present atoms, in nm², the masked residue cross-entropy and the latent penalty, which is 0.01 times the masked mean square of the latent. In the denoiser stage the terms are the coordinate flow loss, in nm², and the latent flow loss. Each is the masked sum of squared residuals divided by the number of present residues and the block width, 3 for coordinates and 8 for the latent. Component means are taken over the successful updates of each logging window.

Shards store coordinates in ångströms. After centring, the code multiplies them by 0.1, so that the networks, the noise and the coordinate losses work in nanometres. The autoencoder reconstruction is reported as a root-mean-square error per coordinate in ångströms. The script averages the coordinate term over the successful updates of each window and converts the mean to a length:

$$\mathrm{RMSE}_{\text{Å}} = 10\sqrt{\mathrm{MSE}_{\mathrm{nm}^2}}.$$

The square root turns a squared error into a length, and the factor 10 turns nanometres into ångströms. A mean squared error of 0.01 nm² gives 1 Å. The result is an error per coordinate, and the root-mean-square distance over atoms is $\sqrt{3}$ times larger. The mean squared error is averaged over the window before the root is taken, and roots of single complexes are not averaged. The mean runs over all present atoms, including the C-alpha atoms, which the decoder receives as input. Because 1 nm² equals 100 Å², the coordinate term is 100 times smaller than in ångström units for the same structure, and its weight against the cross-entropy depends on the unit.

The minimum population squared-error loss is the expected conditional variance of the target velocity given the noisy state and times [55]; this term can be zero. Logged losses fluctuate because noise and times are sampled, so each value is a Monte Carlo estimate and the curve is read over windows.

2.13.2 Comparison with published numbers

Other systems enter only as published numbers, cited as reported. The design runs no competitor and loads no competitor weight. Published numbers of BoltzGen, for example, can be cited as reported [83].

A published number belongs to one panel, one cut-off, one judge and one success rule, and these differ between papers. **Table 14** lists the panels of four published works. **Figure 5**b sets their dates against T^* = 2025-01-01, the temporal cut-off of this study (Section 2.4.3). A number measured on one of these panels cannot be moved to another. A published panel older than T^* is held out of Omniverso training only through the benchmark rule of Section 2.4.2.

Source	Panel	Date rule
Fromm et al. [45]	110 antibody-antigen complexes	Training cut-off of 30 September 2021, shared by all tested methods
FoldBench [103]	172 antibody-antigen pairs	Entries deposited after 13 January 2023 and before 1 November 2024
SNAC-DB benchmark [102]	Public entries	Entries deposited up to 31 March 2025
Origin-1 [88]	Ten targets, fewer than 100 designs per target	Training on entries released before 1 January 2024

Table 14. Panels of published works that the design cites: panel size (complexes, pairs or targets) and calendar date rule, as reported by the sources.

Every comparison reports five items beside the published number, once for the source and once for Omniverso. The target set gives the targets, their number and how they were chosen. The cut-off date gives the deposit date after which a test complex counts as unseen, and the training cut-off of each model. The designs per target give the number generated and the number tested or scored. The judge model gives the model that scored the designs, its version and its training cut-off. The success definition gives the metric, its threshold and its unit, and whether it is a computational score or an assay result. A difference in any item is stated next to the comparison. A published number whose protocol cannot be read from its source is not compared.

The base model generates no design, so only size and budget can sit side by side. The denoiser has 153,102,091 parameters, against about 159 M for the published reference denoiser [71].

That model saw about 374 M training samples, computed from the step counts and batch sizes in its tables (**Table 1**). The training budget of the base model follows the rule of Section 2.6. Size and budget say nothing about accuracy.

The panels are not matched, so a comparison cannot establish equal accuracy or a competitive win, and the report makes no paired claim of superiority or non-inferiority. Sampling noise adds a second limit. With 30 targets, the normal-approximation 95% margin near a 50% hit rate is about 18 percentage points, and 200 independent units give about 7 points. A gap on targets that are not shared does not identify a cause. A comparison sentence takes one form: "System S reports a rate of r on its own panel of m targets. The panels are not matched to those of this study."

2.13.3 Splits and statistical rules for the full system

Splits. Leakage control uses the four axes of Section 2.4.3 (A1 to A4, thresholds set by the design). Records linked on any axis form one component, and a component goes wholly to training or to calibration, except that a component holding a record deposited on or after T^* leaves training as a whole (**Figure 5**a, Section 2.4.3). Only records deposited on or after T^* are test records, and their older partners stay outside the test set. Three populations keep roles apart. Historical data fit models. Held-out calibration objects set thresholds. Outputs created after the generator, the selector and the compute policy are frozen are the only population on which a claim about the generator is read. Further routes of leakage are shared interface geometry [100], overlap with the pretraining corpus of a judge [101] and group-level links such as lineages and assay plates [127]. A claim that a model generalises beyond its training records therefore needs a prospective cohort collected after the model is frozen, as blinded challenges with experimental readouts provide for antibodies [97].

Fixed-horizon record. Every gate fixes its evidence record before an outcome is read. The record names the target population and sampling rule, the independent unit, the committed number of units, the statistic, the interval procedure, the error level, the threshold with its failure direction and the multiplicity rule (Supplementary Section S3):

$$\mathcal{E}_k = (\mathcal{P}_k, \mathcal{U}_k, n_k, T_k, I_k, \alpha_k, \tau_k, \mathcal{M}_{\mathrm{mult},k}).$$

The statistic is computed once, at the committed n_k. Early looks and stopping are not allowed, and sequential methods form a separate research interface with other error rules [179]. A floor reads a lower bound and a ceiling reads an upper bound.

A worked example shows why the bound decides and the point estimate does not. A gate commits 200 independent domains and a ceiling of 0.10 on a failure rate. If 18 domains fail, the point estimate is 0.09. The one-sided 95% exact upper bound is the 0.95 quantile of a beta distribution with parameters 19 and 182:

$$\hat{R}_{200} = \frac{18}{200} = 0.09, \qquad \mathrm{Beta}^{-1}(0.95;\ 19,\ 182) \approx 0.131.$$

The bound lies above the ceiling, so the gate fails. The registry of the full system lists 53 gates. Each carries one of four dispositions. A required gate withholds its capability when it fails. A feature-conditional gate removes its feature. A diagnostic gate is reported and cannot admit anything. A not-applicable gate is a checked absence. Supplementary Section S3 lists the gates.

Sample size. Take a rate with baseline 0.80 and a one-sided exact binomial test at level 0.05. With 200 independent Bernoulli units, the test has 80% power at a true success rate of approximately 0.726, a drop of 7.4 points. Smaller departures from 0.80 have lower power. Failure to reject does not establish equivalence or adequate performance. With 1,000 units, a decrease of approximately 3.3 percentage points gives 80% power, and with 2,400 units a decrease of approximately 2.1 points does. The record marks an arm whose power at the effect of interest is below 80% as underpowered.

