Equivariant Force Field Calibration for Flow-based Protein Design

Fang Wu
Stanford University
fangwu97@stanford.edu

Abstract

Flow models show promise for molecular design due to their fast, expressive sampling capabilities. However, their applicability to complex biological systems remains limited by a lack of physical grounding, which leads to unrealistic or unstable molecular structures. This work presents a framework that integrates force field guidance with consistency-based training to improve the physical fidelity of flow-based generative models. We calibrate pretrained flow models using differentiable physical energy functions to steer generation toward low-energy and sterically valid conformations. A consistency-based training strategy enables accurate, robust generation with very few sampling steps, improving inference efficiency. We evaluate it on three protein design tasks: unconditional backbone generation, full-atom peptide binder co-design, and antibody CDR-H3 redesign. Experiments show that our approach consistently improves physical plausibility and geometric stability, while enabling favorable trade-offs between structural diversity and computational cost. By unifying physical calibration and efficient sampling, we advance the scalability and reliability of flow-based molecular generation.

1 Introduction

Flow models [42, 60] have emerged as a popular generative paradigm [55, 56], widely applied to diverse domains, including images [8] and language [22]. Their utility extends to complex biological and physical systems [9, 70, 37, 26, 44, 73, 28, 38, 19], with potential in designing novel molecules, such as antibodies [74], RNA [49, 1], peptides [69, 71, 35, 72], and enzymes [23, 51].

Despite advances, some methods often overlook fundamental physical constraints during molecule generation. This disconnect leads to unrealistic or energetically unfavorable molecular structures, especially when generalizing to unseen receptors or scaffolds [63, 10]. Lacking built-in awareness of physical principles, such as force fields that govern molecular stability and interaction, these models frequently produce outputs with severe geometric artifacts, including steric clashes and distorted bond geometries [34, 59, 25]. Such issues are especially detrimental in protein-ligand modeling, where generation must adapt to new receptor contexts beyond the training corpus.

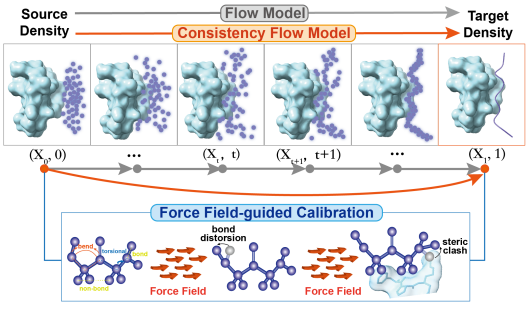


Figure 1: Illustration of our method.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Limitations of Prior Energy-guided Methods. Recent efforts [48, 46, 63, 33, 53, 58, 64, 75] employ energy functions as proxy reward signals to guide structure generation or molecular docking, especially within diffusion models. However, these approaches exhibit key limitations:

- **Energy approximation bias.** Methods like ConfDiff [63] need auxiliary networks to estimate energy scores or condition generation on scalar energy values, causing bias and bad generalizability.
- **Inefficient sampling.** Approaches like DiffForce [33] apply force field guidance only at inference, leading to substantial computational overhead without influencing model parameters directly.
- **Lack of inductive bias.** Models like EnerBridge-DPO [53] treat physical knowledge as an external post hoc guide, rather than embedding it within the generative process. As a result, the learned distributions cannot accurately capture the critical principles of physics and thermodynamics.
- **Model restriction.** Existing methods are largely confined to diffusion-based architectures but leave flow-based models unexplored, despite their efficiency and invertibility.

To address these challenges, we introduce *equivariant force field-guided calibration* (FFC) framework for enhancing flow-based generative models with physical fidelity and sampling efficiency (Fig. 1). We develop a physically grounded post-training calibration method that aligns the terminal outputs of a pretrained flow model with low-energy configurations derived from differentiable force fields, such as those computed by OPENMM [16]. By interpreting the energy landscape as a terminal reward function, FFC uses its gradient to define a force-corrected terminal target velocity, which is then propagated to earlier times via consistency-based training against an exponential-moving-average (EMA) teacher. This reshapes the learned velocity field without backpropagating gradients through the ODE trajectory or solving adjoint equations. Moreover, FFC adopts a consistency paradigm in which the generative process learns to map each noisy input to a common terminal structure in a time-consistent manner. The velocity field is trained to be consistent across interpolation steps and to satisfy boundary conditions that reflect geodesic proximity between sampled noise and clean molecular states. This enables high-quality sample generation in only one or two steps, improving scalability without sacrificing accuracy.

FFC is trained jointly in a single-stage pipeline that optimizes consistency and physical calibration *simultaneously* at every iteration (Alg. 1); there is no separate consistency-distillation phase followed by a calibration phase. This unified procedure avoids the need for costly reverse-time integration during inference while simultaneously embedding physical priors into the flow dynamics. We evaluate our method across three protein-generation tasks, including backbone-structure modeling, peptide binder design, and antibody CDR-H3 redesign. Experimental results show that our approach consistently improves physical plausibility and geometric stability, while largely preserving structural diversity. It also enables efficient large-scale sampling for downstream applications. A discussion of related work is deferred to App. A. Together, these contributions offer a practical approach for integrating physical priors with efficient inference in flow-based molecular generation.

2 Method

2.1 Trajectory-wise Calibration

Energy-guided Distribution Reweighting. Given a pretrained flow model parameterized by $v_\theta(\mathbf{x}_t, t)$ and a differentiable energy function $\mathcal{E}(\mathbf{x})$, we consider an energy-biased target distribution over terminal samples:

$$p(\mathbf{x}_1) \;\propto\; q_\theta(\mathbf{x}_1)\, \exp\big(-\xi \mathcal{E}(\mathbf{x}_1)\big), \tag{1}$$

where $q_\theta(\mathbf{x}_1)$ is the model's original terminal distribution and $\xi > 0$ controls the influence of the energy term. The gradient $\nabla\mathcal{E}(\mathbf{x})$ defines a physical force field $\mathcal{F}(\mathbf{x})$ that encourages lower-energy configurations. In practice, we view $p(\mathbf{x}_1)$ as an *idealized target*: push the learned flow in directions favored by the energy, without modifying the sampling procedure or explicitly estimating p.

Remark 2.1 (Energy versus guidance field)**.** *Eq. 1 is a* motivating special case *that applies when the guidance field is conservative. The regression objective we actually implement (Eq. 4) requires only a vector-valued terminal guidance field $\mathcal{F} : \mathcal{M} \to \mathbb{R}^{3n}$; it never evaluates $\mathcal{E}$ itself. When $\mathcal{F} = -\nabla\mathcal{E}$ for some scalar potential, the calibration admits the Gibbs-reweighting interpretation above. For a non-conservative or learned guidance field, such as the confidence and affinity proxies discussed below, calibration remains well-defined, but the energy-reweighting interpretation of Eq. 1 no longer applies, and no claim about the induced terminal density should be read into it.*

Terminal Vector Field Calibration. We interpret the generative process as a continuous-time transformation driven by a velocity field $v_\theta(\mathbf{x}_t, t)$, evolving from $\mathbf{x}_0 \sim q_0$ to a terminal state $\hat{\mathbf{x}}_1$. Conceptually, there exists an unknown calibrated vector field $u^\star(\mathbf{x}_t, t)$ whose flow would induce an energy-biased terminal distribution $p(\mathbf{x}_1)$ that favors low-energy configurations. Rather than modifying the sampling-time dynamics or solving an optimal-control problem, we perform post-training calibration by *regressing* the terminal velocity toward a physically informed target.

Notation for the Terminal Target. Forces are only numerically meaningful on a structure free of severe steric artifacts, so before evaluating physical forces we apply a short energy minimization to the generated terminal sample using the same force field. We use the following notation consistently throughout the paper. Let $\hat{z}_1$ denote the terminal *latent* state produced by the student and ζ the differentiable full-atom reconstruction module of Sec. 2.4 (App. G). Then

$$\hat{y}_1 = \zeta(\hat{z}_1), \qquad \tilde{y}_1 = \mathrm{sg}\big[\mathrm{Relax}(\hat{y}_1)\big], \tag{2}$$

$$\widetilde{\mathcal{F}}_1 = \mathrm{NormClip}\big[\mathcal{F}(\tilde{y}_1)\big], \qquad y_{\text{target}} = \mathrm{sg}\big[\,\bar{y}_1 + \lambda_{\mathcal{F}}\,\widetilde{\mathcal{F}}_1\,\big], \tag{3}$$

where $\mathrm{sg}[\cdot]$ is a stop-gradient, Relax is the force-field energy minimization of App. E.2, $\mathrm{NormClip}$ is the normalization of Eq. 5, $\bar{y}_1 = \zeta(\bar{z}_1)$ is the terminal structure produced by the EMA teacher, and $\lambda_{\mathcal{F}} > 0$ controls the physical correction strength. Crucially, the force is evaluated at the *relaxed* structure $\tilde{y}_1$, not at $\hat{y}_1$, and the relaxation is a non-differentiable preprocessing step: no gradients are propagated through it. Intuitively, $\bar{y}_1$ preserves the original generative behavior, while the force term tilts the flow toward directions that decrease the energy of terminal states.

How $\hat{z}_1$ Is Drawn and Where Gradients Flow. As Eq. 4 must be differentiable in θ while no gradient traverses the generative trajectory, it is worth stating the mechanism explicitly. At each training iteration, $\hat{z}_1$ is produced by a short K-step Euler integration of the student from $\mathbf{x}_0 \sim p_0$, with K given in App. E.1. Only the *final* velocity evaluation is retained in the computational graph; the preceding $K-1$ evaluations are wrapped in a stop-gradient, as is the teacher rollout that produces $\bar{z}_1$. Consequently $\nabla_\theta \mathcal{L}^\theta_{\text{cal}}$ contains exactly one term, the derivative of ζ composed with the last student evaluation at $t = 1$; there is no recursive differentiation through the solver and no adjoint state. The earlier times are reached not by this gradient but by the consistency pairs.

