BONOBO: EFFICIENT LIBRARY-SCALE GENERATION FOR *De Novo* ANTIBODY DESIGN

Sebastian W. Ober[*]
BigHat Biosciences
sober@bighatbio.com

Nick Bhattacharya[*]
BigHat Biosciences

Phillip Maffettone
BigHat Biosciences

Calvin McCarter
BigHat Biosciences

Hunter Elliott
BigHat Biosciences

ABSTRACT

Recent developments in *de novo* antibody design show promise for generating candidates that bind to drug targets. Methods which have shown success in the lab propose designs with either a diffusion model or hallucination-based sequence optimization. They then filter these candidates with a structure prediction model. Hallucination-based methods rely on expensive backpropagation through a loss function derived from the structure prediction model to perform per-generated-sequence optimization, limiting their ability to generate designs at the scale of the libraries, which may be needed for challenging targets. We propose *Bonobo*, which instead trains a generative model on the structure prediction loss using a GFlowNet formulation in which we transform the loss to act as a reward function. This does not require differentiation of the structure predictor, increasing our computational efficiency and unlocking a broader class of structure-based models for usage. Crucially, our approach amortizes the cost of generation into training time, enabling Bonobo to generate library-scale sets of diverse sequences at a significantly lower cost. We show empirically that our approach can effectively model complex loss functions and generate large numbers of high-performing novel antibody sequences for a range of target proteins.

1 INTRODUCTION

Antibody therapeutics are a versatile and rapidly-expanding drug class, already important in the treatment of many diseases. However, the computational discovery of new antibodies with desirable therapeutic properties remains a challenging problem. Hallucination-based *de novo* design (Norn et al., 2021; Swanson et al., 2025; Mille-Fragoso et al., 2025) has recently shown progress towards generating novel antibodies that bind a target, from the target structure alone. This approach optimizes antibody sequences that minimize an AlphaFold-based loss (Jumper et al., 2021) as a surrogate for the likelihood of binding.

However, there are drawbacks to this approach. First, the optimization of the sequences is a complex, multi-stage process, typically requiring optimization of logits, followed by softmaxed logits, and ending with discrete optimization of the one-hot sequence representations. This heuristic approach can require significant tuning for different targets (Mille-Fragoso et al., 2025). More importantly, design generation is computationally intensive due to the need to run optimization for each new sequence, as well as the need to backpropagate through the structure model.[1] This issue becomes prohibitive for *e.g.,* the design of the large libraries that may be necessary to tackle the significant fraction of targets for which current methods cannot successfully find binders for with a small number of designs.

[*]Equal contribution.

[1]In practice, hallucination-based methods sample multiple (on the order of ten) sequences per optimization trajectory, but the fundamental issue remains.

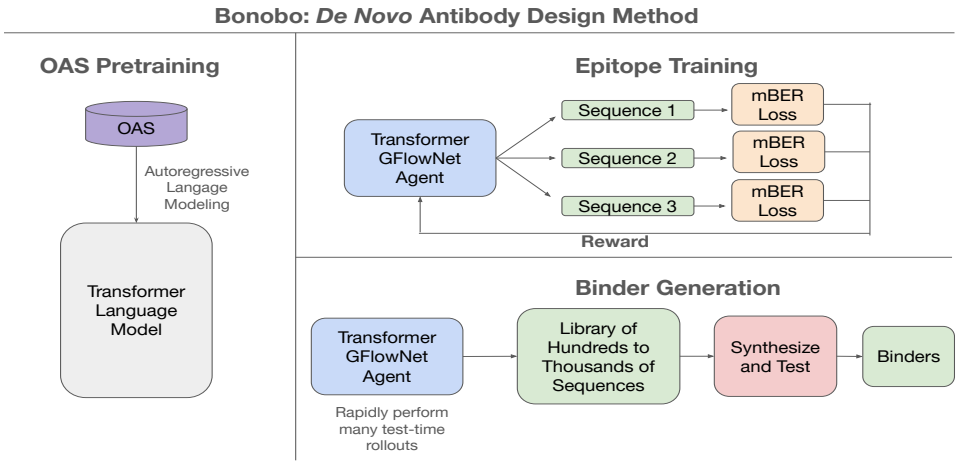

Figure 1: Overview of Bonobo, encompassing three stages: OAS pretraining, epitope-specific GFlowNet training, and test-time generation of potential binders.

