Few-step Cofolding with All-Atom Flow Maps

Gianluca Scarpellini [*1] **Ron Shprints** [2] **Peter Holderrieth** [2] **Juno Nam** [2] **Pranav Murugan** [1]
Rafael Gómez-Bombarelli [2] **Tommi Jaakkola** [2] **Maruan Al-Shedivat** [1] **Nicholas Boffi** [3] **Avishek Joey Bose** [*1 4 5]

Abstract

All-atom generative modeling of 3D biomolecular complexes has emerged as the dominant paradigm for predicting the structure of proteins and protein-ligand systems. Generating structures at the atomic level of fidelity, however, typically requires expensive iterative diffusion rollouts, making both conventional deployment and inference-time search techniques computationally costly. In this paper, we introduce the **DENOISER COFOLDING ALL-ATOM FLOWMAP** (DECAF) framework for distilling state-of-the-art all-atom cofolding models into all-atom flow maps that produce high-quality samples in only a few inference steps. We build DECAF on a denoiser-based formulation of flow maps with endpoint losses that naturally support $\mathrm{SE}(3)$ rigid alignment, which we show is critical for training accurate models. We further derive a simple change of variables that lets DECAF operate in the σ-space noise schedule of EDM-style architectures, enabling direct distillation from pretrained cofolding diffusion models. Equipped with DECAF's flowmap lookahead, we introduce a purpose-built inference-time framework that improves sampling through reward-guided search. Empirically, DECAF statistically improves over Boltz-1x in both accuracy (RMSD) and physical validity scores of protein-ligand poses at strict NFE budgets on the challenging Runs N' Poses, while also showing a more optimal Pareto frontier across all inference compute budgets on PoseBusters.

[*]Equal contribution [1]Genesis Molecular AI [2]Massachusetts Institute of Technology [3]Carnegie Mellon University [4]Imperial College London [5]Mila. Correspondence to: Gianluca Scarpellini <gianluca@genesistherapeutics.ai>, Avishek Joey Bose <joey@genesistherapeutics.ai>.

Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)

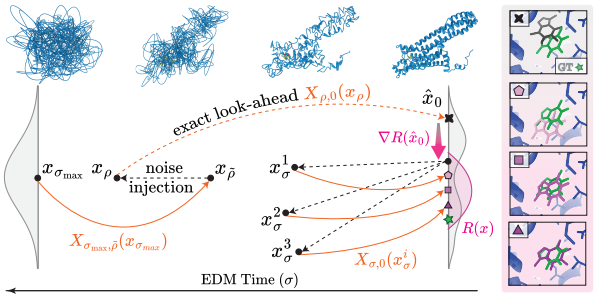

Figure 1. DECAF accelerates all-atom biomolecular structure prediction with a few-step flow-map lookahead across EDM noise levels. Atom-resolution guidance enables candidate search toward high-reward configurations.

1. Introduction

The accurate and efficient computational modeling of biological complexes has the potential to transform both our understanding of biomolecular mechanisms and our ability to catalyze the rational design of novel therapeutics (Chevalier et al., 2017; Ebrahimi & Samanta, 2023). At the core of this challenge is the need to model the intricate 3D intermolecular structures that govern processes such as protein folding, protein–ligand binding, and biocatalysis. This perspective is the thesis of structure-based drug design (SBDD), which seeks to engineer molecular structure to impart a desired downstream biological *function*. Despite its promise, progress in traditional SBDD is limited by the cost and latency of experimental structure determination. In contrast, AI-driven design offers a promising alternative by instead leveraging scalable computational approaches to unlock discoveries of novel therapies (Silva et al., 2019; Murray et al., 2022; Strauch et al., 2017; Gainza et al., 2020; Cao et al., 2020).

The modern *de facto* standard for AI-based SBDD is underpinned by large-scale generative models that represent biomolecular complexes of experimentally resolved structures directly at all-atom resolution, beginning with AlphaFold 3 (Abramson et al., 2024) and followed by other luminary works (Boitreaud et al., 2024; Wohlwend et al., 2025; Passaro et al., 2025; Team et al., 2026; Genesis Research, 2025). This perspective is well-suited to learning the global geometry of the target distribution, reflected in structural features such as relative pose, backbone arrange-

ment, and secondary structure. However, unlike in classical generative modeling domains, success in biomolecular generation fundamentally requires modeling fine-grained local structure. Indeed, in high-impact application settings like protein-ligand cofolding, inaccurate local structure modeling leads to catastrophic failure modes, yielding physically invalid generations that often include steric clashes, incorrect bond lengths and angles, strained side chain placements, and other stereochemical artifacts (Wohlwend et al., 2025).

Strict physical and biological constraints have shifted much of the compute burden to *inference time methods*. In particular, generating structures requires expensive and fine-grained numerical simulation of the learned dynamics, which are needed to accurately predict local structure. In addition, for downstream testing, it remains essential to generate a diverse pool of candidates that increases the transfer rate from in-silico design to wet lab success. Furthermore, refining samples via inference-time search with proxy physical reward models is a critical aspect of the evolving protein generative pipeline and plays an important role in facilitating utility in high-impact downstream applications. Despite this appeal, scaling inference is not a silver bullet and compounds the already expensive cost of numerical integration. For instance, reward functions in the biological setting can often be expensive to query and are only applicable to fully denoised 3D structures. Moreover, employing popular inference-time techniques that leverage multiple particles, such as Sequential Monte Carlo (SMC) (Del Moral et al., 2006), Feynman-Kac steering (FK) (Singhal et al., 2025), and Monte-Carlo Tree Search (MCTS) (Jain et al., 2025), doubly inflict the inference tax as they require multiple reward queries over each particle during the inference trajectory. This raises the natural motivating research question:

Q. *Can we train an all-atom cofolding model that can generate reward-optimized samples efficiently, using only a small number of neural function evaluations (NFEs) at inference time?*

Main contributions. In this paper, we answer the above question affirmatively and introduce the **DE**NOISER **C**OFOLDING **A**LL-ATOM **F**LOWMAP (DECAF) framework. DECAF is built on the flow map framework (Boffi et al., 2025b;a) (Figure 1) and efficiently distills a pretrained Boltz-1 model into the first all-atom cofolding model. Overall, we summarize our contributions as follows:

1. **Training methods**. We design the first all-atom protein flow-map with DECAF, and outline key methodological innovations that enable effective distillation of a pre-trained Boltz-1 teacher. In particular, we construct a novel reparameterization of the flow map in σ space that allows an easy conversion of standard flow map objectives for all-atom modeling. DECAF further exploits a denoiser-based flow map parametrization that crucially enables an $\mathrm{SE}(3)$ weighted rigid alignment of the ground truth structure—softly enforcing $\mathrm{SE}(3)$ symmetry—that is critical for stable training.
2. **Inference methods**. We introduce DECAF-SEARCH, an inference-time framework that leverages DECAF's flow map lookahead that enables higher fidelity reward estimation in comparison to all-atom diffusion models. DECAF-SEARCH shares the benefits of stochastic sampling (Kim et al., 2023), diffusion-SMC samplers (Singhal et al., 2025), Diffusion MCTS (Jain et al., 2025), Diamond maps (Holderrieth et al., 2026b; Potaptchik et al., 2026a), and FMRG (Huang et al., 2026) while being fundamentally an *inference-time search* method generating physically valid structures.
3. **Empirical performance**. We find that DECAF outperforms Boltz-1x *at low NFE budgets* on Runs N' Poses and matches the full-budget Boltz-1x (600 NFE) on PoseBusters with $20\times$ less compute at inference.

2. Background, Preliminaries, and Related Work

2.1. All-Atom Biomolecular Diffusion Modeling

The dominant paradigm for all-atom biomolecular generative modeling is centered around diffusion-based cofolding models such as AlphaFold 3 (Abramson et al., 2024), which operates directly on the native Euclidean coordinates of structures. Through learning, AF3 like models can perform the task of *cofolding*—i.e., simultaneously predict the structure of a protein and a bound ligand. Standard implementations follow the EDM parameterization (Karras et al., 2022), which we review below.

VE Process. Given a target distribution of all-atom structures $p_{\text{data}}(x) \in \mathcal{P}(\mathbb{R}^d)$, the variance-exploding noising process corrupts a sample $x_0 \sim p_0(x_0) := p_{\text{data}}(x)$ with additive Gaussian noise,

$$\begin{aligned} x_\sigma &= x_0 + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \\ p_\sigma(x_\sigma) &= p_0(x_0) * \mathcal{N}(0, \sigma^2 I), \end{aligned} \tag{1}$$

where $\sigma \in [\sigma_{\min}, \sigma_{\max}]$ indexes the noise level. Generation proceeds by solving for the time-reversal of these dynamics, which can be numerically simulated by following an ODE or SDE from high to low noise. In particular, the probability-flow ODE is:

$$\tfrac{dx_\sigma}{d\sigma} = -\sigma^2 \nabla_{x_\sigma} \log p_\sigma(x_\sigma).$$

These reverse ODE dynamics require the Stein score $\nabla_{x_\sigma} \log p_\sigma(x_\sigma)$, which can be estimated through a diffusion model's denoiser D via Tweedie's formula (Tweedie, 1957):

$$s_\sigma(x_\sigma) := \nabla_{x_\sigma} \log p_\sigma(x_\sigma) \approx \frac{D_\sigma(x_\sigma) - x_\sigma}{\sigma^2}. \tag{2}$$

Eq. 2 can then be substituted in the flow ODE to simulate the reverse dynamics. The denoiser itself can be learned by a *simulation-free* ℓ_2-regression objective that performs denoising score matching across all noise levels ith a noise-dependent weighting function $\lambda(\sigma)$ against a clean sample x_0,

$$\mathcal{L}(\theta) = \mathbb{E}_{\sigma,\, x_0,\, \epsilon}\left[\lambda(\sigma)\, \|D_{\theta,\sigma}(x_0 + \sigma\epsilon) - x_0\|_2^2\right], \quad (3)$$

where $\epsilon \sim \mathcal{N}(0, I)$. **Structure prediction head**. For biomolecular all-atom diffusion models, the denoiser commonly leverages the EDM parametrization (Karras et al., 2022) that designs a σ-dependent preconditioning:

$$\begin{aligned}\hat{x}_0^{\mathrm{EDM}} &:= D_{\theta,\sigma}(x_\sigma) \\ &= c_{\mathrm{skip}}(\sigma)\, x_\sigma + c_{\mathrm{out}}(\sigma)\, F_{\theta,\sigma}(c_{\mathrm{in}}(\sigma)\, x_\sigma, c_{\mathrm{noise}}(\sigma))\,. \end{aligned} \quad (4)$$

where F_θ is the raw network, c_{skip} controls the skip connection, c_{in} and c_{out} normalize input and output magnitudes, and c_{noise} embeds the noise level. In practice, the denoiser is implemented as a structure prediction head, typically a diffusion transformer with atom-attention encoder-decoder blocks.

