Guidance of Diffusion-Based Conditional Generative Models for Antibody Design

Francesco Alesiani* NEC Laboratories Europe

Jonathan H Warrell* NEC Laboratories America

Henrik Christiansen NEC Laboratories Europe

Matheus Ferraz NEC OncoImmunity

Abstract

Protein structure generation finds important applications in drug and antibody design. Diffusion models have quickly become one of the most prominent approaches for generative tasks. The diffusion process can be seen as applying the gradient of the log-probability density functions to a time-varying sequence. With this interpretation, it becomes possible to control the diffusion process by manipulating the density function. This idea has motivated the introduction of both classifier-based and classifier-free guidance methods. The score-based interpretation of diffusion models has been used to define alternative methods of conditioning, modifying, and reusing these models for tasks that involve compositional generation and guidance. For protein or drug structure prediction, SE(3)-equivariant message passing has been the predominant approach, while atom types are typically modeled using discrete diffusion models. We introduce a formal logical composition framework for conditional diffusion processes (AND and AND-NOT guidance), which respects Boolean De Morgan's laws, and demonstrate its application to antibody complementarity-determining region design.

1 Introduction

The design of novel molecular structures is a fundamental challenge in computational biology and chemistry, with wide-ranging applications including therapeutic antibody engineering, enzyme design, and small-molecule drug discovery. In recent years, deep generative models have emerged as powerful tools for learning complex, high-dimensional data distributions directly from example data, bypassing the need for hand-crafted energy functions. Among these, diffusion probabilistic models have gained prominence due to their stability, scalability, and ability to produce high-quality samples. Diffusion models frame generation as the reversal of a forward noising process, gradually transforming simple random noise into structured outputs through a series of denoising steps. Importantly, diffusion models naturally support guided sampling mechanisms, such as classifier-based and classifier-free guidance, allowing for fine-grained control over the generation process and enabling the incorporation of external constraints or desired properties during generation.

In this work, we explore the combination of conditional diffusion models for multi-target and anti-target design. We introduce the AND and AND-NOT (ANDN) logic for conditional diffusion models and apply it to the problem of antibody complementarity-determining region (CDR) design. To demonstrate the practical implications of this framework, we instantiate our formulation of logical guidance within an SE(3)-equivariant diffusion model for antibody design. In this setting, the diffusion process is conditioned on multiple biophysical attributes of antibodies, such as antigen-binding affinity, and logical operators are used to compose these conditions during generation. Specifically, we apply AND and ANDN logic to modulate the conditional score fields, enabling

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Principles of Generative Modeling.

explicit control over the joint or antagonistic satisfaction of distinct design objectives. Through this setup, we show that the logical composition not only yields interpretable guidance dynamics in the diffusion trajectories but also produces antibody variants that faithfully reflect the intended logical relationships among their biophysical constraints. This antibody case study thus serves as an experimentally grounded proof of concept for our general theory of logical composition in conditional diffusion processes, illustrating how stochastic guidance can be systematically structured through formal logical operators. Other approaches Du et al. (2020); Liu et al. (2022) offer compositional frameworks. However, their rules do not obey Boolean De Morgan's laws, making their semantics unclear. In contrast, our approach can be shown to satisfy De Morgan's laws (Section D).

Related Work

Diffusion-based generative models Diffusion-based models have emerged as a powerful generative framework Sohl-Dickstein et al. (2015); Song & Ermon (2020), achieving state-of-the-art performance for images, audio, and molecular structures. In molecular applications, they have been adapted to generate both 3D atomic coordinates and graph-based representations. Equivariant diffusion models (Hoogeboom et al., 2022) operate in 3D while respecting SE(3) symmetries, producing physically plausible structures, and torsional diffusion (Jing et al., 2023) sequentially samples torsion angles to capture conformational flexibility. Graph-based formulations (Liu et al., 2023) manipulate nodes and edges to generate chemically valid topologies, bridging discrete graphs and continuous structures. These advances have driven breakthroughs in *de novo* protein design, antibody engineering, and protein-ligand modeling.

Antibody Design Recent advances in diffusion-based generative models (Ho et al., 2020; Dhariwal & Nichol, 2021), have enabled the joint modeling of antibody sequence and 3D structure, capturing both backbone and side-chain geometry, as well as CDRs. Methods such as DiffAb (Luo et al., 2022b) and AbDiffuser (Martinkus et al., 2023) perform co-design of sequence and structure conditioned on antigen complexes. Direct preference from post-training LLM community has been proposed for fine-tuning protein diffusion models for antibody design (Wang et al., 2025). Graph-based frameworks (Jin et al., 2021; Kong et al., 2023; Zhang et al., 2022; Bennett et al., 2025) further refine CDR loops and antibody structures. Despite these advances, accurately modeling antigen-antibody interfaces and generalizing to novel antigens remains challenging, motivating the development of guided diffusion strategies that incorporate orientation, side-chain, and equivariance.

Background

Notation and defintions An amino acid in a protein complex can be represented by its type $s_i \in \{\text{A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y}\}$, C_α atom coordinate $x_i \in \mathbb{R}^3$, and the orientation $\mathbf{O_i} \in \mathbb{SO}(\mathbf{3})$, with $i \in [N] = \{1, \ldots, N\}$, and N the number of amino acids in the protein complex, which can contain multiple chains or proteins. We assume the antigen structure and the antibody framework given in Figure 1, where we focus on designing complementarity-determining regions (CDR)s for the antibody template. We consider that the generated CDR has m amino acids. The position of the CDR is from position $l+1$ to $l+m$ in the protein complex. The CDR is represented by $R = \{(s_j, x_j, O_j) | j = l+1, \ldots, l+m\}$. Therefore, the objective of the diffusion model is to model the distribution of R given the structure of the antibody-antigen complex $C = \{(s_i, x_i, O_i) | i \in [N] \setminus \{l+1, \ldots, l+m\}\}$.

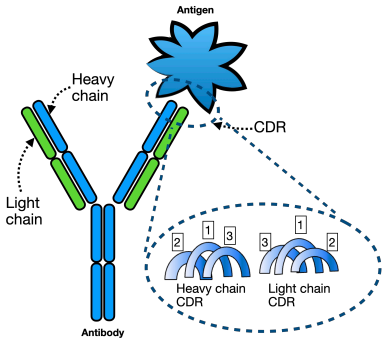

Figure 1: Antibody-antigen protein complex structure, where we highlight the heavy and light chains and the associated CDR structures.

Diffusion model A diffusion probabilistic model is composed of two diffusion processes: The forward process used during training, and the reverse process used during generation. The intermediate sample of the j amino acid at time t is (s_j^t, x_j^t, O_j^t), while the generated sequence and structure at step t is $R_t = \{s_j^t, x_j^t, O_j^t\}_{j=l+1}^{l+m}$. At time $t = 0$, we have the real data, while at time $t = T$ we have

samples from the prior distribution. Therefore, the forward diffusion goes from time $t = 0$ to time $t = T$, while the reverse diffusion proceeds from $t = T$ to $t = 0$.