Endpoint statistics. Two yields per antigen score the top-ranked designs of an arm. For antigen i, arm m and design rank $k \le 5$, let $a_{ikm} = 1$ if the design passes every frozen on-target criterion and 0 otherwise, and let a missing design score 0. Let $h_{ikm} = 1$ if any member of the off-target panel exceeds the frozen alarm threshold or its evaluation is incomplete. The on-target yield of the top five and the yield that also clears the panel are:

$$Y_{im}^{\mathrm{on}} = \frac{1}{5}\sum_{k=1}^{5} a_{ikm}\,, \qquad Y_{im}^{\mathrm{clean}} = \frac{1}{5}\sum_{k=1}^{5} a_{ikm}\,(1 - h_{ikm}).$$

The second yield is a computational surrogate and measures no specificity. For the full arm and a comparator, let Δ^{on} and Δ^{clean} be the means over antigens of the paired differences, and let $L(\cdot)$ be the lower one-sided bound. The acceptance bar is $L(\Delta^{\mathrm{on}}) > -0.10$ and $L(\Delta^{\mathrm{clean}}) > 0$, both required. The margin lets the full arm lose half a passing design per portfolio of five on average, in return for a panel-clean selection.

The bound has level 97.5% on a final panel and 90% on a go or no-go panel, with Holm adjustment across comparisons. A yield takes only six values, so the bound inverts a

conservative test, with a bootstrap-t interval as a check. Coverage is simulated on at least 100,000 data sets per scenario before a protocol is frozen.

Panel size sets the power. With a standard deviation of 0.25 for the paired difference and 25 antigens, the standard error is 0.05. Under a normal approximation, a true gain of 0.10 in the clean yield passes its bound 52% of the time and a gain of 0.15 passes 85% of the time. Any new panel needs its own calculation. The comparators in these tests are the own arms of the ablations below, and other groups enter through published numbers only.

2.13.4 Hypotheses, ablations and validation levels

Hypotheses. Six hypotheses give the programme its questions (**Table 15**). Each names an estimand, a population and split, a comparator, metrics, a refuting condition, a compute tier and a promotion gate. H5 requires assay-family data.

Id	Question	Comparator	Refuting condition
H1	Does the candidate depend on the true antigen?	Framework-only baseline; ungated antigen embedding. Controls: antigen ablation, shuffle, same-antigen epitope swap	A control has a non-positive lower bound, or the framework-only baseline matches the full conditioner on contact quality
H2	Do predicted contact probabilities match observed contacts?	Direct antigen embedding without a contact plan	Calibration slope or error outside the registered limits on a primary split
H3	Does sequence diversity rise without loss of specificity?	Reference decoder, contact-first arm, discrete-sequence arm	Diversity rises while antigen dependence or pose margins fail
H4	Does frame transport add value over C-alpha plus latent transport?	Partially latent flow at matched candidate count and GPU-hours	No lower-bound gain on pose metrics, or a memory or runtime penalty beyond the hardware profile
H5	Do proxy gates track assay outcomes?	Proxy-only triage against an assay-confirmed endpoint	Proxy success with assay failure above the registered ceiling, or a denominator changed after outcomes
H6	Does quality transfer from VHH to paired Fv?	VHH profile, paired-Fv extension	Paired Fv gains on one proxy by breaking pairing, humanness or polyreactivity gates

Table 15. Hypotheses H1 to H6 with their comparators and refuting conditions.

Ablations. Six axes isolate one cause each, and a missing separation on an axis has a stated meaning, so that no arm is read as a general result.

- *Antigen channel:* true antigen, ablation, shuffle and same-antigen epitope swap. A missing gain means that target conditioning is unavailable.
- *Framework shortcut:* full framework, framework dropout and framework only. A missing gain means that antigen evidence is too weak for admission.

- *Contact object:* none, predicted plan, transport-refined plan and contact first. A missing gain means that a miscalibrated plan cannot drive chemistry.
- *Decoder:* continuous latent, discrete sequence and paired Fv. A decoder-specific gain does not transfer across formats.
- *Geometry:* partially latent flow and frame flow. A geometry gain must beat its cost and numerical risk.
- *Evidence mode:* fixed horizon and sequential preview. A preview cannot serve as release evidence.

A failed hypothesis is reported with the split on which it failed and is not averaged into a larger result.

Validation levels. A capability enters the manifest of the full system only through evidence at the level that matches its claim (Supplementary Section S3). Six levels are used. At the algebra level, shapes, units, masks, signs and limiting cases are checked by derivations and numerical fixtures. At the component level, one module is checked against its declared contract on cluster-disjoint examples with named gates. At the interface level, cross-module fixtures check that producer and consumer agree on type, scale, mask and provenance. At the system level, a fixed campaign is replayed end to end under one configuration. At the external level, the frozen system is tested on registered benchmark populations of independent complexes, panels or assay units. At the experimental level, selected candidates are tested against measured biological endpoints under predeclared laboratory protocols.

For the base-model code the specified checks lie at the algebra level (shapes, masks, units and parameter counts) and at parts of the interface and system levels (shard reader and resume). Two outcome classes are defined. A computational nomination has passed every active computational gate and carries no experimental statement. An experimental release needs a laboratory evidence package.

Claim boundary. The boundary holds at every level. Results are computational. A pose score, contact calibration, humanness score, predicted MHC-II rank and generic polyreactivity prediction may rank or quarantine candidates inside their endpoint scope. Binding, affinity, specificity, immunogenicity, aggregation and developability remain assay-family claims, and this manuscript makes none of them for a designed molecule or for the model.

2.14 Computing, software and availability

Hardware and software. The training configuration uses one node with two NVIDIA A100 cards of 40 GB each, with one process per card and data-parallel gradient sums [133]. The code is written in Python 3.11 with PyTorch 2.11 built for CUDA 12.8 [180] and NumPy [181]. The number of updates follows from the measured update time (Section 2.6).

Code. The base-model code is supplied as an archive with the Supplementary Information; Supplementary Section S4 describes its contents and use. The archive holds the training code: the autoencoder, the denoiser, the flow path and losses, the shard reader, the precision policy, the update loop, checkpoints and run records. It also holds three configurations, the scripts that launch a run and the unit tests that run on a CPU. Support code retrieves and cleans PDB entries and performs temporal, duplicate-identity and structural-fingerprint checks. The four-axis split protocol is specified in Section 2.4.3. Numerical checks cover the statements of Supplementary Section S1. The archive contains no trained weight, no structure data and no code of other systems. The code is licensed under AGPL-3.0 (Section 2.14.1).

Data. The base model reads public protein structures. The sources and their licence classes are given in Section 2.4.1. Each dataset stays with its provider, and this paper redistributes none. The unit tests use synthetic complexes that a seeded function generates, so that they need no download.

Weights. No trained weights accompany this paper. A weight set is released only if the licences of all its training sources allow it (Section 2.14.1).

Determinism. Every random draw starts from a SHA-256 hash of the run identifier together with a role label. The initial weights and the global generators use the label for the model, the noise generator of each rank uses the rank, and the batch of update u uses the update index. The run identifier and the update cursor alone therefore fix the length bucket and the draws of an update, and the data order is fixed. Both ranks draw the same indices and take separate slices. A resumed run restores the random state of each rank from the checkpoint, and a skipped update restores the state it started from, so the retry draws the same noise. The seed fixes sample selection and random-number streams. The code does not enable deterministic GPU kernels, and nondeterministic GPU operations can produce differences between repeated training runs. Integer outputs such as shard hashes, manifests, sample order and bucket choice are exact.