The basic loss in Eq. 3 is augmented through $\mathrm{SE}(3)$ weighted rigid alignment of the ground truth structures to the prediction $\hat{x}_0^{\mathrm{EDM}}$. This crucial step serves to simultaneously enforce soft global $\mathrm{SE}(3)$ symmetry and also reduce the variance of the diffusion loss. Finally, additional loss terms, including smooth-LDDT or bond-geometry penalties, encourage generating chemically plausible structures

2.2. Flow Maps

To accelerate inference in diffusion models, rather than simulating infinitesimal dynamics, one can learn a jump operator that directly traverses the probability-flow ODE associated with the diffusion model (Song et al., 2023; Song & Dhariwal, 2024; Boffi et al., 2025b;a). This operator is known as the *flow-map* and constitutes a map $X_{s,t} : [0,1]^2 \times \mathbb{R}^d \to \mathbb{R}^d$, which is the unique solution to the ODE that evolves the state dynamics between times s and t—i.e., the jump condition satisfies $X_{s,t}(x_s) = x_t$, for all $(s,t) \in [0,1]^2$. This leads to a natural parametrization of the flow-map as a displacement using the *average velocity* $u_{s,t}(x_s)$, between the two-time points s and t:

$$\begin{aligned} X_{s,t}(x_s) &= x_s + (t-s)u_{s,t}(x_s), \\ u_{s,t}(x_s) &= \frac{1}{t-s}\int_s^t v_\tau(x_\tau)d\tau, \end{aligned} \quad (5)$$

where $v_\tau(x_\tau)$, is the instantaneous velocity. It is clear from Eq. 5, that in the limit where the two time points converge the average velocity recovers the instantaneous velocity of the ODE, $\lim_{s\to t}\partial_t X_{s,t}(x_s) = u_{t,t}(x_t) := v_t(x_t)$. This is known as the tangent condition (Boffi et al., 2025b), and demonstrates that the flow-map contains an implicit instantaneous velocity in its parametrization.

At optimality, the flow-map simultaneously enforces Lagrangian, Eulerian, and progressive consistency rules (Boffi et al., 2025a;b); each gives rise to a PINN-style loss that can be used to learn or distill a pre-trained diffusion model into a flow-map (Frans et al., 2025; Kim et al., 2024).

3. Method

We now introduce **DENOISER COFOLDING ALL-ATOM FLOWMAP** (DECAF), a novel σ-space flow-map framework that constructs an all-atom protein flow map by distilling a pre-trained all-atom teacher in the vein of AF3 (Abramson et al., 2024). In particular, DECAF outputs a few-step generative model that captures the 3D structure of protein-ligand interactions for the task of cofolding. Critically, DECAF offers two principal advantages over its pre-trained all-atom teacher:

1. *Few-step inference:* DECAF offers accelerated inference that compresses the full simulation of a diffusion trajectory into a few denoising steps without compromising sample quality.
2. *Flowmap lookahead:* DECAF by construction defines a lookahead map over end points that allows higher fidelity terminal reward estimation than a denoiser of an EDM model. As a result, any inference-time technique for reward optimization benefits not only fewer simulation steps that reduce simulation latency but also improved reward estimation and alignment.

We organize the remainder of the section as follows: in §3.1, we introduce our primary learning framework that exploits a denoiser-parametrization to learn DECAF. In §3.2, we exploit our DECAF to unlock new, more efficient mechanisms for inference-time search and reward alignment.

3.1. DENOISER COFOLDING ALL-ATOM FLOWMAP

We first highlight several key technical challenges prevalent in the EDM parametrization that prevent the distillation of all-atom teacher models. EDM's residual form (Eq. 4) is conditioned on a single σ and does not extend cleanly to dual-time flow maps. Importantly, this has already motivated the simplification of the EDM parametrization when distilling to flow-maps (Nie et al., 2026). In addition, for practical noise schedules, the monotone time reparameterization $\sigma : [0,1] \to [\sigma_{\min}, \sigma_{\max}]$ is non-linear, making loss computations that require time partials numerically unstable (Sabour et al., 2025b). More precisely, when computing the chain rule $\partial x/\partial t = (\partial x/\partial\sigma)\,(\partial\sigma/\partial t)$ for

the suggested EDM noise schedule σ_t (Karras et al., 2022), the end points are prone to large magnitudes of $|\partial\sigma/\partial t|$. This in turn leads to numerical instability when computing flow-map based objectives.

To circumvent the numerical instability of flow-map training using a pre-trained EDM protein model, we directly define a flow-map in the σ-noise space, allowing us to redefine all objectives and sampling directly over σ-steps—thus eliminating the problematic factor $\partial\sigma/\partial t$. This leads to our notion of σ-velocity, which represents the σ-instantaneous velocity along the PF-ODE,

$$v_\sigma(x_\sigma) \triangleq \frac{dx_\sigma}{d\sigma},$$
$$v_\sigma^{\text{EDM}}(x_\sigma) = \frac{x_\sigma - D_\sigma(x_\sigma)}{\sigma} = \frac{x_\sigma - \hat{x}_0^{\text{EDM}}}{\sigma}. \quad (6)$$

Our reparametrization of time extends, in a natural way, to now a two-noise level map that denoises $\rho \to \sigma$, for $\rho > \sigma$, using the analogous notion of *average* σ-velocity and flow-map parametrization $X_{\rho,\sigma}(x_\rho)$. This forms the basis of the sampling update at inference:

$$u_{\rho,\sigma}(x_\rho) = \frac{1}{\rho - \sigma}\int_\rho^\sigma v_{\bar\sigma}(x_{\bar\sigma})d\bar\sigma,$$
$$X_{\rho,\sigma}(x_\rho) = x_\rho - (\rho - \sigma)u_{\rho,\sigma}(x_\rho). \quad (7)$$

Denoiser Parametrization. To train our all-atom protein flow map, we follow the mean-flow objective (Geng et al., 2025a), which is also equivalent to the Eulerian objective (Boffi et al., 2025a). This requires the construction of the instantaneous σ-velocity *implied* by the average σ-velocity of Eq. 7. Explicitly, we efficiently compute this quantity using Jacobian-vector products `jvp`,

$$v_\rho(x_\rho) = u_{\rho,\sigma}(x_\rho) + (\rho - \sigma)\,\tfrac{d}{d\rho}\,u_{\rho,\sigma}(x_\rho), \quad (8)$$

$$V(x_\rho, \rho, \sigma) \triangleq u_{\rho,\sigma}(x_\rho) + (\rho - \sigma)\cdot \mathtt{sg}\Big(\mathtt{jvp}(u_{\rho,\sigma}(x_\rho),\, (x_\rho, \rho, \sigma),\, (v, 1, 0))\Big). \quad (9)$$

In Eq. 8, $\mathbf{d}/\mathbf{d}\rho$ represents the total derivative of the average σ-velocity, while `sg` is the stopgrad operator applied to $\mathbf{d}u_{\rho,\sigma}/\mathbf{d}\rho$. In the case of distilling from a pre-trained EDM model, we simply substitute the ground-truth σ-velocity with v_σ^{EDM}. This leads to a natural learning objective of computing the predicted instantaneous σ-velocity using Eq. 9 and matching it to the EDM teacher velocity in Eq. 6 (Geng et al., 2025a;b; Lu et al., 2026; Potaptchik et al., 2026b). We, however, motivate a different approach in the context of all-atom protein models. Specifically, we highlight a standard practice in AF3 diffusion models that leads to a lower variance loss estimate, which is to instead predict the target end-point $\hat{x}_{\text{tgt}}$ and find the closest rigid transformation—i.e., SE(3) rigid alignment—to the target $\hat{x}_0^{\text{EDM}}$. As a result, we parametrize our all-atom flow map as a denoiser that consumes two noise levels $D(x_\rho, \rho, \sigma)$. Specifically, we can recover a $\hat{x}_{\text{tgt}}$-prediction by computing using the predicted σ-instantaneous velocity of Eq. 9:

$$\hat{x}_{\text{tgt}}^{\text{DECAF}} := D(x_\rho, \rho, \sigma) = x_\rho - \rho \cdot V(x_\rho, \rho, \sigma) \quad (10)$$

The above equation gives rise to a *two-time* denoiser (Lu et al., 2026; Lee et al., 2026; Potaptchik et al., 2026b; Roos et al., 2026) that can be leveraged to construct an endpoint loss:

$$\mathcal{L} = \mathbb{E}\left[\tfrac{1}{\sigma^2}\min_{g\in\text{SE}(3)}\left\|\hat{x}_{\text{tgt}}^{\text{DECAF}} - \mathtt{sg}\big(\zeta(g)\circ\hat{x}_0^{\text{EDM}}\big)\right\|^2\right]. \quad (11)$$

Here we minimize over SE(3): $\zeta(g)$ is its matrix representation for the Kabsch rigid alignment, and $\hat{x}_0^{\text{EDM}}$ is the pre-trained EDM teacher's prediction. We emphasize that this formulation fails under a velocity loss, as subtracting translation would lose a degree of freedom.

The loss in Eq. 11 naturally supports both off-diagonal training and diagonal training through sampling of noise levels $(\rho, \sigma) \sim \text{Sampler}(\sigma_{\text{min}}, \sigma_{\text{max}})$. In particular, when $\rho = \sigma$, we learn to match exactly the score associated with the PF-ODE of the EDM teacher, while in all other off-diagonal cases, we learn to take $(\rho - \sigma)$ jumps along the trajectory of the PF-ODE as in Eq. 7.