Amino acid discrete diffusion process For the amino acid, we use a discrete diffusion multimodal process (Luo et al., 2022b; Hoogeboom et al., 2021), where the forward process is $q(s_j^t|s_j^{t-1}) = \text{Multinomial}\left((1-\beta_t^{\text{type}})\text{onehot}(s_j^{t-1}) + \beta_t^{\text{type}}\frac{1}{20}\mathbf{1}\right)$, where onehot$(s)$ converts amino acid type to a 20-dimensional one-hot representation, i.e., a zero vector with 1 at the position s. β_t^{type} is the uniform probability of resampling another amino acid over 20. For $t \to T$, β_t^{type} goes to 1, such that the distribution becomes the uniform distribution. For efficient training β_t^{type} is substituted by $1 - \alpha_t^{\text{type}} = \prod_{\tau=1}^{t} \beta_\tau^{\text{type}}$ and s_j^{t-1} with s_j^0. The reverse discrete diffusion process is implemented as $p(s_j^{t-1}|R_t, C) = \text{Multinomial}\left(F_\theta(R_t, C)_j\right)$, with $F_\theta(R_t, C)$ a neural network that encodes the conditional antibody-antigen complex C and the CDR sequence and structure in the previous step (R_t).

C_α coordinates continuous diffusion process The coordinates of C_α are first normalized. Then the forward diffusion for the normalized C_α coordinates is defined as $q(x_j^t|x_j^{t-1}) = \mathcal{N}(x_j^t; \sqrt{1-\beta_t^{\text{pos}}}x_j^{t-1}, \beta_t^{\text{pos}}\boldsymbol{I})$ where for efficient training $1 - \beta_t^{\text{pos}}$ is substituted with $\alpha_t^{\text{pos}} = \prod_{\tau=1}^{6} \beta_\tau^{\text{pos}}$ and x_j^{t-1} with x_j^0, the data sample. The reverse process is given by $p(x_j^{t-1}|R_t, C) = \mathcal{N}(x_j^t; \mu_\theta(R_t, C), \beta_t^{\text{pos}}\boldsymbol{I})$ with $\mu_\theta(R_t, C) = \frac{1}{\sqrt{\alpha_t^{\text{pos}}}}\left(x_j^t - \frac{\beta_t^{\text{pos}}}{\sqrt{\alpha_t^{\text{pos}}}}G_\theta(R_t, C)_j\right)$ and $G_\theta(R_t, C)_j$ a denoising neural network that predicts the noise $\epsilon_t \sim \mathcal{N}(0, \boldsymbol{I})$, added to the sample $\sqrt{\alpha_t^{\text{pos}}}x_j^0$. Indeed, the sample at time t is transformed as $x_j^t = \sqrt{\alpha_t^{\text{pos}}}x_j^0 + \sqrt{1-\alpha_t^{\text{pos}}}\epsilon_j$.

Residue orientation $SO(3)$ continuous diffusion process The forward diffusion process works on the rotation group (Leach et al., 2022), $q(O_j^t|O_j^0) = \mathcal{IG}_{SO(3)}\left(O_j^t|\text{ScaleRot}\left(\sqrt{\bar{\alpha}_t^{\text{ori}}}, O_j^0\right), 1-\bar{\alpha}_t^{\text{ori}}\right)$ with $\mathcal{IG}_{SO(3)}$ the isotropic Gaussian distribution on $SO(3)$ parametrized by the mean rotation forward scaled by $\sqrt{\bar{\alpha}_t^{\text{ori}}}$ and the variance $1 - \bar{\alpha}_t^{\text{ori}}$. The function ScaleRot scales the rotation angle of the rotation matrix, where the rotation axis is fixed (Gallier & Xu, 2003). $\bar{\alpha}_t^{\text{ori}} = \prod_{\tau=1}^{t} \beta_\tau^{\text{ori}}$, with β_τ^{ori} the variance at time t. The reverse process is given by $p(O_j^{t-1}|R_t, C) = \mathcal{IG}_{SO(3)}\left(O_j^t|H_\theta\left(R_t, C\right), \beta_t^{\text{ori}}\right)_j$ where H_θ is a neural network that denoises the orientation matrix.

2 Guidance of Conditional Diffusion-Based Generative Models

Discrete-time diffusion processes are connected to continuous-time diffusion processes (Song et al., 2021). We use the results from (Didi et al., 2024) to introduce the guided diffusion, which connects an unconditional diffusion process and the condition to reach an end state $\boldsymbol{x}_0 \in Y$.

Proposition 2.1. *(Doob's h-transform Rogers & Williams (2000); Didi et al. (2024)) Consider the reverse SDE:* $\mathrm{d}\boldsymbol{x}_t = b_t(\boldsymbol{x}_t)\,\mathrm{d}t + \sigma_t \mathrm{d}\mathbf{W}_t, \quad \boldsymbol{x}_T \sim \mathcal{P}_T$ *where time flows backwards and with transition densities $p_{t|s}$. It then follows that the conditioned process $\boldsymbol{x}_t|\boldsymbol{x}_0 \in Y$ is a solution of*

$$\mathrm{d}\boldsymbol{x}_t = \left(b_t(\boldsymbol{x}_t) - \sigma_t^2 \nabla_{\boldsymbol{x}_t} \ln p_{0|t}(\boldsymbol{x}_0 \in Y|\boldsymbol{x}_t)\right)\,\mathrm{d}t + \sigma_t \mathrm{d}\mathbf{W}_t, \quad \boldsymbol{x}_T \sim \mathcal{P}_T, \tag{1}$$

such that $\boldsymbol{h}_s|\boldsymbol{h}_t, \boldsymbol{x}_0 \in Y \sim p_{s|t,0}(\boldsymbol{h}_s|\boldsymbol{h}_t, \boldsymbol{x}_0 \in Y)$ and $\mathbb{P}(\boldsymbol{x}_0 \in Y) = 1$.

Theorem 2.1 show that we can obtain a desired state $\boldsymbol{x}_0 \in Y$ by transforming an unconditional diffusion model. The conditional term $\nabla_{\boldsymbol{x}_t} \ln p_{0|t}(\boldsymbol{x}_0 \in Y|\boldsymbol{x}_t)$ substitutes the unconditional term $\nabla_{\boldsymbol{x}_t} \ln p_t(\boldsymbol{x}_t)$ and guarantees the target event $\boldsymbol{x}_0 \in Y$. Therefore, the drift term can be implemented using the conditional model as $\nabla_{\boldsymbol{x}_t} \ln p_{0|t}(\boldsymbol{x}_0 \in Y \mid \boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \ln p_{t|0}(\boldsymbol{x}_t \mid \boldsymbol{x}_0 \in Y) - \nabla_{\boldsymbol{x}_t} \ln p_t(\boldsymbol{x}_t)$.

Combining conditional diffusion processes with AND logic We consider the condition on two independent events $\boldsymbol{x}_0 \in A, \boldsymbol{x}_0 \in B$ happening at the same time $\boldsymbol{x}_0 \in A \wedge \boldsymbol{x}_0 \in B$.

Proposition 2.2. *Given two independent conditions $\boldsymbol{x}_0 \in A$ and $\boldsymbol{x}_0 \in B$, the score function of the $\boldsymbol{x}|\boldsymbol{x}_0 \in A \wedge \boldsymbol{x}_0 \in B$ is given by*

$$\nabla_{\boldsymbol{x}_t} \ln p(\boldsymbol{x}_0 \in A \wedge \boldsymbol{x}_0 \in B|\boldsymbol{x}_t) = \nabla \ln p(\boldsymbol{x}_t|\boldsymbol{x}_0 \in A) + \nabla \ln p(\boldsymbol{x}_t|\boldsymbol{x}_0 \in B)$$

If we want to combine two conditional diffusion models ($\nabla \ln p(\boldsymbol{x}_t|A)$), we have that $\nabla \ln p(\boldsymbol{x}_t|\boldsymbol{x}_0 \in A \wedge \boldsymbol{x}_0 \in B) = \nabla \ln p(\boldsymbol{x}_t|\boldsymbol{x}_0 \in A) + \nabla \ln p(\boldsymbol{x}_t|\boldsymbol{x}_0 \in B) - \nabla \ln p(\boldsymbol{x}_t)$ where we used $\nabla \ln p(\boldsymbol{x}_0 \in A|\boldsymbol{x}_t) = \nabla \ln p(\boldsymbol{x}_t|\boldsymbol{x}_0 \in A) - \nabla \ln p(\boldsymbol{x}_t)$. Proofs are available in the supplementary material.