In this work we propose *Bonobo*, a method for generating large sets of diverse antibody sequences with good AlphaFold-based scores. To do so, we leverage recent advances from the GFlowNet (Bengio et al., 2021) and RL (Zhu et al., 2025) literature and use an AlphaFold-based loss as a *reward* instead. This choice amortizes the cost of generation into training time, thereby allowing us to sample large numbers of high-scoring sequences with a small sequence-only transformer model. Moreover, we are able to avoid costly backpropagation through large structure models such as AlphaFold, as the reward-based formulation requires only forward passes through the structure model.

Concretely, we demonstrate that

- Bonobo models the AlphaFold loss well,
- Bonobo can match the quality of generated sequences of mBER, and
- Bonobo is more computationally efficient for generating large numbers of sequences.

2 Background

While natural antibodies are a core part of the immune system, they have also proven to be a versatile protein platform for drug development. Antibody therapeutics can be complex multi-chain assemblages, but computational design often focuses on the variable domains, which are comprised of largely conserved framework regions (FRs) and the highly-variable complementarity determining regions (CDRs) that enable binding to a target. As with all proteins, we represent our designs as a discrete sequence $x = a_1 a_2 \dots a_n$, where $a_i \in \mathcal{A}$ is one of the twenty canonical amino acids. In this work we focus on variable heavy domains, which can be broken up into seven regions total: four FRs and three CDRs, where $x = \mathrm{FR1}|\mathrm{CDR1}|\mathrm{FR2}|\mathrm{CDR2}|\mathrm{FR3}|\mathrm{CDR3}|\mathrm{FR4}$.

2.1 Hallucination-based *De Novo* Design

To perform hallucination-based *de novo* design, one starts with a structure predictor F – for example, AlphaFold 2 – which maps an input amino acid sequence x to an output structure $F(x)$, represented as a point cloud of atoms in $\mathbb{R}^3$. The quality of the sequence x is then measured by a loss function $\mathcal{L}$, intended to quantify how well the folded $F(x)$ will bind a target T. Naïvely, one would want to take gradients with respect to the input $\nabla_x \mathcal{L}(F(x))$ and use them to modify the input x to get a strong candidate sequence.

Unfortunately, given that the sequence x is discrete, this gradient is not well-posed. Recent methods such as mBER (Swanson et al., 2025) and Germinal (Mille-Fragoso et al., 2025) follow the BindCraft approach (Pacesa et al., 2024) to resolve this. In particular, they start by optimizing the

logit representation of the sequence, followed by optimizing softmaxed logits and subsequently using a straight-through estimator to optimize a discrete sequence. Finally, an optional semi-greedy approach using position-specific scoring matrices (PSSM) was introduced to finalize the sequence. This approach was shown to produce feasible mini-binders. In this work, we focus on mBER, which modifies BindCraft by introducing specialized structure and sequence templating for nanobody design.[2]

2.2 GFlowNets

Generative flow networks (GFlowNets; Bengio et al., 2021) are a class of probabilistic generative models that aim to train a stochastic policy π_θ that generates objects x proportional to a reward $R(x)$. A key assumption is that the underlying sample space $\mathcal{X}$ is compositional, so that we can build up elements of $\mathcal{X}$ through a sequence of actions from an action space $\mathcal{A}$. Fortunately, this assumption is easily satisfied for protein sequences, where the action space can be taken to be the set of amino acids, and generating a new sequence $x \in \mathcal{X}$ amounts to sequentially adding (in the autoregressive case) or unmasking (in the non-autoregressive case) positions one-by-one. We can then define a trajectory τ that starts with an initial state s_0 (corresponding to an empty or fully masked sequence), visits intermediate states s_i (partially completed or masked sequences), and terminates with the final sequence x followed by a sink state s_f.

GFlowNets generally model both the forward (i.e., generative) probability of a trajectory as well as its backward probability, and are trained to match the probability "flow" in the generation process to the reward $R(x)$ by ensuring that these probabilities are consistent. While many losses have been proposed for GFlowNets (e.g., Bengio et al., 2021), in this work we focus on the trajectory balance loss (Malkin et al., 2022), given by

$$\mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\log Z_\phi + \log \pi_\theta(\tau) - \log R(x) - \log p_B(\tau|x)\right)^2\right], \tag{1}$$

where π_θ (abusing notation) is the forward policy and p_B the backward policy, and Z_ϕ is a learnable partition function, representing the total flow through the system.