Flow map sampling algorithm. To sample from DECAF we design a stochastic γ-sampler, which shares inspiration from the consistency models literature (Kim et al., 2023). We present this γ-sampler in algorithm 1, which allows us to toggle deterministic sampling ($\gamma = 0$) to more stochastic sampling for $\gamma > 0$. As we later demonstrate in our experiments §4, the added stochasticity aids overall performance at no added cost and also enjoys seamless integration with our inference strategy in §3.2.

Algorithm 1 DECAF γ-sampling

Require: FlowMap X; N steps; $\gamma \in [0, 1]$.
1: $x_{\sigma_N} \sim \mathcal{N}(0,\, \sigma_N^2 I)$
2: **for** $n = N$ **down to** 1 **do**
3: $\quad\tilde{\sigma}_{n-1} \leftarrow \sqrt{1-\gamma^2}\,\sigma_{n-1}$
4: $\quad x_{\tilde{\sigma}_{n-1}} \leftarrow X(x_{\sigma_n},\, \sigma_n,\, \tilde{\sigma}_{n-1})$ {flowmap}
5: $\quad\epsilon \sim \mathcal{N}(0, I)$
6: $\quad x_{\sigma_{n-1}} \leftarrow x_{\tilde{\sigma}_{n-1}} + \gamma\,\sigma_{n-1}\,\epsilon$ {re-noise}
7: **end for**
8: **return** x_{σ_0}

3.2. Inference-Time Search

The challenge of generating physically plausible biomolecular structures has motivated a vast literature on computationally intensive inference-time correction of pre-trained all-atom diffusion models. While DECAF is primarily designed as a few-step all-atom model that reduces inference latency for conventional deployment, it also enables a second computational advantage. Specifically, DECAF also enables substantial gains in *inference-time search* (i.e., reward alignment (Uehara et al., 2025)) for the cofolding problems we consider.

In biomolecular modeling, a common form of inference-time search defines a terminal reward $R : \mathbb{R}^d \to \mathbb{R}$ that measures physical validity, for example, penalizing steric clashes or violations of bond lengths and angles (Wohlwend et al., 2025, Section 4). The goal is to steer generation toward samples with high $R(x_0)$ while preserving structural accuracy. A central difficulty is that R is only defined on clean structures, whereas steering decisions must be made from noisy intermediate states x_σ. Consequently, we require an efficient estimate of the reward expected after denoising x_σ to a clean structure. DECAF provides such an estimate through its learned two-time flow map: given an intermediate state x_σ, we compute a look-ahead (end point) prediction over clean samples

$$\hat{x}_0 = X(x_\sigma, \sigma, 0) = x_\sigma - \sigma \cdot u_{\sigma,0}(x_\sigma), \tag{12}$$

and use $R(\hat{x}_0)$ as a proxy for the expected reward of x_σ. Similar flow map look-aheads have been used for steering in the image domain (Sabour et al., 2025a; Holderrieth et al., 2026a; Potaptchik et al., 2026a; Huang et al., 2026). We explore their analogue in all-atom biomolecular generation, which differs in two important ways. First, our base sampler is not a standard ODE or SDE integrator but γ-sampler. Second, rewards are based on physical violations whose supervisory signal is highly non-smooth and uninformative at noisy states x_σ. These considerations motivate refining generations through clean-space look-aheads rather than through gradients in noisy state space.

DECAF-SEARCH. We introduce an inference-time search algorithm DECAF-SEARCH for all-atom flow maps (see algorithm 2). DECAF-SEARCH adapts standard inference-time steering methods, including Feynman–Kac (FK) (Singhal et al., 2025; Skreta et al., 2025) and diffusion-based MCTS (Jain et al., 2025), to flow maps and the γ-sampling setting. Starting from a population of particles, we repeatedly denoise each particle from x_σ to a clean look-ahead $\hat{x}_0$ (possibly over several flow map steps), evaluate its reward, and optionally improve it by clean-space gradient ascent

$$\hat{x}_0 \leftarrow \hat{x}_0 + \beta \nabla_{\hat{x}_0} R(\hat{x}_0). \tag{13}$$

Then, we renoise the improved structure to a noisy state $x_{\tilde{\sigma}}$ before continuing to the next γ-sampling iteration. Inspired by Holderrieth et al. (2026a, Proposition 5.1.), we also consider a variant where the gradient is an average of several Monte Carlo samples (MC-GRAD in §A). Across particles, compute is preferentially allocated to promising branches using either SMC resampling or an upper-confidence-bound criterion. With finite-temperature resampling, DECAF-SEARCH resembles FK steering with UCB/UCT selection and recovers a simple MCTS-style variant (Jain et al., 2025). DECAF-SEARCH also extends the FK-style steering used in Boltz-1x (Wohlwend et al., 2025): intermediate clean predictions are obtained using the learned DECAF flow map, which can provide more accurate or more efficient look-aheads than a single-step denoiser.

4. Experiments

We investigate the application of DECAF for the task of cofolding protein-ligand interactions. In particular, we distill a pre-trained Boltz-1 model using Eq. 11 and sample using DECAF-SEARCH (algorithm 2), in contrast to Boltz-1x the physical potential variant (Wohlwend et al., 2025). For fair comparison, DECAF shares architecture, training data, and pretraining with Boltz-1 (see §B for architecture, experimental setup, and hyperparameters). We include additional ablations in §C.

Benchmarks. The primary benchmark we use for analysis is Runs N' Poses (RnP) (Škrinjar et al., 2025). We limit the experiments to a stricter set of 702 structures with a cutoff date of `2023-06-01`. We also conduct additional analysis on PoseBusters (Buttenschoen et al., 2024) with cutoff date of `2021-10-01`, yielding 282 structures that can be handled by Boltz-1 on a single A100-80GB GPU.

Metrics. Following the standard practice, we report the following metrics: (i) *RMSD < 2 Å* defined as the percentage of the test structures for which root mean square distance between ground-truth and generated ligand poses is under 2 Å; (ii) *PB-Valid* defined as the percentage of test structures that are physically valid according to the PoseBusters library; (iii) *lDDT-PLI* defined as local distance difference test (Mariani et al., 2013) on the short-range protein–ligand contacts within a 6 Å protein-ligand pocket, where side-chain atoms typically outnumber backbone atoms. We also define *Success Rate (%)* as a percentage of the test structures that satisfy RMSD < 2 Å and PB-Valid criteria.

Our experiments seek to answer three key questions that test the empirical caliber of DECAF:

(Q1.) Low NFE regime. Does DECAF outperform Boltz-1 with a limited inference budget (§4.1)?

(Q2.) Analysis of compute-optimal frontier. What is the Pareto frontier of DECAF-SEARCH against

Table 1. **DECAF vs Boltz-1x on Runs N' Poses benchmark.** The best recipe at or below the NFE budget is reported per method averaged over 5 poses. **Bold** marks the winner between Boltz-1x and DECAF; stars indicate paired Wilcoxon signed-rank significance vs the other model (two-sided): $^{*}p<0.05$, $^{**}p<0.01$, $^{***}p<0.001$. † Indicates numbers taken from (Genesis Research, 2025)

Method	PB Valid↑	Success Rate↑	lDDT-PLI↑
≤ 10 NFEs (2 steps)			
Boltz-1x (default)	0.0	0.0	0.0
Boltz-1x (tuned)	21.8	17.2	45.8
DECAF-SEARCH	***24.0**	****19.6**	**46.4**
≤ 20 NFEs (5 steps)			
Boltz-1x (tuned)	75.9	50.3	59.8
DECAF-SEARCH	*****83.5**	*****57.4**	*****66.5**
≤ 25 NFEs (6 steps)			
Boltz-1x (tuned)	73.4	51.9	61.8
DECAF-SEARCH	*****80.1**	*****57.0**	*****67.8**
≤ 40 NFEs (10 steps)			
Boltz-1x (tuned)	86.6	55.4	64.2
DECAF-SEARCH	*****91.6**	*****61.8**	*****67.2**
≤ 50 NFEs (12 steps)			
Boltz-1x (tuned)	86.3	56.5	63.7
DECAF-SEARCH	*****92.7**	*****62.6**	****67.1**
≤ 160 NFEs			
Boltz-1x (default, 40 steps)	88.5	58.7	67.6
DECAF-SEARCH (MC-GRAD, 15 steps)	*****92.1**	*****63.7**	*****69.2**
800 NFEs (Full budget)			
Boltz-1x (ref., 200 steps)	**95.9**	64.9	**69.9**
DECAF-SEARCH (MCTS, 10 steps)	95.5	**65.0**	69.5
Frontier cofolding models†			
AlphaFold 3	73.9	60.7	80.9
Pearl	96.1	72.1	81.9

Boltz-1x for inference time scaling across *any* inference compute budget (§4.2)?

(Q3.) Performance analysis. What are the relative quantitative and qualitative differences in generation quality at the sample-level of DECAF in comparison to its teacher Boltz-1 (§4.3)?

4.1. Performance at low NFEs (Q1.)

We evaluate DECAF and Boltz-1 on Runs N' Poses for various NFE inference regimes. We report our main results in table 1 (average over 5 poses) on Runs N' Poses structures drawn entirely from PDB depositions released *after* the `2023-06` cutoff date for training. Specifically, we sample DECAF-SEARCH with SMC resampling over 4 particles—bearing similarity to FK steering (Singhal et al., 2025)—at compute budgets of 10, 20, 25, 40, and 50 NFEs. We also include a high-budget (800 NFEs) comparison between Boltz-1x and DECAF-SEARCH with a variation that is closest to MCTS. As table 1 demonstrates, Boltz-1x fails catastrophically at generating plausible structures at low steps regime under the default sampling configuration. As an orthogonal contribution, we remedy this by tuning the step scale in the default Boltz-1x to $\eta = 1$ when ≤ 15 diffusion steps, which restores stable SDE sampling in the few-step regime. We refer to this setting as Boltz-1x-tuned.

At *every* low compute budget in table 1, DECAF outperforms Boltz-1x-tuned on *every* metric. The improvement

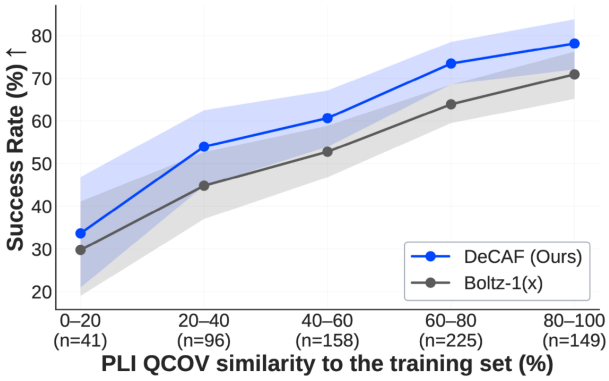

Figure 2. Success rate vs. training-set similarity on the RnP benchmark at 40 NFE.