Combining conditional diffusion processes with AND-NOT logic We consider the condition when the second event is negated, $\boldsymbol{x}_0 \in A \wedge \boldsymbol{x}_0 \in \bar{B}$, where the bar represents the negation $\neg B$. We call this combination ANDN as AND NOT.

Proposition 2.3. *Given two independent conditions A and B, the first-order Taylor expansion of the score function associated to $\boldsymbol{x}_t|\boldsymbol{x}_0 \in A \wedge \boldsymbol{x}_0 \in \bar{B}$ is given by*

$$\nabla \ln p(\boldsymbol{x}_0 \in A \wedge \boldsymbol{x}_0 \in \bar{B}|\boldsymbol{x}_t) \approx \nabla \ln p(\boldsymbol{x}_0 \in A|\boldsymbol{x}_t) - p(\boldsymbol{x}_0 \in B|\boldsymbol{x}_t) \nabla \ln p(\boldsymbol{x}_0 \in B|\boldsymbol{x}_t)$$

If we want to combine two conditional diffusion models, we have that $\nabla \ln p(\boldsymbol{x}_t|\boldsymbol{x}_0 \in A \wedge \boldsymbol{x}_0 \in \bar{B}) \approx \nabla \ln p(\boldsymbol{x}_t|\boldsymbol{x}_0 \in A) - p(\boldsymbol{x}_0 \in B|\boldsymbol{x}_t) \left(\nabla \ln p(\boldsymbol{x}_t|\boldsymbol{x}_0 \in B) - \nabla \ln p(\boldsymbol{x}_t)\right)$. This result defines a guidance framework that respects De Morgan's laws (Section D). The interesting observation is that the unconditional score function can be derived from the empty event, i.e., $\nabla \ln p(\boldsymbol{x}_t) = \nabla \ln p(\boldsymbol{x}_t|\emptyset)$. Furthermore, the absolute probability $p(\boldsymbol{x}_0 \in B|\boldsymbol{x}_t)$ is not accessible; we use an estimate, proportional to the inverse of the distance of two denoising states. For the discrete diffusion process on the residue types, we can not implement the diffusion at the level of the discrete variables; therefore, we implement the combination at the probability distribution of the multinomial distribution (in our model $F_\theta(R_t, C)$ from Section 1), enforcing the probability to stay non-negative.

Experiments

Metrics Following (Luo et al., 2022a), we evaluate the designed antibodies using four metrics: (1) Binding Energy Improvement (IMP): The percentage of CDRs with lower binding energy (ΔG) than the reference, computed with Rosetta (Alford et al., 2017). (2) Root-mean-square deviation (RMSD): The C_α deviation between generated and reference structures, with antibody frameworks aligned. (3) Amino Acid Recovery (AAR): Fraction of residues in the generated CDR sequences that match the reference sequences (Adolf-Bryfogle et al., 2017); (4) $\Delta\Delta G$: The change in ΔG between reference and generated complexes. As in (Didi et al., 2024), we do not use neutralization prediction models.

CDR	Target	Method	AAR (↑)	IMP(↑)	RMSD (↓)	$\Delta\Delta G$(↓)
H1	HAT-11	DiffAb	58.1	53.3	1.5	0.0
		DiffAb-AND	56.7	56.7	3.0	-0.2
		DiffAb-ANDN	57.4	**60.0**	1.4	-0.1
	Omicron	DiffAb	58.5	83.3	1.5	-3.9
		DiffAb-AND	54.8	**93.3**	3.0	-7.1
		DiffAb-ANDN	56.3	83.3	1.5	-6.1
H2	HAT-11	DiffAb	9.1	53.3	0.5	-0.1
		DiffAb-AND	9.1	50.0	1.1	-0.4
		DiffAb-ANDN	**10.3**	**63.3**	0.4	-0.3
	Omicron	DiffAb	8.2	93.3	0.5	-4.5
		DiffAb-AND	8.6	76.7	1.4	-3.2
		DiffAb-ANDN	**10.2**	33.3	0.5	1.2
H3	HAT-11	DiffAb	22.5	86.7	4.9	-20.6
		DiffAb-AND	11.8	40.0	200.6	-9.4
		DiffAb-ANDN	21.5	**90.0**	5.1	-21.4
	Omicron	DiffAb	21.4	60.0	4.2	103.3
		DiffAb-AND	22.2	30.0	200.0	26.8
		DiffAb-ANDN	**25.4**	50.0	6.2	144.7

Table 1: Heavy Chain CDR design based on DiffAb diffusion-based conditioned generative model averaged over 30 samples.

Results Tables 1 and 2 summarize the outcomes of applying logical guidance to two antigen targets: (1) SARS-CoV-2 RBD (Omicron, PDB ID: 7WVN, residues 322-590) and (2) Human Adenovirus type 11 (HAT-11, PDB ID: 2O39). Using DiffAb Luo et al. (2022b) as the baseline, the AND and ANDN combinations provide higher AAR for 2 out of 6 CDR designs and improved IMP in 3 out of 6. The higher RMSD reflects the larger structural rearrangements induced by the guided generation. The $\Delta\Delta G$ values are more favorable across all designs. Comparable or better trends were observed for the light-chain CDRs.

Conclusions This work extends the framework of (Didi et al., 2024) by introducing logical composition of conditional diffusion models through the AND and ANDN operators. While multi-target ligand design (Yang et al., 2024) addresses the AND case, here we demonstrate the ANDN scenario in antibody CDR generation. The proposed framework enables a flexible combination of conditional generative models for precise multi-target control in molecular design. In future work, the authors will consider the extension to a general combination of conditional diffusion flows to realize logic rules, compare with additional baselines, and potentially extend its application to different domains.

References

Jared Adolf-Bryfogle, Oleksandr Kalyuzhniy, Michael Kubitz, Brian D. Weitzner, Xiaozhen Hu, Yumiko Adachi, William R. Schief, and Roland L. Dunbrack. Rosettaantibodydesign (rabd): A general framework for computational antibody design. *PLoS Computational Biology*, 14, 2017. URL `https://api.semanticscholar.org/CorpusID:13658767`.

Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031–3048, 2017.

Nathaniel R. Bennett, Joseph L. Watson, Robert J. Ragotte, Andrew J. Borst, DéJenaé L. See, Connor Weidle, Riti Biswas, Yutong Yu, Ellen L. Shrock, Russell Ault, Philip J. Y. Leung, Buwei Huang, Inna Goreshnik, John Tam, Kenneth D. Carr, Benedikt Singer, Cameron Criswell, Basile I. M. Wicky, Dionne Vafeados, Mariana Garcia Sanchez, Ho Min Kim, Susana Vázquez Torres, Sidney Chan, Shirley M. Sun, Timothy Spear, Yi Sun, Keelan O'Reilly, John M. Maris, Nikolaos G. Sgourakis, Roman A. Melnyk, Chang C. Liu, and David Baker. Atomically accurate de novo design of antibodies with rfdiffusion. *bioRxiv*, 2025. doi: 10.1101/2024.03.14.585103. URL `https://www.biorxiv.org/content/early/2025/02/28/2024.03.14.585103`.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Kieran Didi, Francisco Vargas, Simon V. Mathis, Vincent Dutordoir, Emile Mathieu, Urszula J. Komorowska, and Pietro Lio. A framework for conditional diffusion modelling with applications in motif scaffolding for protein design. *arXiv*, arXiv:2312.09236, March 2024. doi: 10.48550/arXiv.2312.09236. URL `http://arxiv.org/abs/2312.09236`. arXiv:2312.09236 [cs].