Calibration Objective. We then calibrate v_θ by minimizing a terminal squared-error objective in Cartesian space:

$$\mathcal{L}^\theta_{\text{cal}} = \mathbb{E}_{\hat{z}_1 \sim q_1^\theta} \|\zeta(\hat{z}_1) - y_{\text{target}}\|_g^2\,, \tag{4}$$

where $\hat{z}_1$ are terminal samples obtained from the current model and $\|\cdot\|_g$ is the geometry-aware norm induced by the Riemannian metric g on the molecular manifold. The calibration objective regresses the model's terminal output toward this target, without backpropagating gradients through the generative trajectory, through the relaxation step, or through the teacher. The force is evaluated only at the terminal time $t = 1$ and serves as supervision for a terminal regression loss. This one-step calibration objective is differentiable w.r.t. θ, and is optimized jointly with the consistency losses in a single-stage procedure.

Implementation and Generalization. We instantiate $\mathcal{F}(\mathbf{x})$ using differentiable molecular mechanics engines such as OPENMM [16], using a standard all-atom force field derived from classical potentials, such as Lennard-Jones, bonded interactions, electrostatics, etc. Raw force magnitudes vary substantially across atoms and structures, leading to numerical instabilities when used directly as supervision. To stabilize training while preserving the *relative* physical scales within each molecular configuration, we normalize forces using a *per-simulation* scale:

$$\widetilde{\mathcal{F}}_i(\mathbf{x}) = \min\left(1, \frac{c}{\|\mathcal{F}_i(\mathbf{x})\| + \varepsilon}\right) \frac{\mathcal{F}_i(\mathbf{x})}{\max_j \|\mathcal{F}_j(\mathbf{x})\|}, \tag{5}$$

where the denominator is the maximum atomic force magnitude for that structure, i indexes atoms, and c is a fixed clipping threshold for the *force magnitudes*. The resulting field is dimensionless, with per-atom magnitudes in $[0, 1]$. Below the clipping threshold, the scheme preserves the relative strengths of different interactions within a structure exactly; above it, relative magnitudes are deliberately compressed, which is the price paid for bounded supervision in the presence of severe clashes. The normalization avoids extremely large gradients and gives $\lambda_{\mathcal{F}}$ a consistent and interpretable scale

across all examples. As Eq. 5 rescales each atomic force by rotation-invariant quantities (its own norm and the per-structure maximum norm), $\widetilde{\mathcal{F}}$ inherits the $\mathrm{SO}(3)$-equivariance of $\mathcal{F}$.

While classical force fields derived from physical potentials are natural choices, FFC is compatible with any differentiable and computationally efficient reward signal. Examples include: (i) confidence metrics from structure prediction models [27, 2]; (ii) protein–ligand affinity estimators or docking scores; and (iii) learned predictors of stability or solubility, as well as regularizers promoting diversity and novelty. The only requirement is the differentiability of the reward function $r(\mathbf{x})$, which enables gradient-based calibration; as noted in Remark 2.1, such signals are generally non-conservative, so only the regression view carries over. After calibration, the generative model operates independently of these external evaluators during inference, yielding improved efficiency and scalability.

2.2 Consistency-driven Calibration for Flow Models

While energy-guided calibration directly steers the terminal-velocity field, naively enforcing trajectory-wise alignment on finely discretized flows can be computationally expensive. To make calibration practical for high-resolution molecular flows, we adopt a *consistency-driven* formulation that learns a few-step, trajectory-consistent student against an EMA teacher.

The EMA Teacher. Let $\bar{v}$ denote a single stop-gradient teacher network maintained as an exponential moving average of the student, $\bar{\theta} \leftarrow \mu\,\bar{\theta} + (1-\mu)\,\theta$, with $\mu = 0.999$, initialized from the pretrained flow. The *same* teacher is used for the consistency loss and for the terminal target of Eq. 3. Since the teacher tracks the student, the force correction absorbed into $v_\theta(\cdot, 1)$ at one iteration enters $\bar{v}(\cdot, 1)$ at the next, and is thereafter transported to earlier times through the consistency pairs.

Consistency Modeling in Continuous Time. We draw inspiration from recent advances in consistency modeling [56, 54], which enable high-quality generation in only a few forward evaluations. These models learn a mapping $f(\mathbf{x}_t, t)$ that is constant along the generative trajectory,

$$f(\mathbf{x}_t, t) = \mathbf{x}_1 \quad \Leftrightarrow \quad \frac{\mathrm{d}f}{\mathrm{d}t} = 0, \quad f(\mathbf{x}_0, 0) = \mathbf{x}_1, \tag{6}$$

so that all intermediate states deterministically map to the same final data point. This effectively replaces expensive numerical integration with a single-step map while preserving trajectory coherence.

In the flow model, we adapt this idea to the *velocity* field $v_\theta(\mathbf{x}_t, t)$. Instead of requiring that positions map to a common fixed point, we encourage v_θ to be *consistent* across time and different noise levels, so a small number of steps suffices to approximate the original flow.

Velocity-based Consistency Modeling. We define a teacher-student consistency objective over time pairs (t, r) with $t < r < 1$, and construct the loss

$$\mathcal{L}_{\mathrm{cot}} = \mathbb{E}_{t,r,\mathbf{x}_t,\mathbf{x}_r}\Big[w(t,r)\;\mathbf{L}\big(v_\theta(\mathbf{x}_t, t), \bar{v}(\mathbf{x}_r, r)\big)\Big], \tag{7}$$

where $(\mathbf{x}_t, \mathbf{x}_r)$ are states along the same trajectory, $\mathbf{L}$ is a distance metric (e.g., squared geodesic norm), and $w(t,r) = \frac{1}{\max(r-t,\varepsilon)}$ with $\varepsilon = 10^{-3}$ reweighs the objective based on the interpolation gap. We additionally restrict sampling to pairs satisfying $r - t \geq \delta$ with $\delta = 0.05$. Since $\delta > \varepsilon$, the ε-guard is never active under this sampling scheme and is retained only for numerical safety; the operative bound is $w(t,r) \leq 1/\delta = 20$. This loss encourages v_θ evaluated at different times and noise levels to align with a temporally-shifted teacher vector field, effectively flattening the flow into a few-step representation. The bounded weighting prevents exploding gradients, improves training stability, and follows stabilization strategies used in consistency models.

However, consistency alone admits trivial solutions (e.g., constant zero vector fields). To prevent degeneration and preserve the original generative behavior, we impose an explicit initial-condition constraint:

$$\mathcal{L}_{\mathrm{init}} = \mathbf{L}\big(v_\theta(\mathbf{x}_0, 0),\ \log_{\mathbf{x}_0}(\mathbf{x}_1)\big), \tag{8}$$

where $\log_{\mathbf{x}_0}(\mathbf{x}_1) \in \mathcal{T}_{\mathbf{x}_0}\mathcal{M}$ is the Riemannian logarithm map, i.e. the geodesic *displacement vector* from $\mathbf{x}_0$ to a corresponding terminal state $\mathbf{x}_1$ (in Euclidean components this reduces to $\mathbf{x}_1 - \mathbf{x}_0$). Note that this is a tangent vector, not the scalar geodesic distance, so directly comparable to a velocity. This loss encourages the initial velocity to point along a geodesic toward the data point and anchors

Algorithm 1 Force Field-guided Calibration (FFC): single-stage joint training

Require: Source/target distributions p_0, p_1; pretrained flow v_{θ_0}; force field $\mathcal{F}$; reconstruction ζ; training steps N; rollout length K; weights $\lambda_{\mathcal{F}}, \beta$; learning rate η; EMA rate μ; gap δ

$\theta \leftarrow \theta_0$; $\quad \bar{\theta} \leftarrow \theta_0$ ▷ EMA teacher initialized from the pretrained flow

for $i = 1$ to N **do**

  Sample $\mathbf{x}_0 \sim p_0$, $\mathbf{x}_1 \sim p_1$, and $t, r \sim \mathcal{U}(0,1)$ with $r - t \geq \delta$

  Compute interpolants $\mathbf{x}_t, \mathbf{x}_r$ and the consistency loss $\mathcal{L}^{\theta}_{\mathrm{cfc}} = \mathcal{L}_{\mathrm{init}} + \mathcal{L}_{\mathrm{cot}}$ ▷ Eqs. 7, 8

  Roll out K Euler steps from $\mathbf{x}_0$ under v_θ to get $\hat{z}_1$; keep only the last step in the graph

  Roll out $\bar{v}$ to get $\bar{z}_1$ (fully stop-gradient); set $\hat{y}_1 = \zeta(\hat{z}_1)$, $\bar{y}_1 = \zeta(\bar{z}_1)$

  $\tilde{y}_1 \leftarrow \mathrm{sg}[\mathrm{Relax}(\hat{y}_1)]$; $\quad \widetilde{\mathcal{F}}_1 \leftarrow \mathrm{NormClip}[\mathcal{F}(\tilde{y}_1)]$ ▷ no gradient through Relax

  $y_{\text{target}} \leftarrow \mathrm{sg}[\bar{y}_1 + \lambda_{\mathcal{F}} \widetilde{\mathcal{F}}_1]$; $\quad \mathcal{L}^{\theta}_{\mathrm{cal}} \leftarrow \|\hat{y}_1 - y_{\text{target}}\|^2_g$

  $\theta \leftarrow \theta - \eta \nabla_\theta \big(\mathcal{L}^{\theta}_{\mathrm{cfc}} + \beta\, \mathcal{L}^{\theta}_{\mathrm{cal}}\big)$

  $\bar{\theta} \leftarrow \mu\, \bar{\theta} + (1-\mu)\, \theta$ ▷ terminal correction enters the teacher here

end for

return calibrated flow model v_θ

the flow. The consistency-driven flow calibration (CFC) loss is $\mathcal{L}_{\mathrm{cfc}} = \mathcal{L}_{\mathrm{init}} + \mathcal{L}_{\mathrm{cot}}$, which preserves the pretrained model's trajectory and compresses it into a form amenable to few-step sampling.