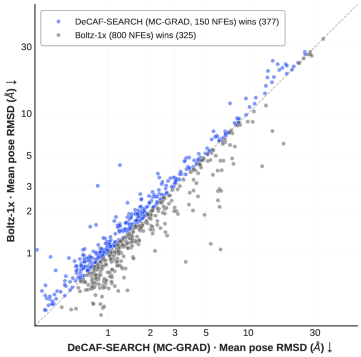

Figure 3. Mean RMSD per structure for DECAF-SEARCH (150 NFEs) (x) vs. Boltz-1x (800 NFEs) (y) on RnP.

is significant in the 20–160 NFE range (paired Wilcoxon signed-rank, $p < 0.001$ on nearly all metrics). At 50 NFEs, DECAF is also competitive with frontier cofolding models that use ≥ 800 NFEs—surpassing even AlphaFold 3 on the Success Rate and PB-Valid metrics. At a matched budget of 800 NFEs, DECAF is on par with Boltz-1x on all three metrics.

Generalization. An important criterion for cofolding models is their ability to handle difficult targets, as it is a proxy for model generalization beyond the training set. In fig. 2, we plot the success rate against the pocket-similarity metric *PLI Q-Coverage* (shaded regions show 95% bootstrap confidence intervals (1000 resamples per bin). We observe that DECAF's margin over Boltz-1x holds across every quartile, including the most out-of-distribution bin (lowest PLI Q-Coverage), confirming that the performance gain is not driven by structures that are close to the training set.

Lastly, in fig. 3 we perform a fine-grained analysis of performance at the sample-level. In particular, we conduct a comparison between DECAF-SEARCH (MC-GRAD) at 150 NFEs and Boltz-1x (800 NFEs) for each of the 702 structures in Runs N' Poses. Each point gives one target's

Table 2. **`Post-2023` Runs N' Poses comparison** (n=196). AlphaFold3, Chai-1, and Boltz-1 were trained with `2021-09` cutoff date. Boltz-2 and Pearl models are trained on `2023-06` and `2023-12` cutoff dates, respectively. Bolding is reserved for only the 40 NFE comparisons and not the frontier models at full budget.

	best@1			best@5		
Method	PB-valid	Success Rate	lDDT-PLI	PB-valid	Success Rate	lDDT-PLI
Full budget						
AF3 †	66.3	53.5	76.1	86.2	69.9	79.3
Chai-1†	71.2	49.5	69.9	90.3	63.3	72.9
Boltz-1x†	93.7	62.2	71.3	95.9	66.3	73.2
Boltz-2† (`2023-06 cutoff`)	67.9	50.1	74.3	79.1	58.2	77.3
Pearl-2026.1.dev (`2023-12 cutoff`)	96.7	67.8	73.0	98.5	77.0	78.9
40 *NFEs (10 steps)*						
Pearl-2026.1.dev (`2023-12 cutoff`)	83.9	52.0	60.6	97.4	75.0	**78.7**
DECAF-Boltz	89.5	55.5	62.3	98.0	66.8	67.7
DECAF-Pearl	**95.1**	**64.8**	**71.1**	**99.5**	**76.0**	77.2

Table 3. **Parameterization ablation.** The x_0-aligned parameterization implements eq. (10); the velocity parameterization predicts $v_\rho(x_\rho)$; consistency distillation (CD) follows (Song et al., 2023). Numbers on PoseBusters, 30 NFEs.

Parametrization	PB-Valid ↑	RMSD<2Å ↑
x_0-aligned	**90.4**	**75.6**
velocity	9.2	49.1
Cons. Distillation	0.0	49.5

mean pose RMSD under DECAF (x) and Boltz-1x (y); points above the $y = x$ diagonal indicate DECAF wins. DECAF matches Boltz-1x on the per-target distribution of mean RMSD at 5.3× less inference compute. The advantage holds in the difficult tail: among the targets where at least one method exceeds 10,Å, DECAF attains the lower RMSD on 71% of them.

Distilling Pearl (Genesis Research, 2025). We apply DECAF to Pearl-2026.1.dev (a Pearl checkpoint with 2023-12 cutoff), yielding DECAF-Pearl. Table 2 evaluates on the `post-2023` Runs N' Poses slice (n=196). At 40 NFEs, DECAF-Pearl achieves 76.0% best@5 success rate, matching Pearl-2026.1.dev's full-budget score (77.0%) at 5× fewer NFEs, and outperforming the 10-step Pearl baseline (75.0%) and DECAF-Boltz (66.8%).

4.2. Analysis of the compute-optimal frontier (Q2.)

We investigate the compute-optimal Pareto frontier on PoseBusters (∼300 structures, cutoff 2021-10-01), where the dense NFE sweep is tractable. Figure 4 compares DECAF-SEARCH and Boltz-1x across PB Valid, RMSD< 2 Å, and Success Rate (best@5 per NFE threshold). DECAF-SEARCH achieves a 20× cost reduction over the 600-NFE Boltz-1x configuration, with its Pareto frontier dominating Boltz-1x at every NFE budget. The optimal recipe shifts with compute: FK-steering at NFE≤ 30, MC-GRAD at 100–250 NFEs, and MCTS at NFE≥ 142.

Fine grained analysis. As we increase the NFE based on the hyperparams in DECAF-SEARCH, however, we note that the head-to-head with Boltz-1x favors different complexes, i.e., the successful complexes at high NFE are not a superset of the successful complexes at low NFE (table 7). This suggests that a more complex interplay between sampling accuracy and reward guidance within DECAF. We highlight representative samples from DECAF-SEARCH (FK) at low NFE, which have higher accuracy than any pose sampled from Boltz-1x at any NFE (fig. 5, additional samples in fig. 9).

4.3. Performance Analysis (Q3.)

Posebusters validity. We qualitatively study the pose quality at various NFE budgets. In fig. 5 we visually depict the samples and observe that, in comparison to Boltz-1x, the MCTS version of DECAF-SEARCH (600 NFEs) can improve pose accuracy. Meanwhile, MC-GRAD (150 NFEs) reduces steric clashes in comparison to Boltz-1x, highlighting improved physical reward alignment.

DECAF-SEARCH methods benefit from having both generally high pose quality (i.e., reward optimization) and more accurate ligand placement. While few-NFE inference with DECAF already yields better failure rates in relation to Boltz-1x (c.f. table 8) we see further refinement in pose quality when scaling NFEs further with the MC-GRAD and MCTS variants. We additionally note that pose quality on PoseBusters is greatly determined by reward design, and leads to common failure modes across DECAF and Boltz-1x that share the same potentials. For instance, we observe that sp2-hybridized bonds are often not planar. In addition, some PoseBusters quality checks have stricter tolerances than are chemically accurate, such as flagging metal ion coordination as a "clash" based solely on the neutral atoms' van der Waals radii (fig. 7). We find majority of these failures can be mitigated through inference search.

Generalization on chemically-relevant targets. We stratify the Runs N' Poses benchmark to evaluate our models on challenging scenarios most relevant to drug discovery tasks. We consider several slices which probe these settings: drug-like ligands only, ligands interacting with ions and cofactors, and ligands at protein-protein interfaces. In these subsets, DECAF-SEARCH outperforms Boltz-1x to a statistically significant degree (table 9). We curate representative poses in fig. 8.

Parametrization. We ablate DECAF's denoiser-aligned parametrization (eq. (10)) against a velocity-predictor student (Geng et al., 2025b) that directly regresses $v_\rho(x_\rho)$ under the same JVP schedule and data budget. We also experimented with consistency distillation parametrization (Song et al., 2023). Table 3 confirms that only our chosen x_0-aligned parametrization can sample valid poses. The denoiser parametrization reuses EDM preconditioning and enables Kabsch alignment on $\hat{x}_0$—which we find critical for stable training.

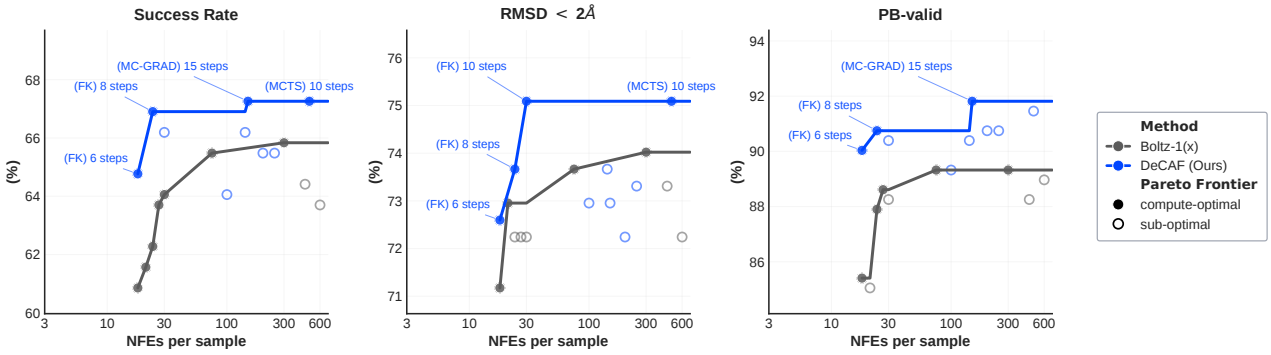

Figure 4. **DECAF is cost-effective as we increase compute.** We compare DECAF-SEARCH and Boltz-1x as we increase NFEs budget. The solid lines are the per-NFE compute optimal frontier for the two methods.