2.14.1 Scope, ethics and licence

Scope. The work is computational and contains no wet-lab step. Predictions of the full system would remain predictions until an experiment tests them. A comparison with another group uses its published numbers, names the differences in protocol and states that the panels are not matched.

Dual use and biosafety. A tool that designs binders can be misused, as can any generative model for biochemistry. A drug-discovery model proposed toxic molecules once its objective was inverted [182]. AI-redesigned variants of proteins of concern evaded the screens of nucleic acid synthesis providers until those screens were patched [183]. Synthesis is therefore the control point, and one proposal keeps synthesis records in repositories that are queried only in emergencies [184]. The study takes three positions. The base model contains no component that conditions on a target or emits an antibody sequence. The manuscript prints no designed sequence, and the code package releases none: its only data are synthetic complexes. Any later release of an antigen-conditioned generator would first need a review of misuse and a screen of requested targets against curated lists of controlled pathogen accessions. These positions reduce risk and do not amount to a safety claim.

Licences and data. The code package is licensed under the GNU Affero General Public License, version 3 (AGPL-3.0). Its section 13 requires that a modified version offered to users over a network make its source available. The licence covers code and grants nothing on weights or data. Third-party libraries keep their own licences, and the main ones are named in Section 2.14. Third-party papers appear as citations only. The licence of trained weights follows the licences of the training data (Section 2.4.1). The base model reads public protein structures only and no personal, clinical or patient data, and each downloaded file is stored with its SHA-256 digest (Section 2.4.4).

3 Results

4 Discussion

5 Conclusions

Declarations

Author contributions. D.K. is the sole author and carried out the conception, design, code and writing of the work.

Data and code availability. The training code, its tests, the configuration files and the container definition are supplied as Supplementary Information and described in Supplementary Section S4. The code is licensed under the GNU Affero General Public License, version 3. Each dataset stays with its provider, and this manuscript redistributes none. The sources and their licences are listed in Section Section 2.4 and Supplementary Section S2.

Artwork. The protein pictures in **Figure 1**, **Figure 2**, **Figure 3** and **Figure 4**, and in Supplementary Figures S1 to S5 and S7, were rendered with UCSF ChimeraX from deposited coordinates of the Protein Data Bank; each caption names the entry.

References

[1] Norman RA, Ambrosetti F, Bonvin AMJJ, Colwell LJ, Kelm S, Kumar S, et al. Computational approaches to therapeutic antibody design: established methods and emerging trends. Briefings in Bioinformatics. 2020;21:1549–1567. doi:10.1093/bib/bbz095

[2] Lefranc MP, Lefranc G. IMGT® and 30 Years of Immunoinformatics Insight in Antibody V and C Domain Structure and Function. Antibodies. 2019;8:29. doi:10.3390/antib8020029

[3] Sela-Culang I, Kunik V, Ofran Y. The Structural Basis of Antibody-Antigen Recognition. Frontiers in Immunology. 2013;4:302. doi:10.3389/fimmu.2013.00302

[4] Hamers-Casterman C, Atarhouch T, Muyldermans S, Robinson G, Hammers C, Songa EB, et al. Naturally occurring antibodies devoid of light chains. Nature. 1993;363:446–448. doi:10.1038/363446a0

[5] Muyldermans S. Nanobodies: Natural Single-Domain Antibodies. Annual Review of Biochemistry. 2013;82:775–797. doi:10.1146/annurev-biochem-063011-092449

[6] Jovčevska I, Muyldermans S. The Therapeutic Potential of Nanobodies. BioDrugs. 2020;34(1):11–26. doi:10.1007/s40259-019-00392-z

[7] Mitchell LS, Colwell LJ. Comparative analysis of nanobody sequence and structure data. Proteins: Structure, Function, and Bioinformatics. 2018;86:697–706. doi:10.1002/prot.25497

[8] Gordon GL, Capel HL, Guloglu B, Richardson E, Stafford RL, Deane CM. A comparison of the binding sites of antibodies and single-domain antibodies. Frontiers in Immunology. 2023;14:1231623. doi:10.3389/fimmu.2023.1231623

[9] Lefranc MP, Pommié C, Ruiz M, Giudicelli V, Foulquier E, Truong L, et al. IMGT unique numbering for immunoglobulin and T cell receptor variable domains and Ig superfamily V-like domains. Developmental & Comparative Immunology. 2003;27(1):55–77. doi:10.1016/s0145-305x(02)00039-3

[10] Dunbar J, Deane CM. ANARCI: antigen receptor numbering and receptor classification. Bioinformatics. 2016;32(2):298–300. doi:10.1093/bioinformatics/btv552

[11] Ahmad ZA, Yeap SK, Ali AM, Ho WY, Alitheen NBM, Hamid M. scFv Antibody: Principles and Clinical Application. Clinical and Developmental Immunology. 2012;2012:980250. doi:10.1155/2012/980250

[12] Abdiche YN, Yeung AY, Ni I, Stone D, Miles A, Morishige W, et al. Antibodies Targeting Closely Adjacent or Minimally Overlapping Epitopes Can Displace One Another. PLOS ONE. 2017;12(1):e0169535. doi:10.1371/journal.pone.0169535

[13] Tsai YX, Chang NE, Reuter K, Chang HT, Yang TJ, von Bülow S, et al. Rapid simulation of glycoprotein structures by grafting and steric exclusion of glycan conformer libraries. Cell. 2024;187(5):1296–1311.e26. doi:10.1016/j.cell.2024.01.034

[14] Xu Y, Roach W, Sun T, Jain T, Prinz B, Yu T-Y, et al. Addressing polyspecificity of antibodies selected from an in vitro yeast presentation system: a FACS-based, high-throughput selection and analytical tool. Protein Engineering Design and Selection. 2013;26:663–670. doi:10.1093/protein/gzt047

[15] Raybould MIJ, Marks C, Krawczyk K, Taddese B, Nowak J, Lewis AP, et al. Five computational developability guidelines for therapeutic antibody profiling. Proceedings of the National Academy of Sciences. 2019;116(10):4025–4030. doi:10.1073/pnas.1810576116

[16] Sydow JF, Lipsmeier F, Larraillet V, Hilger M, Mautz B, Mølhøj M, et al. Structure-Based Prediction of Asparagine and Aspartate Degradation Sites in Antibody Variable Regions. PLoS ONE. 2014;9(6):e100736. doi:10.1371/journal.pone.0100736

[17] van de Bovenkamp FS, Derksen NIL, Ooijevaar-de Heer P, van Schie KA, Kruithof S, Berkowska MA, et al. Adaptive antibody diversification through N-linked glycosylation of the immunoglobulin variable region. Proceedings of the National Academy of Sciences. 2018;115(8):1901–1906. doi:10.1073/pnas.1711720115

[18] Prihoda D, Maamary J, Waight A, Juan V, Fayadat-Dilman L, Svozil D, et al. BioPhi: A platform for antibody design, humanization, and humanness evaluation based on natural antibody repertoires and deep learning. mAbs. 2022;14(1):2020203. doi:10.1080/19420862.2021.2020203

[19] Reynisson B, Alvarez B, Paul S, Peters B, Nielsen M. NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. Nucleic Acids Research. 2020;48(W1):W449–W454. doi:10.1093/nar/gkaa379