Single-stage Joint Optimization. The consistency and calibration objectives are optimized *together* from the first iteration, not in sequence:

$$\mathcal{L}^{\theta}_{\text{total}} \;=\; \mathcal{L}^{\theta}_{\mathrm{cfc}} \;+\; \beta\, \mathcal{L}^{\theta}_{\mathrm{cal}}, \tag{9}$$

with β balancing the two terms ($\beta = 1$ unless otherwise stated). Alg. 1 gives the full procedure: each iteration samples a consistency pair (t, r), draws a terminal sample, evaluates the relaxed force target, and takes a single gradient step on Eq. 9, followed by the EMA update. The student therefore inherits the teacher's trajectory structure and acquires a physically informed terminal correction at the same time, and the teacher in turn inherits the correction through the EMA. This eliminates the need to simulate reverse-time dynamics or solve a full trajectory optimal-control problem.

Equivariance Preservation. All components entering $\mathcal{L}_{\mathrm{cal}}$ are designed to respect $\mathrm{SO}(3)$ symmetry: the reconstruction ζ and the force field $\mathcal{F}$ are equivariant, the normalization of Eq. 5 preserves equivariance, the relaxation operator commutes with rotation, and the norm $\|\cdot\|_g$ is invariant.

Theorem 2.2. *Suppose the energy $\mathcal{E}$ is $\mathrm{SO}(3)$-invariant; the velocity field v_θ, the reconstruction module ζ, and the force field $\mathcal{F} = -\nabla\mathcal{E}$ are $\mathrm{SO}(3)$-equivariant; and the relaxation operator satisfies $\mathrm{Relax}(Qy) = Q\,\mathrm{Relax}(y)$ for every $Q \in \mathrm{SO}(3)$. Then $\mathcal{L}_{\mathrm{cal}}$ is invariant under any rotation $Q \in \mathrm{SO}(3)$.*

Thus, under the stated assumptions, the calibrated model respects physical symmetries in molecular space and produces consistent outputs under coordinate transformations. The proof in App. C.1 discusses when the hypothesis on Relax holds exactly and when only approximately (Remark C.1).

2.3 Propagation of Terminal Calibration

While the physical supervision is applied only at the terminal time $t = 1$, the consistency objective provides a practical mechanism for propagating it backward along the flow, empirically aligning the full velocity field without requiring backward adjoint integration. Recall that consistency enforces $v_\theta(x_t, t) \approx \bar{v}(x_r, r)$ for $0 \leq t < r < 1$, where (x_t, x_r) lie on the same teacher trajectory. Because the teacher is an EMA of the student, once the student's terminal behavior is modified by the force correction, the teacher's terminal behavior follows within a few hundred iterations, and its influence is then inherited by all (t, r) pairs whose teacher signal depends on the terminal state.

The following is an *informal* order-of-magnitude argument, offered for intuition only. If the learned field satisfies a Lipschitz-type temporal regularity $\|v_\theta(x_t, t) - v_\theta(x_r, r)\| \leq L\,|t - r|$, and if a perturbation $\Delta v(\cdot, 1)$ applied at $t{=}1$ propagates with a bounded relative rate, then the induced deviation at earlier times is of order

$$\|v_\theta(\cdot, t) - \bar{v}(\cdot, t)\| \;=\; O\big((1-t)\,\|\Delta v(\cdot, 1)\|\big). \tag{10}$$

We stress that this is a heuristic scaling, not a theorem: we do not claim that learned neural flows satisfy such smoothness uniformly, nor that the constants are controlled in practice. The empirical evidence for propagation is the consistent improvement at *all* sampling budgets reported in Sec. 3.

2.4 Protein Generation with Flow Models

Existing flow models for protein design generally fall into two categories: (i) *single-modality modeling* [14] directly models C_α coordinate trajectories in Euclidean space, and (ii) *multi-modality modeling* [76, 24, 32, 35, 71] factorizes protein structure into multiple components such as backbone frames, torsion angles, and residue types, and defines flows on each manifold separately.

Our method is compatible with both modeling paradigms but particularly enhances multi-modality frameworks, where structure is represented using heterogeneous geometric variables. We now outline the representation scheme, the flow construction over each modality, and how our physical calibration framework integrates into this architecture.

Protein Representations. A protein of n residues is modeled using a set of discrete and continuous latent variables. For each residue i, we represent (1) *Residue identity:* the amino acid type $a^i \in \mathcal{A}$; (2) *Backbone frame:* $\mathbf{x}^i_{\mathrm{C}_\alpha} \in \mathbb{R}^3$ and $O^i \in \mathrm{SO}(3)$ define position and orientation; (3) *Side-chain conformation:* the torsion angles $\boldsymbol{\chi}^i \in [0, 2\pi)^4$ defines side-chain rotations. We write $z^i = (\mathbf{x}^i_{\mathrm{C}_\alpha}, O^i, \boldsymbol{\chi}^i, a^i)$ for the collection of latents.

Multi-modality Flows. We define distinct flows over each modality-specific manifold as (1) A linear Euclidean flow for positions: $\phi^{\text{pos}}_t = t\mathbf{x}^i_1 + (1-t)\mathbf{x}^i_0$, with $\mathbf{x}^i_0 \sim \mathcal{N}(0, I_3)$; (2) A geodesic flow on $\mathrm{SO}(3)$ for orientations: $\phi^{\text{ori}}_t = \exp_{O^i_0}(t \log_{O^i_0}(O^i_1))$, with $O^i_0 \sim \mathcal{U}(\mathrm{SO}(3))$; (3) A circular flow on the 4D torus for torsions: $\phi^{\text{ang}}_t = [t\boldsymbol{\chi}^i_1 + (1-t)\boldsymbol{\chi}^i_0] \bmod 2\pi$, with $\boldsymbol{\chi}^i_0 \sim \mathcal{U}([0, 2\pi)^4)$. The time derivatives of these flows define the conditional velocity fields, containing $u^{\text{pos}}_t(\mathbf{x}^i_t) = \frac{\mathbf{x}^i_1 - \mathbf{x}^i_t}{1-t}$, $u^{\text{ori}}_t(O^i_t) = \frac{\log_{O^i_t}(O^i_1)}{1-t}$, $u^{\text{ang}}_t(\boldsymbol{\chi}^i_t) = \left(\frac{\boldsymbol{\chi}^i_1 - \boldsymbol{\chi}^i_t}{1-t} + \pi\right) \bmod 2\pi - \pi$. Networks v^{pos}_θ, v^{ori}_θ, and v^{ang}_θ approximate these vector fields, and a discrete flow v^{aa}_θ [11] models amino acid types a^i (see App. F).

Calibration over All Modalities. FFC poses a unique challenge in multi-modal models because physical force fields are defined on full-atom Euclidean coordinates, whereas the generative process operates on a mix of Euclidean, rotational, and angular variables. To bridge this gap, we introduce a differentiable reconstruction module $\zeta(z^i) = \zeta(\mathbf{x}^i_{\mathrm{C}_\alpha}, O^i, \boldsymbol{\chi}^i, a^i)$ that reassembles full Cartesian coordinates y^i from latent variables. ζ applies standard geometric operations, including rigid transformations, axis-angle rotations for side chains, and amino-acid-specific internal coordinate templates, to recover full atomic coordinates in a differentiable manner. This enables gradients from force-based objectives to be backpropagated to all continuous modalities during training. Importantly, FFC does not directly optimize discrete residue identities using force-field gradients. Instead, improvements in sequence quality arise indirectly, as calibrated structural flows reshape the joint sequence–structure distribution through consistency regularization and shared latent dynamics. We provide a detailed explanation and differentiability analysis of ζ in App. G.

Unified Calibration Loss. With ζ, the terminal objective of Eq. 4 applies verbatim to the multi-modal setting, with

$$\hat{y}_1 = \zeta(\hat{z}_1), \quad \bar{y}_1 = \zeta(\bar{z}_1), \quad \tilde{y}_1 = \mathrm{sg}[\mathrm{Relax}(\hat{y}_1)], \quad \mathcal{L}^\theta_{\text{cal}} = \left\| \hat{y}_1 - \mathrm{sg}\left[\bar{y}_1 + \lambda_{\mathcal{F}}\, \mathrm{NormClip}[\mathcal{F}(\tilde{y}_1)]\right] \right\|^2_g, \tag{11}$$

where $\hat{z}_1$ and $\bar{z}_1$ are the terminal latents obtained by integrating the current model $(v^{\text{pos}}_\theta, v^{\text{ori}}_\theta, v^{\text{ang}}_\theta, v^{\text{aa}}_\theta)$ and the EMA teacher $(\bar{v}^{\text{pos}}, \bar{v}^{\text{ori}}, \bar{v}^{\text{ang}}, \bar{v}^{\text{aa}})$, respectively, and the force is evaluated in Cartesian space on the relaxed student structure. The force-corrected teacher structure is treated as a stop-gradient pseudo-target. This formulation incorporates physical supervision and multi-modal consistency within a single-stage training procedure. By calibrating across all structure-defining modalities, we ensure that generated proteins are physically plausible and geometrically coherent.

2.5 Theoretical Analysis of Calibration

Rather than deriving an explicit closed form for the calibrated flow, FFC is seen as a regression toward an ideal calibrated vector field in a Hilbert space of time-dependent velocities. For a vector field v, let ϕ_t^v be the solution flow of ODE $\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = v(\mathbf{x}_t, t), \quad \mathbf{x}_0 \sim q_0$, and write $q_t^v = [\phi_t^v]_{\#} q_0$ for the induced marginal at time t. The pretrained model is v_{θ_0}, with terminal marginal $q_1^{\theta_0}$. We consider an energy-biased target density $p(\mathbf{x}_1) \propto q_1^{\theta_0}(\mathbf{x}_1) \exp\big(-\xi\, \mathcal{E}(\mathbf{x}_1)\big)$, which encodes a preference for low-energy configurations; per Remark 2.1, this analysis is specific to the conservative case. We make the following idealized assumption.