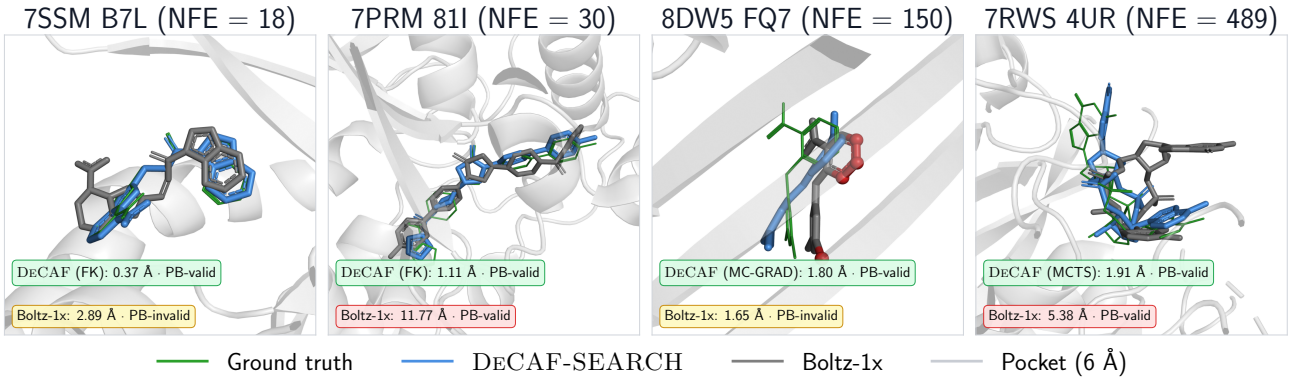

Figure 5. Each panel overlays the **ground-truth** crystal ligand against **DECAF-SEARCH** and **Boltz-1x** samples at the specified NFE. The protein pocket is shown as a light-gray cartoon, and predicted ligand atoms that clash with the protein in red. At the bottom, we report the RMSD and posebusters validity of the pose.

5. Related Work

Protein generation. Early generative models focused on backbone design (Yim et al., 2023; Watson et al., 2023; Bose et al., 2023; Huguet et al., 2024; Geffner et al., 2025b;a). AlphaFold 3 (Abramson et al., 2024) established EDM-style denoising over all-atom coordinates as the dominant paradigm for cofolding, and several systems have followed: Boltz-1(x) (Wohlwend et al., 2025), ProteniX (Team et al., 2026), Chai-1 (Boitreaud et al., 2024), Pearl (Genesis Research, 2025), Complexa (Didi et al., 2026), and DISCO (Rector-Brooks et al., 2026). All share a costly $\mathcal{O}(200)$-NFE inference bottleneck.

Few-step generative models. Trajectory compression methods include consistency models (Song et al., 2023; Song & Dhariwal, 2024), Consistency Trajectory Models (Kim et al., 2024), shortcut models (Frans et al., 2025), and MeanFlow (Geng et al., 2025a; Boffi et al., 2025a). Despite this literature, distillation of all-atom cofolding models remains underexplored, with only DCFold as concurrent closed-source work (Zhang et al., 2026).

Inference-time steering. Inference-time techniques include classifier guidance (Ho & Salimans, 2021), Universal Guidance (Bansal et al., 2024), and particle-based SMC (Del Moral et al., 2006) methods such as Feynman–Kac steering (Singhal et al., 2025; Skreta et al., 2025) and diffusion MCTS (Jain et al., 2025). In the protein setting, AF3-family models routinely apply physics-informed potentials (Wohlwend et al., 2025) whose cost grows linearly in denoising steps and particle count.

6. Conclusion

We introduced DECAF, a flow map distilling a pretrained all-atom cofolding model into a few-step generator via σ-space reparameterization and a denoiser parametrization that preserves $\mathrm{SE}(3)$ rigid alignment. On top of DECAF, we built DECAF-SEARCH, an inference-time search framework exploiting flow-map lookahead for reward alignment. DECAF-SEARCH matches the 600-NFE Boltz-1 teacher with a $20\times$ reduction in NFEs on PoseBusters and improves over Boltz-1x on Runs N' Poses at every low-NFE setting. Limitations and next steps: our reward signal is inherited from Boltz-1x; the σ-space formulation should extend to nucleic acids and larger assemblies; and the non-monotone NFE–success relationship points to adaptive search and joint reward training as natural future directions.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630 (8016):493–500, 2024.

Bansal, A., Chu, H.-M., Schwarzschild, A., Sengupta, S., Goldblum, M., Geiping, J., and Goldstein, T. Universal guidance for diffusion models. In *ICLR*, 2024.

Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, January 2012. ISSN 1755-4349. doi: 10.1038/nchem.1243. URL http://dx.doi.org/10.1038/nchem.1243.

Boffi, N. M., Albergo, M. S., and Vanden-Eijnden, E. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025a. ISSN 2835-8856.

Boffi, N. M., Albergo, M. S., and Vanden-Eijnden, E. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Boitreaud, J., Dent, J., McPartlon, M., Meier, J., Reis, V., Rogozhnikov, A., and Wu, K. Chai-1: Decoding the molecular interactions of life. *bioRxiv*, 2024. doi: 10.1101/2024.10.10.615955. URL https://www.biorxiv.org/content/10.1101/2024.10.10.615955v1.

Bose, A. J., Akhound-Sadegh, T., Huguet, G., Fatras, K., Rector-Brooks, J., Liu, C.-H., Nica, A. C., Korablyov, M., Bronstein, M., and Tong, A. Se (3)-stochastic flow matching for protein backbone generation. *arXiv preprint arXiv:2310.02391*, 2023.

Buttenschoen, M., Morris, G. M., and Deane, C. M. PoseBusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024. doi: 10.1039/D3SC04185A.

Cao, L., Goreshnik, I., Coventry, B., Case, J. B., Miller, L., Kozodoy, L., Chen, R. E., Carter, L., Walls, A. C., Park, Y.-J., Strauch, E.-M., Stewart, L., Diamond, M. S., Veesler, D., and Baker, D. De novo design of picomolar sars-cov-2 miniprotein inhibitors. *Science*, 370(6515): 426–431, 2020.

Chevalier, A., Silva, D.-A., Rocklin, G. J., Hicks, D. R., Vergara, R., Murapa, P., Bernard, S. M., Zhang, L., Lam, K.-H., Yao, G., et al. Massively parallel de novo protein design for targeted therapeutics. *Nature*, 550(7674):74–79, 2017.

Del Moral, P., Doucet, A., and Jasra, A. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B*, 68(3):411–436, 2006.

Didi, K., Zhang, Z., Zhou, G., Reidenbach, D., Cao, Z., Cha, S., Geffner, T., Dallago, C., Tang, J., Bronstein, M. M., et al. Scaling atomistic protein binder design with generative pretraining and test-time compute. *arXiv preprint arXiv:2603.27950*, 2026.

Ebrahimi, S. B. and Samanta, D. Engineering protein-based therapeutics through structural and chemical design. *Nature Communications*, 14(1):2411, 2023.

Frans, K., Hafner, D., Levine, S., and Abbeel, P. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Gainza, P., Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M. M., and Correia, B. E. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2): 184–192, 2020.

Geffner, T., Didi, K., Cao, Z., Reidenbach, D., Zhang, Z., Dallago, C., Kucukbenli, E., Kreis, K., and Vahdat, A. La-proteina: Atomistic protein generation via partially latent flow matching. *arXiv preprint arXiv:2507.09466*, 2025a.

Geffner, T., Didi, K., Zhang, Z., Reidenbach, D., Cao, Z., Yim, J., Geiger, M., Dallago, C., Kucukbenli, E., Vahdat, A., et al. Proteina: Scaling flow-based protein structure generative models. *arXiv preprint arXiv:2503.00710*, 2025b.

Genesis Research, T. Pearl: A foundation model for placing every atom in the right location. *arXiv preprint arXiv:2510.24670*, 2025.

Geng, Z., Deng, M., Bai, X., Kolter, J. Z., and He, K. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Geng, Z., Lu, Y., Wu, Z., Shechtman, E., Kolter, J. Z., and He, K. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025b.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Holderrieth, P., Chen, D., Eyring, L., Shah, I., Anantharaman, G., He, Y., Akata, Z., Jaakkola, T., Boffi, N. M., and Simchowitz, M. Diamond maps: Efficient reward

alignment via stochastic flow maps. *arXiv preprint arXiv:2602.05993*, 2026a.

Holderrieth, P., Singer, U., Jaakkola, T., Chen, R. T. Q., Lipman, Y., and Karrer, B. GLASS flows: Efficient inference for reward alignment of flow and diffusion models. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026b.

Huang, J. Y., Lin, J., Shah, S., Nair, K., and Boffi, N. M. How to guide your flow: Few-step alignment via flow map reward guidance. *arXiv preprint arXiv:2604.27147*, 2026.

Huguet, G., Vuckovic, J., Fatras, K., Thibodeau-Laufer, E., Lemos, P., Islam, R., Liu, C.-H., Rector-Brooks, J., Akhound-Sadegh, T., Bronstein, M., et al. Sequence-augmented se (3)-flow matching for conditional protein generation. *Advances in neural information processing systems*, 37:33007–33036, 2024.

Jain, V., Sareen, K., Pedramfar, M., and Ravanbakhsh, S. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. *arXiv preprint arXiv:2506.20701*, 2025.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Kim, D., Lai, C.-H., Liao, W.-H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y., and Ermon, S. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

Kim, D., Lai, C.-H., Liao, W.-H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y., and Ermon, S. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

Lee, C., Yoo, J., Agarwal, M., Shah, S., Huang, J., Raghunathan, A., Hong, S., Boffi, N. M., and Kim, J. Flow map language models: One-step language modeling via continuous denoising. *arXiv preprint arXiv:2602.16813*, 2026.

Lu, Y., Lu, S., Sun, Q., Zhao, H., Jiang, Z., Wang, X., Li, T., Geng, Z., and He, K. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158*, 2026.

Mariani, V., Biasini, M., Barbato, A., and Schwede, T. lddt: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

Murray, K. A., Hu, C. J., Griner, S. L., Pan, H., Bowler, J. T., Abskharon, R., Rosenberg, G. M., Cheng, X., Seidler, P. M., and Eisenberg, D. S. De novo designed protein inhibitors of amyloid aggregation and seeding. *Proceedings of the National Academy of Sciences*, 119 (34):e2206240119, 2022.

Nie, W., Berner, J., Liu, C., and Vahdat, A. Nvidia fastgen: Fast generation from diffusion models, 2026. URL https://github.com/NVlabs/FastGen.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Potaptchik, P., Saravanan, A., Mammadov, A., Prat, A., Albergo, M. S., and Teh, Y. W. Meta flow maps enable scalable reward alignment. *arXiv preprint arXiv:2601.14430*, 2026a.