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020.

Jean Gallier and Dianna Xu. Computing exponentials of skew-symmetric matrices and logarithms of orthogonal matrices. *International Journal of Robotics and Automation*, 18(1):10–20, 2003.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d, 2022. URL `https://arxiv.org/abs/2203.17003`.

Wengong Jin, Jeremy Wohlwend, Regina Barzilay, and Tommi Jaakkola. Iterative refinement graph neural network for antibody sequence-structure co-design. *arXiv preprint arXiv:2110.04624*, 2021.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation, 2023. URL `https://arxiv.org/abs/2206.01729`.

Xiangzhe Kong, Wenbing Huang, and Yang Liu. End-to-end full-atom antibody design. *arXiv preprint arXiv:2302.00203*, 2023.

Adam Leach, Sebastian M Schmon, Matteo T Degiacomi, and Chris G Willcocks. Denoising diffusion probabilistic models on so (3) for rotational alignment. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022.

Chengyi Liu, Wenqi Fan, Yunqing Liu, Jiatong Li, Hang Li, Hui Liu, Jiliang Tang, and Qing Li. Generative diffusion models on graphs: Methods and applications, 2023. URL https://arxiv.org/abs/2302.02591.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European conference on computer vision*, pp. 423–439. Springer, 2022.

Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022a. URL https://openreview.net/forum?id=jSorGn2Tjg.

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, 2022b.

Karolis Martinkus, Jan Ludwiczak, Wei-Ching Liang, Julien Lafrance-Vanasse, Isidro Hotzel, Arvind Rajpal, Yan Wu, Kyunghyun Cho, Richard Bonneau, Vladimir Gligorijevic, and Andreas Loukas. Abdiffuser: full-atom generation of in-vitro functioning antibodies. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=7GyYpomkEa.

Dmitry I Nikolayev and Tatjana I Savyolov. Normal distribution on the rotation group so (3). *Texture, Stress, and Microstructure*, 29(3-4):201–233, 1997.

L Chris G Rogers and David Williams. *Diffusions, Markov processes, and martingales*, volume 2. Cambridge university press, 2000.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL https://arxiv.org/abs/1503.03585.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020. URL https://arxiv.org/abs/1907.05600.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

Rubo Wang, Fandi Wu, Jiale Shi, Yidong Song, Yu Kong, Jian Ma, Bing He, Qihong Yan, Tianlei Ying, Peilin Zhao, Xingyu Gao, and Jianhua Yao. A generative foundation model for antibody design. *bioRxiv*, 2025. doi: 10.1101/2025.09.12.675771. URL https://www.biorxiv.org/content/early/2025/09/16/2025.09.12.675771.

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, 2024.

Yiren Yang, Yi Mou, Lin-Xi Wan, Shiou Zhu, Guan Wang, Huiyuan Gao, and Bo Liu. Rethinking therapeutic strategies of dual-target drugs: An update on pharmacological small-molecule compounds in cancer. *Medicinal Research Reviews*, 44(6):2600–2623, 2024.

Xu Zhang, Yiwei Liu, Yaming Wang, Liang Zhang, Lin Feng, Bo Jin, and Hongzhe Zhang. Multistage combination classifier augmented model for protein secondary structure prediction. *Frontiers in Genetics*, 13:769828, May 2022. ISSN 1664-8021. doi: 10.3389/fgene.2022.769828.

Zheng Zhao, Ziwei Luo, Jens Sjölund, and Thomas Schön. Conditional sampling within generative diffusion models. *Philosophical Transactions A*, 383(2299):20240329, 2025.

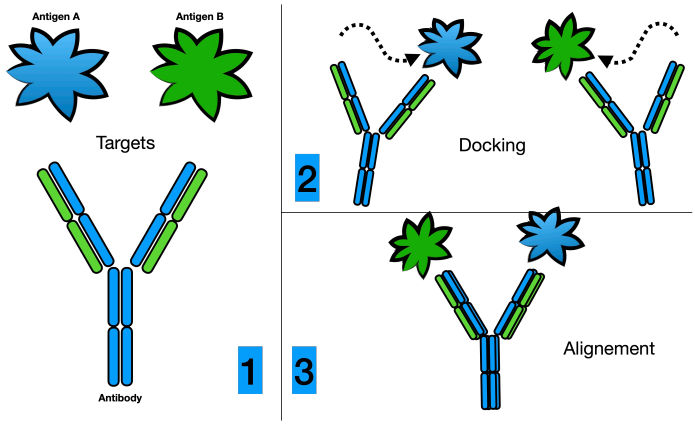

Figure 2: For each target, first dock the Antigen to the Antibody template. We then align the targets with respect to the Antibody template.

Supplementary Material of Guidance of Diffusion-Based Conditional Generative Models for Antibody Design

A Target preparation for diffusion model composition

Before we combine different conditional diffusion models, we need to align the targets. As shown in Figure 2, this is done using a template antibody. We first dock the antigen to the antibody template and then align all targets with respect to the same template.

B Continuous diffusion model

Forward process The *forward process* or diffusion process is a Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$:

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I})$$

Since the variance and mean are known, we can build the distribution at any time t as

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I})$$

$$\alpha_t = 1 - \beta_t, \quad \prod_{\tau=1}^{t} \alpha_\tau$$

Reverse process The forward process is used to train the *reverse process*, which is defined as a Markov chain starting from a fixed Gaussian prior $x_T \sim p(\boldsymbol{x}_t) = \mathcal{N}(x_T; \mathbf{0}, \boldsymbol{I})$, described by:

$$q(\boldsymbol{x}_{0:T}) = q(\boldsymbol{x}_T) \prod_{t=1}^{T} q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$$

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t))$$

The simplified training objective (Ho et al., 2020)

$$\mathcal{L} = \mathbb{E}_{t, \boldsymbol{x}_0 \sim D, \boldsymbol{\epsilon} \sim \mathcal{N}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|\right]$$

Denoise process During the denoise process, we first estimate the noise and then denoise the sample

$$\boldsymbol{\epsilon}_t = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$$

$$\boldsymbol{x}'_t = (1-\beta_t)^{-1/2}\big(\boldsymbol{x}_t - \beta_t(1-\bar{\alpha}_t)^{-1/2}\boldsymbol{\epsilon}_t\big) \quad \text{Denoise}$$

$$\boldsymbol{x}_t = \boldsymbol{x}_t + \sigma_t \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \quad \text{Brownian motion}$$

The denoise step is summarized here

$$\boxed{\boldsymbol{x}_{t-1} = \alpha_t^{-1/2}\big(\boldsymbol{x}_t - \beta_t\bar{\beta}_t^{-1/2}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\big) + \sigma_t\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \tag{2}$$

Composition rules The $A \wedge B$ conditional score function is given by

$$\hat{\boldsymbol{\epsilon}}_{\text{AND}}(\boldsymbol{x}_t, t) = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + w_A\big(\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t|A) - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\big) + w_B\big(\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t|B) - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\big),$$

w_A, w_B are an hyper-parameters. With $w_A = w_B = 1$

$$\hat{\boldsymbol{\epsilon}}_{\text{AND}}(\boldsymbol{x}_t, t) = -\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t|A) + \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t|B),$$