2.3 FlowRL

In recent work, FlowRL (Zhu et al., 2025) takes inspiration from GFlowNets to derive an alternative loss function that additionally employs a reference model π_{ref} as a regularizer, while also modifying the loss to take into account sequence length:

$$\mathbb{E}_{x\sim\pi_\theta}\left[\left(\log Z_\phi + \frac{1}{L}\log \pi_\theta(x) - \beta \log \hat{R}_i(x) - \frac{1}{L}\log \pi_{ref}(x)\right)^2\right], \tag{2}$$

Here, β is a tunable hyperparameter, L is the length of the generated sequence, and $\hat{R}$ is the reward function normalized using group normalization (Wu & He, 2020). Note that the loss no longer includes a backward policy term due to the focus on autoregressive generation, for which the backward policy becomes deterministic. We also note that other works have proposed similar losses, particularly involving reference models (Guo et al., 2021; Hu et al., 2023; Lee et al., 2025; Bartoldson et al., 2025).

3 Methods

We introduce Bonobo, a GFLowNet-based approach to *de novo* nanobody binder design. Our approach builds on the recent backpropagation-based binder design method mBER (Swanson et al., 2025), but instead amortizes sampling cost over a single upfront training phase. This allows for library-scale design of potential binders more efficiently than backpropagation-based methods, which require a successful optimization trajectory for each molecule they design.

More concretely, given a nanobody framework sequence, desired CDR lengths, a target protein, and optionally a desired epitope, Bonobo constructs a generative model over CDR sequences. In

[2]We refer the reader to App. B for a more detailed treatment of related works.

particular, our generative model is given by a GFlowNet which yields a policy π_θ that guides the generation of these CDRs. We opt for an autoregressive generation procedure, both to simplify the modeling task (as the backward policy is deterministic), and relatedly to avoid credit assignment problems known to affect non-autoregressive GFlowNets (Shen et al., 2023). To facilitate autoregressive generation of CDRs conditional on a given framework sequence, we choose to format our sequences as

$$[\text{BOS}]\,\text{FR}|\text{CDR1}|\text{CDR2}|\text{CDR3}\,[\text{EOS}]\,, \tag{3}$$

where FR denotes all the framework regions concatenated simply, and BOS and EOS are beginning- and end-of-string tokens, respectively.

For the reward function, we transform the mBER loss function with a simple parameterization:

$$R(x) = \exp\left(-\text{softplus}\left(\mathcal{L}_{mBER}\left(x\right)\right)\right). \tag{4}$$

Note that we introduce a softplus, as we found that the mBER loss could achieve large negative values with minor changes in the sequence, leading to sharp modes. Importantly, as this is a reward function and not a loss function, this setup does *not* require backpropagation through a structure prediction model. In addition, we follow FlowRL (Zhu et al., 2025) and use a reference model in the loss: we choose AbLang2 (Tobias H. Olsen & Deane, 2024) for this purpose. However, unlike FlowRL, we did not find group normalization to aid optimization and so exclude it. Our proposed loss function is then

$$\mathbb{E}_{x\sim\pi_\theta}\left[\left(\log Z_\phi + \frac{1}{L_{\text{CDR}}}\log\pi_\theta\left(x\right) - \beta\log R\left(x\right) - \frac{1}{L_{\text{CDR}}}\log p_{AbLang2}\left(x\right)\right)^2\right] \tag{5}$$

where $p_{AbLang2}\left(\cdot\right)$ is the pseudo log-likelihood of AbLang2, and the length L_{CDR} refers to the total generated CDR length. Once Bonobo is trained for a specific epitope, generation can be done by directly sampling from the model. This sampling procedure is incredibly efficient compared to backpropagation-based methods, as we will explore in Section 4.

3.1 OAS Pretraining

Given that the length of typical CDRs ranges from about 25 to 45 residues, and that each residue can take one of 20 amino acids, the design problem is intractable when tackled naïvely. Therefore, we pretrain a transformer model on a subset of Observed Antibody Space (OAS; Olsen et al., 2022) heavy chains, and use this as the initialization for our GFlowNet model. Note that in order to achieve this, we need to pre-process the sequences to match the formatting of Eq. 3.

3.2 Off-Policy Training

In order to encourage broad exploration within the space of antibody-like sequences, we incorporate off-policy training. The GFlowNet literature partially encourages exploration by using a train-time sampling temperature and mixing in a uniform action space with probability ϵ (e.g., Jain et al., 2022). To encourage exploration of antibody-like sequences, we also sample from AbLang2, with a probability of p_{off}. Therefore, during the rollout, our effective policy for a given action is given by

$$\left(1 - p_{\text{off}} - \epsilon\right)\pi_{\theta,T} + p_{\text{off}}\pi_{ref} + \epsilon\cdot\text{Uniform}, \tag{6}$$

where T is the train-time sampling temperature.

