ABPILOT: STEERING DIFFUSION CO-FOLDING MODELS FOR ANTIBODY DESIGN VIA END-TO-END DIFFERENTIATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern all-atom co-folding models have sharply improved antibody–antigen structure prediction, yet their stochastic diffusion structure decoders make direct gradient-based antibody design difficult. Existing methods therefore either rely on simpler predictors or bypass the diffusion process, underexploiting the full capacity of co-folding models. We present AbPilot, which enables end-to-end differentiation through the native diffusion sampling process and propagates atomic-level structural and confidence objectives back to antibody sequence for steered generation. The gradient reaches the coordinates the model actually emits, so the interface can be controlled at atomic resolution. Using *adjoint-state recursion*, AbPilot reduces activation memory to $\mathcal{O}(1)$ with respect to diffusion depth, and further introduces *orthogonal rectification* to mitigate predictor hacking using an external structural verification as a search signal. On a new 12-target *de novo* antibody design benchmark, AbPilot achieves *in silico* success rates more than eightfold those of the strongest baseline for both VHH and scFv design. Finally, prospective VHH design against mICAM-1, a target with no previously reported VHH binder, yields nanomolar binders, demonstrating that the proposed end-to-end differentiation can turn modern diffusion co-folding models into effective antibody design engines.

1 INTRODUCTION

Antibodies are capable of recognizing diverse targets with high specificity and favorable therapeutic properties, they have become a dominant class of binders in biomedicine and biotechnology (Hozumi & Tonegawa, 1976). Discovering antibodies against a desired antigen, however, has traditionally relied on laborious experimental pipelines such as animal immunization and display-based screening (Alexander & Leong, 2024; Wilson & Andrews, 2012). Computational antibody design provides an appealing alternative by directly co-designing binding sequences and structures conditioned on a target epitope (Luo et al., 2022; Kong et al., 2023b; Zhu et al., 2024).

Despite rapid progress, reliable epitope-specific antibody design remains challenging, with experimental success rates substantially lower than those achieved in *de novo* miniprotein design (Pacesa et al., 2025; Watson et al., 2023). A key advance underlying the success of miniprotein design is accurate co-folding models. In particular, AlphaFold-Multimer (AFM) enables methods such as BindCraft to directly optimize protein sequences by backpropagating structural and confidence objectives through the predictor, efficiently generating designs that can be folded with high confidence (Pacesa et al., 2025). This establishes a powerful paradigm where a sufficiently accurate co-folding model can itself be turned into a design model through gradient-based optimization.

Transferring this paradigm to antibodies, however, is considerably more difficult. Antibody–antigen complexes are typically harder to predict than generic protein–protein interactions, and AFM remains less reliable in this regime. Only recently have modern all-atom co-folding models (Abramson et al., 2024; Passaro et al., 2025) achieved substantially improved precision. Yet it comes with a fundamental architectural change where the deterministic structural decoder of AFM is replaced by a stochastic diffusion process. Backpropagating through this long generative trajectory is computationally prohibitive under standard automatic differentiation. Existing approaches therefore make

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

compromises, either relying on simplified AFM-like predictors (Mille-Fragoso et al., 2026; Zhang et al., 2026), or detaching the diffusion process from gradient computation (Cho et al., 2025). This raises a natural question: **can we better exploit modern diffusion co-folding models for antibody design by making them differentiable in an end-to-end manner?**

To this end, we introduce AbPilot, which steers diffusion co-folding models for antibody design via end-to-end differentiation. AbPilot propagates gradients from objectives defined on atomic structures and confidences through the complete diffusion sampling process back to the antibody sequence as the signal for generation. These structural gradients give us fine-grained control over the designed atomic structure. To make this tractable, we reformulate diffusion backpropagation using an adjoint variable and derive a recursive update over adjoint states, reducing activation memory cost to $\mathcal{O}(1)$ with respect to diffusion depth. We further introduce *orthogonal rectification* to mitigate reward hacking by *post hoc* ranking discretized candidates with an independent predictor every few rounds. This periodically re-anchors the design process to model-consistent structural regions rather than overfitting a single co-folding model.

We establish a comprehensive 12-target antibody design benchmark and systematically evaluate design models under matched computational budgets. Our AbPilot achieves $8 \sim 17\times$ higher success rates and obtains $5 \sim 8\times$ more successful designs than the state-of-the-art antibody design approaches, demonstrating substantial improvements in both performance and computational efficiency. Finally, we prospectively apply AbPilot to VHH design against mICAM-1 and experimentally validate binders with nanomolar affinities. Together, these results demonstrate that modern diffusion co-folding models can be transformed into effective antibody design engines through end-to-end differentiation.

Our main contributions are:

- **End-to-end differentiation through diffusion co-folding models**. We formulate antibody design as gradient optimization through all-atom diffusion co-folding models and derive an adjoint-based backward procedure that propagates objectives through the complete diffusion trajectory with constant memory complexity with respect to diffusion depth.
- **Orthogonal rectification for robust design**. We introduce a rectification mechanism that repeatedly corrects the optimization trajectory, reducing model hacking and improving the consistency of designs across different co-folding models.
- **Systematic computational and wetlab evaluation**. We establish a 12-target benchmark for *de novo* antibody design under matched budgets and independent evaluation criteria, demonstrate substantial gains of AbPilot over existing methods, and experimentally validate it by obtaining nanomolar VHH binders against mICAM-1.

2 Related Work

Antibody design. A major line of work co-designs CDR sequence and structure given an antibody–antigen complex with the loops masked out (Luo et al., 2022; Kong et al., 2023a; Zhu et al., 2024), typically by diffusion or iterative refinement over the masked region. They model the interface well, but the bound complex they require is what one lacks before a binder exists: they optimize a known interaction rather than design a new one. Antibody language models (Olsen et al., 2022) relax the structural requirement but provide no mechanism for targeting a chosen epitope. Structure-based *de novo* approaches remove the complex assumption: RFantibody (Bennett et al., 2026) adapts RFdiffusion (Watson et al., 2023) to generate CDR loops on a fixed framework against a target with specified hotspots, and Germinal (Mille-Fragoso et al., 2026) couples generation with filtering. Both are limited by the accuracy of the structure predictor used to screen candidates, and both report a small number of targets for *in silico* evaluation, which motivates the benchmark we introduce.

Co-folding models and gradient-based generation. Co-folding models predict biomolecular complexes with a trunk that encodes the input and a structural module that generates coordinates (Jumper et al., 2021; Passaro et al., 2025; Team et al., 2026). Design methods invert them by optimizing an input with the gradient of predefined objectives so that the model's own outputs report a favourable complex. Hallucination-style approaches (Frank et al., 2024) and BindCraft (Pacesa et al., 2025) use AlphaFold Multimer as the base model, where the structural module is deterministic and directly differentiable. Modern all-atom co-folding models, which have substantially improved antibody–antigen complex prediction, break this property by leveraging diffusion in the structural module (Abramson

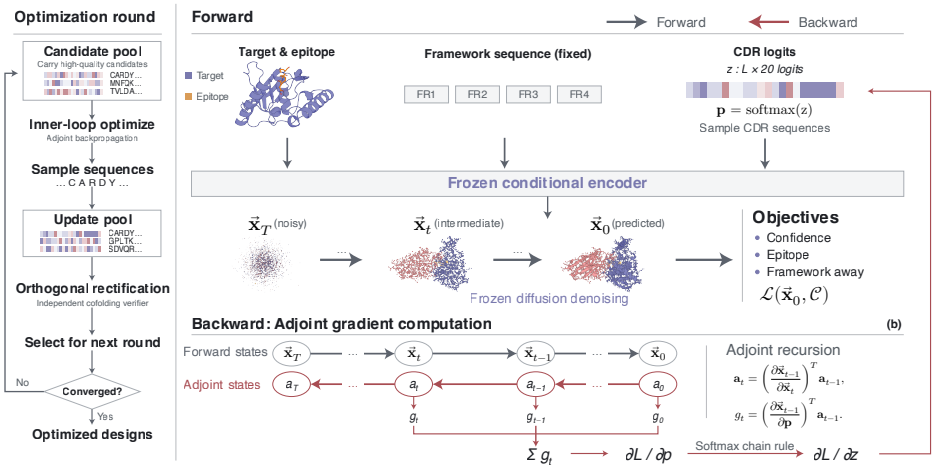

Figure 1: **Overview of AbPilot. Left:** one design run. Each round optimizes the CDR logits, samples discrete sequences, and adds high-quality ones to the pool; every few rounds an independent co-folding model rectifies the pool and the next round restarts from a candidate it certifies. **Top right:** the forward pass, from the inputs through the frozen trunk and T denoising steps to the objective. **Bottom right:** the backward pass, propagating the adjoint a_t through the sampler and accumulating g_t to reach the logits.

et al., 2024). Consequently, BoltzDesign1 (Cho et al., 2025) restricts gradients to the Pairformer and confidence heads, optimizing the distogram rather than the sampled structure. DCFold (Zhang et al., 2026) takes the alternative route of distilling the model into a single-step generator for direct differentiation, leading to a performance drop and additional training cost. AbPilot instead differentiates the diffusion process itself, leaving the base model untouched. Further discussion of backpropagation through diffusion models is provided in Appendix A.

3 Method

3.1 Task Formulation

Sampling in diffusion cofolding models. AF3-like cofolding models predict the corresponding antibody–antigen complex in two stages. First, a trunk network E_θ encodes the input sequences S into a collection of conditioning representations $\mathcal{H}$:

$$\mathcal{H} = E_\theta(S) = \left(\boldsymbol{s}_{\text{trunk}}, \boldsymbol{s}_{\text{inputs}}, \boldsymbol{q}, \boldsymbol{c}, \boldsymbol{b}_{\text{enc}}, \boldsymbol{b}_{\text{dec}}, \boldsymbol{b}_{\text{tok}}\right), \tag{1}$$

where $\boldsymbol{s}_{\text{trunk}}$ and $\boldsymbol{s}_{\text{inputs}}$ are token-level single representations; $\boldsymbol{q}$ and $\boldsymbol{c}$ denote atom- and token-level conditioning features used by the diffusion module; and $\boldsymbol{b}_{\text{enc}}$, $\boldsymbol{b}_{\text{dec}}$, and $\boldsymbol{b}_{\text{tok}}$ are attention biases for the atom encoder, atom decoder, and token transformer, respectively. Conditioned on $\mathcal{H}$, a diffusion decoder generates the all-atom coordinates through sequential denoising steps. Let $\vec{\boldsymbol{x}}_t \in \mathbb{R}^{N\times 3}$ denote the coordinates of N atoms at diffusion step t, each sampling step is written as:

$$\vec{\boldsymbol{x}}_{t-1} = D_\theta\left(\vec{\boldsymbol{x}}_t, \mathcal{H}; \sigma_t, \epsilon_t\right) = D_\theta\left(\vec{\boldsymbol{x}}_t, E_\theta(S); \sigma_t, \epsilon_t\right), \qquad t = T, \ldots, 1, \tag{2}$$

where σ_t is the churn noise level and ϵ_t collects the stochastic variables introduced at step t. In AF3-like samplers, ϵ_t includes a random rigid-body augmentation with random rotation and translation. Applying Eq. 2 recursively yields the final predicted structure:

$$\vec{\boldsymbol{x}}_0 = D_\theta^{(1)} \circ D_\theta^{(2)} \circ \cdots \circ D_\theta^{(T)}\left(\vec{\boldsymbol{x}}_T, \mathcal{H}; \sigma_T, \epsilon_T\right). \tag{3}$$

Throughout the sampling process, the sequence-dependent conditioning $\mathcal{H}$ remains fixed, while the atomic coordinates are iteratively refined from $\vec{\boldsymbol{x}}_T$ to $\vec{\boldsymbol{x}}_0$. A confidence module C_θ further takes the sampled, detached structure together with the trunk representations and predicts a collection of confidences $\mathcal{C} = C_\theta\left(\text{sg}(\vec{\boldsymbol{x}}_0), \mathcal{H}\right)$, where $\text{sg}(\boldsymbol{x})$ denotes the stop-gradient of $\boldsymbol{x}$, and $\mathcal{C}$ collectively denotes structural and interaction confidence predictions such as pLDDT, ipAE, and ipTM scores. Therefore, the complete co-folding model can be written compactly as a function F_θ mapping sequences to structures and confidences:

$$(\vec{\boldsymbol{x}}_0, \mathcal{C}) = F_\theta(S) = \left[D_\theta^{(1)} \circ \cdots \circ D_\theta^{(T)}\left(\vec{\boldsymbol{x}}_T, E_\theta(S); \sigma_T, \epsilon_T\right), C_\theta\left(\text{sg}(\vec{\boldsymbol{x}}_0), E_\theta(S)\right)\right]. \tag{4}$$