Potaptchik, P., Yim, J., Saravanan, A., Holderrieth, P., Vanden-Eijnden, E., and Albergo, M. S. Discrete flow maps. *arXiv preprint arXiv:2604.09784*, 2026b.

Rector-Brooks, J., Lambert, T., Skreta, M., Roth, D., Long, Y., Li, Z.-Q., Zhang, X., Cretu, M., Li, F.-Z., Ganapathy, T., et al. General multimodal protein design enables dna-encoding of chemistry. *arXiv preprint arXiv:2604.05181*, 2026.

Roos, D., Davis, O., Eijkelboom, F., Bronstein, M., Welling, M., Ceylan, İ. İ., Ambrogioni, L., and van de Meent, J.-W. Categorical flow maps. *arXiv preprint arXiv:2602.12233*, 2026.

Sabour, A., Albergo, M. S., Domingo-Enrich, C., Boffi, N. M., Fidler, S., Kreis, K., and Vanden-Eijnden, E. Test-time scaling of diffusions with flow maps. *arXiv preprint arXiv:2511.22688*, 2025a.

Sabour, A., Fidler, S., and Kreis, K. Align your flow: Scaling continuous-time flow map distillation, 2025b. URL https://arxiv.org/abs/2506.14603.

Silva, D.-A., Yu, S., Ulge, U. Y., Spangler, J. B., Jude, K. M., Labão-Almeida, C., Ali, L. R., Quijano-Rubio, A., Ruterbusch, M., Leung, I., et al. De novo design of potent and selective mimics of il-2 and il-15. *Nature*, 565(7738): 186–191, 2019.

Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K., and Ranganath, R. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Skreta, M., Akhound-Sadegh, T., Ohanesian, V., Bondesan, R., Aspuru-Guzik, A., Doucet, A., Brekelmans, R., Tong,

A., and Neklyudov, K. Feynman–Kac correctors in diffusion: Annealing, guidance, and product of experts. In *International Conference on Machine Learning*, 2025.

Škrinjar, P., Eberhardt, J., Durairaj, J., and Schwede, T. Have protein-ligand co-folding methods moved beyond memorisation? *BioRxiv*, pp. 2025–02, 2025.

Song, Y. and Dhariwal, P. Improved techniques for training consistency models. In *International Conference on Learning Representations (ICLR)*, 2024.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Strauch, E.-M., Bernard, S. M., La, D., Bohn, A. J., Lee, P. S., Anderson, C. E., Nieusma, T., Holstein, C. A., Garcia, N. K., Hooper, K. A., et al. Computational design of trimeric influenza-neutralizing proteins targeting the hemagglutinin receptor binding site. *Nature biotechnology*, 35(7):667–671, 2017.

Team, P., Zhang, Y., Gong, C., Zhang, H., Ma, W., Liu, Z., Chen, X., Guan, J., Wang, L., Yang, Y., et al. Protenix-v1: Toward high-accuracy open-source biomolecular structure prediction. *bioRxiv*, pp. 2026–02, 2026.

Tweedie, M. C. Statistical properties of inverse gaussian distributions. ii. *The Annals of Mathematical Statistics*, pp. 696–705, 1957.

Uehara, M., Zhao, Y., Wang, C., Li, X., Regev, A., Levine, S., and Biancalani, T. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976): 1089–1100, 2023.

Wohlwend, J., Corso, G., Passaro, S., Getz, N., Reveiz, M., Leidal, K., Swiderski, W., Atkinson, L., Portnoi, T., Chinn, I., et al. Boltz-1 democratizing biomolecular interaction modeling. *BioRxiv*, pp. 2024–11, 2025.

Yim, J., Campbell, A., Foong, A. Y., Gastegger, M., Jiménez-Luna, J., Lewis, S., Satorras, V. G., Veeling, B. S., Barzilay, R., Jaakkola, T., et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

Zhang, Z., Feng, Y., Song, Y., Qiu, K., Zhou, H., and Ma, W.-Y. DCFold: Efficient protein structure generation with single forward pass. In *International Conference on Learning Representations (ICLR)*, 2026.

Appendix

A. DECAF-SEARCH

Algorithm 2 Inference-time search with DECAF-SEARCH

Require: FlowMap X; noise schedule $\sigma_N > \sigma_{N-1} > \cdots > \sigma_0 = 0$; particles P; resampling interval L; steering weight λ; number of search iterations S; potential R; $\gamma \in [0, 1]$; rollout horizon K (with $K = 1$ for FK).

Ensure: Sample $\hat{x}_0$.

1: Draw $\{x_{\sigma_N}^{(p)}\}_{p=1}^{P} \sim \mathcal{N}(0, \sigma_N^2 I)$ {initialize P particles}
2: $\mathcal{X} \leftarrow \{x_{\sigma_N}^{(p)}\}_{p=1}^{P}, \quad n \leftarrow 0$ {global search tree, current noise index}
3: **for** $s = 0, 1, \ldots, S-1$ **do**
4: $\quad \mathcal{T} \leftarrow \emptyset$ {rollout trajectories}
5: $\quad n_{\text{end}} \leftarrow \min(N,\ n + K)$
6: $\quad$ **for** $m = n, \ldots, n_{\text{end}} - 1$ **do** {denoise rollout (all particles in parallel)}
7: $\quad\quad \hat{x}_0^{(p)} \leftarrow X(x_{\sigma_m}^{(p)}, \sigma_m, 0), \quad \forall p$
8: $\quad\quad$ *Optional:* gradient steps on $\hat{x}_0^{(p)}$ w.r.t. R {see eq. (13)}
9: $\quad\quad \tilde{\sigma}_{m+1} \leftarrow \sqrt{1-\gamma^2}\,\sigma_{m+1}$ {γ renoise}
10: $\quad\quad \tilde{x}^{(p)} \leftarrow \hat{x}_0^{(p)} + \tilde{\sigma}_{m+1}\,\dfrac{x_{\sigma_m}^{(p)} - \hat{x}_0^{(p)}}{\sigma_m}, \quad \forall p$
11: $\quad\quad \epsilon^{(p)} \sim \mathcal{N}(0, I), \quad \forall p$
12: $\quad\quad x_{\sigma_{m+1}}^{(p)} \leftarrow \tilde{x}^{(p)} + \gamma\,\sigma_{m+1}\,\epsilon^{(p)}, \quad \forall p$
13: $\quad\quad \mathcal{T} \leftarrow \mathcal{T} \cup \left\{(x_{\sigma_{m+1}}^{(p)},\ m+1)\right\}_{p=1}^{P}$
14: $\quad$ **end for**
15: $\quad$ Compute particle scores $R^{(p)}\big(\hat{x}_0^{(p)}\big), \forall p$
16: $\quad$ **if** SEARCH = MCTS **then**
17: $\quad\quad \mathcal{X} \leftarrow \mathcal{X} \cup \mathcal{T}$
18: $\quad\quad$ Update weights of all $(x, m) \in \mathcal{T}$ {backup}
19: $\quad\quad$ Draw $(x_{\sigma_i}^{(p)}, i) \sim \mathcal{X}$ according to UCT, $\forall p$
20: $\quad$ **else if** SEARCH = FK **then** {$K = 1$, so $\mathcal{T}$ holds one entry per particle}
21: $\quad\quad (x_{\sigma_i}^{(p)}, i) \leftarrow$ the entry of $\mathcal{T}$ for particle p, $\forall p$
22: $\quad\quad$ **if** $s \bmod L = 0$ **then**
23: $\quad\quad\quad$ Resample $\{x_{\sigma_i}^{(p)}\}_{p=1}^{P}$ with weights $\propto R^{(p)}$
24: $\quad\quad$ **end if**
25: $\quad$ **end if**
26: $\quad n \leftarrow i + 1$ {advance to next noise level}
27: **end for**
28: **return** $x_0^{(p^\star)}$ with $p^\star = \arg\max_p \big\{R(x_0^{(p)}) : (x_0^{(p)}, 0) \in \mathcal{X}\big\}$

DECAF-SEARCH (MC-Grad). As a slight extension of algorithm 2, we realized that the gradient estimate in $\nabla_{x_{\sigma_0}} R(x_{\sigma_0})$ can be a noisy estimate of the optimal guidance direction. We hypothesize that a Monte Carlo average of several gradients might be more favorable, as observed previously (Holderrieth et al., 2026a). In order to do so, we set

$$x_0 \leftarrow x_0 + \frac{\beta}{L} \sum_{l=1}^{L} w_l \nabla_{x_0^l} R(x_0^l) \tag{14}$$

where weights w_l are the softmax of importance logits derived from the local reward and the renoise prior, $\beta = (\sigma/\sigma_{\text{data}}^2)$, and x_0^l is obtained by renoising x_σ back to $x_{\tilde{\sigma}}$ and then performing the rollouts step in line 7 (i.e. this induces one extra loop that we omit here for readability).

Table 4. Shared inference-parameter settings (constant across NFE budgets and methods).

Parameter	Value
EDM noise schedule	Karras et al. (ρ=7)
$\sigma_{\min}, \sigma_{\max}, \sigma_{\text{data}}$	0.0004, 160, 16
Stochastic-step noise scale (`noise_scale`)	0.901
Per-step gradient updates	20
MCTS selection rule	UCT
UCT exploration constant (c)	1.0
Progressive widening (k, α)	2.0, 0.5

B. Experimental Setup

B.1. Architecture and Training

We adopt Boltz-1 original architecture and codebase (Wohlwend et al., 2025) except for the implementation of dual-time conditioning. The denoiser parametrization $u(x_\rho, \rho, \sigma)$ requires fusing two noise levels into the score network's conditioning stream. We adopt a dual-time conditioning module from FastGen (Nie et al., 2026) that departs from the single-time conditioning of the EDM teacher. We train DECAF as a distillation head with our $x0$-aligned loss (eq. (11)), while the trunk and EDM modules are frozen and initialized from the Boltz-1 open-source checkpoint. Training uses RCSB PDB structures released before 2021-09-30, filtered to resolution $\leq$ 9.0 Å. We train for 100 epochs with 51,200 samples per epoch on 64 H200 GPUs, using a per-GPU batch size of 2 (effective batch size 128) and diffusion multiplicity 16. The optimizer is Muon with momentum 0.95 and Nesterov updates, with an AdamW fallback ($\beta_1 = 0.9$, $\beta_2 = 0.95$) for 1D parameters at learning rate ratio 0.1; weight decay is 0.01 throughout. The learning rate follows an AlphaFold 3-style schedule with linear warmup over 1,000 steps to a peak of $1.8e^{-3}$. The σ-weighting follows the Karras EDM schedule ($\sigma_{\min} = 4e^{-4}$, $\sigma_{\max} = 160$, $\rho = 7$, $\sigma_{\text{data}} = 16$), with 10% diagonal samples ($\sigma_r = \sigma_t$) and 90% off-diagonal. We further re-scale loss $\mathcal{L}$ in eq. (11) by $\frac{1}{\sqrt{(\mathcal{L}+1e^{-6})}}$

Table 5. Per-method settings on Runs N' Poses and PoseBusters. `Steps`: number of sampling steps. `Params`: per-method hyperparameters that scale the per-pose NFE. All other sampler hyperparameters are fixed across NFE budgets.