We currently use:

$$\hat{\boldsymbol{\epsilon}}_{\text{AND}}(\boldsymbol{x}_t, t) = -\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t|A) + w\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t|B),$$

While for the NOT, the process is defined by

$$\hat{\boldsymbol{\epsilon}}_{\text{ANDN}}(\boldsymbol{x}_t, t) = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + w\big(\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t|A) - p(B|\boldsymbol{x}_t)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t|B)\big).$$

C Logic rules

If we have two conditions A, B we can create logic rules for:

- $A \wedge B$
- $\neg A = \bar{A}$
- $A \wedge \neg B = A \wedge \bar{B}$

Proposition C.1. *Given an event A, we can write the gradient conditional distribution equalities*

$$\boxed{\nabla \ln p(x|A) = \nabla \ln p(x) + \nabla \ln p(A|x)}$$

$$\boxed{\nabla \ln p(A|x) = \nabla \ln p(x|A) - \nabla \ln p(x)}$$

Proof. First, we remind that Bayes' rule

$$p(x|A) = p(A|x)p(x)/p(A)$$

then we compute the gradient with respect to x,

$$\nabla \ln p(x|A) = \nabla \ln p(x) + \nabla \ln p(A|x) - \nabla_x \ln p(A)$$

but since $\nabla_x \ln p(A) = 0$, we have that

$$\nabla \ln p(x|A) = \nabla \ln p(x) + \nabla \ln p(A|x)$$

□

In the following we remove the condition on x, therefore instead of studying $p(A|x)$ we consider only $p(A)$. We then write the condition for the general case.

AND clause ($A \wedge B$)

Proposition C.2. *Given two independent events A and B, we can write the conditional distribution of x with respect the event $A \wedge B$ as*

$$\boxed{\nabla \ln p(x|A \wedge B) = \nabla \ln p(x) + \nabla \ln p(A|x) + \nabla \ln p(B|x)}$$

Proof. For diffusion models, we have that, if A, B are independent

$$\ln p(A \wedge B) = \ln p(A)p(B) = \ln p(A) + \ln p(B)$$

$$\nabla \ln p(x|A \wedge B) = \nabla \ln p(x) + \nabla \ln p(A \wedge B|x) = \nabla \ln p(x) + \nabla \ln p(A|x) + \nabla \ln p(B|x)$$

□

NOT clause ($\bar{A}, \neg A$) We of course have that $p(\bar{A}) = 1 - p(A)$.

Proposition C.3. *If A is an event, then we can write the first-order approximation for the gradient of the log probability of the conditional distribution*

$$\boxed{\nabla \ln p(x|\bar{A}) \approx \nabla \ln p(x) - p(A|x)\nabla \ln p(A|x)}$$

Proof. We first remind that the Taylor series for $\ln 1 + x$ is

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$$

therefore, for $|p| < 1$ we have

$$\ln(1-p) = -(p + \frac{p^2}{2} + \frac{p^3}{3} + \frac{p^4}{4} + \dots)$$

we can now compute the gradient

$$\begin{aligned}
\nabla_x \ln(1-p) &= -\nabla_x(p + \frac{p^2}{2} + \frac{p^3}{3} + \frac{p^4}{4} + \dots) \\
&= -(\nabla_x p + p\nabla_x p + p^2\nabla_x p + p^3\nabla_x p + \dots) \\
&= -(p\nabla_x \ln p + p^2\nabla_x \ln p + p^3\nabla_x \ln p + p^4\nabla_x \ln p \dots) \\
&= -p\nabla_x \ln p + O(p)
\end{aligned}$$

On the other hand

$$\nabla_x \ln(1-p) = -\nabla_x p \frac{1}{1-p} = -(1+p)\nabla_x p + O(p^2) = -\nabla_x p - p\nabla_x p + O(p^2) = -\nabla_x p + O(p) = -p\nabla_x \ln p + O(p)$$

since $(1-p)^{-1} = 1 + p + O(p^2)$

In conclusion, we consider the following first order approximation

$$\nabla_x \ln(1-p) \approx -p\nabla_x \ln p$$

then

$$\nabla_x \ln p(\bar{A}) = \nabla_x \ln(1 - p(A)) \approx -p(A)\nabla_x \ln p(A)$$

□

AND NOT $A \wedge \neg B = A \wedge \bar{B}$

Proposition C.4. *Given two independent events A and B, we can write the gradient of the condidional distribution to $A \wedge \neg B$ as*

$$\boxed{\nabla \ln p(x|A \wedge \bar{B}) \approx \nabla \ln p(x) + \nabla \ln p(A|x) - p(B|x)\nabla \ln p(B|x)}$$

Proof. The result is obtained by combining the two previous results. □

In the absence of the normalization function, we can scale the negative term

$$\boxed{\nabla \ln p(x|A \wedge \bar{B}) \approx \nabla \ln p(x) + \nabla \ln p(A|x) - \epsilon\frac{p(B|x)}{p(A|x)}\nabla \ln p(B|x)}$$

D De Morgan's laws

We now show that the proposed guidance framework adheres to De Morgan's laws.

De Morgan's laws De Morgan's laws:

$$\neg(A \wedge B) = \neg A \vee \neg B \tag{3}$$
$$\neg(A \vee B) = \neg A \wedge \neg B \tag{4}$$

Energy Models: For energy models (Liu et al., 2022; Du et al., 2020), the De Morgan's laws do not hold. For instance, for the LHS of Equation (3), we have:

$$\begin{aligned} p(x|A \wedge B) &\propto \exp(-E_A(x) - E_B(x)) \\ p(x|\neg(A \wedge B)) &\propto \exp(-E_\emptyset(x) + E_A(x) + E_B(x)), \end{aligned}$$

while, for the RHS we have:

$$\begin{aligned} p(x|\neg Y) &\propto \exp(-E_\emptyset(x) + E_Y(x)) \\ p(x|\neg A \vee \neg B) &\propto Z_A^{-1} \exp(-E_\emptyset(x) + E_A(x)) + Z_B^{-1} \exp(-E_\emptyset(x) + E_B(x)) \\ &= \exp(-E_\emptyset(x))(Z_A^{-1} \exp(E_A(x)) + Z_B^{-1} \exp(E_B(x))) \end{aligned}$$

Hence, $\exists A, B$ s.t. $p(x|\neg(A \wedge B)) \neq p(x|\neg A \vee \neg B)$.

Our diffusion model framework: For our approach, De Morgan's laws can be shown to hold. For instance, for the LHS of Eq. Equation (3) we have:

$$\begin{aligned} p(x|A \wedge B) &= \frac{p(A|x)p(B|x)p(x)}{Z} \\ p(x|\neg(A \wedge B)) &\propto p(x)p(\neg(A \wedge B)|x) = p(x)(1 - p(A \wedge B|x)) \\ Z &= \sum_x p(A|x)p(B|x)p(x) = p(A \wedge B), \end{aligned}$$

while for the RHS, we have:

$$p(x|\neg A \vee \neg B) \propto p(x)p(\neg A \vee \neg B|x) = p(x)(1 - p(A \wedge B|x)).$$

Similarly, for Equation (4) we have:

$$\begin{aligned} p(x|\neg(A \vee B)) \propto p(x)p(\neg(A \vee B)|x) &= p(x)(1 - p(A \vee B|x)) \\ p(x|\neg A \wedge \neg B) \propto p(x)p(\neg A \wedge \neg B|x) &= p(x)(1 - p(A|x))(1 - p(B|x)) \\ &= p(x)(1 - p(A|x) - p(B|x) + p(A \wedge B|x)) \\ &= p(x)(1 - p(A \vee B|x)) \end{aligned}$$