4 Results

We compare Bonobo to mBER for four targets that are typically used in the *de novo* literature (Swanson et al., 2025; Mille-Fragoso et al., 2025): PD-L1, IL-3, IL-20, and BHRF1. For each target we use the same framework sequence as mBER, and use the same hotspots for both methods. For mBER, we request 100 sequences for each target, requiring iPTM and pLDDT (Jumper et al., 2021) greater than 0.7 to be accepted, as the former has been found to correlate well with wet-lab binding (Swanson et al., 2025). To emphasize the generality of Bonobo, we use the same hyperparameter settings for each target. We provide full experimental details in App. A.

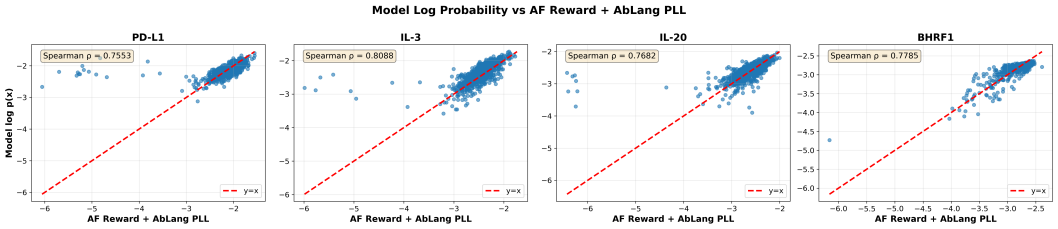

Figure 2: Bonobo GFlowNet probabilities versus AlphaFold reward plus AbLang2 PLLs show strong agreement across four different targets.

Table 1: Comparison of Bonobo and mBER on sequences with iPTM > 0.7. The number next to the method name indicates the number of passing and unique variants for that method.

Target	Method	iPTM ($\mu \pm \sigma$) ↑	iPTM (min, max)	Loss ($\mu \pm \sigma$) ↓	Loss (min, max)
PD-L1	Bonobo (763)	$\mathbf{0.784 \pm 0.029}$	$\mathbf{(0.700, 0.845)}$	$\mathbf{-1.211 \pm 0.258}$	$\mathbf{(-1.730, -0.135)}$
	mBER (108)	0.750 ± 0.032	$(0.700, 0.824)$	-0.035 ± 0.725	$(-0.956, 3.032)$
IL-3	Bonobo (738)	$\mathbf{0.757 \pm 0.024}$	$\mathbf{(0.701, 0.835)}$	$\mathbf{0.077 \pm 0.260}$	$(-0.777, 0.880)$
	mBER (100)	0.756 ± 0.033	$(0.700, 0.824)$	0.124 ± 0.599	$\mathbf{(-1.054, 3.329)}$
IL-20	Bonobo (771)	$\mathbf{0.792 \pm 0.027}$	$\mathbf{(0.701, 0.855)}$	$\mathbf{-0.371 \pm 0.245}$	$(-0.996, 0.517)$
	mBER (100)	0.774 ± 0.034	$(0.710, 0.845)$	-0.193 ± 0.819	$\mathbf{(-1.568, 3.955)}$
BHRF1	Bonobo (548)	$\mathbf{0.736 \pm 0.016}$	$(0.700, 0.776)$	$\mathbf{1.207 \pm 0.241}$	$(0.558, 2.299)$
	mBER (100)	0.728 ± 0.026	$\mathbf{(0.700, 0.803)}$	1.878 ± 0.959	$\mathbf{(0.217, 4.446)}$

We first investigate whether the GFlowNet probabilities accurately model the AlphaFold–plus–AbLang scores for the sequences they generate. We generate 800 sequences for each target with a temperature of 1, filter for uniqueness, and plot scatterplots of the trained models' probabilities (with learned Z_ϕ) versus the sum of the AlphaFold reward and AbLang2 PLLs in Fig. 2. This figure shows remarkable agreement between the two, with high Spearman correlations across all targets.

Next, we investigate Bonobo's ability to generate high-quality sequences. For this comparison we threshold on sequences with iPTM > 0.7, the threshold used by mBER. We compare results, showing both iPTMs and losses, in Fig. 1. This table shows that across all four targets, Bonobo is generally able to generate sequences with comparable or better metrics. For a more complete comparison, we point the reader to plots showing the distributions of these metrics in App. C.

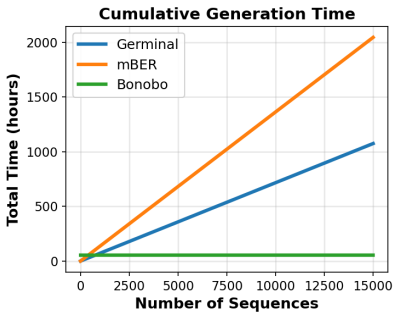

Figure 3: Total per-target generation time for Bonobo compared to mBER and Germinal. Note that only Bonobo has an upfront training cost, but its scaling is far superior.