Method	Steps	Params	NFE per pose
Runs N' Poses (4 FK particles)			
Boltz-1x / DECAF (FK)	2	4 particles	8
Boltz-1x / DECAF (FK)	5	4 particles	20
Boltz-1x / DECAF (FK)	6	4 particles	24
Boltz-1x / DECAF (FK)	10	4 particles	40
Boltz-1x / DECAF (FK)	12	4 particles	48
DECAF-SEARCH (MC-GRAD)	15	10 samples	150
Boltz-1x	40	4 particles	160
Boltz-1x *(ref.)*	200	4 particles	800
DECAF-SEARCH (MCTS)	10	4 children, 50 simulations	800
PoseBusters (3 FK particles)			
Boltz-1x	$[6, \ldots, 200]$	3 particles	$[18, \ldots, 600]$
Boltz-1x *(ref.)*	200	3 particles	600
DECAF (FK)	6	3 particles	18
DECAF (FK)	8	3 particles	24
DECAF (FK)	10	3 particles	30
DECAF (MC-GRAD)	10	10 samples	100
DECAF (MCTS)	10	5 sims, 4 children	142
DECAF (MC-GRAD)	15	10 samples	150
DECAF (MC-GRAD)	20	10 samples	200
DECAF (MC-GRAD)	25	10 samples	250
DECAF (MCTS)	10	15 sims, 4 children	489

B.2. Hyperparameters

We report the inference hyperparameters in tables 4 and 5. Values in table 4 are constant across all NFE budgets and across both methods unless otherwise noted. Values in table 5 are the only knobs that change with the compute budget.

The key sampler tuning we apply to Boltz-1x in the few-step regime is setting the step scale $\sigma_{\text{scale}} = 1.0$ (vs. the default 1.638 used at 200 steps). With the default 1.638, the SDE diverges under aggressive step-size schedules, producing 0.0% on every metric at 2 steps (table 1, "Boltz-1x (default)"). Setting noise scale $\eta = 1.0$ recovers stable sampling and is what we report as "Boltz-1x (tuned)" throughout.

C. Additional Ablations

C.1. Sampler ablations

Stochasticity γ sweep. Table 6 sweeps the CTM stochasticity parameter $\gamma \in \{0.3, 0.5, 0.7, 1.0\}$ for algorithm 2 on PoseBusters, holding the rest of the sampler fixed (10 denoising steps, DECAF-SEARCH (FK), $P = 3$ FK particles). We adopt $\gamma = 0.5$, which we hypothesize balances the determinism of $\gamma \to 0$ against the variance of sampling at γ=1. Empirically, Success Rate peaks at γ=0.5 (65.9%), with a 4.7-pt spread between the best (γ=0.5) and worst (γ=1.0) settings, while PB-validity is essentially saturated across all γ ($\Delta \leq 1.5$ pp).

Table 6. Sweep of γ for Alg. 2 on PoseBusters. 10 sampling steps, DECAF-SEARCH (FK), $P = 3$.

γ	RMSD<2 ↑	PB Valid ↑	Success Rate ↑
0.3	75.3	89.2	65.2
0.5	**75.6**	90.3	**65.9**
0.7	72.8	90.3	63.8
1.0	70.3	**90.7**	61.3

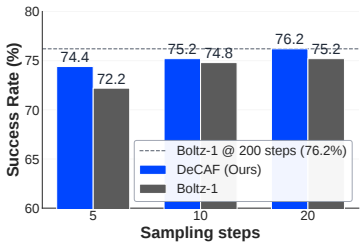

Figure 6. RMSD$<$ 2 Å best@5 on PoseBusters. DECAF (FlowMap sampler Alg. 1, γ=0) vs Boltz-1 (ODE) at matched sampling steps; dashed line marks Boltz-1 at 200 steps.

DECAF vs Boltz-1 without inference scaling. Figure 6 isolates the sampler axis: both methods run unguided ODE-style integration so the comparison reflects the underlying sampler quality in isolation rather than any inference-time augmentation. DECAF's FlowMap sampler (taking velocity predictions directly from the trained flow map at each step) outperforms Boltz-1's standard ODE solver at every sampling-step count — at 5 / 10 / 20 steps respectively by +2.2, +0.4, and +1.0 pp on RMSD$<$ 2 Å. Most notably,

DECAF at 20 steps (76.2%) *matches* Boltz-1 at 200 steps (76.2%, dashed reference), a $10\times$ reduction in inference compute.

C.2. Qualitative analysis

PoseBusters quality checks. Table 8 collates statistics of the different PoseBusters sub-check failures for DECAF-SEARCH and Boltz-1x. As noted in the main text section 4.3, we see high failure rates due to lack of sp2-hybridized bond flatness across all models, likely due to suboptimal reward design. Other checks are failed much less frequently and generally improve with increasing NFE (consider e.g. the bond angles failure mode). In fig. 7 we highlight some notable PoseBusters failures, including examples of the pervasive sp2 planarity issue (middle row). We also flag some false positives of the PoseBusters checks, where accurate poses which capture ionic coordination are flagged as having incorrect distance and volume overlap.

Practically-relevant slices of Runs N' Poses. As we note in section 4.3, it is critical to validate the performance of DECAF on systems that are relevant for users of cofolding models. To emulate this setting, we slice Runs N' Poses to a subset of structures that highlight common settings in small-molecule drug discovery and report results in table 9. We note that DECAF-SEARCH has strong performance at both low and high NFE, and matches or outperforms Boltz-1x at similar NFEs. We also highlight select structures from each category in fig. 8.

Table 7. **Success rate breakdown for DECAF-SEARCH as NFEs grow.** We report the fraction of Runs N' Poses that succeed along the DECAF-SEARCH Pareto front, highlighting that while increasing NFEs does improve accuracy, the low-NFE variants do offer complementary successes to the most expensive MCTS-based sampling method.

Outcome	count	%
All three succeed	479	68.23%
Only DECAF FK (NFE=40) + DECAF MCTS (NFE=800) succeed	24	3.42%
Only DECAF FK (NFE=20) + DECAF MCTS (NFE=800) succeed	7	1.00%
Only DECAF FK (NFE=20) + DECAF FK (NFE=40) succeed	7	1.00%
Only DECAF MCTS (NFE=800) succeeds	25	3.56%
Only DECAF FK (NFE=40) succeeds	7	1.00%
Only DECAF FK (NFE=20) succeeds	7	1.00%
None succeeds	146	20.80%
Total	702	100.00%

Table 8. **Per-complex failure rates by PoseBusters sub-check** (best@5 over PoseBusters metrics). DECAF-SEARCH exceeds Boltz-1x at 600 NFE across all NFE levels, and we see generally increasing pose quality with increasing NFE. Lower is better for all entries in this table.

Sub-check	**DECAF-SEARCH**				**Boltz-1x**
	NFE = 20	NFE = 40	NFE = 75	NFE = 489	NFE = 600
double bond flatness	6.34%	8.10%	7.27%	7.29%	7.83%
volume overlap with protein	2.82%	2.11%	0.69%	2.43%	3.20%
bond angles	3.87%	2.11%	0.35%	0.00%	0.00%
bond lengths	1.76%	1.41%	0.00%	0.00%	0.00%
volume overlap with organic cofactors	1.06%	1.76%	0.00%	1.07%	1.07%
minimum distance to protein	0.35%	0.00%	0.35%	0.00%	0.36%
minimum distance to organic cofactors	0.00%	0.00%	0.35%	0.00%	0.36%
minimum distance to waters	0.00%	0.00%	0.35%	0.00%	0.36%
internal steric clash	0.70%	0.00%	0.00%	0.00%	0.00%
PB-valid pass rate	89.79%	90.14%	**92.04%**	90.62%	88.97%

Table 9. **Method performance on challenging and chemically-relevant targets.** We measure best@5 joint RMSD < 2 Å and PB-valid performance on the following slices of the Runs N' Poses set (higher is better). Drug-likeness is determined using a Quantitative Estimate of Drug-likeness (QED) score threshold of 0.65, as is typical for a drug-like small molecule (Bickerton et al., 2012). Methods marked with an asterisk have a statistically-significant improvement relative to the best Boltz-1x version as measured by a two-sided paired Wilcoxon signed-rank test with $p < 0.01$.