E Tweedie's formula

With a diffusion process, we can estimate (Ho et al., 2020)[eq.15] the end sample from the current sample by

$$\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t, \boldsymbol{\epsilon}_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\big(\boldsymbol{x}_t - \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_t\big), \quad \boldsymbol{\epsilon}_t = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t), \quad \bar{\alpha}_t = \prod_{\tau=1}^{t} \alpha_\tau, \quad \alpha_t = 1 - \beta_t$$

$$\boxed{\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t, \boldsymbol{\epsilon}_t) = \bar{\alpha}_t^{-1/2}\big(\boldsymbol{x}_t - \bar{\beta}_t^{1/2}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\big)} \tag{5}$$

F Discrete diffusion process

Forward process Let's note $\mathcal{C} =$ Multinomial, the forward diffusion process (Hoogeboom et al., 2021) and the long forward distribution, where the sample has β_t probability of sampling form the uniform distribution:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{C}(\boldsymbol{x}_t; (1-\beta_t)\boldsymbol{x}_{t-1} + \beta_t \frac{\mathbf{1}}{K})$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{C}(\boldsymbol{x}_t; \bar{\alpha}_t \boldsymbol{x}_0 + (1-\bar{\alpha}_t)\frac{\mathbf{1}}{K}), \quad \bar{\alpha}_t = \prod_{\tau=1}^{t} \alpha_\tau, \quad \alpha_t = 1 - \beta_t$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{C}(\boldsymbol{x}_t; \bar{\alpha}_t \boldsymbol{x}_0 + \bar{\beta}_t \frac{\mathbf{1}}{K}), \quad \bar{\beta}_t = 1 - \alpha_t$$

We therefore have two forward equations for the multinomial distribution

$$\boxed{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{C}(\boldsymbol{x}_t; \alpha_t \boldsymbol{x}_{t-1} + \beta_t \frac{\mathbf{1}}{K})} \tag{6}$$

$$\boxed{q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{C}(\boldsymbol{x}_t; \bar{\alpha}_t \boldsymbol{x}_0 + \bar{\beta}_t \frac{\mathbf{1}}{K}).} \tag{7}$$

Reverse process By combining the previous equations (Ho et al., 2020), we can compute the posterior distribution $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ as

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{C}(\boldsymbol{x}_{t-1}; \tilde{\boldsymbol{p}}_\theta(\boldsymbol{x}_t, \boldsymbol{x}_0, t))$$

$$\tilde{\boldsymbol{p}}_\theta(\boldsymbol{x}_t, \boldsymbol{x}_0, t) = \boldsymbol{p}_\theta(\boldsymbol{x}_t, \boldsymbol{x}_0, t) / \sum_{k=1}^{K} \boldsymbol{p}_\theta(\boldsymbol{x}_t, \boldsymbol{x}_0, t)_k$$

$$\boldsymbol{p}_\theta(\boldsymbol{x}_t, \boldsymbol{x}_0, t) = \big[\alpha_t \boldsymbol{x}_t + (1-\alpha_t)\frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_0 + (1-\bar{\alpha}_{t-1})\frac{\mathbf{1}}{K}\big]$$

$$\boldsymbol{p}_\theta(\boldsymbol{x}_t, \boldsymbol{x}_0, t) = \big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_0 + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big], \quad \bar{\beta}_{t-1} = 1 - \bar{\alpha}_{t-1}$$

and to summarize

$$\boxed{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{C}\big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_0 + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big].} \tag{8}$$

Or alternatively, but not usefully

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \frac{p(\boldsymbol{x}_{t-1})}{p(\boldsymbol{x}_t)} p(\boldsymbol{x}_t|\boldsymbol{x}_0)$$

$$\approx \mathcal{C} \frac{\bar{\alpha}_{t-1} + \bar{\beta}_{t-1}}{\alpha_t + \beta_t} \big[\alpha_t \boldsymbol{x}_{t-1} + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_t \boldsymbol{x}_0 + \bar{\beta}_t \frac{\mathbf{1}}{K}\big]$$

$$\frac{p(\boldsymbol{x}_{t-1})}{p(\boldsymbol{x}_t)} = \frac{\bar{\alpha}_{t-1}\mathbb{E}[\boldsymbol{x}_0] + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}}{\alpha_t \mathbb{E}[\boldsymbol{x}_0] + \beta_t \frac{\mathbf{1}}{K}} \approx \frac{\bar{\alpha}_{t-1} + \bar{\beta}_{t-1}}{\alpha_t + \beta_t}$$

We can think of inverting the forward equation as follows

$$\mathcal{C}\boldsymbol{x}_t = \mathcal{C}\alpha_t \boldsymbol{x}_{t-1} + \beta_{t-1}\frac{\mathbf{1}}{K}$$

$$\mathcal{C}\boldsymbol{x}_{t-1} = \mathcal{C}\alpha_t^{-1}(\boldsymbol{x}_t - \beta_{t-1}\frac{\mathbf{1}}{K}) \quad \text{Reverse direction}$$

If we use this reversion, we can write the posterior distribution as

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) p(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{C}\big[\alpha_t^{-1}(\boldsymbol{x}_t - \beta_{t-1}\frac{\mathbf{1}}{K})\big] \odot \big[\bar{\alpha}_t \boldsymbol{x}_0 + \bar{\beta}_t \frac{\mathbf{1}}{K}\big]$$

and to summarize

$$\boxed{p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{C}\big[\alpha_t^{-1}(\boldsymbol{x}_t - \beta_{t-1}\frac{\mathbf{1}}{K})\big] \odot \big[\bar{\alpha}_t \boldsymbol{x}_0 + \bar{\beta}_t \frac{\mathbf{1}}{K}\big]} \tag{9}$$

Denoise process Since during the reverse process $\boldsymbol{x}_0$ is not available, we use the prediction from a neural network $\boldsymbol{f}_\theta$, as

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{f}_\theta(\boldsymbol{x}_t, t)$$

where $\boldsymbol{f}_\theta$ includes a soft-max to ensure the positive definiteness. We then can interpret $\hat{\boldsymbol{x}}_0(x_t, t)$ as the (un-normalized) probability of the classes at the end of the diffusion process.

G The logic of discrete diffusion processes

We can then combine the predictions to form the AND and AND-NOT logic.

G.1 Posterior distribution for the AND logic

We first derive the following relationship applying Bayes' theorem:

$$\begin{aligned} p(t|A \wedge B) &= p(A \wedge B|t)\frac{p(t)}{p(A \wedge B)} = p(A|t)p(B|t)\frac{p(t)}{p(A)p(B)} \\ &= \frac{p(A|t)}{p(A)}p(t)\frac{p(B|t)}{p(B)}p(t)\frac{1}{p(t)} = \frac{p(t|A)p(t|B)}{p(t)} \end{aligned}$$

if we consider $p(t-1|t, A \wedge B)$ and map $t \leftarrow t-1|t$ we have $p(t-1|t, A, B) = p(t-1|t, A)p(t-1|t, B)/p(t-1|t)$, where $p(t-1|t)$ is the unconditional conditional posterior distribution, while $p(t-1|t, A)p(t-1|t, B)$ are the two conditional posterior distributions. Let's consider the following relationship by using Equation (8),

$$\begin{aligned} p(t-1|t, A) &= \mathcal{C}\big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_{0|A} + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big] \\ p(t-1|t, B) &= \mathcal{C}\big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_{0|B} + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big] \\ p(t-1|t) &= \mathcal{C}\big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_0 + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big] \end{aligned}$$

where we estimate $\boldsymbol{x}_{0|A} = \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, A)$, and $\boldsymbol{x}_{0|B} = \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, B)$.