Assumption 2.3 (Existence of a calibrated flow)**.** *There exists a vector field $u^\star(\mathbf{x}_t, t)$ such that the flow $\phi_t^{u^\star}$ satisfies $[\phi_1^{u^\star}]_{\#} q_0 = p$. Let $q_t^\star = q_t^{u^\star}$ denote the corresponding path marginals.*

This assumption is standard in flow-based formulations and does not require an explicit characterization of $u^\star$. It simply states that, in principle, some calibrated vector field could realize the energy-biased terminal distribution p. We stress that such a $u^\star$ is *not unique*: infinitely many vector fields transport q_0 to p, and the analysis below fixes one of them together with its path marginals $q_t^\star$, which in turn define the Hilbert space. The uniqueness claimed in Prop. 2.4 is therefore uniqueness of the minimizer *given* that choice, not of the calibrated flow itself.

Calibration as L^2 Regression. Consider the Hilbert space $\mathcal{H} = L^2\big([0,1] \times \mathcal{M},\, q_t^\star;\, \mathbb{R}^d\big)$, with inner product $\langle v, w\rangle_{\mathcal{H}} = \int_0^1 \mathbb{E}_{\mathbf{x}_t \sim q_t^\star}\big[\, \langle v(\mathbf{x}_t, t),\, w(\mathbf{x}_t, t)\rangle_g \big]\, \mathrm{d}t$. In this idealized population setting, the energy-guided calibration objective becomes

$$\mathcal{L}_{\mathrm{cal}}(v) = \tfrac{1}{2} \int_0^1 \mathbb{E}_{\mathbf{x}_t \sim q_t^\star} \big\| v(\mathbf{x}_t, t) - u^\star(\mathbf{x}_t, t) \big\|_g^2 \, \mathrm{d}t, \tag{12}$$

which is a strictly quadratic functional uniquely minimized at $v = u^\star$. We emphasize that Eq. 12 is the *population* objective over the whole time horizon, and is not identical to the practical terminal loss of Eq. 4; Sec. 2.6 discusses the gap. Within this idealized setting we obtain the following.

Proposition 2.4 (Well-posedness of the regression view)**.** *Fix $u^\star$ as in Assumption 2.3. The population objective $\mathcal{L}_{\mathrm{cal}}$ of Eq. 12 is 1-strongly convex on $\mathcal{H}$ with 1-Lipschitz gradient, so exact gradient descent in $\mathcal{H}$, $v^{(k+1)} = v^{(k)} - \eta\, \nabla \mathcal{L}_{\mathrm{cal}}(v^{(k)})$ with $0 < \eta < 2$, converges linearly to $u^\star$: $\big\| v^{(k)} - u^\star \big\|_{\mathcal{H}} \le |1 - \eta|^k \, \big\| v^{(0)} - u^\star \big\|_{\mathcal{H}}$.*

Prop. 2.4 is an immediate consequence of writing Eq. 12 as a shifted squared norm (App. C.2), and we claim nothing more for it than that: it records that the regression view is well-posed and free of spurious minima. It is not a statement about the trained model. Neither $u^\star$ nor the path marginals $q_t^\star$ are accessible; the vector field is a finite-capacity network estimated from finite samples, and the objective we actually optimize is terminal rather than full-path.

2.6 Practical Approximation of the Ideal L^2 Regression

Prop. 2.4 analyzes the ideal regression problem in which $u^\star$ is recovered by fitting the entire time-dependent flow along $t \in [0, 1]$. Our method approximates this objective using two complementary components: (i) a trajectory-consistency loss that preserves the teacher flow for $t < 1$, and (ii) a terminal calibration loss that explicitly aligns the terminal output with a physically informed, force-corrected target.

Informally, if consistency training induces a Lipschitz-type temporal regularity of the learned velocity field as $|v_\theta(x_t, t) - v_\theta(x_1, 1)\| \le L(1-t)$, for some smoothness constant L, then the time-averaged discrepancy between a terminal-only regression and the ideal full-path regression is bounded by

$$\int_0^1 L(1-t)\, \mathrm{d}t \;=\; \frac{L}{2}, \tag{13}$$

i.e. the deviation is at most $L/2$, which is small whenever consistency training succeeds in flattening the flow. We present this as motivation rather than as a formal guarantee, since our Lipschitz assumption is not an established property. It does, however, explain why FFC is observed to improve the learned velocity field at all intermediate times, even though the physical supervision is applied solely at the terminal time.

3 Experiments

We evaluate three design tasks: (*i*) unconditional generation of monomeric protein backbones, (*ii*) conditional sequence-structure co-design of receptor-binding peptides, and (*iii*) conditional antibody CDR-H3 redesign. Implementation details, experimental setups, and runtime analysis are in App. E.1. Additional ablation studies on the force magnitude and consistency strategy are in App. H.

3.1 Unconditional Protein Design

Setup. We apply FFC to FrameFlow [76] and train on the SCOPe dataset [17] comprising 3,938 monomeric proteins of up to 128 residues. During evaluation, we sample 10 backbones for each length between 60 and 128 residues and design 8 sequences per backbone using ProteinMPNN [15]. This experiment is an ablation of FFC against its own uncalibrated base model under matched sampling budgets; cross-method comparisons are reported for the peptide task (Tab. 2, 3).

Metrics. (1) **Designability**, the fraction of sequences that fold back to within 2.0 Å RMSD of their original backbone using ESMFold [41]; (2) **Diversity**, defined as the number of structural clusters (via MaxCluster [20]) normalized by the number of designable samples. Numbers in parentheses indicate the absolute number of clusters; and (3) **Novelty**, measured as one minus the average TM-score to the closest structure in PDB using FoldSeek [61], with higher scores indicating greater novelty.

Results. In Tab. 1, the calibrated model consistently outperforms the uncalibrated baseline in designability, with absolute gains of +0.13, +0.06, and +0.04 at 10, 100, and 500 steps, respectively. The gain is largest in the few-step regime: with only 10 timesteps, FFC reaches 0.80 designability, approaching the uncalibrated baseline's 0.81 at 500 steps, a 50× reduction in sampling cost at comparable quality. Novelty is maintained or slightly improved across all budgets. Normalized diversity decreases modestly, but the absolute cluster counts move far less (124→121 at 10 steps, 147→135 at 100 steps, 123→126 at 500 steps), indicating that much of the apparent drop reflects the larger designable denominator rather than mode collapse. We nonetheless acknowledge a genuine, if modest, quality–diversity trade-off; the sensitivity study in App. H shows that it is governed by $\lambda_{\mathcal{F}}$, with moderate force weights improving quality and excessive force strength causing marked deterioration.

Table 1: Evaluation of unconditional backbone generation, comparing FrameFlow with and without FFC at matched budgets. **Bold** indicates the better value within each pair. Absolute differences attributable to FFC are shown in gray. Note that the normalized diversity metric is divided by the number of designable samples, so it falls partly because designability rises; absolute cluster counts are given in parentheses.

FFC	Timesteps	Des. ↑	Div. (clusters) ↑	Nov. ↑
✗	10	0.67	**0.27** (124)	0.37
✓	10	**0.80** (+0.13)	0.22 (121) (-0.05)	**0.38** (+0.01)
✗	100	0.77	**0.28** (147)	0.33
✓	100	**0.83** (+0.06)	0.24 (135) (-0.02)	**0.36** (+0.03)
✗	500	0.81	**0.22** (123)	0.31
✓	500	**0.85** (+0.04)	0.21 (126) (-0.01)	**0.33** (+0.02)

3.2 Peptide Design

Setup and Metrics. We select PepFlow [37] as the backbone trained on PepMerge, a curated dataset derived from PepBDB [67] and Q-BioLip [66]. It provides 8,207 non-homologous complexes for training and validation and a held-out evaluation set of 158 complexes grouped into 10 clusters; App. E.1 gives the split details. We also include the adapted energy-guidance baselines described in App. E.3: energy-conditioned flow [63], inference-time force guidance [33], reconstruction-based energy [79], and a consistency-only ablation. Metrics include:

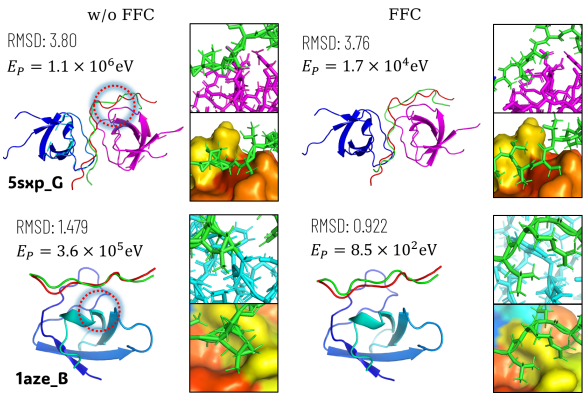


Figure 2: Instances with and w.o. FFC, where RMSD and energy are reported. FFC substantially reduces steric clashes between side chains of generated peptides and receptors, leading to favorable energy profiles.

Table 2: Evaluation for peptide co-design. In block (A), **bold** marks the best value across *all* methods in the table and underline the best value among the block-(A) baselines; in block (B), **bold** marks the better value within each matched-step PepFlow/+FFC pair.