Finally, we compare the generation efficiency of Bonobo to that of backpropagation-based methods, as the amortization of generation into training time should yield much faster generation. In order to compare the time necessary for Bonobo to generate a fixed number of sequences, we took the following factors into consideration: first, the upfront training cost for a given target. Second, filters are often used to select top candidates, so the reward function was calculated to get ipTMs and other valuable statistics. Nonetheless, a single evaluation of the reward function is much simpler than the many evaluations (and backpropagations) involved in a full optimization trajectory. We display the results in Fig. 3, which shows far superior scaling against mBER, as well as Germinal. See App. A.2 for more.

5 CONCLUSION

In this work we introduced Bonobo, a method for *de novo* antibody design that leverages a structure prediction model to create a reward function for guiding GFlowNet training. This method combines the conceptual simplicity of hallucination/backpropagation-based *de novo* design methods with the powerful generation capabilities of GFlowNets to learn a distribution over antibodies for a given epitope and target. We demonstrated that our GFlowNets trained successfully and achieved similar metrics to mBER at a much lower computational cost. Because we can generate high quality sequences using forward passes, our approach also opens up opportunities for using diffusion-based models, where backpropagation is much more intensive, and simulation-based models, where it may not even be possible. We leave this, as well as comparisons to additional *de novo* methods and *in vitro* validation, to future work.

REFERENCES

Brian R. Bartoldson, Siddarth Venkatraman, James Diffenderfer, Moksh Jain, Tal Ben-Nun, Seanie Lee, Minsu Kim, Johan Obando-Ceron, Yoshua Bengio, and Bhavya Kailkhura. Trajectory balance with asynchrony: Decoupling exploration and learning for fast, scalable LLM post-training. *arXiv preprint arXiv:2503.18929*, 2025.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Chai, Jacques Boitreaud, Jack Dent, Danny Geisz, Matthew McPartlon, Joshua Meier, Zhuoran Qiao, Alex Rogozhnikov, Nathan Rollins, Paul Wollenhaupt, and Kevin Wu. Zero-shot antibody design in a 24-well plate. *bioRxiv*, 2025. doi: 10.1101/2025.07.05.663018.

Han Guo, Bowen Tan, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. Text generation with efficient (soft) Q-learning. *arXiv preprint arXiv:2106.07704*, 2021.

Edward J. Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. *arXiv preprint 2310.04363*, 2023.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with GFlowNets. In *International Conference on Machine Learning (ICML)*, 2022.

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.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *nternational Conference on Learning Representations ICLR*, 2015.

Latent, Henry Kenlay, Daniella Pretorius, Jonathan Crabbé, Alex Bridgland, Sebastian M. Schmon, Agrin Hilmkil, James Vuckovic, Simon Mathis, Tomas Matteson, Rebecca Bartke-Croughan, Amir Motmaen, Robin Rombach, Mária Vlachynská, Alexander W. R. Nelson, David Yuan, Annette Obika, and Simon A. A. Kohl. Drug-like antibodies with low immunogenicity in human panels designed with latent-x2, 2025.

Seanie Lee, Minsu Kim, Lynn Cherif, David Dobre, Juho Lee, Sung Ju Hwang, Kenji Kawaguchi, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Moksh Jain. Learning diverse attacks on large language models for robust red-teaming and safety tuning. *International Conference on Learning Representations (ICLR)*, 2025.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in GFlowNets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Luis Santiago Mille-Fragoso, John N. Wang, Claudia L. Driscoll, Haoyu Dai, Talal M. Widatalla, Xiaowe Zhang, Brian L. Hie, and Xiaojing J. Gao. Efficient generation of epitope-targeted de novo antibodies with germinal. *bioRxiv*, 2025. URL https://www.biorxiv.org/content/10.1101/2025.09.19.677421v1.

Nabla. JAM-2: Fully computational design of drug-like antibodies with high success rates, 2025. URL https://nabla-public.s3.us-east-1.amazonaws.com/2025_Nabla_JAM2.pdf.

Christoffer Norn, Basile IM Wicky, David Juergens, Sirui Liu, David Kim, Doug Tischer, Brian Koepnick, Ivan Anishchenko, Foldit Players, David Baker, et al. Protein sequence design by conformational landscape optimization. *Proceedings of the National Academy of Sciences*, 118 (11):e2017228118, 2021.

Tobias H Olsen, Fergus Boyles, and Charlotte M Deane. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1):141–146, 2022.