Antibody design as gradient-based optimization. Given a target with a specified epitope and an antibody framework (FR), antibody design can be formalized as optimizing the amino-acid identities at designable positions in the complementarity-determining regions (CDRs), while keeping the remaining residues fixed, to achieve desired structural patterns and high confidences. Namely,

$$\hat{S} = (\hat{S}_{\text{CDR}}, S_{\text{other}}) = \arg\min_{S} \left[\mathcal{L}_x(\vec{\boldsymbol{x}}_0) + \mathcal{L}_c(\mathcal{C})\right] = \arg\min_{S} \mathcal{L}(F_\theta(S)), \tag{5}$$

where $\mathcal{L}$ denotes the design objective that consists of $\mathcal{L}_x$ and $\mathcal{L}_c$, defined on the structure $\vec{\boldsymbol{x}}_0$ and confidences $\mathcal{C}$, respectively. To enable gradient-based optimization over the discrete sequence space, we parameterize the L designable residues with learnable logits $\boldsymbol{z} \in \mathbb{R}^{L\times 20}$ over the 20 standard amino-acid types, and relax them into probability distributions $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{z})$, from which discrete CDRs can be sampled as $S_{\text{CDR}} \sim \boldsymbol{p}$. During optimization, $\boldsymbol{p}$ is directly fed into the trunk network E_θ, allowing the sequence logits to be updated using the gradient $\partial\mathcal{L}/\partial\boldsymbol{z}$. Since $\mathcal{L}$ depends on both the final structure and the predicted confidences, computing this gradient requires backpropagation through the confidence module, the trunk network, and, critically, the complete diffusion trajectory. Efficiently obtaining this end-to-end gradient is therefore the central technical challenge.

3.2 End-to-End Differentiation through Co-folding Models

Gradient composition. The key challenge is to propagate the design objective through the complete co-folding model back to the sequence logits $\boldsymbol{z}$. Given $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{z})$, we have:

$$\mathcal{H} = E_\theta(\boldsymbol{p}), \qquad \mathcal{C} = C_\theta(\vec{\boldsymbol{x}}_0, \mathcal{H}), \qquad \mathcal{L} = \mathcal{L}(\vec{\boldsymbol{x}}_0, \mathcal{C}). \tag{6}$$

Applying the chain rule, the end-to-end gradient can be decomposed as:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{z}} &= \left(\frac{\partial\boldsymbol{p}}{\partial\boldsymbol{z}}\right)^\top \left(\frac{\partial\mathcal{H}}{\partial\boldsymbol{p}}\right)^\top \left[\frac{\partial\mathcal{L}_c}{\partial\mathcal{H}} + \frac{\partial\mathcal{L}_x}{\partial\mathcal{H}}\right] \\ &= \left(\frac{\partial\boldsymbol{p}}{\partial\boldsymbol{z}}\right)^\top \left(\frac{\partial\mathcal{H}}{\partial\boldsymbol{p}}\right)^\top \Bigg[\underbrace{\left(\frac{\partial\mathcal{C}}{\partial\mathcal{H}}\right)^\top \frac{\partial\mathcal{L}_c}{\partial\mathcal{C}}}_{\text{shortcut differentiation}} + \underbrace{\sum_{t=1}^{T} \left(\frac{\partial\vec{\boldsymbol{x}}_{t-1}}{\partial\mathcal{H}}\right)^\top \frac{\partial\mathcal{L}_x}{\partial\vec{\boldsymbol{x}}_{t-1}}}_{\text{diffusion differentiation}} \Bigg].\end{aligned} \tag{7}$$

The shortcut pathway can be computed using standard automatic differentiation, while traversing the diffusion pathway requires differentiating through the complete sampling trajectory. Naively, automatic differentiation retains the complete computation graph and incurs $\mathcal{O}(T)$ activation memory and is computationally prohibitive for gradient calculation. All derivatives are conditioned on a fixed realization of the diffusion randomness.

Discrete adjoint differentiation. To address the problem, we first define an additional variable called the adjoint state: $\boldsymbol{a}_t = \partial\mathcal{L}_x/\partial\vec{\boldsymbol{x}}_t$. Since $\vec{\boldsymbol{x}}_{t-1}$ is derived from $\vec{\boldsymbol{x}}_t$ during the diffusion denoising steps, the chain rule gives the following backward recurrence:

$$\boldsymbol{a}_t = \left(\frac{\partial\vec{\boldsymbol{x}}_{t-1}}{\partial\vec{\boldsymbol{x}}_t}\right)^\top \boldsymbol{a}_{t-1} = \left(\frac{\partial D_\theta(\vec{\boldsymbol{x}}_t, \mathcal{H}; \cdot)}{\partial\vec{\boldsymbol{x}}_t}\right)^\top \boldsymbol{a}_{t-1}, \qquad t = 1, \ldots, T. \tag{8}$$

The recursion starts from the gradient on the final sampled structure $\boldsymbol{a}_0 = \frac{\partial\mathcal{L}_x}{\partial\vec{\boldsymbol{x}}_0}$. Starting from $\boldsymbol{a}_0$, Eq. 8 recursively propagates the structural gradient backward through the entire diffusion trajectory from $\vec{\boldsymbol{x}}_0$ to $\vec{\boldsymbol{x}}_T$. With these in hand, we can rewrite the contribution of denoising step t to the gradient with respect to the trunk representations $\mathcal{H}$ as:

$$\boldsymbol{g} = \frac{\partial\mathcal{L}_x}{\partial\mathcal{H}} = \sum_{t=1}^{T} \boldsymbol{g}_t, \qquad \boldsymbol{g}_t = \left(\frac{\partial\vec{\boldsymbol{x}}_{t-1}}{\partial\mathcal{H}}\right)^\top \frac{\partial\mathcal{L}_x}{\partial\vec{\boldsymbol{x}}_{t-1}} = \left(\frac{\partial D_\theta(\vec{\boldsymbol{x}}_t, \mathcal{H}; \cdot)}{\partial\mathcal{H}}\right)^\top \boldsymbol{a}_{t-1}. \tag{9}$$

Thus, the diffusion gradient is obtained by accumulating the local gradients over all denoising steps.

Memory-efficient trajectory replay. The discrete adjoint formulation above decomposes differentiation through the entire diffusion trajectory into a sequence of local one-step gradient computations. In particular, given the adjoint state $\boldsymbol{a}_{t-1}$, a single differentiation through the t-th denoising step is sufficient to compute both the next adjoint state $\boldsymbol{a}_t$ in Eq. 8 and its gradient contribution $\boldsymbol{g}_t$ to the

trunk representations in Eq. 9. Therefore, backpropagation does not require retaining and traversing the computational graphs of all T denoising steps simultaneously. Based on this observation, we perform *trajectory replay* during backward propagation. In the forward pass, diffusion sampling is executed without retaining the computational graph, while the coordinate states $\{\vec{\boldsymbol{x}}_t\}_{t=0}^{T}$ and the stochastic variables $\{\epsilon_t\}_{t=0}^{T}$ required to reproduce each transition are recorded. Starting from $\boldsymbol{a}_0$, we then replay the denoising steps one at a time in the reverse direction. At step t, we reconstruct:

$$\vec{\boldsymbol{x}}_{t-1} = D_\theta(\vec{\boldsymbol{x}}_t, \mathcal{H}; \sigma_t, \epsilon_t), \tag{10}$$

and locally differentiate this transition with $\boldsymbol{a}_{t-1}$ to simultaneously obtain

$$\boldsymbol{a}_t = \left(\frac{\partial D_\theta(\vec{\boldsymbol{x}}_t, \mathcal{H}; \cdot)}{\partial \vec{\boldsymbol{x}}_t}\right)^\top \boldsymbol{a}_{t-1}, \qquad \boldsymbol{g}_t = \left(\frac{\partial D_\theta(\vec{\boldsymbol{x}}_t, \mathcal{H}; \cdot)}{\partial \mathcal{H}}\right)^\top \boldsymbol{a}_{t-1}. \tag{11}$$

The local computational graph is immediately released after these two vector–Jacobian products are obtained, and $\boldsymbol{a}_t$ is then passed to the next replayed step. Consequently, only the activations of a single denoising step need to reside in memory at any time. Compared with standard backpropagation, which requires $\mathcal{O}(T)$ denoiser activation memory, trajectory replay reduces the activation memory to $\mathcal{O}(1)$ with respect to diffusion depth. The recorded coordinate trajectory itself still requires $\mathcal{O}(T)$ storage, but can be neglected as it is substantially smaller than retaining the neural-network activations of the complete diffusion process. We compare the memory cost in Appendix B.

3.3 Design Objectives

With end-to-end gradients available, AbPilot performs continuous optimization over the residue logits $\boldsymbol{z}$. The objective combines $\mathcal{L}_c$ and $\mathcal{L}_x$, denoting confidences from the co-folding model and elaborated geometric constraints defined directly on the sampled structure. For $\mathcal{L}_c$, we combine confidence scores from the co-folding model to favor well-folded and confidently interacting complexes:

$$\mathcal{L}_c = w_1\ell_{\text{pLDDT}} + w_2\ell_{\text{ab-pLDDT}} + w_3\ell_{\text{cdr-pLDDT}} + w_4\ell_{\text{ipTM}} - w_5\ell_{\text{ipAE}} \tag{12}$$

The first three terms encourage confident folding of the entire complex, the antibody, and the designable CDRs, respectively, while the latter two promote a confident antibody–antigen interface. For $\mathcal{L}_x$, we encourage CDR-mediated interaction geometries on the desired epitope:

$$\mathcal{L}_x = w_6\ell_{\text{epi}} + w_7\ell_{\text{FW-away}} \tag{13}$$

The optimization of both objectives necessarily requires the gradient through the diffusion process to control interface accurately, as they act on coordinates $\vec{\boldsymbol{x}}_0$ only, which are defined as follows.

Epitope attraction. We introduce an explicit objective on the sampled coordinates to enforce binding to the desired epitope. Let $\mathcal{E}$ and $\mathcal{D}$ denote the epitope and the designable CDR. For each epitope residue i, we identify its k nearest designable CDR residues, denoted by $\mathcal{N}_k(i;\mathcal{D})$, and define:

$$d_{\text{epi}} = \frac{1}{k|\mathcal{E}|}\sum_{i\in\mathcal{E}}\sum_{j\in\mathcal{N}_k(i;\mathcal{D})}\left\|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\right\|_2, \qquad \ell_{\text{epi}} = \text{ReLU}\left(\frac{d_{\text{epi}}}{\hat{d}_{\text{epi}}^2} - 1\right). \tag{14}$$

where $\vec{\boldsymbol{x}}_i \in \mathbb{R}^3$ denotes the center-of-mass coordinate of residue i, and $\hat{d}_{\text{epi}}$ is a predefined distance threshold. Selecting the nearest k CDR residues for each epitope residue focuses the optimization on the local interaction, while the distance threshold stops attraction once the CDRs are sufficiently close to the epitope, allowing further optimization on other objectives.

Framework-away constraint. A common pitfall is that the cofolding model may attach the frameworks instead of the CDRs to the antigen, which could achieve favorable confidence scores yet is not reasonable, as only the CDRs are generated and decide the binding specificity. We thus explicitly repel the framework from the antigen. Let $\mathcal{F}$ and $\mathcal{T}$ denote the framework and target residues, for each residue $i \in \mathcal{F}$, we identify its k nearest target residues $\mathcal{N}_k(i;\mathcal{T})$ and define:

$$\ell_{\text{FW-away}} = \frac{1}{k|\mathcal{F}|}\sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{N}_k(i;\mathcal{T})}\exp\left(\frac{\hat{d}_{\text{FW-away}^2} - \left\|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\right\|_2}{\beta}\right), \tag{15}$$

where $\hat{d}_{\text{FW-away}}$ specifies the desired separation and β controls the sharpness of the repulsion. The exponential form imposes a strong penalty on close framework–target contacts while rapidly decaying for sufficiently separated residues. Together with Eq. 14, this produces a directional constraint: the specified epitope is attracted toward the designable CDRs while the non-designable framework is kept away from the target, encouraging the desired CDR-mediated binding geometry.