[20] Bradbury ARM, Sidhu S, Dübel S, McCafferty J. Beyond natural antibodies: the power of in vitro display technologies. Nature Biotechnology. 2011;29:245–254. doi:10.1038/nbt.1791

[21] Olsen TH, Boyles F, Deane CM. Observed Antibody Space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. Protein Science. 2022;31(1):141–146. doi:10.1002/pro.4205

[22] Dunbar J, Krawczyk K, Leem J, Baker T, Fuchs A, Georges G, et al. SAbDab: the structural antibody database. Nucleic Acids Research. 2014;42(D1):D1140–D1146. doi:10.1093/nar/gkt1043

[23] Capel HL, Vavourakis O, Williams BH, Taylor CR, Deane CM. SAbDab2: The structural antibody database in the age of machine learning. bioRxiv. 2026. doi:10.64898/2026.06.16.732554

[24] Brenke R, Hall DR, Chuang GY, Comeau SR, Bohnuud T, Beglov D, et al. Application of asymmetric statistical potentials to antibody–protein docking. Bioinformatics. 2012;28:2608–2614. doi:10.1093/bioinformatics/bts493

[25] Ambrosetti F, Jiménez-García B, Roel-Touris J, Bonvin AMJJ. Modeling Antibody-Antigen Complexes by Information-Driven Docking. Structure. 2020;28:119–129.e2. doi:10.1016/j.str.2019.10.011

[26] Sircar A, Gray JJ. SnugDock: Paratope Structural Optimization during Antibody-Antigen Docking Compensates for Errors in Antibody Homology Models. PLoS Computational Biology. 2010;6:e1000644. doi:10.1371/journal.pcbi.1000644

[27] Lippow SM, Wittrup KD, Tidor B. Computational design of antibody-affinity improvement beyond in vivo maturation. Nature Biotechnology. 2007;25:1171–1176. doi:10.1038/nbt1336

[28] Adolf-Bryfogle J, Kalyuzhniy O, Kubitz M, Weitzner BD, Hu X, Adachi Y, et al. RosettaAntibodyDesign (RAbD): A general framework for computational antibody design. PLOS Computational Biology. 2018;14:e1006112. doi:10.1371/journal.pcbi.1006112

[29] Ruffolo JA, Gray JJ, Sulam J. Deciphering antibody affinity maturation with language models and weakly supervised learning. arXiv:2112.07782. 2021.

[30] Shuai RW, Ruffolo JA, Gray JJ. IgLM: Infilling language modeling for antibody sequence design. Cell Systems. 2023;14:979–989.e4. doi:10.1016/j.cels.2023.10.001

[31] Olsen TH, Moal IH, Deane CM. AbLang: an antibody language model for completing antibody sequences. Bioinformatics Advances. 2022;2(1):vbac046. doi:10.1093/bioadv/vbac046

[32] Hie BL, Shanker VR, Xu D, Bruun TUJ, Weidenbacher PA, Tang S, et al. Efficient evolution of human antibodies from general protein language models. Nature Biotechnology. 2024;42:275–283. doi:10.1038/s41587-023-01763-2

[33] Olsen TH, Moal IH, Deane CM. Addressing the antibody germline bias and its effect on language models for improved antibody design. Bioinformatics. 2024;40:btae618. doi:10.1093/bioinformatics/btae618

[34] Ruffolo JA, Sulam J, Gray JJ. Antibody structure prediction using interpretable deep learning. Patterns. 2022;3:100406. doi:10.1016/j.patter.2021.100406

[35] Ruffolo JA, Chu LS, Mahajan SP, Gray JJ. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. Nature Communications. 2023;14:2389. doi:10.1038/s41467-023-38063-x

[36] Abanades B, Wong WK, Boyles F, Georges G, Bujotzek A, Deane CM. ImmuneBuilder: Deep-Learning models for predicting the structures of immune proteins. Communications Biology. 2023;6:575. doi:10.1038/s42003-023-04927-7

[37] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596:583–589. doi:10.1038/s41586-021-03819-2

[38] Evans R, O'Neill M, Pritzel A, Antropova N, Senior A, Green T, et al. Protein complex prediction with AlphaFold-Multimer. bioRxiv. 2021. doi:10.1101/2021.10.04.463034

[39] Yin R, Feng BY, Varshney A, Pierce BG. Benchmarking AlphaFold for protein complex modeling reveals accuracy determinants. Protein Science. 2022;31:e4379. doi:10.1002/pro.4379

[40] Abramson J, Adler J, Dunger J, Evans R, Green T, Pritzel A, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. Nature. 2024;630(8016):493–500. doi:10.1038/s41586-024-07487-w

[41] Wohlwend J, Corso G, Passaro S, Getz N, Reveiz M, Leidal K, et al. Boltz-1 Democratizing Biomolecular Interaction Modeling. bioRxiv. 2024. doi:10.1101/2024.11.19.624167

[42] Chai Discovery, Boitreaud J, Dent J, McPartlon M, Meier J, Reis V, et al. Chai-1: Decoding the molecular interactions of life. bioRxiv. 2024. doi:10.1101/2024.10.10.615955

[43] Passaro S, Corso G, Wohlwend J, Reveiz M, Thaler S, Somnath VR, et al. Boltz-2: Towards Accurate and Efficient Binding Affinity Prediction. bioRxiv. 2025. doi:10.1101/2025.06.14.659707

[44] Hitawala FN, Gray JJ. What does AlphaFold3 learn about antibody and nanobody docking, and what remains unsolved? mAbs. 2025;17:2545601. doi:10.1080/19420862.2025.2545601

[45] Fromm S, Ludaic M, Elofsson A. Evaluating deep learning based structure prediction methods on antibody–antigen complexes. Bioinformatics. 2026;42(4):btag136. doi:10.1093/bioinformatics/btag136

[46] Smorodina E, Ali M, Kropivšek Brumat K, Salicari L, Miklavc S, Kappassov A, et al. Structural Plausibility Without Binding Specificity: Limits of AI-Based Antibody-Antigen Structure Prediction Confidence Scores. bioRxiv. 2026. doi:10.64898/2026.03.02.709004

[47] Dauparas J, Anishchenko I, Bennett N, Bai H, Ragotte RJ, Milles LF, et al. Robust deep learning–based protein sequence design using ProteinMPNN. Science. 2022;378:49–56. doi:10.1126/science.add2187

[48] Hsu C, Verkuil R, Liu J, Lin Z, Hie B, Sercu T, et al. Learning inverse folding from millions of predicted structures. bioRxiv. 2022. doi:10.1101/2022.04.10.487779

[49] Høie MH, Hummer AM, Olsen TH, Aguilar-Sanjuan B, Nielsen M, Deane CM. AntiFold: improved structure-based antibody design using inverse folding. Bioinformatics Advances. 2025;5:vbae202. doi:10.1093/bioadv/vbae202

[50] Kotelnikov DD, Tatarinova KS, Zhdanov DD. NanoDeNovo: De Novo Design of Anti-Poliovirus I Sabin Strain Nanobodies by Semi-Automated Computational Pipeline. International Journal of Molecular Sciences. 2025;26(19):9262. doi:10.3390/ijms26199262

[51] Ho J, Jain A, Abbeel P. Denoising Diffusion Probabilistic Models. arXiv:2006.11239. 2020.