Method	Steps	Geometry					Energy		Design	
		Valid↑	AAR↑	RMSD↓	SSR↑	BSR↑	Stb↑	Aff↑	Des↑	Div↑
(A) Baseline models										
RFdiffusion [65]	200	**66.04**	40.14	4.17	63.86	26.71	**26.82**	16.53	78.52	0.38
ProteinGenerator [43]	200	65.88	45.82	4.35	29.15	24.62	23.48	13.47	71.82	0.54
DiffAb [47]	200	34.62	47.04	3.28	74.89	49.83	15.34	17.13	48.54	0.57
PPFlow [39]	100	36.09	48.35	3.59	68.13	25.94	15.77	12.08	46.53	0.51
PepGLAD [31]	100	55.20	–	3.83	80.24	19.34	20.39	10.47	75.07	0.32
PepHAR [36]	–	59.31	48.35	2.68	84.91	86.74	16.62	20.53	**79.11**	0.29
(B) PepFlow [37] vs. PepFlow+FFC across matched steps										
PepFlow	1	19.08	40.27	3.76	62.58	61.38	10.69	4.81	31.41	**0.68**
+ FFC	1	**31.48**	**46.73**	**2.89**	**72.36**	**72.58**	**13.83**	**9.41**	**44.20**	0.66
PepFlow	2	26.72	39.51	3.53	65.14	65.52	11.76	5.88	34.96	0.63
+ FFC	2	**34.92**	**45.66**	**2.64**	**73.55**	**74.23**	**14.42**	**10.71**	**46.02**	0.63
PepFlow	5	29.59	40.08	3.43	66.70	67.94	12.29	6.42	38.52	0.61
+ FFC	5	**38.66**	**47.20**	**2.41**	**74.97**	**76.81**	**15.07**	**12.16**	**49.83**	0.60
PepFlow	10	33.10	44.69	3.02	69.32	70.05	12.83	9.62	43.07	0.62
+ FFC	10	**41.83**	**50.09**	**2.17**	**78.44**	**79.76**	**16.24**	**13.77**	**56.24**	0.60
PepFlow	200	40.27	51.25	2.07	83.46	86.89	18.15	21.37	65.22	0.42
+ FFC	200	**56.83**	**51.89**	**1.97**	**86.34**	**87.01**	**20.46**	**21.59**	**65.87**	0.44

Table 3: Peptide design with physical signal injection under the **1-step sampling regime**. Time is wall-clock seconds per sample at inference. * indicates $p < 0.01$ vs. the PepFlow baseline (paired t-test on the per-complex metric, $n = 158$ complexes; for the proportion-valued metrics Valid and Stb we report the same test for consistency with prior work).

Method	Valid↑	AAR↑	RMSD↓	Stb↑	Time(s)↓
PepFlow	19.08	40.27	3.76	10.69	0.05
+ Consistency only	24.31	42.18	3.42	11.22	0.05
+ Inference-time forces [33]	22.47	41.63	3.51	12.84	1.52
+ Idempotent flow [79]	26.15	43.91	3.28	11.87	1.46
+ Energy conditioning [63]	28.52	44.85	3.08	12.43	0.07
+ FFC (ours)	**31.48***	**46.73***	**2.89***	**13.83***	**0.05**

(1) **Geometry:** Validity (%), amino acid recovery (AAR), backbone RMSD, secondary structure retention (SSR), and binding site accuracy (BSR). (2) **Energy:** Proportion of peptides with better stability and binding affinity than the native complex, evaluated using Rosetta [5]. (3) **Design:** Designability via ESMFold refolding (RMSD $<$ 2.0 Å), and diversity based on pairwise TM-scores among designable peptides.

Results. Tab. 2 shows that our method improves nearly all metrics across all inference regimes. At the extreme setting of a single inference step, the valid rate improves from 19.08% to 31.48% (+12.40), RMSD decreases by 0.87 Å, and the AAR increases by over 6%. Importantly, designability increases from 31.41% to 44.20%, all in just one forward pass. As steps increase, our method maintains consistent advantages, culminating at 200 steps in the best RMSD (1.97 Å), SSR (86.34%), BSR (87.01%) and binding-affinity (21.59%) figures in the table. We do not claim uniform state of the art: RFdiffusion and PepHAR, which build on pretrained structure-prediction backbones, retain higher Valid (66.04%) and Des (79.11%) at 200 steps, at an order of magnitude more compute per sample, and RFdiffusion also attains the best stability proportion (26.82% vs. 20.46% for FFC). Tab. 3 isolates the comparison against other ways of injecting the same physical signal under a 1-step budget: FFC reaches 31.48% validity versus 28.52% for energy conditioning and 22.47% for inference-time forces, while inference-time guidance incurs $30\times$ higher sampling cost (1.52s vs 0.05s per sample) because the force field must be evaluated at every solver step. Because FFC moves the physical signal into training, its inference cost is identical to the uncalibrated base model. This confirms that direct force field calibration during training outperforms both learned approximations and inference-time corrections.

These results indicate that our calibration enhances structural plausibility and physical realism, while substantially reducing inference cost. Fig. 2 shows that uncalibrated models often produce physically implausible geometries, *e.g.*, atomic clashes at binding sites—due to delayed reconstruction of full-atom structures. Our method mitigates this by directly guiding the generative vector field

using physical forces throughout training, resulting in tighter atomic packing, reduced clashes, and improved interaction geometry.

3.3 Antibody Design

Task Definition. We choose UniMoMo [32] as the backbone architecture, a latent flow model for binder design. Following Luo et al. [47], Kong et al. [29], Wu and Li [68], Kong et al. [32], Wu et al. [74], the task is **conditional CDR-H3 sequence–structure redesign**, not simultaneous generation of all six CDRs or of all three nanobody CDRs. In each test complex, the CDR-H3 loop (defined by the Chothia numbering scheme) is masked and is the sole design target; the antigen, the framework region, and the five remaining CDRs are all provided as fixed conditioning context.

Metrics. **AAR** is *exact* amino acid recovery on the designed loop. It is an identity statistic, not a homology statistic: conservative substitutions receive no credit. **RMSD** is computed on CDR-H3 only, over its backbone heavy atoms including the carbonyl oxygen, after Kabsch superposition of the *framework* backbone atoms of the heavy chain (the loop itself is not used for alignment, so the value measures placement as well as conformation). Because the metric uses backbone atoms only, it is well-defined regardless of whether the designed residue identity matches the native one. **IMP** is the fraction of designs whose binding energy, computed with pyRosetta, improves on the native complex. For each test complex we draw k candidate designs and report best-of-k statistics.

Results. In Tab. 4, FFC provides consistent but modest improvements across all recovery metrics for CDR-H3 redesign. When applied to UniMoMo, FFC yields gains at every sampling budget: exact AAR increases from 52.34% to 54.03% at 100 steps (+1.69 percentage points), RMSD decreases from 1.04 Å to 0.99 Å (a 4.8% reduction), and IMP improves from 65.00% to 68.33%. Even in the challenging 1-step regime, FFC enhances both sequence recovery (from 21.44% to 22.18%) and geometric accuracy (RMSD 2.52 Å $\rightarrow$ 2.39 Å). We note that the sequence-level gains are small in absolute terms, which is expected: as discussed in App. F, no force-field gradient reaches the residue-identity logits, so sequence improvements arise only indirectly through improved geometry. The geometric and energetic improvements are the direct effect of calibration. Comparisons within the UniMoMo rows are like-for-like, since those rows share a single implementation and evaluation script. These results suggest that physically grounded calibration enhances structural fidelity and proxy-binding energetics in antibody loop generation, complementing our findings in peptide and backbone design.

Table 4: Performance for antibody CDR-H3 sequence-structure redesign. The second column is the number of *sampling steps* used by each generative model; the predictive models of block (A) are single-pass and have no such parameter. **Bold** denotes the best, and underline marks the best non-FFC baseline.

Model	Steps	AAR↑	RMSD↓	IMP↑
(A) Predictive models				
MEAN [29]	–	29.13%	1.87	6.67%
dyMEAN [30]	–	31.65%	8.21	11.86%
GeoAB-R [40]	–	32.04%	1.67	6.67%
(B) Generative models across matched steps				
DiffAb [47]	1	24.60%	2.77	10.34%
	10	38.42%	2.08	34.48%
	100	49.74%	1.46	60.34%
GeoAB-D [40]	1	29.74%	1.73	6.67%
	10	38.20%	1.58	20.00%
	100	45.96%	1.50	40.00%
UniMoMo-single	1	20.44%	2.71	15.00%
	10	39.04%	1.90	35.00%
	100	48.78%	1.39	63.33%
UniMoMo-all	1	21.44%	2.52	13.33%
	10	42.05%	1.44	41.67%
	100	52.34%	1.04	65.00%
UniMoMo-all+FFC	1	22.18%	2.39	16.67%
	10	44.76%	1.35	45.00%
	100	**54.03%**	**0.99**	**68.33%**

4 Conclusion

We introduce a unified calibration framework that integrates physically grounded force fields into pretrained flow models. Our method enhances structural stability and physical realism across three protein design tasks, while reducing the number of inference steps required for high-quality generation.

Limitations and Future Work

We highlight the following limitations of FFC.

1. **Dependence on the fidelity of the guidance signal.** FFC inherits whatever biases the underlying force field or learned guidance field carries. If the energy function fails to capture the relevant

physics or chemistry, calibration will faithfully steer generation toward the wrong configurations. We use a fixed classical force field (Amber14) and do not assess sensitivity to that choice.