3.4 ITERATIVE POLICY REFINEMENT

While end-to-end differentiation enables efficient optimization of the continuous logits, repeatedly optimizing against a single co-folding model can lead to model-specific reward hacking. We thus organize antibody design as an iterative policy search process alternating between gradient-based sequence optimization and orthogonal rectifications. In each round, AbPilot first optimizes the policy (*i.e.*, residue logits $\boldsymbol{z}$) for M steps with the objectives in § 3.3, then discretizes the distributions into K candidate sequences [1] to a candidate pool that carries high-quality and diverse solutions across rounds. An independent structure predictor, serving as the orthogonal recification, periodically re-ranks the pool to correct model-specific drift. Ablations on M and K are further provided in Appendix E.3.

Inner-loop optimization. At each iteration, $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{z})$ is passed to the co-folding model, which produces the atomic structure and confidences used to evaluate the objective $\mathcal{L}$. The end-to-end differentiation described in § 3.2 then provides $\partial\mathcal{L}/\partial\boldsymbol{z}$, and the logits are updated to minimize $\mathcal{L}$:

$$\boldsymbol{z}^{(n+1)} = \boldsymbol{z}^{(n)} - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(n)}}, \qquad n = 0, \ldots, M-1, \tag{16}$$

where η denotes the learning rate. Thus, the inner loop continuously refines the residue distribution toward sequences that are predicted to form high-confidence, epitope-specific complexes, with the differentiable cofolding model serving as the source of the optimization signal.

Candidate generation. After the inner-loop optimization, the continuous logits are discretized to draw a population of K candidate antibody sequences. These candidates are collected into the candidate pool $\mathcal{P}$, which serves as a replay buffer for subsequent rounds. Those achieving promising objectives from $\mathcal{P}$ are used to initialize the residue distributions for the next round, allowing the search to iteratively refine competitive sequence regions while maintaining diverse search trajectories.

Orthogonal rectification. Although gradient optimization efficiently improves candidates according to the design objectives, repeatedly optimizing against the same co-folding model may gradually exploit its model-specific biases. We therefore periodically perform *orthogonal rectification* by evaluating the discretized candidates with an independent structure predictor. Specifically, after every few rounds, candidates in the pool are ranked according to the verifier's structural and interface confidence, and only the top-ranked candidates are retained to initialize the subsequent search. The verifier is not involved in gradient computation and thus provides an independent criterion for candidate selection. Periodical rectification according to this orthogonal signal prevents the search from drifting toward model-specific artifacts and re-anchors subsequent gradient optimization to structural regions consistently supported by an independent predictor (Algorithm 1).

4 EXPERIMENTS

We instantiate AbPilot with Boltz-2 (Passaro et al., 2025) and evaluate it from three aspects, with protocols in § 4.1. We first ask whether differentiating through the diffusion sampler provides a stronger design signal than the shortcut differentiation or distogram surrogate in § 4.2. We then establish a 12-target benchmark to compare open-source approaches for *de novo* antibody design under matched computational budgets in § 4.3. Finally, we validate AbPilot experimentally on a novel target, mICAM-1, in § 4.4, and extend the framework to non-protein targets in § 4.5.

4.1 BENCHMARK AND EVALUATION PROTOCOL

Test set. Existing *de novo* antibody design studies typically evaluate *in silico* success rates with different targets and protocols, making direct comparison difficult. We thus construct a benchmark of 12 targets adopted from previous protein binder design literature (Pacesa et al., 2025; Zambaldi et al., 2024). Ten targets are associated with predefined binding hotspots, while the remaining two are designed without an epitope constraint, covering both epitope-specific and unconstrained settings.

Baselines. We compare AbPilot against representative open-source methods that generate antibodies without requiring a predefined antibody–antigen complex. **RFantibody** (Bennett et al., 2026) and

[1] Discretization is required, not just a readout: AF3-like models gather atom features through an atom-to-token index built from discrete residue identities, which cannot be interpolated because residue types differ in atom count. Running the inner loop longer would keep optimizing $\boldsymbol{z}$ against an atom layout it no longer matches.

BoltzGen (Stark et al., 2025) directly generate CDR structures and sequences by diffusion and inverse folding, whereas Germinal (Mille-Fragoso et al., 2026) and BoltzDesign1 (Cho et al., 2025) perform gradient-based sequence optimization through co-folding models. **Germinal** is adapted from BindCraft (Pacesa et al., 2025), with AlphaFold-Multimer as the base model, while **BoltzDesign1** adopts shortcut differentiation via a distogram, neglecting the diffusion trajectories. We report three variants: **AbPilot** does not involve an external verifier, with **AbPilot + AF3 rect.** and **AbPilot + PX rect.** adopting AlphaFold3 and Protenix V1 (Team et al., 2026) for orthogonal rectifications, respectively. Details of hyperparameters are presented in Appendix E.

Evaluation protocol. All methods receive the same target structure and two typical frameworks, VHH and scFv, distributed with Germinal. The same computational budgets are assigned for each method for fair comparison, allocating 24 GPU hours on a single NVIDIA L40S for each target. All generated sequences are evaluated using AlphaFold3[2]. A candidate is considered successful if it simultaneously satisfies ipTM > 0.7, antibody pLDDT > 80, and scRMSD < 5 Å (Bennett et al., 2023), where scRMSD measures the antibody RMSD after aligning the predicted complex by the target. We primarily report the target-averaged success rates and number of successful designs, together with ipTM, antibody pLDDT, and scRMSD.

4.2 Effects of Differentiation through Diffusion

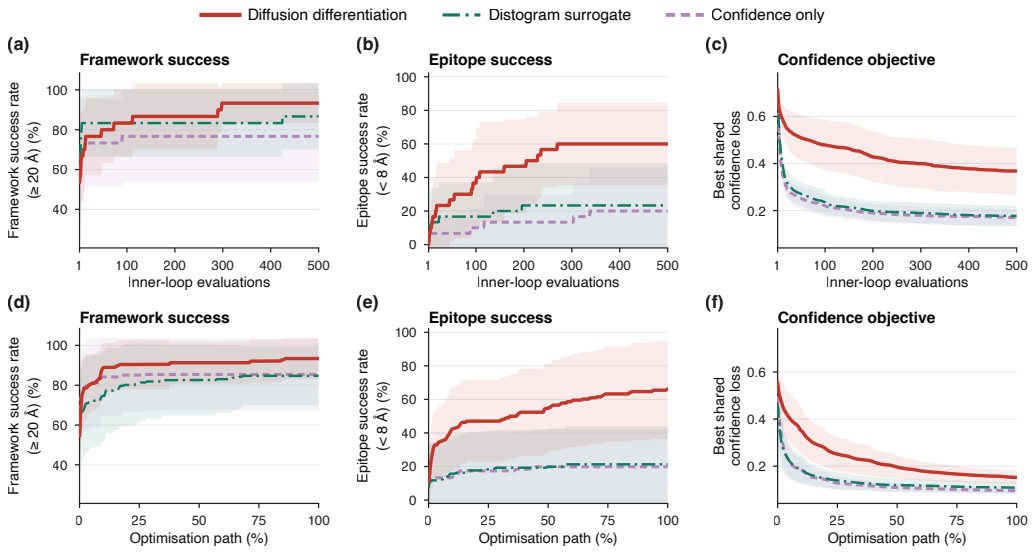

Figure 2: **Diffusion differentiation against the distogram surrogate**, **(a–c)** a fixed 500 inner-loop evaluations, **(d–f)** the full pipeline at $M = 25$ steps per round, trajectories resampled to 0–100 % progress. All quantities are best-so-far, so curves are monotone by construction.

We first isolate the role of diffusion differentiation by comparing three variants: **AbPilot** differentiates both confidence objectives $\mathcal{L}_c$ and coordinate objectives $\mathcal{L}_x$ through the complete co-folding model; **Shortcut** retains only the shortcut pathway, and consequently, $\mathcal{L}_x$, which is defined solely on the sampled coordinates, will not take effect; **Distogram surrogate** also removes diffusion differentiation, but replaces $\mathcal{L}_x$ with analogous objectives defined on the trunk-level distogram, providing an indirect geometric signal (see Appendix G). We evaluate the three variants on the ten benchmark targets with specified epitopes. Figure 2 runs one inner loop for 500 steps in the top row and the full pipeline at $M = 25$ steps per round in the bottom row.

Figure 2 shows a clear trade-off between confidence and structure loss. Without diffusion differentiation, $\mathcal{L}_c$ decreases fastest because the optimizer completely ignores $\mathcal{L}_x$. The distogram surrogate partially restores geometric control, thus slowing confidence optimization, but only weakly moves the antibody toward the desired epitope. In contrast, AbPilot directly optimizes the atomic coordinates and achieves substantially better epitope and framework geometry, indicating that differentiating the

[2]The target experimental structure is provided as a template without MSA, while the antibody only uses precomputed framework-only MSAs with the CDR positions masked, which are identical for each method.

sampler substantially improves geometric controllability. Despite stronger competition between $\mathcal{L}_c$ and $\mathcal{L}_x$, this raises the success rate on the benchmark from 19.9% to 30.7% and the accepted count from 613 to 862 (Table 5). These results show that trunk-level surrogates provide only limited control over the final structure, whereas diffusion differentiation is essential for solving the intended antibody design problem where high confidences are meaningful only when the desired geometry is achieved. (Detailed analysis in Appendix H)

4.3 De novo antibody design

Table 1: De novo antibody design on 12 targets. All numbers are target-balanced: computed per target, then averaged over targets. #Succ is the absolute count of accepted candidates and #Cand the total number evaluated; SR is target-balanced and is therefore not #Succ/#Cand. Best value is **bold** for success rate, #Succ, and the all-candidate metrics. AF3 rect. and PX rect. denote rectification against AlphaFold3 and Protenix V1 respectively.

Method	SR	#Succ	#Cand	ipTM ↑		scRMSD (Å) ↓		pLDDT ↑	
	(%)	(n)	(n)	all	succ	all	succ	all	succ
VHH									
BoltzDesign1	0.1	1	1162	0.231	0.710	34.05	2.97	75.5	85.7
BoltzGen	1.3	150	12000	0.228	0.761	20.73	2.72	83.4	90.3
Germinal	1.6	4	116	0.394	0.775	27.18	2.86	82.1	86.8
RFantibody	0.3	22	4974	0.182	0.806	35.56	3.37	**86.3**	93.3
AbPilot	21.6	944	3990	0.518	0.756	**7.05**	2.29	85.5	90.4
AbPilot + AF3 rect.	**28.4**	926	2742	**0.542**	0.767	7.99	2.53	85.5	90.0
AbPilot + PX rect.	26.1	**1138**	4232	0.517	0.768	8.70	2.52	85.2	89.8
scFv									
BoltzDesign1	0.0	0	500	0.254	–	40.54	–	76.9	–
BoltzGen	1.4	169	12000	0.234	0.771	32.44	3.45	**87.9**	91.7
Germinal	3.6	27	384	0.270	0.782	46.03	2.86	81.5	87.4
RFantibody	0.4	41	8546	0.517	0.784	41.21	3.47	86.7	91.2
AbPilot	26.0	**925**	3342	0.567	0.756	9.01	3.33	85.7	88.1
AbPilot + AF3 rect.	**28.8**	635	2202	**0.577**	0.766	**8.54**	3.13	84.8	87.3
AbPilot + PX rect.	25.3	826	2904	0.535	0.771	10.60	3.45	84.7	87.5

Table 1 reports the benchmark results for different models under the matched 24-hour GPU budget. AbPilot achieves success rates of 21.6% for VHH and 26.0% for scFv, compared with 1.6% and 3.6% for the strongest baselines, respectively. Orthogonal rectification further increases the success rates to 28.4% and 28.8%. Under the same compute budget, AbPilot also produces hundreds of successful candidates (*e.g.*, 944 VHHs and 925 scFvs without rectification), whereas state-of-the-art baselines yield at most 169 successful designs.