[52] Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B. Score-Based Generative Modeling through Stochastic Differential Equations. arXiv:2011.13456. 2020.

[53] Watson JL, Juergens D, Bennett NR, Trippe BL, Yim J, Eisenach HE, et al. De novo design of protein structure and function with RFdiffusion. Nature. 2023;620(7976):1089–1100. doi:10.1038/s41586-023-06415-8

[54] Ingraham JB, Baranov M, Costello Z, Barber KW, Wang W, Ismail A, et al. Illuminating protein space with a programmable generative model. Nature. 2023;623:1070–1078. doi:10.1038/s41586-023-06728-8

[55] Lipman Y, Chen RTQ, Ben-Hamu H, Nickel M, Le M. Flow Matching for Generative Modeling. arXiv:2210.02747. 2022.

[56] Albergo MS, Vanden-Eijnden E. Building Normalizing Flows with Stochastic Interpolants. arXiv:2209.15571. 2022.

[57] Liu X, Gong C, Liu Q. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. arXiv:2209.03003. 2022.

[58] Geffner T, Didi K, Zhang Z, Reidenbach D, Cao Z, Yim J, et al. Proteina: Scaling Flow-based Protein Structure Generative Models. arXiv:2503.00710. 2025.

[59] Albergo MS, Boffi NM, Vanden-Eijnden E. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. arXiv:2303.08797. 2023.

[60] Tong A, Fatras K, Malkin N, Huguet G, Zhang Y, Rector-Brooks J, et al. Improving and generalizing flow-based generative models with minibatch optimal transport. arXiv:2302.00482. 2023.

[61] Yim J, Campbell A, Foong AYK, Gastegger M, Jiménez-Luna J, Lewis S, et al. Fast protein backbone generation with SE(3) flow matching. arXiv:2310.05297. 2023.

[62] Yim J, Trippe BL, De Bortoli V, Mathieu E, Doucet A, Barzilay R, et al. SE(3) diffusion model with application to protein backbone generation. arXiv:2302.02277. 2023.

[63] Chen RTQ, Lipman Y. Flow Matching on General Geometries. arXiv:2302.03660. 2023.

[64] Bose AJ, Akhound-Sadegh T, Huguet G, Fatras K, Rector-Brooks J, Liu CH, et al. SE(3)-Stochastic Flow Matching for Protein Backbone Generation. arXiv:2310.02391. 2023.

[65] Campbell A, Yim J, Barzilay R, Rainforth T, Jaakkola T. Generative Flows on Discrete State-Spaces: Enabling Multimodal Flows with Applications to Protein Co-Design. arXiv:2402.04997. 2024.

[66] Huguet G, Vuckovic J, Fatras K, Thibodeau-Laufer E, Lemos P, Islam R, et al. Sequence-Augmented SE(3)-Flow Matching For Conditional Protein Backbone Generation. arXiv:2405.20313. 2024.

[67] Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752. 2021.

[68] Geffner T, Didi K, Cao Z, Reidenbach D, Zhang Z, Dallago C, et al. La-Proteina: Atomistic Protein Generation via Partially Latent Flow Matching. arXiv:2507.09466. 2025.

[69] Lu AX, Yang J, Long Y, Lambert T, Yan W, Robinson SA, et al. Controllable All-Atom Protein Generation with Latent Diffusion. bioRxiv. 2024. doi:10.1101/2024.12.02.626353

[70] Yim J, Jaakik M, Liu G, Gershon J, Kreis K, Baker D, et al. Hierarchical protein backbone generation with latent and structure diffusion. arXiv:2504.09374. 2025.

[71] Didi K, Zhang Z, Zhou G, Reidenbach D, Cao Z, Cha S, et al. Scaling Atomistic Protein Binder Design with Generative Pretraining and Test-Time Compute. arXiv:2603.27950. 2026.

[72] Luo S, Su Y, Peng X, Wang S, Peng J, Ma J. Antigen-Specific Antibody Design and Optimization with Diffusion-Based Generative Models for Protein Structures. bioRxiv. 2022. doi:10.1101/2022.07.10.499510

[73] Martinkus K, Ludwiczak J, Cho K, Liang WC, Lafrance-Vanasse J, Hotzel I, et al. AbDiffuser: Full-Atom Generation of in vitro Functioning Antibodies. arXiv:2308.05027. 2023.

[74] Cutting D, Dreyer FA, Errington D, Schneider C, Deane CM. De novo antibody design with SE(3) diffusion. arXiv:2405.07622. 2024.

[75] Shanehsazzadeh A, Bachas S, McPartlon M, Kasun G, Sutton JM, Steiger AK, et al. Unlocking de novo antibody design with generative artificial intelligence. bioRxiv. 2023. doi:10.1101/2023.01.08.523187

[76] Wang R, Wu F, Gao X, Wu J, Zhao P, Yao J. IgGM: A Generative Model for Functional Antibody and Nanobody Design. bioRxiv. 2024. doi:10.1101/2024.09.19.613838

[77] Wang R, Wu F, Shi J, Song Y, Kong Y, Ma J, et al. A Generative Foundation Model for Antibody Design. bioRxiv. 2025. doi:10.1101/2025.09.12.675771

[78] Bennett NR, Watson JL, Ragotte RJ, Borst AJ, See DL, Weidle C, et al. Atomically accurate de novo design of antibodies with RFdiffusion. Nature. 2026;649(8095):183–193. doi:10.1038/s41586-025-09721-5

[79] Bennett NR, Watson JL, Ragotte RJ, Borst AJ, See DL, Weidle C, et al. Atomically accurate de novo design of antibodies with RFdiffusion. bioRxiv. 2024. doi:10.1101/2024.03.14.585103

[80] Mille-Fragoso LS, Driscoll CL, Wang JN, Dai H, Widatalla T, Zhang JL, et al. Efficient generation of epitope-targeted antibodies with Germinal. Nature Biotechnology. 2026. doi:10.1038/s41587-026-03187-0

[81] Mille-Fragoso LS, Driscoll CL, Wang JN, Dai H, Widatalla T, Zhang JL, et al. Efficient generation of epitope-targeted de novo antibodies with Germinal. bioRxiv. 2025. doi:10.1101/2025.09.19.677421

[82] Swanson E, Nichols M, Ravichandran S, Ogden P. mBER: Controllable de novo antibody design with million-scale experimental screening. bioRxiv. 2025. doi:10.1101/2025.09.26.678877

[83] Stark H, Faltings F, Choi M, Xie Y, Hur E, O'Donnell T, et al. BoltzGen: Toward Universal Binder Design. bioRxiv. 2025. doi:10.1101/2025.11.20.689494

[84] Chai Discovery Team, Boitreaud J, Dent J, Geisz D, McPartlon M, Meier J, et al. Zero-shot antibody design in a 24-well plate. bioRxiv. 2025. doi:10.1101/2025.07.05.663018

[85] Chai Discovery Team, Boitreaud J, Chen R, Dent J, Fairweather L, Geisz D, et al. Drug-like antibody design against challenging targets with atomic precision. bioRxiv. 2025. doi:10.1101/2025.11.29.691346

[86] Nabla Bio, Biswas S. De novo design of epitope-specific antibodies against soluble and multipass membrane proteins with high specificity, developability, and function. bioRxiv. 2025. doi:10.1101/2025.01.21.633066

[87] Latent Labs Team, Kenlay H, Pretorius D, Crabbé J, Bridgland A, Schmon SM, et al. Drug-like antibodies with low immunogenicity in human panels designed with Latent-X2. arXiv:2512.20263. 2025.