Slice	**N**	**DECAF-SEARCH**		**Boltz-1x**	
		NFE = 40	NFE = 800	NFE = 40	NFE = 800
Drug-like ligands	283	****0.809**	0.806	0.787	0.783
Ion/cofactor coordination	134	0.754	****0.761**	0.749	0.760
Ligand at PPI	292	0.777	****0.784**	0.779	0.780

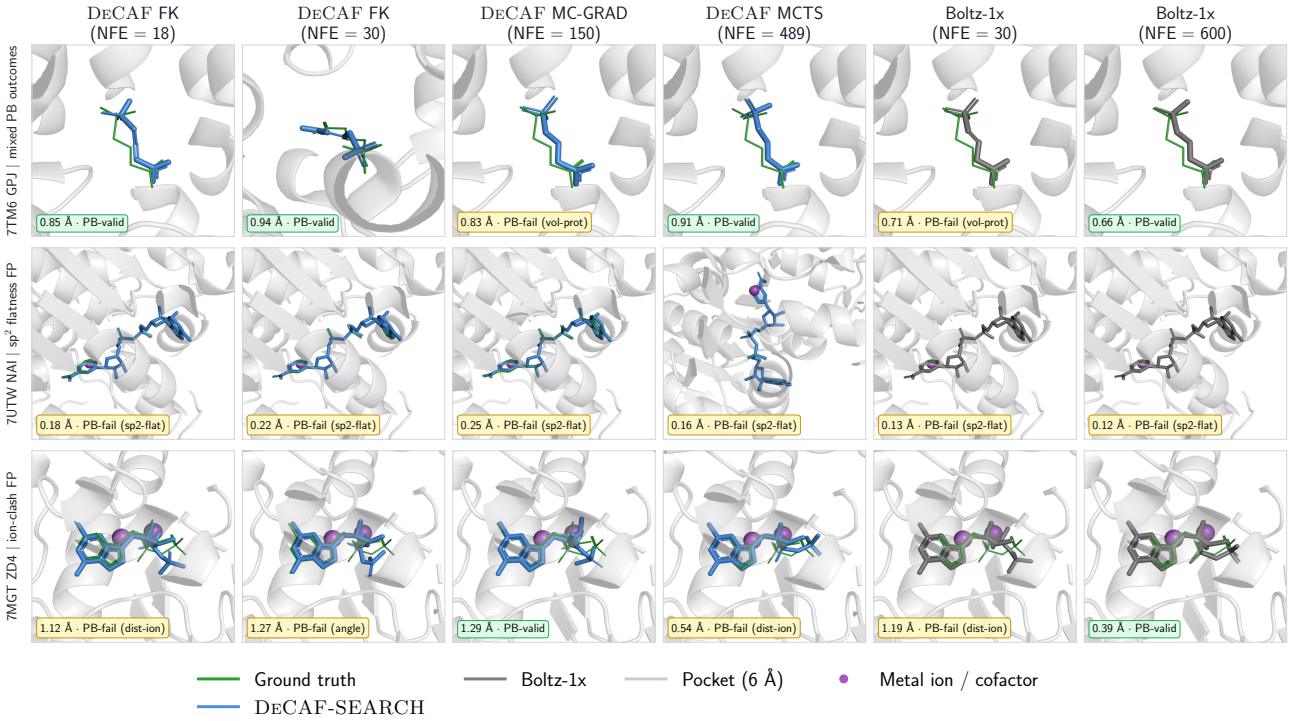

Figure 7. **Example failure modes due to PoseBusters checks** DECAF predictions blue, Boltz-1 gray. Pocket cartoon (gray) is the residues within 6 Å of the crystal ligand (green).

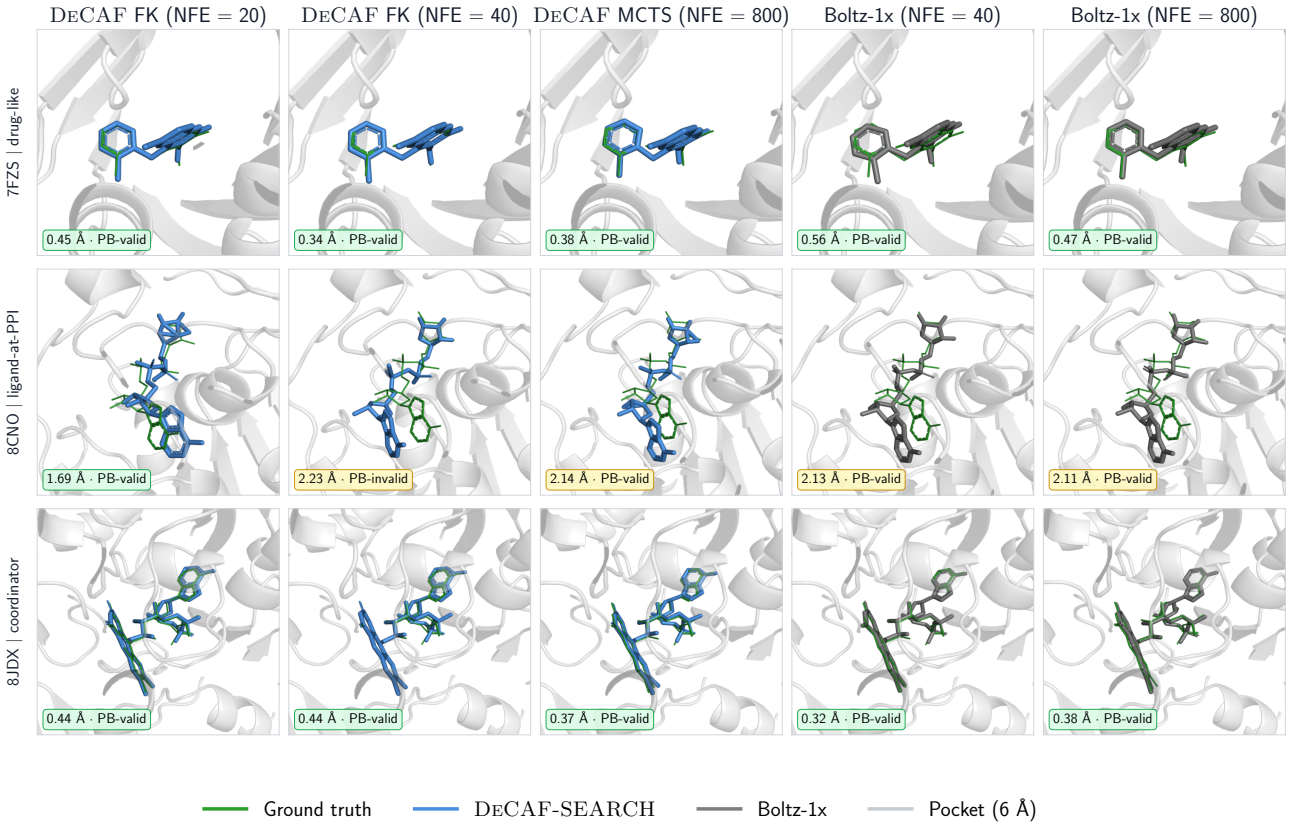

Figure 8. **Qualitative grid for chemically-relevant subsets of Runs N' Poses.** DECAF predictions blue, Boltz-1 gray. Pocket cartoon (gray) is the residues within 6 Å of the crystal ligand (green).

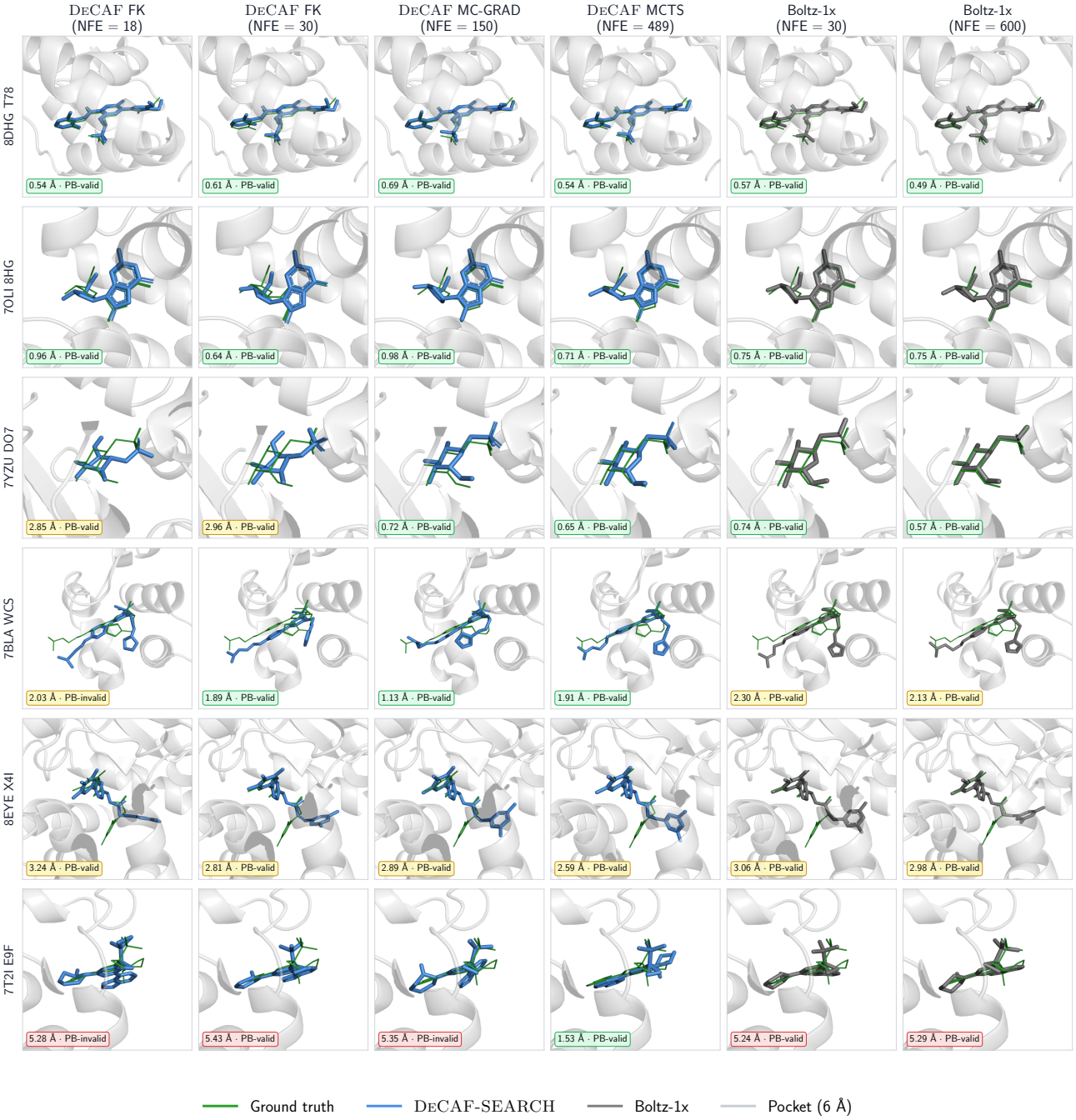

Figure 9. **Multi-method qualitative grid for PoseBusters complexes.** DECAF predictions blue, Boltz-1 gray. Pocket cartoon (gray) is the residues within 6 Å of the crystal ligand (green).