The all-candidate statistics further explain this gap. AbPilot reduces the average scRMSD to 7.05 Å for VHH and 9.01 Å for scFv, compared with 20.7–46.0 Å across the baselines. In contrast, once candidates satisfy the success criteria, all the methods exhibit much more similar ipTM and scRMSD values. The primary advantage of AbPilot is therefore that it reaches the successful region of sequence and structure space far more frequently. Note that some baselines, such as RFantibody, achieve stronger metrics among their accepted designs; however, these averages are computed over very few successful candidates and therefore provide limited evidence of consistently better design quality. Appendix I resolves these results by target and reports the sequence and structure diversity of the accepted designs.

External rectification provides an additional improvement in several settings. Notably, Protenix rectification increases the VHH success rate from 21.6% to 26.1% under AlphaFold3 evaluation, showing that the gain is not specific to using the final evaluator as the verifier. The effect is verifier- and format-dependent, as Protenix rectification remains comparable to the unrectified model for scFv.

4.4 Wet-Lab Validation

We next evaluate whether the computational gains of AbPilot translate into experimentally functional antibodies through prospective *de novo* VHH design against mouse ICAM-1 (mICAM-1, UniProt:

P13597), a target for which no VHH binder has previously been reported (Eswaran et al., 2025). Starting from the same VHH framework as before, we generate 100 computationally successful candidates and experimentally test the top 50 designs with the highest AlphaFold3 ipTM. Six designs exhibit ELISA signals more than threefold above the negative control (*i.e.* show reliable binding signal), corresponding to a 12% experimental hit rate (Fig. 11a). The strongest design, nb45, binds mICAM-1 with $K_{\mathrm{D}} = 65.8$ nM (Fig. 11b). AlphaFold3 independently supports the designed complex with ipTM 0.86 and scRMSD 1.51 Å, showing strong agreement between the AbPilot-generated structure and an external predictor (Fig. 11c). These results demonstrate that the improved *in silico* success of AbPilot can translate into nanomolar antibody binders in a prospective *de novo* setting.

4.5 Beyond protein targets

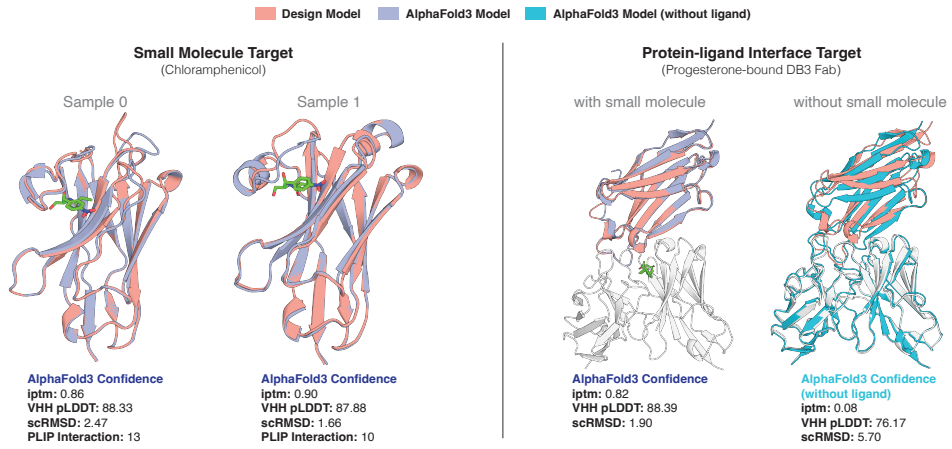

Figure 3: **Design against non-protein targets.** Design models (red) superposed on AlphaFold3 predictions (blue). **Left:** two nanobodies designed against chloramphenicol, with the number of PLIP contacts to the ligand. **Right:** a VHH designed against the progesterone-bound DB3 Fab (grey).

A key advantage of steering all-atom cofolding models is that the same design framework naturally extends beyond protein-only targets. We first design **VHHs against the small molecule**, chloramphenicol (Swofford et al., 2022). Independent AlphaFold3 predictions reproduce two representative designed complexes with ipTM values of 0.86 and 0.90 and scRMSDs of 2.47 and 1.66 Å, respectively (Fig. 3. PLIP (Schake et al., 2025) identifies 13 and 10 non-covalent contacts between the designed CDRs and chloramphenicol, indicating that the antibodies form structured binding pockets around the ligand rather than merely approaching its position. We further consider **recognition of a protein–ligand interface** on the progesterone-bound DB3 Fab (Marchand et al., 2025), with the specified epitope spanning both the protein and ligand. With progesterone (*i.e.*, the ligand) present, the designed complex is well supported by AlphaFold3 with ipTM 0.82 and scRMSD 1.90 Å. Yet removing progesterone reduces ipTM to 0.08 and increases scRMSD to 5.70 Å (Fig. 3, right). This control suggests that the designed VHH recognizes the desired ligand-bound state rather than the Fab surface alone, demonstrating that AbPilot can target composite molecular states beyond conventional protein epitopes.

5 Conclusion

We introduced AbPilot, which enables antibody design by end-to-end differentiation through diffusion co-folding models. Using discrete adjoint differentiation, orthogonal rectification, and confidence as well as geometric objectives, AbPilot turns a frozen co-folding model into an effective design engine and substantially improves *de novo* antibody design on a 12-target benchmark, with prospective validation yielding nanomolar VHH binders against mICAM-1. Importantly, AbPilot is model-agnostic: as stronger co-folding models emerge, they can be directly incorporated as new design backbones without retraining a task-specific generator. This provides a scalable route for continuously translating advances in biomolecular structure prediction into improved molecular design.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript for clarity, grammar, and readability. We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims or proofs, design research methodology or experiments, implement methods, or interpret experimental results. Other tasks requiring disclosure are not applicable to this work. All AI-assisted edits were reviewed and verified by the authors. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024.

Elena Alexander and Kam W Leong. Discovery of nanobodies: a comprehensive review of their applications and potential over the past five years. *Journal of Nanobiotechnology*, 22(1):661, 2024.

Nathaniel R Bennett, Brian Coventry, Inna Goreshnik, Buwei Huang, Aza Allen, Dionne Vafeados, Ying Po Peng, Justas Dauparas, Minkyung Baek, Lance Stewart, et al. Improving de novo protein binder design with deep learning. *Nature Communications*, 14(1):2625, 2023.

Nathaniel R Bennett, Joseph L Watson, Robert J Ragotte, Andrew J Borst, DéJenaé L See, Connor Weidle, Riti Biswas, Yutong Yu, Ellen L Shrock, Russell Ault, et al. Atomically accurate de novo design of antibodies with rfdiffusion. *Nature*, 649(8095):183–193, 2026.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Zander W Blasingame and Chen Liu. Adjointdeis: Efficient gradients for diffusion models. *Advances in Neural Information Processing Systems*, 37:2449–2483, 2024.

Yehlin Cho, Martin Pacesa, Zhidian Zhang, Bruno E Correia, and Sergey Ovchinnikov. Boltzdesign1: Inverting all-atom structure prediction model for generalized biomolecular binder design. *BioRxiv*, pp. 2025–04, 2025.

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024.

Sreepradha Eswaran, Laura Gebert, Sarah Schraven, Nicole Treichel, Thomas Ritz, Sabine Hamm, Agnes Seeger, Fabian Kiessling, Thomas Clavel, Stephan Dreschers, et al. Intercellular adhesion molecule-1 protects against adipose tissue inflammation and insulin resistance but promotes liver disease activity in western-diet fed mice. *Scientific reports*, 15(1):25884, 2025.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in neural information processing systems*, 36:79858–79885, 2023.

Christopher Frank, Ali Khoshouei, Lara Fuβ, Dominik Schiwietz, Dominik Putz, Lara Weber, Zhixuan Zhao, Motoyuki Hattori, Shihao Feng, Yosta de Stigter, et al. Scalable protein design using optimization in a relaxed sequence space. *Science*, 386(6720):439–445, 2024.

Nobumichi Hozumi and Susumu Tonegawa. Evidence for somatic rearrangement of immunoglobulin genes coding for variable and constant regions. *Proceedings of the National Academy of Sciences*, 73(10):3628–3632, 1976.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Xiangzhe Kong, Wenbing Huang, and Yang Liu. End-to-end full-atom antibody design. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 17409–17429, 2023a.

Xiangzhe Kong, Wenbing Huang, and Yang Liu. Conditional antibody design as 3d equivariant graph translation. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=LFHFQbjxIiP.

Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35:9754–9767, 2022.

Anthony Marchand, Stephen Buckley, Arne Schneuing, Martin Pacesa, Maddalena Elia, Pablo Gainza, Evgenia Elizarova, Rebecca M Neeser, Pao-Wan Lee, Luc Reymond, et al. Targeting protein–ligand neosurfaces with a generalizable deep learning tool. *Nature*, 639(8054):522–531, 2025.

Luis S Mille-Fragoso, Claudia L Driscoll, John N Wang, Haoyu Dai, Talal Widatalla, Jim L Zhang, Xiaowei Zhang, Bing Rao, Liang Feng, Brian L Hie, et al. Efficient generation of epitope-targeted antibodies with germinal. *Nature biotechnology*, pp. 1–10, 2026.

Tobias H. Olsen, Iain H. Moal, and Charlotte M. Deane. AbLang: an antibody language model for completing antibody sequences. *Bioinformatics Advances*, 2(1):vbac046, 2022.

Martin Pacesa, Lennart Nickel, Christian Schellhaas, Joseph Schmidt, Ekaterina Pyatova, Lucas Kissling, Patrick Barendse, Jagrity Choudhury, Srajan Kapoor, Ana Alcaraz-Serna, et al. One-shot design of functional protein binders with bindcraft. *Nature*, 646(8084):483–492, 2025.

Jiachun Pan, Jun Hao Liew, Vincent Tan, Jiashi Feng, and Hanshu Yan. Adjointdpm: Adjoint sensitivity method for gradient backpropagation of diffusion probabilistic models. In *International Conference on Learning Representations*, volume 2024, pp. 2653–2674, 2024.

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

Philipp Schake, Sarah Naomi Bolz, Katja Linnemann, and Michael Schroeder. Plip 2025: introducing protein–protein interactions to the protein–ligand interaction profiler. *Nucleic Acids Research*, 53 (W1):W463–W465, 2025.

Hannes Stark, Felix Faltings, MinGyu Choi, Yuxin Xie, Eunsu Hur, Timothy O'Donnell, Anton Bushuiev, Talip Uçar, Saro Passaro, Weian Mao, et al. Boltzgen: Toward universal binder design. *BioRxiv*, 2025.

Charles A Swofford, Sarah A Nordeen, Lu Chen, Mahaam M Desai, Joanna Chen, Stacy L Springs, Thomas U Schwartz, and Anthony J Sinskey. Structure and specificity of an anti-chloramphenicol single domain antibody for detection of amphenicol residues. *Protein Science*, 31(11):e4457, 2022.

Protenix Team, Yuxuan Zhang, Chengyue Gong, Hanyu Zhang, Wenzhi Ma, Zhenyu Liu, Xinshi Chen, Jiaqi Guan, Lan Wang, Yanping Yang, et al. Protenix-v1: Toward high-accuracy open-source biomolecular structure prediction. *bioRxiv*, pp. 2026–02, 2026.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Patrick C Wilson and Sarah F Andrews. Tools to therapeutically harness the human antibody response. *Nature Reviews Immunology*, 12(10):709–719, 2012.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Vinicius Zambaldi, David La, Alexander E Chu, Harshnira Patani, Amy E Danson, Tristan OC Kwan, Thomas Frerix, Rosalia G Schneider, David Saxton, Ashok Thillaisundaram, et al. De novo design of high-affinity protein binders with alphaproteo. *arXiv preprint arXiv:2409.08022*, 2024.

Zhe Zhang, Yuanning Feng, Yuxuan Song, Keyue Qiu, Hao Zhou, and Wei-Ying Ma. DCFold: Efficient protein structure generation with single forward pass. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LMsdys7t1L.

Tian Zhu, Milong Ren, and Haicang Zhang. Antibody design using a score-based diffusion model guided by evolutionary, physical and geometric constraints. In *Forty-first International Conference on Machine Learning*, 2024.