2. **Force-evaluation cost during calibration.** Inference cost is unchanged, but calibration is expensive: a relaxation and force evaluation is required for *every sampled structure at every training iteration*, at roughly 1.5 s per structure (App. E.2). For the peptide setting (5K iterations, batch size 32) this amounts to on the order of 1.6×10^5 force evaluations, i.e. tens of GPU-hours of OPENMM time on top of the network training itself, and the per-structure cost scales unfavorably with system size. This constrains applicability to large complexes or multi-molecule systems.
3. **Sensitivity to $\lambda_{\mathcal{F}}$.** Performance depends non-monotonically on the force weight: in peptide design, validity peaks at $\lambda_{\mathcal{F}} = 1.0$ and collapses from 52.66% to 12.55% at $\lambda_{\mathcal{F}} = 10.0$ (App. H). The optimum differs across tasks and must be tuned.
4. **Quality–diversity trade-off.** Calibration modestly reduces normalized structural diversity, particularly at short sampling budgets (Tab. 1), and large force weights shift the quality–novelty balance further.
5. **Classical force fields do not guarantee biological function.** Low potential energy and the absence of steric clashes are necessary but far from sufficient for binding or function. Our energy metrics are proxies; Rosetta and pyRosetta scores are themselves approximations.
6. **No wet-lab validation.** All results are computational. We make no claim about experimentally measured binding or expression.
7. **Theoretical scope.** The convergence result (Prop. 2.4) applies to an idealized population objective over the full time horizon, not to the terminal loss we actually optimize with a finite-capacity network; $u^\star$ is moreover not unique (Sec. 2.5). The propagation arguments in Sec. 2.3 and 2.6 are heuristic scalings rather than guarantees.
8. **Incomplete hyperparameter study.** We ablate $\lambda_{\mathcal{F}}$ and the form of the consistency objective, but not the loss balance β (fixed to 1), the EMA rate μ, the minimum gap δ, or the weighting $w(t, r)$. We also report results from a single training run per configuration; the differences of a few tenths of a point in Tab. 1 should be read with that in mind.

Looking ahead, force-field-inspired calibration could be used for post-training on large unlabeled datasets, in the spirit of RLHF. Even without labeled data, this approach could enhance the generalizability of flow models to a broader range of unseen protein families, improving their versatility across protein design tasks. Replacing the classical force field with a learned, differentiable surrogate is a natural route to addressing limitations 1 and 2 simultaneously.

References

[1] A. R. Abir and L. Zhang. Rna-efm: energy based flow matching for protein-conditioned rna sequence-structure co-design. *Bioinformatics Advances*, page vbaf258, 2025.

[2] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3, 2024.

[3] M. S. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[4] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[5] R. F. Alford, A. Leaver-Fay, J. R. Jeliazkov, M. J. O'Meara, F. P. DiMaio, H. Park, M. V. Shapovalov, P. D. Renfrew, V. K. Mulligan, K. 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.

[6] F. Bao, M. Zhao, Z. Hao, P. Li, C. Li, and J. Zhu. Equivariant energy-guided sde for inverse molecular design. *arXiv preprint arXiv:2209.15408*, 2022.

[7] A. Berlaga, M. S. Jones, and A. L. Ferguson. Flowback-adjoint: Physics-aware and energy-guided conditional flow-matching for all-atom protein backmapping. *arXiv preprint arXiv:2508.03619*, 2025.

[8] L. Bogensperger, D. Narnhofer, A. Falk, K. Schindler, and T. Pock. Flowsdf: Flow matching for medical image segmentation using distance transforms. *International Journal of Computer Vision*, pages 1–13, 2025.

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

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

[11] A. Campbell, J. Yim, R. Barzilay, T. Rainforth, and T. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

[12] D. A. Case, H. M. Aktulga, K. Belfon, I. Ben-Shalom, S. R. Brozell, D. S. Cerutti, T. E. Cheatham III, V. W. D. Cruzeiro, T. A. Darden, R. E. Duke, et al. *Amber 2021*. University of California, San Francisco, 2021.

[13] R. T. Chen and Y. Lipman. Riemannian flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

[14] A. E. Chu, J. Kim, L. Cheng, G. El Nesr, M. Xu, R. W. Shuai, and P.-S. Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences*, 121(27): e2311500121, 2024.

[15] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. Wicky, A. Courbet, R. J. de Haas, N. Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[16] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, et al. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology*, 13(7):e1005659, 2017.

[17] N. K. Fox, S. E. Brenner, and J.-M. Chandonia. Scope: Structural classification of proteins—extended, integrating scop and astral data and classification of new structures. *Nucleic acids research*, 42(D1):D304–D309, 2014.

[18] I. Gat, T. Remez, N. Shaul, F. Kreuk, R. T. Chen, G. Synnaeve, Y. Adi, and Y. Lipman. Discrete flow matching. *arXiv preprint arXiv:2407.15595*, 2024.

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

[20] A. Herbert and M. Sternberg. Maxcluster: a tool for protein structure comparison and clustering, 2008.

[21] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[22] V. Hu, D. Wu, Y. Asano, P. Mettes, B. Fernando, B. Ommer, and C. Snoek. Flow matching for conditional text generation in a few sampling steps. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 380–392, 2024.

[23] C. Hua, Y. Liu, D. Zhang, O. Zhang, S. Luan, K. K. Yang, G. Wolf, D. Precup, and S. Zheng. Enzymeflow: Generating reaction-specific enzyme catalytic pockets through flow matching and co-evolutionary dynamics. *arXiv preprint arXiv:2410.00327*, 2024.

[24] G. Huguet, J. Vuckovic, K. Fatras, E. Thibodeau-Laufer, P. Lemos, R. Islam, C.-H. Liu, J. Rector-Brooks, T. Akhound-Sadegh, M. Bronstein, et al. Sequence-augmented se (3)-flow matching for conditional protein backbone generation. *arXiv preprint arXiv:2405.20313*, 2024.

[25] Y. Jiang, X. Li, Y. Zhang, J. Han, Y. Xu, A. Pandit, Z. Zhang, M. Wang, M. Wang, C. Liu, et al. Posex: Ai defeats physics approaches on protein-ligand cross docking. *arXiv preprint arXiv:2505.01700*, 2025.

[26] B. Jing, B. Berger, and T. Jaakkola. Alphafold meets flow matching for generating protein ensembles. *arXiv preprint arXiv:2402.04845*, 2024.

[27] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[28] L. Klein and F. Noé. Transferable boltzmann generators. *arXiv preprint arXiv:2406.14426*, 2024.

[29] X. Kong, W. Huang, and Y. Liu. Conditional antibody design as 3d equivariant graph translation. In *The Eleventh International Conference on Learning Representations*, 2023a.

[30] X. Kong, W. Huang, and Y. Liu. End-to-end full-atom antibody design. In *International Conference on Machine Learning*, pages 17409–17429. PMLR, 2023b.

[31] X. Kong, W. Huang, and Y. Liu. Full-atom peptide design with geometric latent diffusion. *arXiv preprint arXiv:2402.13555*, 2024.

[32] X. Kong, Z. Zhang, Z. Zhang, R. Jiao, J. Ma, W. Huang, K. Liu, and Y. Liu. Unimomo: Unified generative modeling of 3d molecules for de novo binder design. *arXiv preprint arXiv:2503.19300*, 2025.

[33] P. Kulytė, F. Vargas, S. V. Mathis, Y. G. Wang, J. M. Hernández-Lobato, and P. Liò. Improving antibody design with force-guided sampling in diffusion models. *arXiv preprint arXiv:2406.05832*, 2024.

[34] H. Lai, L. Wang, R. Qian, J. Huang, P. Zhou, G. Ye, F. Wu, F. Wu, X. Zeng, and W. Liu. Interformer: an interaction-aware model for protein-ligand docking and affinity prediction. *Nature Communications*, 15(1):10223, 2024.

[35] G. Li, X. Zhao, F. Wu, and S. Laue. Joint design of protein surface and backbone using a diffusion bridge model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[36] J. Li, T. Chen, S. Luo, C. Cheng, J. Guan, R. Guo, S. Wang, G. Liu, J. Peng, and J. Ma. Hotspot-driven peptide design via multi-fragment autoregressive extension. *arXiv preprint arXiv:2411.18463*, 2024.

[37] J. Li, C. Cheng, Z. Wu, R. Guo, S. Luo, Z. Ren, J. Peng, and J. Ma. Full-atom peptide design based on multi-modal flow matching. *arXiv preprint arXiv:2406.00735*, 2024.

[38] S. Li, M. Li, Y. Wang, X. He, N. Zheng, J. Zhang, and P.-A. Heng. Improving alphaflow for efficient protein ensembles generation. *arXiv preprint arXiv:2407.12053*, 2024.

[39] H. Lin, O. Zhang, H. Zhao, D. Jiang, L. Wu, Z. Liu, Y. Huang, and S. Z. Li. Ppflow: Target-aware peptide design with torsional flow matching. *bioRxiv*, pages 2024–03, 2024.

[40] H. Lin, L. Wu, Y. Huang, Y. Liu, O. Zhang, Y. Zhou, R. Sun, and S. Z. Li. Geoab: Towards realistic antibody design and reliable affinity maturation. In *Forty-first International Conference on Machine Learning*, 2024b.

[41] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[42] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[43] S. L. Lisanza, J. M. Gershon, S. Tipps, L. Arnoldt, S. Hendel, J. N. Sims, X. Li, and D. Baker. Joint generation of protein sequence and structure with rosettafold sequence space diffusion. *bioRxiv*, pages 2023–05, 2023.

[44] J. Liu, S. Li, C. Shi, Z. Yang, and J. Tang. Design of ligand-binding proteins with atomic flow matching. *arXiv preprint arXiv:2409.12080*, 2024.

[45] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[46] C. Lu, H. Chen, J. Chen, H. Su, C. Li, and J. Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning*, pages 22825–22855. PMLR, 2023.

[47] S. Luo, Y. Su, X. Peng, S. Wang, J. Peng, and J. Ma. Antigen-specific antibody design and optimization with diffusion-based generative models. *bioRxiv*, 2022.

[48] M. Masters, A. Mahmoud, and M. Lill. Fusiondock: Physics-informed diffusion model for molecular docking. In *ICML2023 CompBio Workshop*, 2023.