Sergey Ovchninnikov, Shihao Feng, Justas Dauparas, Weikun Wu, and Christopher Frank. Colabdesign, 2022. URL https://github.com/sokrypton/ColabDesign.

Martin Pacesa, Lennart Nickel, Christian Schellhaas, Joseph Schmidt, Ekaterina Pyatova, Lucas Kissling, Patrick Barendse, Jagrity Choudhury, Srajan Kapoor, Ana Alcaraz-Serna, et al. Bindcraft: one-shot design of functional protein binders. *bioRxiv*, 2024. URL https://www.biorxiv.org/content/10.1101/2024.09.30.615802v3.

Max W. Shen, Emmanuel Bengio, Ehsan Hajiramezanali, Andreas Loukas, Kyunghyun Cho, and Tommaso Biancalani. Towards understanding and improving GFlowNet training. In *International Conference on Machine Learning ICML*, 2023.

Hannes Stark, Felix Faltings, MinGyu Choi, Yuxin Xie, Eunsu Hur, Timothy O'Donnell, Anton Bushuiev, Talip Uçar, Saro Passaro, Weian Mao, Mateo Reveiz, Roman Bushuiev, Tomáš Pluskal, Josef Sivic, Karsten Kreis, Arash Vahdat, Shamayeeta Ray, Jonathan T. Goldstein, Andrew Savinov, Jacob A. Hambalek, Anshika Gupta, Diego A. Taquiri-Diaz, Yaotian Zhang, A. Katherine Hatstat, Angelika Arada, Nam Hyeong Kim, Ethel Tackie-Yarboi, Dylan Boselli, Lee Schnaider, Chang C. Liu, Gene-Wei Li, Denes Hnisz, David M. Sabatini, William F. DeGrado, Jeremy Wohlwend, Gabriele Corso, Regina Barzilay, and Tommi Jaakkola. BoltzGen: Toward universal binder design. *bioRxiv*, 2025. doi: 10.1101/2025.11.20.689494.

Xingyu Su, Xiner Li, Masatoshi Uehara, Sunwoo Kim, Yulai Zhao, Gabriele Scalia, Ehsan Hajiramezanali, Tommaso Biancalani, Degui Zhi, and Shuiwang Ji. Iterative distillation for reward-guided fine-tuning of diffusion models in biomolecular design. *arXiv preprint arXiv:2507.00445*, 2025.

Erik Swanson, Michael Nichols, Supriya Ravichandran, and Pierce Ogden. mBER: Controllable de novo antibody design with million-scale experimental screening. *bioRxiv*, 2025. URL https://www.biorxiv.org/content/early/2025/09/28/2025.09.26.678877.

Iain H. Moal Tobias H. Olsen and Charlotte M. Deane. Addressing the antibody germline bias and its effect on language models for improved antibody design. *bioRxiv*, 2024. doi: https://doi.org/10.1101/2024.02.02.578678.

Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, Shuiwang Ji, Sergey Levine, and Tommaso Biancalani. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and DNA design. In *International Conference on Machine Learning (ICML)*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Yuxin Wu and Kaiming He. Group normalization. *Int. J. Comput. Vis.*, 2020.

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, et al. FlowRL: Matching reward distributions for LLM reasoning. *arXiv preprint arXiv:2509.15207*, 2025.

A EXPERIMENTAL DETAILS

For all our experiments, we use a transformer architecture (Vaswani et al., 2017) as our base model. More specifically, we use an decoder-only autoregressive architecture with 8 layers, embedding dimension of 512, and 8 heads. We use a standard learned embedding as well as sinusoidal position encodings. For each target, we generate 32 CDR sequences from a rollout of the GFlowNet training policy. We train for 8,000 steps,[3] using the Adam optimizer with a learning rate of $3e$-5 (Kingma & Ba, 2015), using a cosine learning rate scheduler with a warmup of 200 steps. We find that setting β to 1, an off-policy rate (p_{off}) of 0.1, and a training time temperature of 1 performs well across all targets. We follow Jain et al. (2022) in setting the uniform policy rate ϵ to 0.001.

A.1 OAS PRETRAINING

We pretrain the transformer on 2,076,420 heavy domain human sequences taken from the OAS database (Olsen et al., 2022) in an unsupervised manner, where we set aside 5% of the dataset for validation. We use the Adam optimizer, a cosine learning rate scheduler with a warmup period of 1,000 steps, and use gradient clipping set to a value of 0.25. We perform hyperparameter sweeps on the number of epochs in $\{5, 10, 30\}$, learning rate in $\{1e\text{-}5, 1e\text{-}4, 1e\text{-}3\}$, batch size in $\{32, 64\}$, and dropout rate in $\{0, 0.1, 0.2\}$. Note that unlike with the GFlowNet, we train on the *full* heavy chain sequence, instead of conditioning on frameworks.