A Discussion on Backpropagation through Diffusion Models

A diffusion model can be aligned to a downstream objective by policy gradient over the denoising chain (Black et al., 2024; Fan et al., 2023), or, when the objective is differentiable, by backpropagating through sampling. The chain is deep, so methods of the latter kind differentiate only part of it (Xu et al., 2023; Clark et al., 2024). A second line leaves the model untouched and solves the *continuous* adjoint equations instead: AdjointDPM (Pan et al., 2024) through an augmented ODE, AdjointDEIS (Blasingame & Liu, 2024) through bespoke exponential-integrator solvers.

Two things separate AbPilot from this second line. Their gradient is the gradient of the continuous problem, recovered by a solver of their own, whereas ours is the discrete adjoint of the discretization the forward pass ran and is exact for the map actually applied. Their solvers are moreover derived for the flow of a diffusion ODE or SDE, and the AF3 sampler is not such a flow: a freshly drawn rigid-body augmentation is applied between updates, so the trajectory is a composition of discrete maps. We replay that randomness and differentiate the composition directly, which Appendix C shows to be a condition for the result to be a gradient at all.

B Memory scaling of the adjoint backward pass

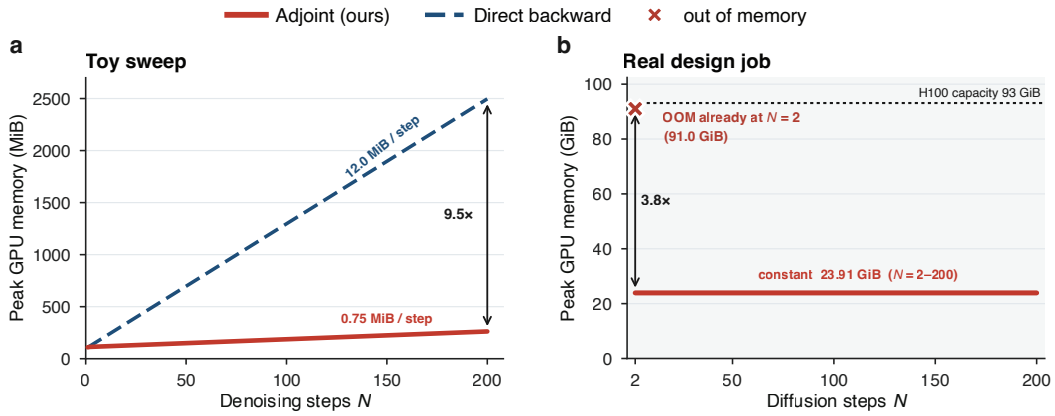

Figure 4: **The adjoint decouples memory from denoising depth.** Peak allocated GPU memory for the two ways of computing $\partial \ell / \partial Z$, on a toy model **(a)** and a real design job **(b)**, where direct backward runs out of memory at $N = 2$.

Section 3.2 claims that the adjoint removes the $\mathcal{O}(T)$ activation cost of differentiating through the sampler. We verify this first on a small surrogate, since unrolling an AF3-scale sampler is already prohibitive, and then on a design job itself. Both implementations compute the same $\partial \ell / \partial Z$ from identical inputs and identical diffusion noise, and we record peak allocated GPU memory on an NVIDIA H100 over three runs.

On the surrogate the two agree to a maximum absolute difference of 6×10^{-11}, so the comparison is between two routes to one quantity. Their scaling is not comparable. Unrolling grows at 12.0 MiB per denoising step, since the activations of every step are held until the backward pass reaches them, whereas the adjoint grows at 0.75 MiB, the cost of the cached x_t and ε_t from which each step is recomputed (Fig. 4a). Only one step's graph is alive at any moment. The gap widens from 217 against 119 MiB at $T = 10$ to 2497 against 262 MiB at $T = 200$, a factor of 9.5.

Absolute values depend on the size of the surrogate; the scaling is what transfers. On a real design job, an scFv (248 amino acids) against DerF7 (196 amino acids), unrolling never gets off the ground: at $T = 2$, the shallowest schedule we measure, it has already allocated 91.0 GiB of the 93 GiB on an H100 and runs out of memory there, so every deeper schedule is unreachable—the schedules of Appendix E.2 and the $T = 200$ default of the base model (Fig. 4b) alike. The adjoint holds at 23.91 GiB across the whole sweep, $T = 2$–200: the peak is identical at all 199 depths, and 3.8× below unrolling at the one depth where the two can be compared at all. Differentiating the sampler

on complexes of this size is therefore not a matter of how much memory one is willing to spend, but of removing the term that scales with the schedule.

C CORRECTNESS OF THE ADJOINT GRADIENT

Eq. 8 is a hand-written backward pass, so we check it against full backpropagation through the unrolled sampler, which is exact by construction. The check runs on a small surrogate, the same as Appendix B.

The surrogate keeps the structure of Eq. 8 and nothing else: a trunk maps E_θ to a conditioning pair held fixed thereafter, each step centres the state, applies a random rotation and translation, injects noise and passes the result through a nonlinearity, and a readout is taken on the terminal state and on the conditioning. Both routes of Eq. 7 are therefore present, and $\vec{x}_t$ is passed as an argument exactly as `random_aug` and `eps` are in our implementation. Table 2 reports agreement at `float32` round-off, with no degradation as T grows.

We also repeat it with the backward pass redrawing ϵ_t instead of replaying it. This is the natural implementation, and it is silently wrong: the recomputed step becomes a different function. The error rises by six orders of magnitude while the cosine similarity stays near one, so the gradient is not mis-scaled but does not points elsewhere. That gap is why record-and-replay appears in Section 3.2 as a requirement.

Table 2: Adjoint gradient against full backpropagation through the unrolled sampler. Left: the noise recorded in the forward pass is replayed in the backward pass. Right: the backward pass redraws it.

	replay (ours)		redraw (control)	
T	rel. L_2	cosine	rel. L_2	cosine
5	1.5×10^{-8}	1.0000	4.9×10^{-2}	0.9988
10	4.3×10^{-8}	1.0000	1.4×10^{-2}	0.9999
25	2.7×10^{-8}	1.0000	1.7×10^{-1}	0.9862
50	3.7×10^{-8}	1.0000	3.2×10^{-2}	0.9997

D TEST SET TARGET INFORMATION

Table 3 lists the targets with their design templates and hotspot residues.

E HYPERPARAMETER SENSITIVITY

Four quantities control the computational behavior of AbPilot: the number of encoder recycling iterations R, the number of diffusion steps T, the number of gradient steps M per round, and the number of sequences K sampled at each discretization. The main benchmark uses a single global configuration, $R = 1$, $T = 25$, $M = 25$, and $K = 5$, for every target without target-specific tuning. The experiments in this section characterize the sensitivity of AbPilot around this operating point and the associated quality–compute trade-offs, rather than seeking a separate configuration for individual benchmark targets.

We analyze the four quantities in two groups because they affect different parts of the design loop. R and T determine the cost and gradient behavior of each inner optimization step, and are therefore studied with the outer loop disabled so that their effect can be read directly from the continuous optimization trajectory. M and K govern how frequently the continuous policy is discretized and how many candidates are evaluated in the outer loop, and are therefore studied end to end. Overall, the experiments reveal broad operating regimes rather than sharply localized optima, indicating that the main conclusions are not sensitive to precise choices of these quantities.

Table 3: Targets and epitope definitions in the VHH benchmark.

Target ID	Biological target and local structural input	Origin	Source-reported hotspot residues
BHRF1	Epstein–Barr virus BHRF1, a viral BCL-2 homolog (2WH6, chain A, residues 2–158)	AlphaProteo	A65, A74, A77, A82, A85, A93
CLN1-14	Engineered soluble analogue of human claudin-1 (local AlphaFold2 model, chain A, residues 1–188)	BindCraft	A31, A46, A55, A152
CrSAS-6	N-terminal domain of *Chlamydomonas reinhardtii* SAS-6 (3Q0Y-derived local input, chain A, residues 15–159)	BindCraft	None
DerF7	Group 7 allergen from the house-dust mite *Dermatophagoides farinae* (AlphaFold2 model using 3UV1 as a template; local chain A, residues 1–196)	BindCraft	A132 or A178 or A201[a]
EGFR	Human epidermal growth factor receptor extracellular fragment (5XWD, chain A, residues 310–501)	BindCraft extension[b]	Not reported
IFNAR2	Human interferon alpha/beta receptor 2 ectodomain (2LAG, chain A, residues 8–110)	BindCraft	A52, A80, A82, A84, A96, A98
IL7RA	Human interleukin-7 receptor subunit alpha ectodomain (3DI3, chain B, residues 17–209)	AlphaProteo	B58, B80, B139
Insulin receptor	Human insulin receptor L1 domain (4ZXB, chain E, residues 6–155)	AlphaProteo	E64, E88, E96
PD-1	Human programmed cell death protein 1 ectodomain (AlphaFold2 model of source residues 32–146; local chain A, residues 1–115)	BindCraft	A64, A126, A129, A133
PD-L1	Human programmed death-ligand 1 ectodomain (5O45, local chain A, residues 18–132)	AlphaProteo	A56, A115, A123
SC2_RBD	SARS-CoV-2 spike receptor-binding domain (6M0J, chain E, residues 333–526)	AlphaProteo	E485, E489, E494, E500, E505
TrkA	Human tropomyosin receptor kinase A extracellular immunoglobulin-like domain (1WWW, chain X, residues 282–382)	AlphaProteo	X294, X296, X333

[a] BindCraft reports the three Der f 7 hotspots as alternative single-site settings ("132 or 178 or 201"); we select all three simultaneously.
[b] EGFR is a local extension of the BindCraft set with public validation records, but is not among the targets in BindCraft Supplementary Table 1.

E.1 Encoder recycling

We vary $R \in \{1, 2, 3\}$ with the sampler fixed at $T = 20$, and optimize the residue logits for 200 continuous steps with the outer loop disabled on the ten targets with specified epitopes (Fig. 5). Increasing the number of recycling iterations consistently increases computational cost, from 9.3 ± 2.0 s per inner step at $R = 1$ to 15.9 ± 4.0 s at $R = 3$, because every additional recycle contributes to both the forward computation and the trunk backward pass in Eq. 7.

Across the tested range, additional recycling does not improve the optimization trajectories: $R = 1$ achieves comparable or better interface ipTM, binder pLDDT, interface pAE, and epitope distance than deeper recycling (Fig. 5a–d). Thus, the operating point used in the main experiments does not rely on additional encoder recycling and lies in the most compute-efficient regime of this sweep (Fig. 5f).

E.2 Diffusion steps

We varied $T \in \{10, 20, 25, 30, 35, 40, 50, 100\}$ at $R = 1$ over 200 inner steps (Fig. 6).

Cost rises from 8.9 s per inner step at $T = 10$ to 13.5 s at $T = 50$, then jumps to 20.1 s at $T = 100$ (Fig. 6e). The two families of objectives disagree about the best schedule. Confidence improves monotonically with T, with the interface ipTM rising from 0.24 to 0.52 (Fig. 6a–c), while the epitope distance is lowest in the middle of the range and drifts upward for $T \geq 40$ (Fig. 6d). At $T = 10$, everything fails, including the fold.

The gradient reaching the logits arrives by two routes (Eq. 7). One passes through the sampler and carries the geometry; the other runs from the confidence head straight to the trunk and does not. We measured the size of each. The route through the sampler is $98\times$ the other at $T = 10$ and $11\times$ at $T = 20$, but $24\times$ smaller at $T = 50$ and $887\times$ smaller at $T = 100$ (Fig. 6f), because every step adds a Jacobian factor to Eq. 9 and the adjoint decays geometrically along the chain. Long schedules thus leave the optimizer working through the trunk alone, exactly as the distogram surrogate of Section 4.2

Algorithm 1 ABPILOT Pipeline

Require: Target epitope E; fixed residues S_{other}; L designable CDR positions; frozen $\{E_\theta, D_\theta, C_\theta\}$; T, M, K, η; independent structure predictor $\mathcal{V}$; ranking score $r_{\mathcal{V}}(S)$; retention size B.
Ensure: Optimized antibody designs $\hat{S} = (\hat{S}_{\text{CDR}}, S_{\text{other}})$.