[49] D. Nori and W. Jin. Rnaflow: Rna structure & sequence design via inverse folding-based flow matching. *arXiv preprint arXiv:2405.18768*, 2024.

[50] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[51] X. Qi, D. Wang, Z. Shi, X. Liao, and H. Ma. Generative artificial intelligence for function-driven de novo enzyme design. *Synthetic Biology and Engineering*, 3(3):10015, 2025.

[52] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[53] D. Rong, H. Lu, W. Zheng, F. Zhang, S. Zheng, and N. Liu. Enerbridge-dpo: Energy-guided protein inverse folding with markov bridges and direct preference optimization. *arXiv preprint arXiv:2506.09496*, 2025.

[54] Y. Song and P. Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

[55] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[56] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[57] M. Steinegger and J. Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

[58] M. Sun, H. Cao, Z. Zhang, C. Wei, L. Wang, T. Jia, Z. Liu, T. Fu, X. Tang, Y. Choi, et al. Ribopo: Preference optimization for structure-and stability-aware rna design. *arXiv preprint arXiv:2510.21161*, 2025.

[59] X. Tang, H. Dai, E. Knight, F. Wu, Y. Li, T. Li, and M. Gerstein. A survey of generative ai for de novo drug design: new frontiers in molecule and protein generation. *Briefings in Bioinformatics*, 25(4), 2024.

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

[61] M. van Kempen, S. S. Kim, C. Tumescheit, M. Mirdita, C. L. Gilchrist, J. Söding, and M. Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pages 2022–02, 2022.

[62] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

[63] Y. Wang, L. Wang, Y. Shen, Y. Wang, H. Yuan, Y. Wu, and Q. Gu. Protein conformation generation via force-guided se (3) diffusion models. *arXiv preprint arXiv:2403.14088*, 2024.

[64] Z. Wang, J. Fan, R. Guo, T. Nguyen, H. Ji, and G. Liu. Proteinzero: Self-improving protein generation via online reinforcement learning. *arXiv preprint arXiv:2506.07459*, 2025.

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

[66] H. Wei, W. Wang, Z. Peng, and J. Yang. Q-biolip: A comprehensive resource for quaternary structure-based protein–ligand interactions. *Genomics, Proteomics & Bioinformatics*, 22(1), 2024.

[67] Z. Wen, J. He, H. Tao, and S.-Y. Huang. Pepbdb: a comprehensive structural database of biological peptide–protein interactions. *Bioinformatics*, 35(1):175–177, 2019.

[68] F. Wu and S. Z. Li. A hierarchical training paradigm for antibody structure-sequence co-design. *Advances in Neural Information Processing Systems*, 36:31140–31157, 2023.

[69] F. Wu and S. Z. Li. Surface-vqmae: Vector-quantized masked auto-encoders on molecular surfaces. In *Forty-first International Conference on Machine Learning*, 2024.

[70] F. Wu, D. Radev, and S. Z. Li. Molformer: Motif-based transformer on 3d heterogeneous molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5312–5320, 2023.

[71] F. Wu, Z. Zhou, S. Jin, X. Zeng, J. Leskovec, and J. Xu. Surface-based molecular design with multi-modal flow matching. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 3192–3203, 2025.

[72] F. Wu, S. Jin, X. Tang, M. Gerstein, X. Zeng, J. Leskovec, Y. Choi, and J. Xu. Surfdesign: Effective protein design on molecular surfaces. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 12396–12407, 2026.

[73] F. Wu, S. Jin, X. Tang, J. Xu, M. Gerstein, L. E. Li, and J. Zou. D-flow: Multi-modality flow matching for d-peptide design. *IEEE Journal of Biomedical and Health Informatics*, 2026.

[74] J. Wu, X. Kong, N. Sun, J. Wei, S. Shan, F. Feng, F. Wu, J. Peng, L. Zhang, Y. Liu, et al. Flowdesign: Improved design of antibody cdrs through flow matching and better prior distributions. *Cell Systems*, 16(6), 2025.

[75] J. Xu, Z. Gao, X. Zhou, J. Hu, X. Cheng, L. Song, G. Chen, P.-A. Heng, and J. Qiu. Protein inverse folding from structure feedback. *arXiv preprint arXiv:2506.03028*, 2025.

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

[77] J. Yim, A. Campbell, E. Mathieu, A. Y. Foong, M. Gastegger, J. Jiménez-Luna, S. Lewis, V. G. Satorras, B. S. Veeling, F. Noé, et al. Improved motif-scaffolding with se (3) flow matching. *ArXiv*, 2024.

[78] J. Zhang, Z. Liu, S. Bai, H. Cao, Y. Li, and L. Zhang. Efficient antibody structure refinement using energy-guided se (3) flow matching. *arXiv preprint arXiv:2410.16673*, 2024.

[79] W. Zhou, C. I. Sprague, V. Viliuga, M. Tadiello, A. Elofsson, and H. Azizpour. Energy-based flow matching for generating 3d molecular structure. *arXiv preprint arXiv:2508.18949*, 2025.

A Related Work

Flow-based Protein Generation. Flow matching [42, 4, 45] has become prevalent for molecular modeling. FrameFlow [76] introduced SE(3)-equivariant frame-based flows for fast backbone modeling, with subsequent extensions enabling improved motif-scaffolding and stability [77]. Others extend flows to multimodal or discrete protein representations. For instance, MultiFlow [11] learns flows over both coordinates and amino-acid types, while PepFlow [37], PPFlow [39], and FlowAB [78] incorporate side-chain generative flows for peptides, protein interfaces, and antibodies. Recent models further broaden the application landscape, including mirror-image D-peptide design [73], RNA-protein co-design (RNAFlow) [49], enzyme pocket generation (EnzymeFlow) [23], and partially latent flow representations for joint sequence–structure modeling (La-Proteina) [19]. Our work complements these efforts and is a post-training calibration layer on top of any pretrained flow, without modifying architectures or sampling-time solvers.

Energy-guided Molecular Modeling. A major challenge in molecular generation is to enforce physical plausibility while maintaining efficient sampling. Energy-guided SDE and flow approaches, including EEGSDE [6] and FlowBack-Adjoint [7], incorporate $\nabla E(x)$ into reverse-time dynamics or backpropagate through SDE/ODE trajectories to bias samples toward low-energy configurations. While effective, they typically modify sampling-time solvers, rely on auxiliary energy or force predictors, or require solving adjoint equations, which is computationally expensive and unstable for full-atom biomolecular systems. Other efforts apply force fields only during inference, such as ConfDiff [63], DiffForce [33], or alignment strategies inspired by reinforcement learning from human feedback (RLHF) [50, 62, 75, 64]. However, they fail to update the generative model itself and thus cannot propagate physical corrections into the learned dynamics. Tab. 5 positions FFC against these methods along the axes of energy signal, training-time use, and inference-time modification.

B Preliminary and Background

Probability Path and Flow. Let $\mathbb{P}(\mathcal{M})$ be the space of probability distributions on a manifold $\mathcal{M}$ with a Riemannian metric g. A probability path $\{q_t\}_{t\in[0,1]} : \mathcal{M} \to \mathbb{P}(\mathcal{M})$ interpolates between two distributions $q_0, q_1 \in \mathbb{P}(\mathcal{M})$. A flow on $\mathcal{M}$ is defined by a family of diffeomorphisms $\phi_t : \mathcal{M} \to \mathcal{M}$, induced by integrating a vector field $u \in \mathcal{U}$. $u(\mathbf{x}_t, t) \in \mathcal{T}_{\mathbf{x}_t}\mathcal{M}$ is the gradient vector of the path q_t on $\mathbf{x}$ at time t. For an initial point $\mathbf{x}_0$, the flow map $\phi_t : \mathcal{M} \to \mathcal{M}$ is defined by an ordinary differential equation (ODE) $\frac{\mathrm{d}\phi_t}{\mathrm{d}t}(\mathbf{x}_0) = u(\phi_t(\mathbf{x}_0), t)$. Given a source density q_0, $\phi_t(\mathbf{x}_t)$ induces a push-forward operation $q_t = [\phi_t]_{\#}q_0$. It reshapes the density of points $\mathbf{x}_0 \sim q_0$ to a more complicated one along $u(\mathbf{x}_t, t)$, and the change-of-variable operator $\#$ is defined by $[\phi_t]_{\#}q_0(\mathbf{x}_t) = q_0\left(\phi_t^{-1}(\mathbf{x}_t)\right) \det\left[\frac{\partial \phi_t^{-1}}{\partial \mathbf{x}_t}(\mathbf{x}_t)\right]$. The time-varying density q_t is characterized by the continuity equation as $\frac{\partial q_t(\mathbf{x}_t)}{\partial t} = -\mathrm{div}(u(\mathbf{x}_t, t)q_t(\mathbf{x}_t))$. Under these conditions, the vector field u is the probability flow for q_t, and q_t is the marginal probability path generated by u.

Riemannian Flow Matching. Flow matching [3] trains a continuous normalizing flow (CNF) by fitting a vector field $v_\theta \in \mathcal{U}$ with parameters θ to the target u that produces a probability path q_t. Its objective falls at the tangent space as $\mathcal{L}^\theta_{\text{RFM}} = \mathbb{E}_{t\sim\mathcal{U}[0,1],\mathbf{x}_t\sim q_t(\mathbf{x}_t)}\|v_\theta(\mathbf{x}_t, t) - u(\mathbf{x}_t, t)\|_g^2$, As u is intractable, an alternative is to construct conditional probability path $q_t(\mathbf{x}_t|\mathbf{x}_1)$ with conditional vector field $u(\mathbf{x}_t, t|\mathbf{x}_1)$, and the objective becomes $\mathcal{L}^\theta_{\text{CRFM}} = \mathbb{E}_{t\sim\mathcal{U}[0,1],\mathbf{x}_1\sim q_1,\mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_1)}\|v_\theta(\mathbf{x}_t, t) - u(\mathbf{x}_t, t|\mathbf{x}_1)\|_g^2$, which share the same gradients as $\mathcal{L}_{\text{RFM}}$ [42, 60, 13]. During inference, we can solve the ODE associated with the neural vector field v_θ to push $\mathbf{x}_0 \in \mathcal{M}$ from the source distribution q_0 to the data distribution q_1 in time.