[88] Levine S, King JE, Stern J, Grayson D, Wang R, Yin R, et al. Origin-1: a generative AI platform for de novo antibody design against novel epitopes. bioRxiv. 2026. doi:10.64898/2026.01.14.699389

[89] Wu F, Zhao Y, Wu J, Jiang B, He B, Huang L, et al. De novo design of epitope-specific antibodies via a structure-driven computational workflow. Nature Communications. 2025;17(1):625. doi:10.1038/s41467-025-67361-9

[90] BioGeometry Team, Tang J. Rapid De Novo Antibody Design with GeoFlow-V3. bioRxiv. 2025. doi:10.1101/2025.10.20.682964

[91] Cha H, Cho K, Gu J, Gwak D, Ham SW, Hong M, et al. Precision at Every Scale: Efficiency in AI-Driven De Novo Antibody Design. bioRxiv. 2025. doi:10.1101/2025.11.21.689414

[92] Ma J, Wu F, Xu T, Xu S, Liu W, Yan L. An adaptive autoregressive diffusion approach to design active humanized antibodies and nanobodies. Nature Machine Intelligence. 2025. doi:10.1038/s42256-025-01120-9

[93] Kong Y, Shi J, Wu F, Zhao T, Wang R, Zhu X, et al. A synergistic generative-ranking framework for tailored design of therapeutic single-domain antibodies. Cell Discovery. 2025;11(1):85. doi:10.1038/s41421-025-00843-8

[94] Kong X, Zhang Z, Zhang Z, Jiao R, Ma J, Huang W, et al. UniMoMo: Unified Generative Modeling of 3D Molecules for De Novo Binder Design. arXiv:2503.19300. 2025.

[95] Ahmed M, Patterson M. AgForce Enables Antigen-conditioned Generative Antibody Design. arXiv:2605.21610. 2026.

[96] Zhu Y, Ma J, Yin M, Wu J, Tang L, Zhang Z, et al. Ophiuchus-Ab: A Versatile Generative Foundation Model for Advanced Antibody-Based Immunotherapy. bioRxiv. 2026. doi:10.64898/2026.02.02.703197

[97] Erasmus MF, Bedinger D, Hopkins E, Ferguson G, Strickler J, Graff CP, et al. A blinded, prospective benchmark of in silico antibody discovery anchored to experimental affinity and developability. Nature Biotechnology. 2026. doi:10.1038/s41587-026-03238-6

[98] Ahmed M, Taj N, Khan IU, Venkateswara H, Patterson M. CHIMERA-Bench: A Benchmark Dataset for Epitope-Specific Antibody Design. arXiv:2603.13431. 2026.

[99] Bennett NR, Coventry B, Goreshnik I, Huang B, Allen A, Vafeados D, et al. Improving de novo protein binder design with deep learning. Nature Communications. 2023;14:2625. doi:10.1038/s41467-023-38328-5

[100] Bushuiev A, Bushuiev R, Sedlar J, Pluskal T, Damborsky J, Mazurenko S, et al. Revealing data leakage in protein interaction benchmarks. arXiv:2404.10457. 2024.

[101] Hermann L, Fiedler T, Nguyen HA, Nowicka M, Bartoszewicz JM. Beware of Data Leakage from Protein LLM Pretraining. bioRxiv. 2024. doi:10.1101/2024.07.23.604678

[102] Gupta A, Rivero BM, Li R, Roel-Touris J, Nanfack YF, Wendt M, et al. SNAC-DB: An ML-ready database for antibody and NANOBODY® VHH–antigen complexes with expanded structural diversity and real-world benchmarking. Protein Science. 2026;35(7):e70655. doi:10.1002/pro.70655

[103] Xu S, Feng Q, Qiao L, Wu H, Shen T, Cheng Y, et al. Benchmarking all-atom biomolecular structure prediction with FoldBench. Nature Communications. 2025;17(1):442. doi:10.1038/s41467-025-67127-3

[104] Dunbrack RL. Res ipSAE loquuntur: What's wrong with AlphaFold's ipTM score and how to fix it. bioRxiv. 2025. doi:10.1101/2025.02.10.637595

[105] Overath MD, Rygaard A, Jacobsen CP, Brasas V, Morell O, Sormanni P, et al. Predicting Experimental Success in De Novo Binder Design: A Meta-Analysis of 3,766 Experimentally Characterised Binders. bioRxiv. 2025. doi:10.1101/2025.08.14.670059

[106] Liang L, Qian J, Ma X, Zou M, Xu Y, Xu Z. Revisiting Structure-Based Filtering for AI-Designed Antibodies: Opportunities, Limits, and What Matters. arXiv:2610.23136. 2026.

[107] Fannjiang C, Bates S, Angelopoulos AN, Listgarten J, Jordan MI. Conformal prediction under feedback covariate shift for biomolecular design. Proceedings of the National Academy of Sciences. 2022;119(43):e2204569119. doi:10.1073/pnas.2204569119

[108] Harvey EP, Shin JE, Skiba MA, Nemeth GR, Hurley JD, Wellner A, et al. An in silico method to assess antibody fragment polyreactivity. Nature Communications. 2022;13(1):7554. doi:10.1038/s41467-022-35276-4

[109] Jedlitzke B, Wissett O, Nguyen TH, Ghedin E, Sormanni P. SurfDiff: protein surface profiling for selective or broadly reactive epitope prioritisation in binder and immunogen design. bioRxiv. 2025. doi:10.64898/2025.12.23.696170

[110] Brasas V, Christensen CR, Björnsson KH, Møller VE, Scapolo B, Benard-Valle M, et al. Specificity-driven protein binder design with Odin-Multi. bioRxiv. 2026. doi:10.64898/2026.09.08.749745

[111] Jain T, Sun T, Durand S, Hall A, Houston NR, Nett JH, et al. Biophysical properties of the clinical-stage antibody landscape. Proceedings of the National Academy of Sciences. 2017;114(5):944–949. doi:10.1073/pnas.1616408114

[112] Chungyoun M, Gray JR. FLAb2: Benchmarking reveals that protein AI models cannot yet consistently predict developability properties. bioRxiv. 2025. doi:10.64898/2025.12.27.696706

[113] van Niekerk L, Moller J, Ritter S, Quintero-Cadena P, Cohen R, Channing G. 2025 Ginkgo Datapoints Antibody Developability Competition outcomes: limited model performance and a call for data standardization. mAbs. 2026;18(1):2634216. doi:10.1080/19420862.2026.2634216

[114] Gokemeijer J, Jawa V, Mitra-Kaushik S. How Close Are We to Profiling Immunogenicity Risk Using In Silico Algorithms and In Vitro Methods?: an Industry Perspective. The AAPS Journal. 2017;19:1587–1592. doi:10.1208/s12248-017-0143-z