Initialize residue logits $\boldsymbol{z} \in \mathbb{R}^{L\times 20}$ and candidate pool $\mathcal{P} \leftarrow \varnothing$.
for each optimization round **do**
  $\boldsymbol{z}^{(0)} \leftarrow \boldsymbol{z}$
  for $n = 0, \dots, M-1$ **do** ▷ *Inner-loop optimization*
    $\boldsymbol{p}^{(n)} \leftarrow \mathrm{softmax}(\boldsymbol{z}^{(n)}), \quad \mathcal{H} \leftarrow E_\theta(\boldsymbol{p}^{(n)})$
    Sample $\boldsymbol{x}_T$ and $\{\epsilon_t\}_{t=1}^T$; generate $\boldsymbol{x}_{t-1} \leftarrow D_\theta(\boldsymbol{x}_t, \mathcal{H}; \sigma_t, \epsilon_t), \quad t = T, \dots, 1$
    Cache $\{\boldsymbol{x}_t\}_{t=0}^T$ and $\{\epsilon_t\}_{t=1}^T$
    $C \leftarrow C_\theta(\mathrm{sg}(\boldsymbol{x}_0), \mathcal{H}), \quad \mathcal{L} \leftarrow \mathcal{L}_x + \mathcal{L}_c$
    $\boldsymbol{a}_0 \leftarrow \dfrac{\partial \mathcal{L}_x}{\partial \boldsymbol{x}_0}, \quad \boldsymbol{g} \leftarrow 0$ ▷ *Adjoint replay*
    for $t = 1, \dots, T$ **do**
      Replay $\boldsymbol{x}_{t-1} \leftarrow D_\theta(\boldsymbol{x}_t, \mathcal{H}; \sigma_t, \epsilon_t)$
      $\boldsymbol{a}_t \leftarrow \left(\dfrac{\partial D_\theta(\boldsymbol{x}_t, \mathcal{H}; \cdot)}{\partial \boldsymbol{x}_t}\right)^\top \boldsymbol{a}_{t-1}$
      $\boldsymbol{g}_t \leftarrow \left(\dfrac{\partial D_\theta(\boldsymbol{x}_t, \mathcal{H}; \cdot)}{\partial \mathcal{H}}\right)^\top \boldsymbol{a}_{t-1}, \quad \boldsymbol{g} \leftarrow \boldsymbol{g} + \boldsymbol{g}_t$
      Free the local computation graph
    end for
    $\dfrac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(n)}} \leftarrow \left(\dfrac{\partial \boldsymbol{p}^{(n)}}{\partial \boldsymbol{z}^{(n)}}\right)^\top \left(\dfrac{\partial \mathcal{H}}{\partial \boldsymbol{p}^{(n)}}\right)^\top \left[\left(\dfrac{\partial C}{\partial \mathcal{H}}\right)^\top \dfrac{\partial \mathcal{L}_c}{\partial C} + \boldsymbol{g}\right]$
    $\boldsymbol{z}^{(n+1)} \leftarrow \boldsymbol{z}^{(n)} - \eta \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(n)}}$
  end for
  $\boldsymbol{p}^\star \leftarrow \mathrm{softmax}(\boldsymbol{z}^{(M)})$; sample $S_{\text{CDR}}^{(k)} \sim \boldsymbol{p}^\star, \quad k = 1, \dots, K$
  $\mathcal{P} \leftarrow \mathcal{P} \cup \left\{(S_{\text{CDR}}^{(k)}, S_{\text{other}})\right\}_{k=1}^K$
  if rectification is performed **then** ▷ *Orthogonal rectification*
    Run $\mathcal{V}$ on each $S \in \mathcal{P}$
    $r_{\mathcal{V}}(S) \leftarrow$ ranking score from $\mathcal{V}(S), \quad \forall\, S \in \mathcal{P}$
    $\mathcal{P} \leftarrow \mathrm{Top}_B(\mathcal{P}; r_{\mathcal{V}})$
  end if
  Initialize $\boldsymbol{z}$ for the next round from a retained candidate in $\mathcal{P}$
  if converged or computational budget exhausted **then**
    break
  end if
end for
return selected optimized designs from $\mathcal{P}$.

does, which is why confidence keeps improving while the epitope is abandoned. Short ones let the geometric gradient arrive undamped, and the updates destroy the fold. We use $T = 25$, where the two routes stay within an order of magnitude of each other at 10.5 s per inner step.

E.3 GRADIENT STEPS AND CANDIDATES PER ROUND

We next examine the two quantities controlling the outer-loop discretization. Using $R = 1$ and $T = 20$[3], we evaluate $M \in \{10, 25, 50, 100\}$ and $K \in \{1, 5, 10\}$ end to end on the 12-target benchmark (Fig. 7). This analysis is intended to measure robustness to the discretization frequency and candidate throughput; all targets share the same (M, K) configuration within each run.

Performance is broadly stable across the tested range. After marginalizing over K, the AlphaFold3 success rate varies only from 19.8% to 21.1% across a ten-fold range of M, showing limited sensitivity to the number of gradient steps between discretization events (Fig. 7c). Increasing K has a larger effect when moving from one to several candidates per round, but the improvement

[3]This sweep predates the schedule sweep of Appendix E.2, which later settled on $T = 25$. The two schedules are adjacent, and the objectives are flat between them (Fig. 6)

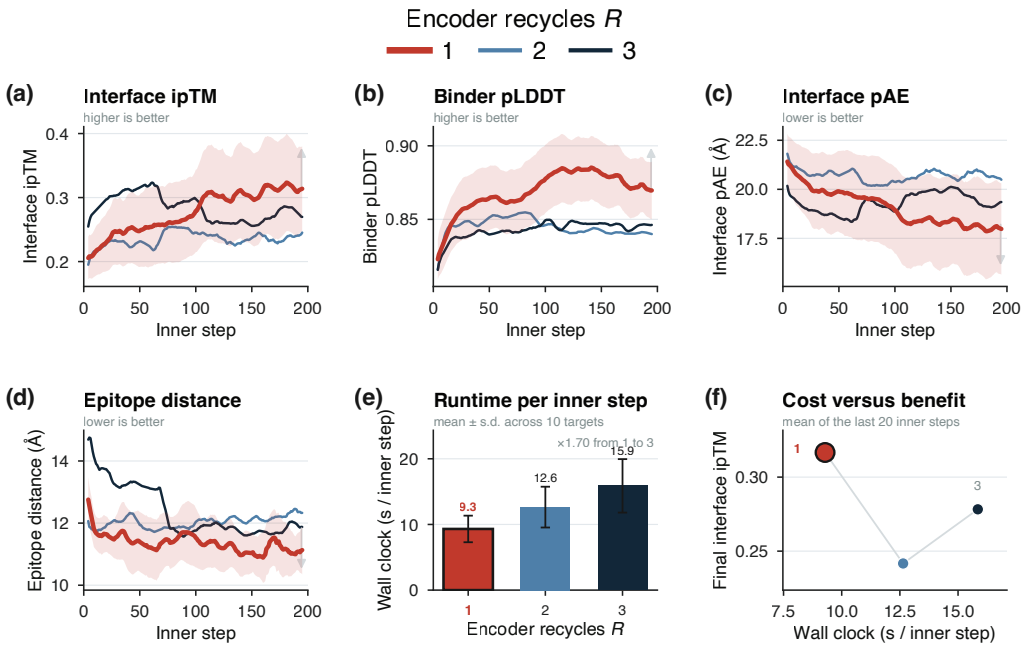

Figure 5: **Sensitivity to encoder recycling R.** The sweep is performed at $T = 20$ over 200 inner steps with the outer loop disabled on the ten targets with specified epitopes. (a–d) Objective trajectories averaged over targets; the band shows the spread across targets for $R = 1$. (e) Wall-clock time per inner step. (f) Final interface ipTM against computational cost. Increasing recycling does not improve the optimization trajectory over the tested range while substantially increasing runtime.

saturates quickly: the marginal success rates are 16.9%, 22.4%, and 22.3% for $K = 1$, 5, and 10, respectively (Fig. 7d). The two-dimensional grid therefore exhibits a broad plateau rather than a narrowly localized optimum (Fig. 7a).

The same trend is visible when accounting for candidate-evaluation cost. Larger K increases the number of sequences screened by AlphaFold3 nearly proportionally, whereas the gain in accepted designs saturates beyond moderate candidate counts (Fig. 7b). The main configuration, $M = 25$ and $K = 5$, lies within this broad high-performing regime and provides a moderate point on the cost–yield trade-off, producing 1,204 successful designs from 4,404 AlphaFold3-screened candidates in this analysis.

F Baseline details

All methods are run from their public repositories. To keep the comparison fair, CDR lengths are fixed to the natural lengths of the framework for every method, although AbPilot itself supports variable lengths.

AbPilot. Our implementation builds on the Boltz-2 repository (Passaro et al., 2025).[4] Loss weights are given in Table 4. Three rectification models are reported: Boltz-2 under its own default inference settings in the base configuration, AlphaFold3 in + *AF3 rect.*, and Protenix V1 in + *PX rect.* Each receives the target structure as a template together with the sequences of the full complex. Boltz-2 and AlphaFold3 are given no MSA; Protenix V1 is additionally given the precomputed MSA of the VHH or scFv framework, without which it does not fold the antibody into an Ig-like domain.

[4] https://github.com/jwohlwend/boltz

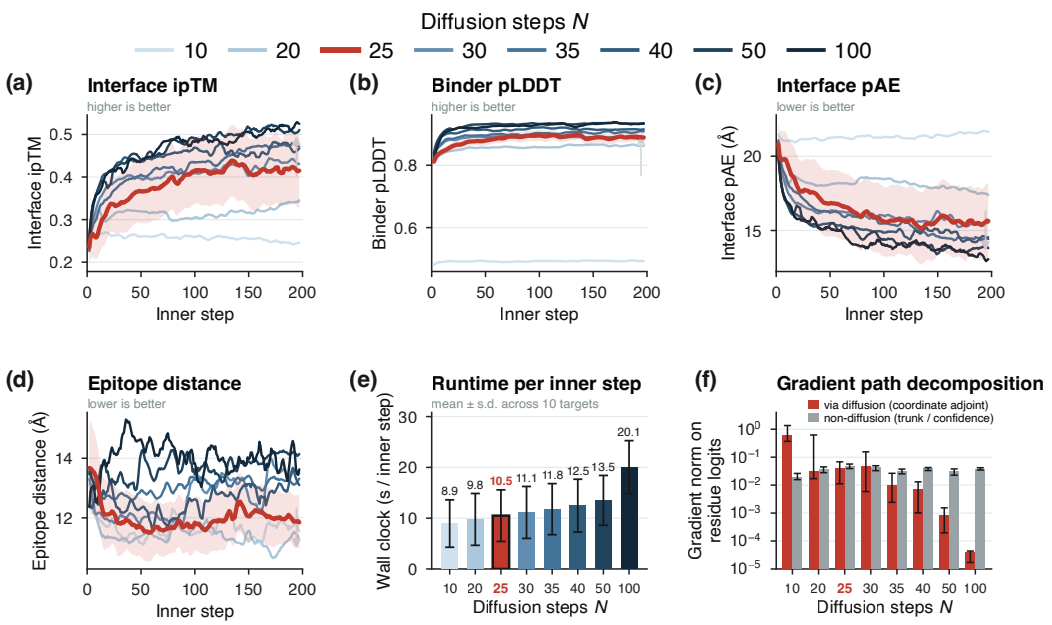

Figure 6: **Sensitivity to diffusion depth T.** The sweep is performed at $R = 1$ over 200 inner steps with the outer loop disabled on the ten targets with specified epitopes. (a–d) Objective trajectories averaged over targets; the band shows the spread across targets for $T = 25$. (e) Wall-clock time per inner step. (f) Norm of the gradient reaching the residue logits, decomposed into the pathway through the sampler (red) and the pathway from the confidence head to the trunk (grey), shown on a log scale. Intermediate diffusion depths balance direct geometric control and confidence optimization.

Table 4: Terms of training objectives and their weights.

Term	Path	Quantity	Weight
pLDDT	confidence	global predicted lDDT	0.1
Chain pLDDT	confidence	pLDDT restricted to the designed chain	0.3
Generated-part pLDDT	confidence	pLDDT on the designed CDR positions	0.5
Cross-chain ipTM	confidence	ipTM over the specified chain pairs	0.5
Interface pAE	confidence	predicted aligned error across the interface	0.2
Framework clearance	coordinate	penalty on framework–target proximity, $k = 3$, $\hat{d}_{\text{FW-away}} = 8$	0.5
Epitope	coordinate	soft epitope distance, $k = 3$, $\hat{d}_{\text{epi}} = 12$ Å	1.0

BoltzDesign1. We follow the Example 2 workflow of the design notebook in the official repository,[5] modifying only the design mask so that the CDRs of a VHH or scFv are the designed positions. All other settings are left at their defaults.