A.2 TIMING EXPERIMENTS

All timing experiments were performed on a single NVIDIA A10G. We fixed the target to be PD-L1 and calculated the time for Germinal and mBER to generate a single trajectory by averaging over 30 successful trajectories. The time for Bonobo to generate a single successful sequence was similarly obtained. The training time was based on the average of five similar runs. Runtimes to more sequences were extrapolated linearly, assuming fixed compute and ideal scaling.

B RELATED WORK

De novo antibody design has seen a surge in interest in recent years. The hallucination-based methods mBER (Swanson et al., 2025) and Germinal (Mille-Fragoso et al., 2025) build on BindCraft (Pacesa et al., 2024) and ColabDesign (Ovchinnikov et al., 2022), reporting positive results for both library design and batch design of nanobodies and scFvs, respectively. More specifically, ColabDesign (Ovchinnikov et al., 2022), was the first to propose using a continuous logit representation for the sequence x, enabling meaningful gradient computations for sequence optimization. Follow-up work, in particular BindCraft (Pacesa et al., 2024), refined this approach by introducing additional stages after initial logit-based optimization, namely by optimizing softmaxed logits followed by optimizing a discrete sequence with gradients derived using a straight-through estimator. Finally, an optional semi-greedy approach using position-specific scoring matrices (PSSM) was introduced to finalize the sequence. This approach was shown to produce feasible mini-binders.

While BindCraft demonstrated successful minibinder design, it was not aimed at producing antibodies specifically. Key to the success of all hallucination-based approaches is proper engineering of the loss $\mathcal{L}$ to capture the various tradeoffs or complementary properties that are needed for a successful binder. Two recent concurrent works, Germinal and mBER, demonstrated that modifying the loss function could enable the design of antibodies. Germinal (Mille-Fragoso et al., 2025) introduces loss terms for secondary structure and filtering that ensures that CDR residues contact the antigen, ensuring that CDRs are predicted to play a sufficient role in the binding of any passing designs. mBER (Swanson et al., 2025), on the other hand, largely inherited its loss function from ColabDesign and instead developed specialized structure and sequence templating.

Diffusion-based methods have likewise been of great recent interest, with BoltzGen (Stark et al., 2025) reporting nanobody design results on both standard and novel targets, with *in vitro* testing of binding affinity. Finally, the commercial teams Nabla Bio, Chai Discovery, and Latent Labs

[3] for IL-20 we use 6,000 due to its size increasing the AlphaFold-2 prediction time

have issued technical reports claiming the ability to design strong binders with good developability properties (Nabla, 2025; Chai et al., 2025; Latent et al., 2025).

The trajectory balance loss for GFlowNets was introduced in Malkin et al. (2022) as an alternative to the flow-matching loss of Bengio et al. (2021). Autoregressive GFlowNets for biological sequence design were proposed in Jain et al. (2022), where they were incorporated into an active learning loop for multi-batch optimization. This work incorporated trajectory balance over flow matching due to observing faster convergence with the former. Shen et al. (2023) perform a detailed study of GFlowNet training dynamics, unpacking underfitting behavior of existing losses and introducing guided trajectory balance as a solution to the substructure credit assignment problem. However, none of these works attempt to use GFlowNets in a *de novo* protein design setting.

We note that some recent works proposed using AlphaFold-derived rewards in a similar way to Bonobo, in particular Uehara et al. (2025); Su et al. (2025). However, these works do not extend their methods to antibody therapeutics, and use diffusion models, as opposed to much smaller transformer models, as the basis for the method.