[115] Wright A, Tao MH, Kabat EA, Morrison SL. Antibody variable region glycosylation: position effects on antigen binding and carbohydrate structure. The EMBO Journal. 1991;10:2717–2723. doi:10.1002/j.1460-2075.1991.tb07819.x

[116] Chuang GY, Asokan M, Ivleva VB, Pegu A, Yang ES, Zhang B, et al. Removal of variable domain N-linked glycosylation as a means to improve the homogeneity of HIV-1 broadly neutralizing antibodies. mAbs. 2020;12:1836719. doi:10.1080/19420862.2020.1836719

[117] Greenshields-Watson A, Agarwal P, Robinson SA, Williams BH, Gordon GL, Capel HL, et al. ANARCII enables alignment-free antigen receptor numbering using a generalised language model. Communications Biology. 2026;9(1):1085. doi:10.1038/s42003-026-10186-z

[118] Ye J, Ma N, Madden TL, Ostell JM. IgBLAST: an immunoglobulin variable domain sequence analysis tool. Nucleic Acids Research. 2013;41(W1):W34–W40. doi:10.1093/nar/gkt382

[119] Guo D, Dunn-Walters DK, Fraternali F, Ng JCF. ImmunoMatch learns and predicts cognate pairing of heavy and light immunoglobulin chains. Nature Methods. 2026;23(1):106–117. doi:10.1038/s41592-025-02913-x

[120] Berman HM. The Protein Data Bank. Nucleic Acids Research. 2000;28(1):235–242. doi:10.1093/nar/28.1.235

[121] Varadi M, Anyango S, Deshpande M, Nair S, Natassia C, Yordanova G, et al. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Research. 2022;50(D1):D439–D444. doi:10.1093/nar/gkab1061

[122] Lau AM, Bordin N, Kandathil SM, Sillitoe I, Waman VP, Wells J, et al. Exploring structural diversity across the protein universe with The Encyclopedia of Domains. Science. 2024;386:eadq4946. doi:10.1126/science.adq4946

[123] Robert PA, Akbar R, Frank R, Pavlović M, Widrich M, Snapkov I, et al. Unconstrained generation of synthetic antibody–antigen structures to guide machine learning methodology for antibody specificity prediction. Nature Computational Science. 2022;2(12):845–865. doi:10.1038/s43588-022-00372-4

[124] Wu Y, Liu X, Hrovatin K, Wu D, Linker SM, Winkel M. A Unified Dataset for Antibody and Nanobody Design Including Sequence, Structure, and Binding Affinity Data. Scientific Data. 2026;13. doi:10.1038/s41597-026-06878-0

[125] Steinegger M, Söding J. Clustering huge protein sequence sets in linear time. Nature Communications. 2018;9(1):2542. doi:10.1038/s41467-018-04964-5

[126] van Kempen M, Kim SS, Tumescheit C, Mirdita M, Lee J, Gilchrist CLM, et al. Fast and accurate protein structure search with Foldseek. Nature Biotechnology. 2024;42(2):243–246. doi:10.1038/s41587-023-01773-0

[127] Joeres R, Blumenthal DB, Kalinina OV. Data splitting to avoid information leakage with DataSAIL. Nature Communications. 2025;16(1):3337. doi:10.1038/s41467-025-58606-8

[128] Chen RTQ, Rubanova Y, Bettencourt J, Duvenaud D. Neural Ordinary Differential Equations. arXiv:1806.07366. 2018.

[129] Peebles W, Xie S. Scalable Diffusion Models with Transformers. arXiv:2212.09748. 2022.

[130] Shazeer N. GLU Variants Improve Transformer. arXiv:2002.05202. 2020.

[131] Grathwohl W, Chen RTQ, Bettencourt J, Sutskever I, Duvenaud D. FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models. arXiv:1810.01367. 2018.

[132] Loshchilov I, Hutter F. Decoupled Weight Decay Regularization. arXiv:1711.05101. 2017.

[133] Li S, Zhao Y, Varma R, Salpekar O, Noordhuis P, Li T, et al. PyTorch Distributed: Experiences on Accelerating Data Parallel Training. arXiv:2006.15704. 2020.

[134] Kalamkar D, Mudigere D, Mellempudi N, Das D, Banerjee K, Avancha S, et al. A Study of BFLOAT16 for Deep Learning Training. arXiv:1905.12322. 2019.

[135] Chen T, Xu B, Zhang C, Guestrin C. Training Deep Nets with Sublinear Memory Cost. arXiv:1604.06174. 2016.

[136] Korthikanti V, Casper J, Lym S, McAfee L, Andersch M, Shoeybi M, et al. Reducing Activation Recomputation in Large Transformer Models. arXiv:2205.05198. 2022.

[137] Chizat L, Peyré G, Schmitzer B, Vialard FX. Scaling algorithms for unbalanced optimal transport problems. Mathematics of Computation. 2018;87(314):2563–2609. doi:10.1090/mcom/3303

[138] Peyré G, Cuturi M. Computational Optimal Transport. arXiv:1803.00567. 2018.

[139] Séjourné T, Feydy J, Vialard FX, Trouvé A, Peyré G. Sinkhorn Divergences for Unbalanced Optimal Transport. arXiv:1910.12958. 2019.

[140] Assran M, Duval Q, Misra I, Bojanowski P, Vincent P, Rabbat M, et al. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture. arXiv:2301.08243. 2023.

[141] Rabinowitz S, Nigam P, Santolla N, Ford CT. Kukulu: Diffusion-Based Reconstruction of Antibody CDR Loops using a Structure-Aware Joint Embedding Predictive Architecture. bioRxiv. 2026. doi:10.64898/2026.08.03.742168

[142] Bradley RA, Terry ME. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. Biometrika. 1952;39(3/4):324. doi:10.2307/2334029

[143] Bardes A, Ponce J, LeCun Y. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. arXiv:2105.04906. 2021.

[144] Arvanitidis G, Hansen LK, Hauberg S. Latent Space Oddity: on the Curvature of Deep Generative Models. arXiv:1710.11379. 2017.

[145] Désidéri JA. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. Comptes Rendus. Mathématique. 2012;350(5-6):313–318. doi:10.1016/j.crma.2012.03.014

[146] Sener O, Koltun V. Multi-Task Learning as Multi-Objective Optimization. arXiv:1810.04650. 2018.

[147] Yao Y, Pan Y, Li J, Tsang I, Yao X. PROUD: PaRetO-gUided diffusion model for multi-objective generation. Machine Learning. 2024;113:6511–6538. doi:10.1007/s10994-024-06575-2

[148] Naesseth CA, Lindsten F, Schön TB. Elements of Sequential Monte Carlo. Foundations and Trends® in Machine Learning. 2019;12(3):187–306. doi:10.1561/2200000074

[149] Wu L, Trippe BL, Naesseth CA, Blei DM, Cunningham JP. Practical and Asymptotically Exact Conditional Sampling in Diffusion Models. arXiv:2306.17775. 2023.

[150] Skreta M, Akhound-Sadegh T, Ohanesian V, Bondesan R, Aspuru-Guzik A, Doucet A. Feynman-Kac Correctors in Diffusion: Annealing, Guidance, and Product of Experts. arXiv:2503.02819. 2025.