BoltzGen. We use the `nanobody-anything` design script from the official repository[6] at its default configuration. BoltzGen produces hundreds of thousands of candidates within the compute budget, so we rank them by its own score and evaluate the top 1,000 for each target.

RFantibody. We use the default scripts and settings of the official repository.[7]

Germinal. We use the default scripts and settings of the official repository.[8]

[5] https://github.com/yehlincho/BoltzDesign1
[6] https://github.com/HannesStark/boltzgen
[7] https://github.com/RosettaCommons/RFantibody
[8] https://github.com/SantiagoMille/germinal

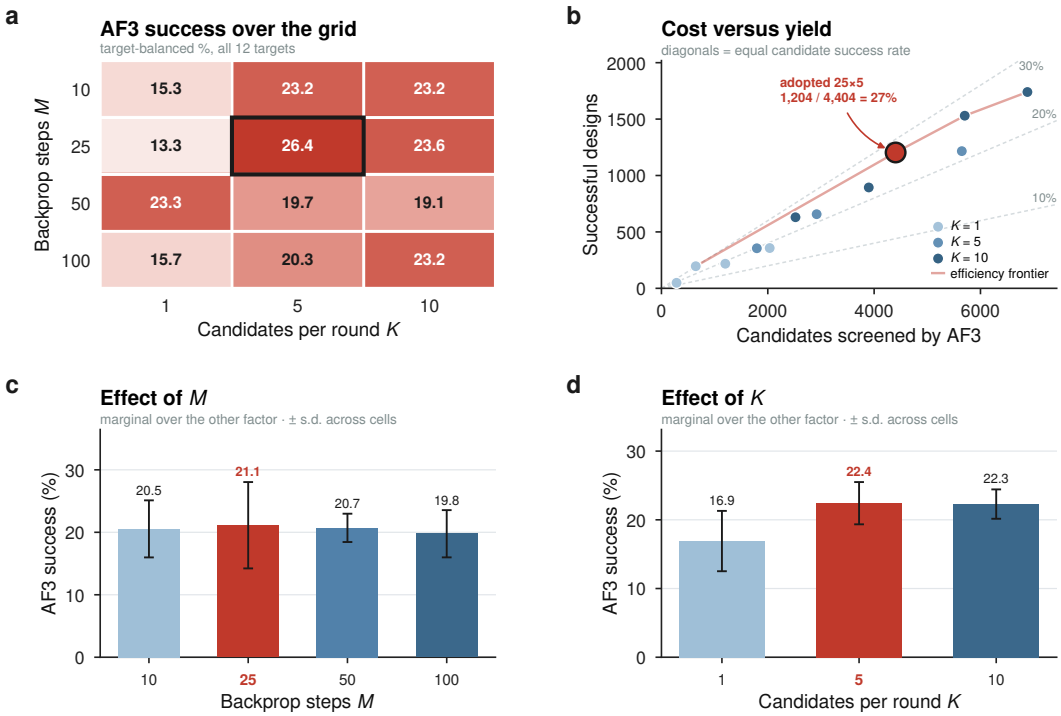

Figure 7: **Sensitivity to the outer-loop schedule.** Gradient steps M and candidates per round K are evaluated end to end at $R = 1$ and $T = 20$ on all twelve targets. (a) AlphaFold3 success rate over the grid, computed per target and then averaged across targets; the configuration corresponding to the main operating regime is outlined for reference. (b) Number of accepted designs against the number of candidates screened by AlphaFold3, the dominant cost of the outer loop; diagonals indicate equal candidate success rates. (c,d) Each factor marginalized over the other, with standard deviation across cells. The results show a broad performance plateau rather than a sharply localized optimum.

G DISTOGRAM SURROGATE OBJECTIVES

The control in Section 4.2 needs geometric objectives that stay differentiable without reaching the sampled coordinates. BoltzDesign1 (Cho et al., 2025) cannot serve this purpose, since it is built on Boltz-1 and its confidence objectives are not comparable to ours. We instead re-express $\ell_{\text{FW-away}}$ and ℓ_{epitope} on the trunk distogram, leaving every other component of AbPilot in place.

Let p_{ijb} be the predicted probability that token pair (i, j) falls in distance bin b, and c_b the representative distance of that bin. The expected pairwise distance

$$\hat{d}_{ij} = \sum_b p_{ijb}\, c_b \tag{17}$$

is taken over the 64 bins of the head, with open-ended terminal bins. Substituting $\hat{d}_{ij}$ for d_{ij} throughout Eq. 14 and Eq. 15 defines the surrogates $\ell^{\text{disto}}_{\text{FW-away}}$ and $\ell^{\text{disto}}_{\text{epitope}}$, with $\mathcal{N}_k$ now taken under $\hat{d}$ and the constants unchanged. Gradients then run from the geometric terms to the distogram logits and back through the trunk, never entering the sampler. The five confidence terms of Eq. 12 are untouched, so the comparison differs in one respect only: whether the geometric gradient originates in the sampled coordinates or in the predicted distance distribution.

This substitution costs two things. First, $\hat{d}_{ij}$ is an average, not a distance: when the model is undecided between a bound and an unbound arrangement, $p_{ij\cdot}$ has two peaks and the average falls between them, at a value neither takes. Second, the entries are computed pair by pair with nothing tying them together, so $\hat{d}$ need not describe any structure that exists in three dimensions. Optimizing the surrogates can therefore improve a distance matrix that no coordinates can produce, which is what Section 4.2 measures.

H DIFFUSION DIFFERENTIATION AGAINST THE DISTOGRAM SURROGATE

Table 5: VHH design comparison on 10 epitope-defined targets. All continuous metrics and success rates are target averages, computed per target and then averaged over targets. #Succ is the absolute number of accepted candidates across all targets.

Method	SR	#Succ	ipTM ↑		scRMSD (Å) ↓		pLDDT ↑	
	(%)	(n)	all	succ	all	succ	all	succ
Diffusion differentiation	**30.7**	**862**	**0.550**	0.771	8.32	2.56	85.5	89.0
Distogram surrogate	19.9	613	0.525	0.767	**7.35**	2.47	84.7	88.4
Shortcut	20.7	597	0.545	0.758	7.55	2.36	**86.0**	89.7

The three settings separate along the objective they can reach. Both ablations drive the confidence loss below diffusion differentiation (Fig. 2c, f) yet fall far short on the geometric objectives, at 23 and 20% epitope success against 60% (Fig. 2b), and under an independent evaluator the ordering follows the geometry: the distogram surrogate accepts 19.9% of its designs against 30.7% (Table 5).

The confidence head reports how the co-folding model scores its own output, and it can be reached from the trunk without settling the coordinates. Optimizing it therefore improves the calibration of one network, and the two networks agree only weakly: with each target centred, so that only agreement within a target counts, the Pearson correlation over all candidates [9] is $r = 0.37$ on interface ipTM and $r = 0.38$ on binder pLDDT (Fig. 8). Geometry does not depend on the model: a sequence whose loops reach the epitope reaches it under either predictor. Differentiating the sampler therefore wins under AlphaFold3 even though it scores worse on Boltz-2's own confidence.

The same weak correlation is what makes orthogonal rectification worth its cost. Were the two predictors closely aligned, an external verifier would only repeat what the inner loop already knows; at $r \approx 0.37$ it supplies evidence the optimizer cannot manufacture, which is why it raises the success rate in Table 1 despite consuming budget that would otherwise fund more rounds.

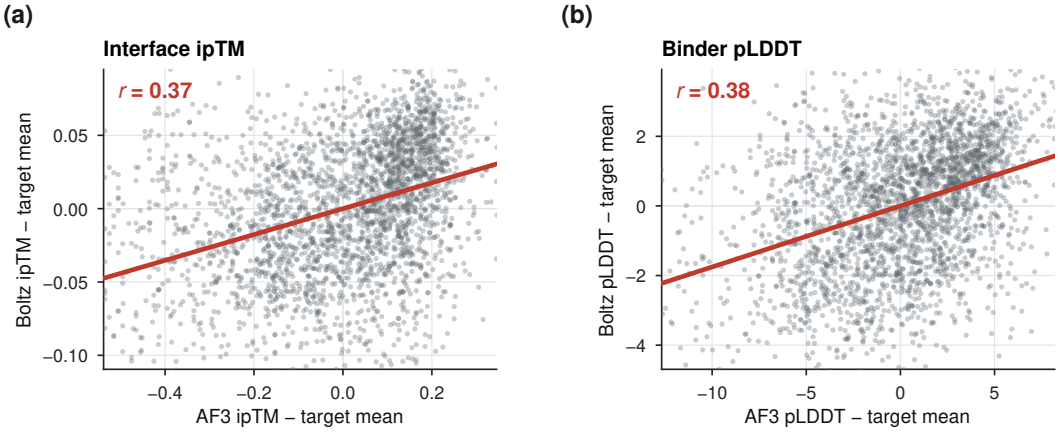

Figure 8: **Boltz-2 and AlphaFold3 confidences agree only weakly on the same designs.** Each point is one candidate, with the per-target mean subtracted so that the comparison is within a target rather than across targets. Pearson r over all candidates: **(a)** interface ipTM, **(b)** binder pLDDT.

I DENOVO ANTIBODY DESIGN

Per-target success and metric distributions Figure 9 resolves Table 1 by target and reports the distribution behind each average. The advantage is broad rather than concentrated: AbPilot

[9] All candidates first pass a Boltz-2 filter, requiring CDR pLDDT > 0.7 and target–binder ipTM > 0.7, and the survivors are then evaluated by AlphaFold3.

with rectification accepts designs on all twelve targets in both settings and without it on eleven, while the baselines are at zero on most cells, and what success they have is confined to PD-1 and PD-L1 (Fig. 9a, b). The distributions locate the separation. Interface ipTM does not separate the methods sharply, with AbPilot at a median of 0.58 for VHH against 0.52 for Germinal and the boxes overlapping (Fig. 9c, d), whereas AbPilot's median scRMSD is near 3 Å with most of the box below the threshold and every baseline sits between 13 and 40 Å entirely above it (Fig. 9e, f). Baseline designs are therefore not marginal failures of confidence but structures an independent predictor places somewhere else entirely.