C DISTRIBUTIONAL PLOTS

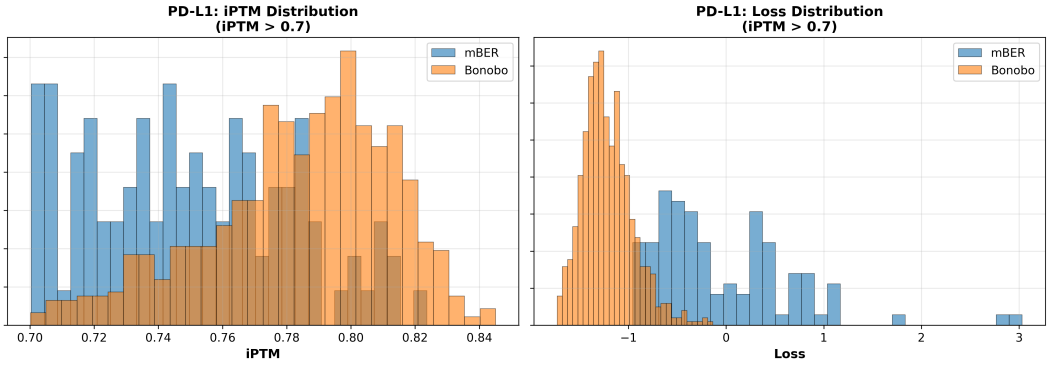

Figure 4: Comparing the distributions of iPTMs and AlphaFold losses for PD-L1

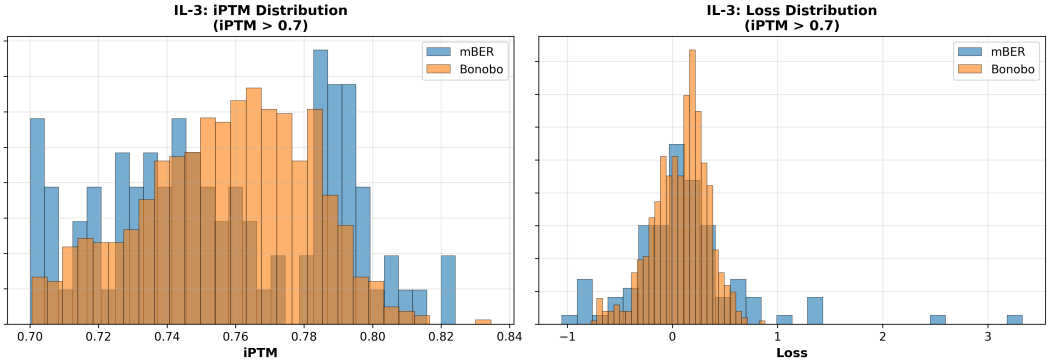

Figure 5: Comparing the distributions of iPTMs and AlphaFold losses for IL-3

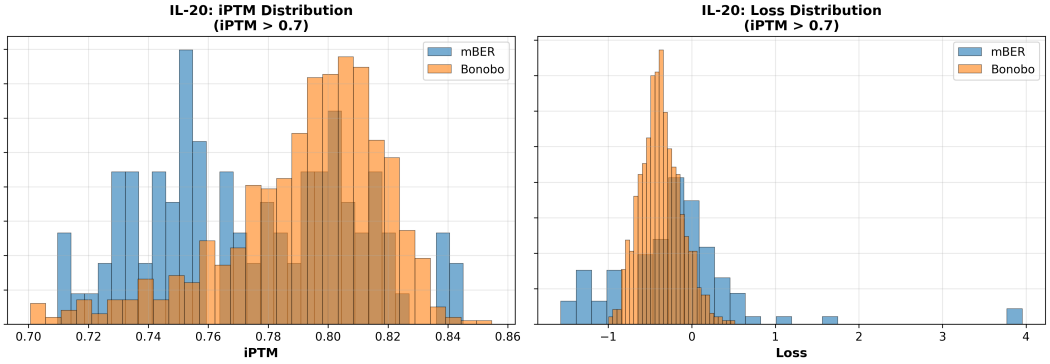

Figure 6: Comparing the distributions of iPTMs and AlphaFold losses for IL-20

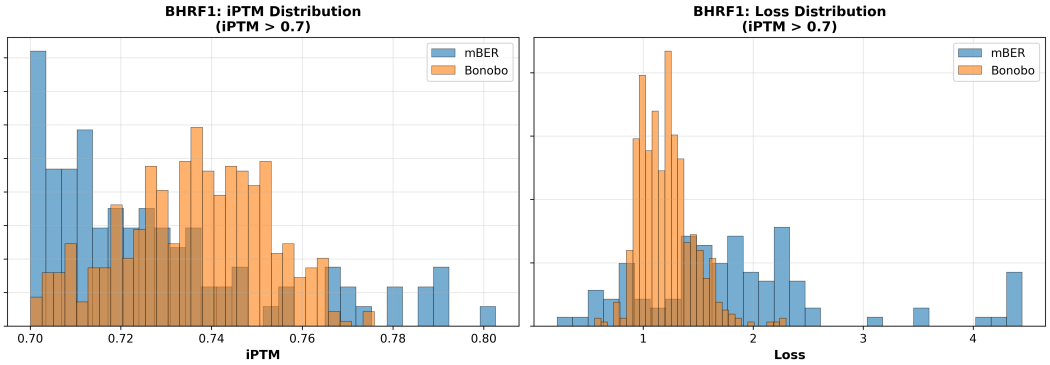

Figure 7: Comparing the distributions of iPTMs and AlphaFold losses for BHRF1