[151] Mark K, Galustian L, Kovar MP-P, Heid E. Feynman-Kac-Flow: Inference Steering of Conditional Flow Matching to an Energy-Tilted Posterior. arXiv:2509.01543. 2025.

[152] Hartman E, Wallin J, Malmström J, Olsson J. Controllable protein design with particle-based Feynman-Kac steering. arXiv:2511.09216. 2025.

[153] Stocco F, Garibbo M, Ferruz N. Steering generative models for protein design: Aligning and conditioning strategies. Current Opinion in Structural Biology. 2026;98:103250. doi:10.1016/j.sbi.2026.103250

[154] Kalaivanan A, Zhao Z, Sjölund J, Lindsten F. Interacting particle guidance for sampling reward-tilted generative priors. arXiv:2609.37227. 2026.

[155] Gerber M, Chopin N, Whiteley N. Negative association, ordering and convergence of resampling methods. The Annals of Statistics. 2019;47(4):2236–2260. doi:10.1214/18-aos1746

[156] Elvira V, Martino L, Robert CP. Rethinking the Effective Sample Size. International Statistical Review. 2022;90(3):525–550. doi:10.1111/insr.12500

[157] Meng C, He Y, Song Y, Song J, Wu J, Zhu JY, et al. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. arXiv:2108.01073. 2021.

[158] Xiong J, Gaur I, Lukarska M, Nisonoff H, Oltrogge LM, Savage DF, et al. ProteinGuide: On-the-fly property guidance for protein sequence generative models. arXiv:2505.04823. 2025.

[159] Sanders J, Giancardo L, Guo L, Zhao Y, Sonmez K, Cheng N, et al. Conditional generation of antibody sequences with classifier-guided germline-absorbing discrete diffusion. arXiv:2605.06720. 2026.

[160] Sinha I, Stanton S, Lillington S, Robinson SA, Nerli S, Zadorozhny K, et al. CDR Conformation Aware Antibody Sequence Design with ConformAb. bioRxiv. 2025. doi:10.1101/2025.11.12.688095

[161] Yaari G, Vander Heiden JA, Uduman M, Gadala-Maria D, Gupta N, Stern JNH, et al. Models of Somatic Hypermutation Targeting and Substitution Based on Synonymous Mutations from High-Throughput Immunoglobulin Sequencing Data. Frontiers in Immunology. 2013;4. doi:10.3389/fimmu.2013.00358

[162] Zhao X, Tang YC, Singh A, Cantu VJ, An K, Lee J, et al. AbBiBench: A Benchmark for Antibody Binding Affinity Maturation and Design. arXiv:2506.04235. 2025.

[163] Basu S, Wallner B. DockQ: A Quality Measure for Protein-Protein Docking Models. PLOS ONE. 2016;11(8):e0161879. doi:10.1371/journal.pone.0161879

[164] Eastman P, Galvelis R, Peláez RP, Abreu CRA, Farr SE, Gallicchio E, et al. OpenMM 8: Molecular Dynamics Simulation with Machine Learning Potentials. The Journal of Physical Chemistry B. 2024;128:109–116. doi:10.1021/acs.jpcb.3c06662

[165] Miller BR, McGee TD, Swails JM, Homeyer N, Gohlke H, Roitberg AE. MMPBSA.py: An Efficient Program for End-State Free Energy Calculations. Journal of Chemical Theory and Computation. 2012;8:3314–3321. doi:10.1021/ct300418h

[166] Barlow KA, Ó Conchúir S, Thompson S, Suresh P, Lucas JE, Heinonen M, et al. Flex ddG: Rosetta Ensemble-Based Estimation of Changes in Protein–Protein Binding Affinity upon Mutation. The Journal of Physical Chemistry B. 2018;122:5389–5399. doi:10.1021/acs.jpcb.7b11367

[167] Barducci A, Bussi G, Parrinello M. Well-Tempered Metadynamics: A Smoothly Converging and Tunable Free-Energy Method. Physical Review Letters. 2008;100:020603. doi:10.1103/PhysRevLett.100.020603

[168] Angelopoulos AN, Bates S, Fisch A, Lei L, Schuster T. Conformal Risk Control. arXiv:2208.02814. 2022.

[169] Angelopoulos AN, Bates S, Candès EJ, Jordan MI, Lei L. Learn then Test: Calibrating Predictive Algorithms to Achieve Risk Control. arXiv:2110.01052. 2021.

[170] Jin Y, Candès EJ. Selection by Prediction with Conformal p-values. arXiv:2210.01408. 2022.

[171] Nilsson JB, Kaabinejadian S, Yari H, Kester MGD, van Balen P, Hildebrand WH, et al. Accurate prediction of HLA class II antigen presentation across all loci using tailored data acquisition and refined machine learning. Science Advances. 2023;9(47):eadj6367. doi:10.1126/sciadv.adj6367

[172] Ramon A, Ali M, Atkinson M, Saturnino A, Didi K, Visentin C, et al. Assessing antibody and nanobody nativeness for hit selection and humanization with AbNatiV. Nature Machine Intelligence. 2024;6(1):74–91. doi:10.1038/s42256-023-00778-3

[173] Ramon A, Ni M, Predeina O, Gaffey R, Kunz P, Onuoha S, et al. Prediction of protein biophysical traits from limited data: a case study on nanobody thermostability through NanoMelt. mAbs. 2025;17:2442750. doi:10.1080/19420862.2024.2442750

[174] Alvarez JAE, Dean SN. TEMPRO: nanobody melting temperature estimation model using protein embeddings. Scientific Reports. 2024;14:19074. doi:10.1038/s41598-024-70101-6

[175] Kuhn D, Esfahani PM, Nguyen VA, Shafieezadeh-Abadeh S. Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning. arXiv:1908.08729. 2019.

[176] Huangfu Q, Hall JAJ. Parallelizing the dual revised simplex method. Mathematical Programming Computation. 2018;10:119–142. doi:10.1007/s12532-017-0130-5

[177] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods. 2020;17(3):261–272. doi:10.1038/s41592-019-0686-2

[178] Nemhauser GL, Wolsey LA, Fisher ML. An analysis of approximations for maximizing submodular set functions—I. Mathematical Programming. 1978;14(1):265–294. doi:10.1007/bf01588971

[179] Ramdas A, Grünwald P, Vovk V, Shafer G. Game-Theoretic Statistics and Safe Anytime-Valid Inference. Statistical Science. 2023;38(4):576–601. doi:10.1214/23-sts894

[180] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703. 2019.

[181] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. 2020;585:357–362. doi:10.1038/s41586-020-2649-2

[182] Urbina F, Lentzos F, Invernizzi C, Ekins S. Dual use of artificial-intelligence-powered drug discovery. Nature Machine Intelligence. 2022;4:189–191. doi:10.1038/s42256-022-00465-9

[183] Wittmann BJ, Alexanian T, Bartling C, Beal J, Clore A, Diggans J, et al. Strengthening nucleic acid biosecurity screening against generative protein design tools. Science. 2025;390:82–87. doi:10.1126/science.adu8578

[184] Baker D, Church G. Protein design meets biosecurity. Science. 2024;383:349. doi:10.1126/science.ado1671