When can we then combine and get the following relationship

$$\begin{aligned} p(t-1|t, A \wedge B) &= p(t-1|t, A)p(t-1|t, B)/p(t-1|t) \\ &= \mathcal{C}\big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_{0|A} + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big] \\ &\odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_{0|B} + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_0 + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big]^{-1} \end{aligned}$$

to summarize

$$\boxed{\begin{aligned} p(t-1|t, A \wedge B) &= \mathcal{C}\big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_{0|A} + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big] \\ &\odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_{0|B} + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1}\boldsymbol{x}_0 + \bar{\beta}_{t-1}\frac{\mathbf{1}}{K}\big]^{-1} \end{aligned}} \tag{10}$$

G.2 Posterior distribution for the AND NOT logic

We have:

$$\begin{aligned} p(B|x_{t-1}, x_t) &= \frac{p(x_{t-1}|B, x_t)p(B|x_{t-1})}{p(x_{t-1}|x_t)} \\ p(\neg B|x_{t-1}, x_t) &= 1 - p(B|x_{t-1}, x_t) \\ &= \frac{p(x_{t-1}|x_t) - p(x_{t-1}|B, x_t)p(B|x_{t-1})}{p(x_{t-1}|x_t)} \end{aligned}$$

Hence, by Bayes' rule:

$$\begin{aligned} p(x_{t-1}|\neg B, x_t) &= \frac{p(\neg B|x_{t-1}, x_t)p(x_{t-1}|x_t)}{p(\neg B|x_t)} \\ &= \frac{p(x_{t-1}|x_t)^2 - p(x_{t-1}|x_t)p(x_{t-1}|B, x_t)p(B|x_t)}{p(x_{t-1}|x_t)p(\neg B|x_t)} \\ &= \frac{p(x_{t-1}|x_t) - p(x_{t-1}|B, x_t)p(B|x_t)}{1 - p(B|x_t)} \end{aligned} \tag{11}$$

We note that the only quantity in Eq. 11 not immediately available from the denoising model for B is $p(B|x_t)$, the probability of B under the unconditional model given x_t. This quantity may be estimated by Monte Carlo sampling, i.e. by repeatedly sampling x_0 by denoising x_t under the unconditional model, and observing the fraction of samples with property B. Alternatively, a neural network may be pre-trained to estimate this quantity by repeated off-line sampling from the unconditional model, and minimizing the cross-entropy between a binary vector of predicted properties (for A and B) given x_t and t, and the properties of the generated x_0. Lastly, we may use a heuristic estimate of this quantity, for instance by evaluating $p(B|f(x_t, t))$, or letting $p(B|x_t) = Z_{x_t}^{-1} \exp(-\tau \text{Hamming}(f(x_t, t), f(x_t, t, B)))$, where Z_{x_t} is a normalization factor, and τ is a temparature parameter. We then define the following weights:

$$v_t^B = \frac{1}{1 - p(B|x_t)}, \;\; w_t^B = \frac{p(B|x_t)}{1 - p(B|x_t)}, \;\; v_t^B - w_t^B = 1$$

Using the above heuristic, we may thus set:

$$\begin{aligned} v_t^B &= Z_{x_t}/(Z_{x_t} - \exp(-\tau \text{Hamming}(f(x_t, t), f(x_t, t, B))) \\ &= \sigma(\tau' \text{Hamming}(f(x_t, t), f(x_t, t, B))) \end{aligned}$$

where $\tau' = \tau / \exp(Z_{x_t})$, $\sigma(a) = 1/(1 - \exp(-a))$, and $w_t^B = v_t^B - 1$. We therefore have:

$$\begin{aligned} p(t-1|t, \neg B) &= v_t^B \mathcal{C}\big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1} \boldsymbol{x}_0 + \bar{\beta}_{t-1} \frac{\mathbf{1}}{K}\big] - \\ &\quad w_t^B \mathcal{C}\big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1} \boldsymbol{x}_{0|B} + \bar{\beta}_{t-1} \frac{\mathbf{1}}{K}\big] \\ &= C\big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[v_t^B \bar{\alpha}_{t-1} \boldsymbol{x}_0 - w_t^B \bar{\alpha}_{t-1} \boldsymbol{x}_{0|B} + (v_t^B - w_t^B) \bar{\beta}_{t-1} \frac{\mathbf{1}}{K}\big] \end{aligned}$$

Then, to summarize, we have:

$$\begin{aligned} p(t-1|t, A \wedge \bar{B}) &= \mathcal{C}\big[\alpha_t \boldsymbol{x}_t + \beta_t \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1} \boldsymbol{x}_{0|A} + \bar{\beta}_{t-1} \frac{\mathbf{1}}{K}\big] \\ &\quad \odot \big[v_t^B \bar{\alpha}_{t-1} \boldsymbol{x}_0 - w_t^B \bar{\alpha}_{t-1} \boldsymbol{x}_{0|B} + \bar{\beta}_{t-1} \frac{\mathbf{1}}{K}\big] \odot \big[\bar{\alpha}_{t-1} \boldsymbol{x}_0 + \bar{\beta}_{t-1} \frac{\mathbf{1}}{K}\big]^{-1} \end{aligned} \tag{12}$$

Heuristic Another way is to combine the prediction only

$$\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t, t, A \wedge B) = \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, A) + w_t^{\text{AND}}(\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, B) - \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, \emptyset)) \tag{13}$$

$$\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t, t, A \wedge \neg B) = \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, A) - w_t^{\text{ANDN}}(\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, B) - \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, \emptyset)) \tag{14}$$

$$w_t^{\text{AND}} = \big[\|\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, A) - \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, B)\|\big]_{\bar{w}_{\text{MAX}}^{\text{AND}}}$$

$$w_t^{\text{ANDN}} = \big[1/\|\boldsymbol{f}_\theta(\boldsymbol{x}_t, t, A) - \boldsymbol{f}_\theta(\boldsymbol{x}_t, t, B)\|\big]_{\bar{w}_{\text{MAX}}^{\text{ANDN}}}$$

with $[.]_{w_{\text{MAX}}}$ is the the $\min(., w_{\text{MAX}})$.

H $SO(3)$ rotation diffusion process

We consider the $SO(3)$ diffusion process (Hoogeboom et al., 2022) defined using the isotropic Gaussian distribution on $SO(3)$ ($\mathcal{IG}$) (Nikolayev & Savyolov, 1997) $g \sim \mathcal{IG}_{SO(3)}(\mu, \sigma^2)$, with μ, σ the mean and variance parameters. The density parametrized by the axis-angle form, with uniformly sampled axes, and the rotation angle $\omega \in [0, \pi]$, whose density is given by $f(\omega) = \frac{1-\cos\omega}{\pi} \sum_{l=0}^{\infty}(2l+1)e^{-l(l+1)\sigma^2}\frac{\sin(l+1)\omega}{\sin\omega/2}$. Interestingly, the $\mathcal{IG}$ distribution is closed under convolution. We need to train the model sampling from $\boldsymbol{x}_0 \in SO(3)$. If the rotation angle of a rotation matrix R is ω, than $\ln R = \frac{\omega}{2\sin\omega}(R^T - R)$ (or $A = 1/2(R - R^T), \ln R = \frac{\sin^{-1}\|A\|}{\|A\|}A, \|A\|^2 = -1/2\mathrm{tr}A^2$), where ω satisfies $\mathrm{tr}R = 1 + 2\cos\omega$. The lie algebra $so(3)$ of $SO(3)$ is composed of skew-symmetric matrices $S(v) = \begin{pmatrix} 0 & v_3 & -v_2 \\ -v_3 & 0 & v_1 \\ v_2 & -v_1 & 0 \end{pmatrix}, \boldsymbol{v} = [v_1, v_2, v_3], \|v\|_2 = \omega$. The scaling by α, is implemented in the $so(3)$ by the function $\boldsymbol{\lambda}(\alpha, \boldsymbol{x}) = \exp\alpha \ln \boldsymbol{x}$, which represents the scaling along the geodesic from $\boldsymbol{I}$ to $\boldsymbol{s}$ by the amount α. The exponential map is defined as $e^{\boldsymbol{x}} = \boldsymbol{I} + \frac{\sin\omega}{\omega}\boldsymbol{x} + 2\frac{\sin^2\omega/2}{\omega^2}\boldsymbol{s}^2, \omega = \|\boldsymbol{x}\|$.