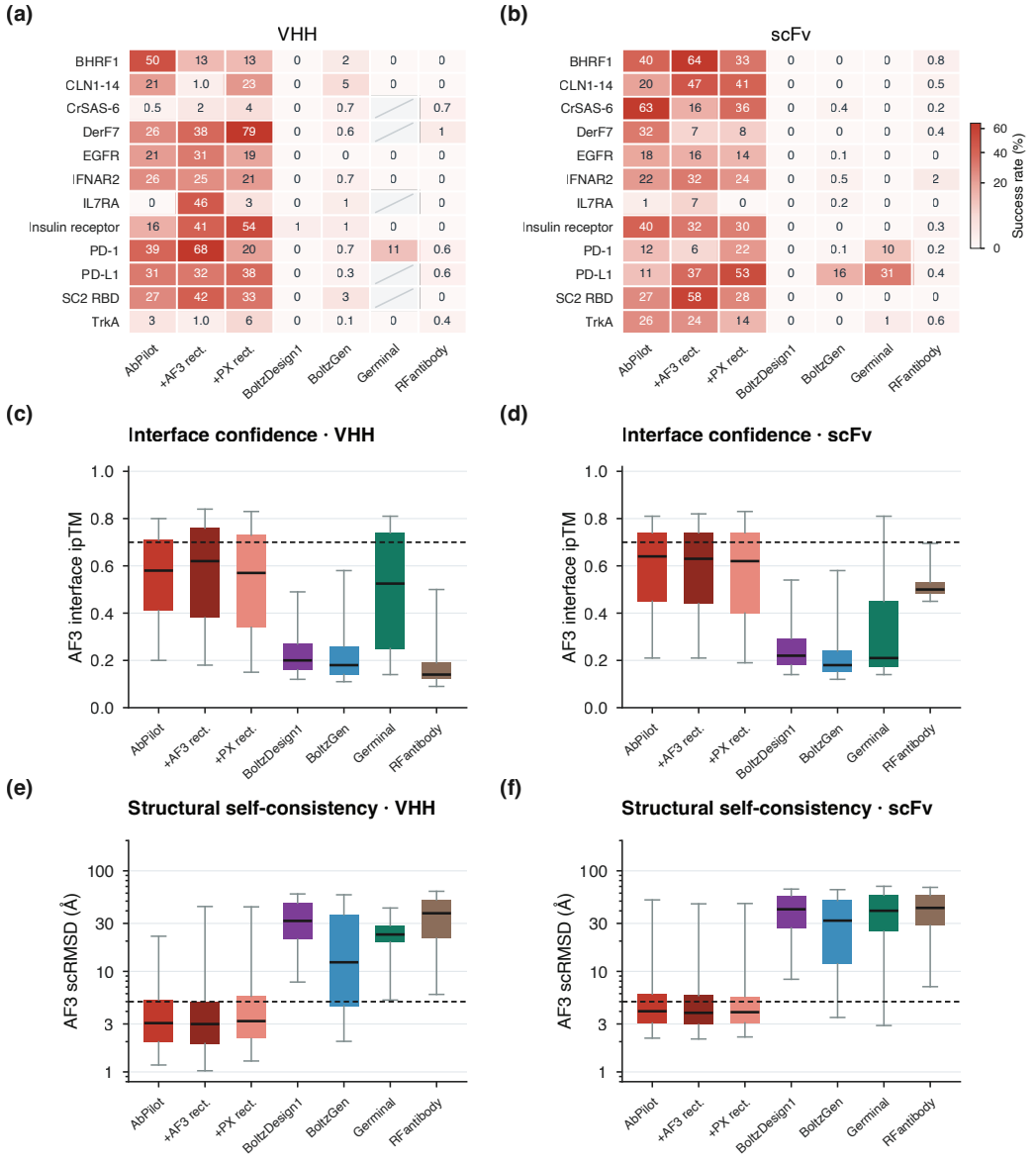

Figure 9: **Per-target results behind Table 1. (a, b)** AlphaFold3 success rate for each target and method. Hatched cells mark targets on which the model ran for the full budget but returned no candidate AlphaFold3 could evaluate; the success rate is 0%. **(c–f)** Distributions over all generated candidates, with the dashed line at the acceptance threshold.

Sequence and structure diversity We cluster the accepted candidates for each target and report the number of clusters as a fraction of the candidate count (Fig. 10). Sequences are clustered at 70%

identity. Structures are clustered after superposing each complex on the target and taking the RMSD over the CDR backbone at 2 Å, so the measure separates binding modes rather than folds.

Sequence diversity averages 57, 55 and 48% across the three rectification settings for VHH and 57, 56 and 52% for scFv. Structural diversity reaches 31–44% and 35–41% respectively, so a substantial fraction of the accepted designs engage the epitope through binding modes that no other design in the panel shares.

Absolute cluster counts tell the same story without depending on the candidate count. On the targets with the largest yields a single run returns of the order of a hundred distinct sequence clusters and tens of distinct binding modes, and no target in any setting collapses to a handful. The panel AbPilot delivers is therefore diverse in both senses, at every scale of output.

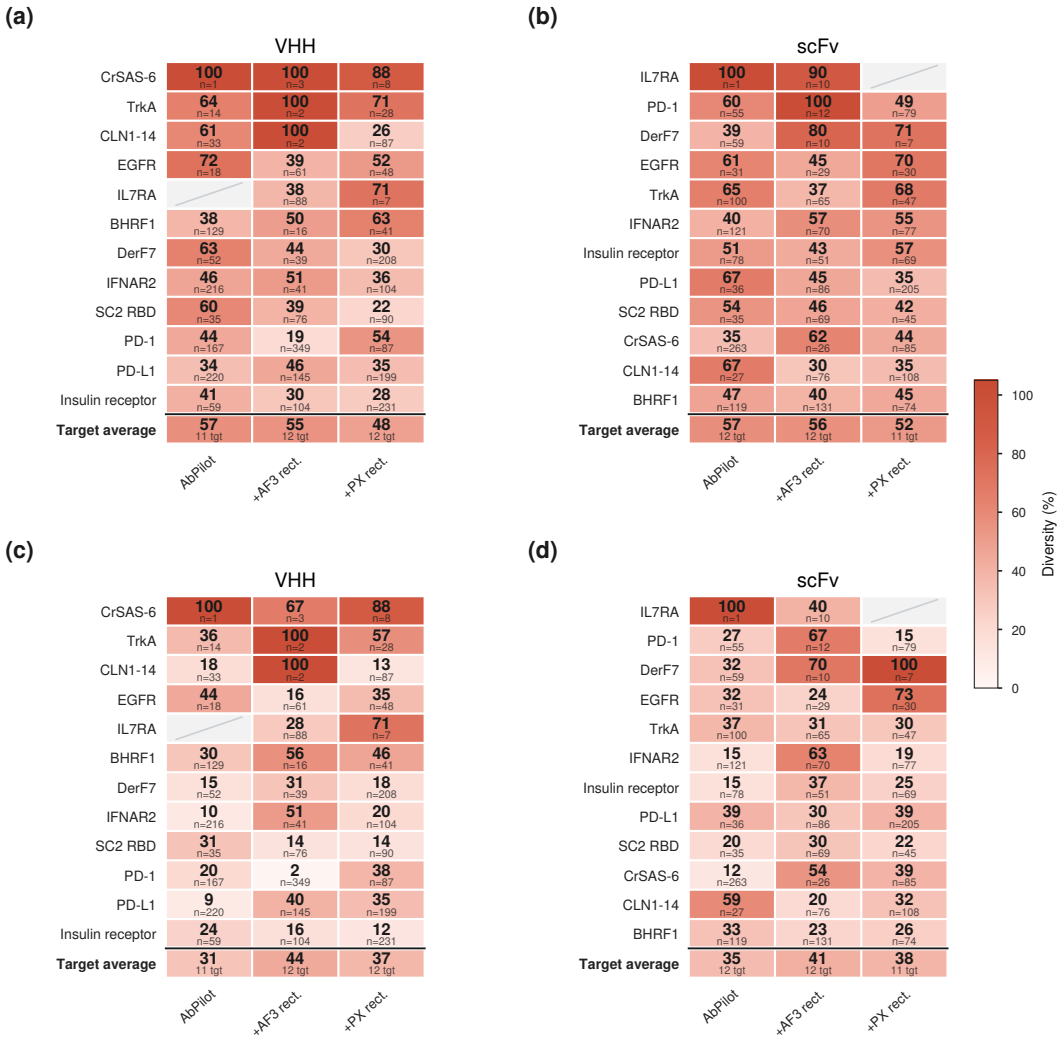

Figure 10: **Per-target sequence and structure diversity. (a, b)** Sequence clusters at 70 % identity divided by candidates; **(c, d)** the same designs clustered by CDR backbone RMSD at 2 Å. n is the candidate count. Diagonals: no designs. The bottom row averages the per-target ratios.

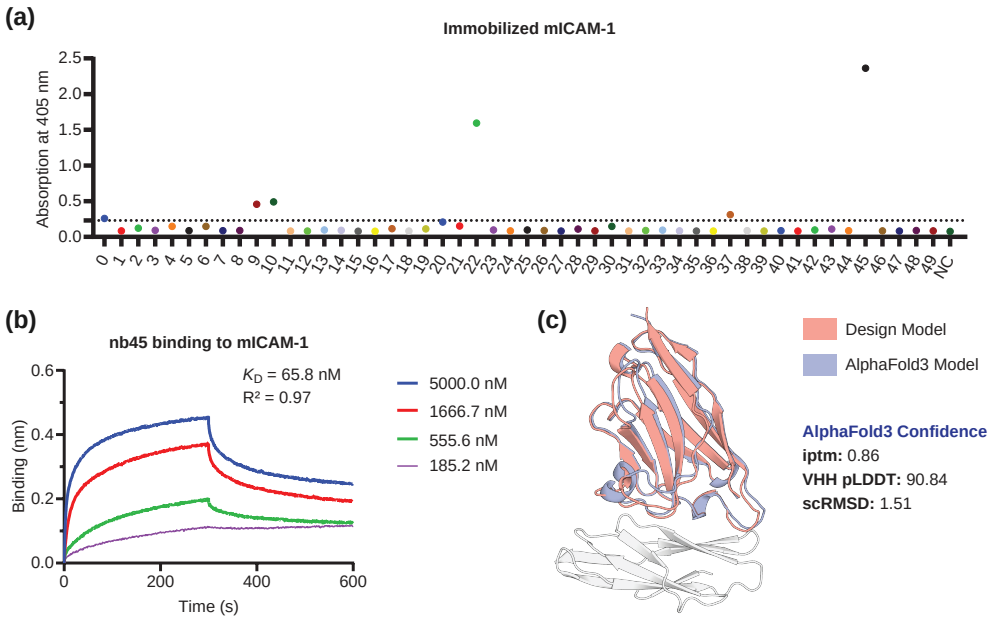

Figure 11: **De novo VHH against mICAM-1. (a)** ELISA of the 50 expressed designs on immobilized mICAM-1; dotted line, background. NC, negative control. **(b)** BLI of nb45, the strongest hit, giving $K_{\mathrm{D}} = 65.8\,\mathrm{nM}$. **(c)** Design model (red) superposed on the AlphaFold3 prediction (blue), with the target in grey.

J WET-LAB PROTOCOLS

Target The de novo target is mouse ICAM-1 (Eswaran et al., 2025). No experimental structure is available, so we used the AlphaFold3 web server prediction as the design target [10]. We use AbPilot+ AF3 rect. to design these VHHs.

Monoclonal expression and ELISA screening. Genes encoding designed nanobodies targeting mICAM-1 were synthesized and inserted into the customized plasmid pComb3x by Genewiz. Recombinant plasmids were transformed into *E. coli* HB2151 and plated on LB agar containing 0.1% ampicillin (Amp) for overnight culture at 37 °C. Single colonies were picked into 2YT medium (100 μL per well, containing 0.1% Amp), cultured at 37 °C for 6 h, and induced with IPTG at 30 °C for 14–16 h. Polymyxin B was then added to lyse the cells at 30 °C and 220 rpm for 40 min. In parallel, 96-well microplates (Corning, catalog no. 3690) were coated with the target antigen (100 ng per well) overnight at 4 °C and blocked with 3% MPBS (PBS containing 3% (w/v) skim milk) at 37 °C for 1 h. Fifty microliters of the cell lysate supernatant was added to each well and incubated at 37 °C for 1.5 h. After washing, HRP-conjugated anti-FLAG secondary antibody (1:5,000, STARTER Antibodies, catalog no. SOB1648) was added and incubated at 37 °C for 45 min. Following five washes with PBS containing 0.05% (v/v) Tween-20 (PBST), 50 μL of ABTS substrate solution (Thermo Fisher Scientific, catalog no. 002024) was added to each well for colour development at 37 °C for 10 min, and the absorbance at 405 nm was measured on a microplate reader.

Bio-layer interferometry. Binding kinetics to mICAM-1 was measured by bio-layer interferometry on an Octet-RED96 system (ForteBio). mICAM-1 (15 μg mL^{-1}) was covalently immobilized onto AR2G biosensors (Sartorius, catalog no. 18-5092). Biosensors were incubated with threefold serially

[10] We classify mouse ICAM-1 (UniProt: P13597) as having no known nanobody binder under our benchmark definition, which requires direct experimental validation against mouse ICAM-1. Although the anti-ICAM-1 sdAb clone 11-4 has been evaluated in mouse disease models, its reported ELISA and SPR measurements used recombinant human ICAM-1; no affinity measurement or complex structure with mouse ICAM-1 has been reported. We therefore do not consider it a validated mouse ICAM-1 nanobody binder.

diluted nanobodies in PBST (containing 0.02% Tween-20) for $300\,\mathrm{s}$ of association, followed by $300\,\mathrm{s}$ of dissociation in PBST at $37\,^{\circ}\mathrm{C}$. All curves were fitted with a $1{:}1$ binding model in Data Analysis software 10.0, and K_{D} values were determined with R^2 above the 95% confidence level.