Forward process The diffusion model is then defined by a forward diffusion process as

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{IG}_{SO(3)}(\boldsymbol{\lambda}(\bar{\alpha}_t^{1/2}, \boldsymbol{x}_0), \bar{\beta}_t)$$

Reverse process The reverse diffusion process is implemented as

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{IG}_{SO(3)}(\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, \boldsymbol{x}_0, t), \tilde{\beta}_t)$$
$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, \boldsymbol{x}_0, t) = \boldsymbol{\lambda}\big(\bar{\beta}_t^{-1}\bar{\alpha}_t^{1/2}\beta_t, \boldsymbol{x}_0\big)\boldsymbol{\lambda}\big(\bar{\beta}_t^{-1}\alpha_t^{1/2}\bar{\beta}_{t-1}, \boldsymbol{x}_t\big)$$

In practice, $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, \boldsymbol{x}_0, t)$ is a neural network, using the vector representation of $\boldsymbol{x}$.

AND To implement $A \wedge B$ we apply the two rotations

$$\boldsymbol{x}_{t|A\wedge B} = \boldsymbol{x}_{t|A}\boldsymbol{x}_{t|B}$$

with $\boldsymbol{v}_t \in so(3)$ are computed as

$$\boldsymbol{v}_{t|A\wedge B} = \ln(\exp\boldsymbol{v}_{t|A}\exp\boldsymbol{v}_{t|B})$$

Heuristic Since the $\boldsymbol{v}$ vectors are defined in the Lie Algebra, we can implement with

$$\boldsymbol{v}_{t|A\wedge B} = \boldsymbol{v}_{t|A} + \boldsymbol{v}_{t|B}$$

AND NOT To implement $A \wedge \neg B$, we apply the two rotations

$$\boldsymbol{x}_{t|A\wedge\neg B} = \boldsymbol{x}_{t|A}\boldsymbol{x}_{t|B}^T$$

with $\boldsymbol{v}_t \in so(3)$ are computed as

$$\boldsymbol{v}_{t|A\wedge\neg B} = \ln(\exp\boldsymbol{v}_{t|A}\exp\boldsymbol{v}_{t|B}^T)$$

Heuristic Similarly to the previous case, since the $\boldsymbol{v}$ vectors are defined in the Lie Algebra, we can implement with

$$\boldsymbol{v}_{t|A\wedge\neg B} = \boldsymbol{v}_{t|A} - \boldsymbol{v}_{t|B}$$

FKC correction To allow our sampling methods to better approximate the conditional distribution $p(\boldsymbol{x}_0|A \wedge B)$ or $p(\boldsymbol{x}_0|A \wedge \neg B)$, we may introduce Feynman-Kac corrections (FKC) at each sampling step. Following Zhao et al. (2025), a Sequential Monte Carlo (SMC) sampler for the Feynman-Kac model may be defined by introducing a proposal kernel, $M_{t-1|t}(.|\boldsymbol{x}_t)$, and weighting potentials, $G_{t-1,t}(.,.)$ and $G_T(.)$. The SMC sampler then proceeds by first drawing J samples $\boldsymbol{x}_T^{j=1\ldots J}$ from M_T (the reference noise distribution), and calculating weights $w_T^j = G_T(\boldsymbol{x}_T^j)/\sum_{j'}(G_T(\boldsymbol{x}_T^{j'}))$. For steps $t = T, ..., 1$, the sampler then: (a) resamples $\{(w_t^j, \boldsymbol{x}_t^j)\}_{j=1}^J$ if $\exists j.w_t^j \approx 0$; (b) draws new samples $\boldsymbol{x}_{t-1}^j \sim M_{t-1|t}(.|\boldsymbol{x}_t^j)$ and computes the unnormalized weights $\bar{w}_{t-1}^j = w_t^j G_{t-1,t}(\boldsymbol{x}_{t-1}^j, \boldsymbol{x}_t^j)$; (c)

normalizes the weights $w_{t-1}^j = \bar{w}_{t-1}^j / \sum_{j'} \bar{w}_{t-1}^{j'}$. The algorithm then outputs the weighted samples $\{(w_0^j, \boldsymbol{x}_0^j)\}_{j=1}^J$.

Following Wu et al. (2024), we set:

$$G_{t-1,t}^Y(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t) = \frac{p(Y|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{p(Y|\boldsymbol{x}_t)M_{t-1|t}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}$$

and $G_T^Y(\boldsymbol{x}_T) = p(Y|\boldsymbol{x}_T)$, where $M_{t-1|t}^Y(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is the proposal distribution generated by our framework, and Y is the condition, in our case $Y = \{A \wedge B, A \wedge \neg B\}$. Here, $p(Y|\boldsymbol{x}_t) = p(A|\boldsymbol{x}_t)p(B|\boldsymbol{x}_t)$ and $p(Y|\boldsymbol{x}_t) = p(A|\boldsymbol{x}_t)(1 - p(B|\boldsymbol{x}_t))$ for the cases $Y = A \wedge B$ and $Y = A \wedge \neg B$ respectively, and the requisite probabilities may be estimated using the sampling, neural network or heuristic approaches noted above.

I Light Chain CDR

We report the additional results for the light chain CDR in Table 2. The corresponding metrics and test systems are discussed in the main text.

CDR	Target	Method	AAR (↑)	IMP(↑)	RMSD (↓)	$\Delta\Delta G$(↓)
L1	HAT-11	DiffAb	74.1	60.0	0.9	-2.3
		DiffAb-AND	61.3	60.0	134.9	252.1
		DiffAb-ANDN	74.6	**73.3**	0.8	-5.4
	Omicron	DiffAb	71.8	86.7	0.9	-13.3
		DiffAb-AND	61.4	26.7	128.9	1452.3
		DiffAb-ANDN	**74.6**	**96.7**	0.9	-22.3
L2	HAT-11	DiffAb	57.5	80.0	1.8	-5.1
		DiffAb-AND	28.0	63.3	100.7	55.6
		DiffAb-ANDN	54.3	73.3	1.8	-7.0
	Omicron	DiffAb	57.1	30.0	2.0	3.9
		DiffAb-AND	38.7	30.0	118.0	364.1
		DiffAb-ANDN	55.9	**76.7**	1.9	-5.5
L3	HAT-11	DiffAb	42.6	20.0	1.2	0.5
		DiffAb-AND	42.5	36.7	2.7	0.6
		DiffAb-ANDN	**44.3**	**36.7**	1.4	-0.0
	Omicron	DiffAb	46.6	96.7	1.3	-51.5
		DiffAb-AND	42.5	76.7	2.3	-22.9
		DiffAb-ANDN	44.2	86.7	1.2	-44.0

Table 2: Light Chain CDR design based on DiffAb diffusion-based conditioned generative model averaged over 30 samples.