C Theoretical Analysis and Proofs

C.1 Invariance of the Calibration Loss

Equivariance under spatial transformations such as 3D rotations is a fundamental property for generative models of molecular structures. Preserving $\mathrm{SO}(3)$-equivariance ensures that the output of

the model transforms consistently with the input, a critical requirement for robust physical modeling. In this section, we prove Theorem 2.2.

Problem Setup. Let $y \in \mathbb{R}^{3n}$ represent a full-atom molecular conformation and let $Q \in \mathrm{SO}(3)$ be a global rotation applied to all coordinates, so that Qy denotes the rotated configuration (acting blockwise on each atom). We make the following assumptions, which correspond to the hypotheses of Theorem 2.2. The velocity field and the reconstruction module are $\mathrm{SO}(3)$-*equivariant*:

$$v_\theta(Q\mathbf{x}_t, t) = Q\, v_\theta(\mathbf{x}_t, t), \qquad \zeta(Q \cdot z) = Q\, \zeta(z).$$

The energy is $\mathrm{SO}(3)$-*invariant*, $\mathcal{E}(Qy) = \mathcal{E}(y)$, from which the force field $\mathcal{F} = -\nabla\mathcal{E}$ is equivariant:

$$\mathcal{F}(Qy) = Q\, \mathcal{F}(y).$$

Note that invariance of $\mathcal{E}$ and equivariance of $\mathcal{F}$ are not the same property: the former is the scalar statement, the latter the induced statement on its gradient. We state both explicitly to avoid the ambiguity.

Two Auxiliary Facts. Two further ingredients of the terminal target preserve equivariance. *(i) Normalization.* The map $\mathrm{NormClip}$ of Eq. 5 rescales each atomic force by $\min(1, c/(\|\mathcal{F}_i\| + \varepsilon)) \,/\, \max_j \|\mathcal{F}_j\|$. Since $\|Q\mathcal{F}_i\| = \|\mathcal{F}_i\|$ for $Q \in \mathrm{SO}(3)$, every scalar factor is rotation-invariant, and therefore $\mathrm{NormClip}[\mathcal{F}(Qy)] = Q\,\mathrm{NormClip}[\mathcal{F}(y)]$. *(ii) Relaxation.* Whenever $\mathcal{E}$ is rotation-invariant, the energy minimization Relax follows the gradient of an invariant energy from an initial structure and therefore commutes with rotation, $\mathrm{Relax}(Qy) = Q\,\mathrm{Relax}(y)$, up to the numerical tolerance of the minimizer. This is the hypothesis on Relax stated in Theorem 2.2; Remark C.1 discusses when it holds exactly. Consequently, the full target y_{target} of Eq. 3 is equivariant.

Remark C.1 (Exactness of the invariance hypothesis under PME)**.** *Particle Mesh Ewald, used for long-range electrostatics in App. E.2, requires a periodic simulation box. A global rotation of the molecule inside a fixed box is not a symmetry of the periodic system, so with PME the energy is invariant only up to the box convention, and the hypotheses of Theorem 2.2 hold exactly only for a non-periodic (e.g.* NoCutoff *or implicit-solvent) force evaluation. In practice, the deviation is small for a solute well inside the box, but the theorem should be read as exact in the non-periodic case and approximate otherwise.*

Invariance of the Loss. Applying a rotation $Q \in \mathrm{SO}(3)$ to the entire molecular structure, and writing $\hat{y}_1 = \zeta(\hat{z}_1)$, $\bar{y}_1 = \zeta(\bar{z}_1)$ and $\tilde{y}_1 = \mathrm{Relax}(\hat{y}_1)$ for brevity:

$$\begin{aligned}
\mathcal{L}_{\mathrm{cal}}(Q\hat{y}_1) &= \|Q\hat{y}_1 - (Q\bar{y}_1 + \lambda_{\mathcal{F}}\mathrm{NormClip}[\mathcal{F}(Q\tilde{y}_1)])\|^2 \\
&= \|Q\hat{y}_1 - (Q\bar{y}_1 + \lambda_{\mathcal{F}}\, Q\,\mathrm{NormClip}[\mathcal{F}(\tilde{y}_1)])\|^2 \\
&= \|Q\,[\hat{y}_1 - (\bar{y}_1 + \lambda_{\mathcal{F}}\mathrm{NormClip}[\mathcal{F}(\tilde{y}_1)])]\|^2 \\
&= [\hat{y}_1 - (\bar{y}_1 + \lambda_{\mathcal{F}}\mathrm{NormClip}[\mathcal{F}(\tilde{y}_1)])]^\top\, Q^\top Q\, [\hat{y}_1 - (\bar{y}_1 + \lambda_{\mathcal{F}}\mathrm{NormClip}[\mathcal{F}(\tilde{y}_1)])] \\
&= \|\hat{y}_1 - (\bar{y}_1 + \lambda_{\mathcal{F}}\mathrm{NormClip}[\mathcal{F}(\tilde{y}_1)])\|^2 \;=\; \mathcal{L}_{\mathrm{cal}}(\hat{y}_1).
\end{aligned}$$

Since the inner difference vector transforms linearly under Q, and $Q^\top Q = I$, the norm is invariant under $\mathrm{SO}(3)$ rotations. The same argument applies verbatim with $\|\cdot\|_g$ in place of the Euclidean norm whenever g is itself rotation-invariant. Thus, the loss is rotation-invariant. □

Discussion. In contrast to earlier calibration objectives that propagate physical gradients along an entire trajectory (and thus depend on intermediate directional derivatives), our one-stage loss relies only on the force field at the terminal state. This design eliminates equivariance-breaking terms associated with time-dependent vector fields or reverse-time integration. As a result, the calibration loss remains invariant under any global rigid-body rotation of the molecular structure, which keeps model training consistent with physical symmetry.

C.2 Proof Sketch for Proposition 2.4

Recall the Hilbert space

$$\mathcal{H} = L^2\big([0,1] \times \mathcal{M}, q_t^\star; \mathbb{R}^d\big)$$

equipped with the inner product

$$\langle v, w\rangle_{\mathcal{H}} = \int_0^1 \mathbb{E}_{\mathbf{x}_t\sim q_t^\star}\left[\langle v(\mathbf{x}_t,t), w(\mathbf{x}_t,t)\rangle_g\right] \mathrm{d}t.$$

In this space, the population calibration functional of Eq. 12 can be written as

$$\mathcal{L}_{\mathrm{cal}}(v) = \tfrac{1}{2}\int_0^1 \mathbb{E}_{\mathbf{x}_t\sim q_t^\star}\left\|v(\mathbf{x}_t,t) - u^\star(\mathbf{x}_t,t)\right\|_g^2 \mathrm{d}t = \tfrac{1}{2}\|v-u^\star\|_{\mathcal{H}}^2 + \text{const},$$

so it is a shifted squared norm around $u^\star$.

Proof sketch of Proposition 2.4. From the expression above, we immediately obtain $\nabla\mathcal{L}_{\mathrm{cal}}(v) = v - u^\star$ in $\mathcal{H}$, so the Hessian of $\mathcal{L}_{\mathrm{cal}}$ is the identity operator on $\mathcal{H}$. Hence $\mathcal{L}_{\mathrm{cal}}$ is 1-strongly convex and its gradient is 1-Lipschitz with respect to the $\mathcal{H}$-norm. For a strongly convex functional with L-Lipschitz gradient, exact gradient descent with step size $0 < \eta < 2/L$ converges linearly to the unique minimizer; specializing to $L = 1$ and minimizer $u^\star$ gives the claim. □

We reiterate that this statement concerns the idealized population objective of Eq. 12, which is *not* the objective optimized in Alg. 1: the latter applies supervision only at $t = 1$, uses a finite-capacity parameterization, and replaces $q_t^\star$ with model samples. Moreover the Hilbert space itself is defined through $q_t^\star$, which depends on the particular $u^\star$ fixed by Assumption 2.3. Prop. 2.4 should therefore be read as establishing well-posedness of the regression view, not as a convergence guarantee for the trained network.

D Summarization of Energy-guided Molecular Generation

Tab. 5 summarizes representative energy-guided molecular generation methods across different generative paradigms. Several prior methods do use the energy signal during training; what distinguishes FFC is the combination of a *direct, non-surrogate* force signal with an *unmodified sampler* and no backpropagation through the generative trajectory. To our knowledge it is the only entry in the table with all three properties.

Table 5: Taxonomy of energy-guided molecular generation methods. The **Training** column indicates whether the energy signal is used to *update model parameters* ("Yes") or is applied only at sampling time without changing the model ("No"); the **Inference** column indicates whether the sampler itself is modified relative to the unguided base model. FFC is the only entry that is Yes/Standard with a direct force signal, i.e. it pays the cost of the physical signal during training and none at inference.

Method	Paradigm	Energy Signal	Training	Inference	Primary Limitation
EEGSDE [6]	SDE	∇E in dynamics	No	Modified	Unstable integration
ConfDiff [63]	Diffusion	Energy predictor	Yes	Standard	Approximation bias
DiffForce [33]	Diffusion	Direct forces	No	Modified	High inference cost
EnerBridge [53]	Bridge	RLHF-style	Yes	Standard	Requires preference pairs
FlowBack [7]	Flow	Adjoint backprop	No	Modified	Expensive adjoint solve
IDFlow [79]	Flow	Implicit energy	Yes	Modified	Requires iterative refinement
FFC (ours)	**Flow**	**Direct forces**	**Yes**	**Standard**	**Relies on force field accuracy**

E Experimental Details

E.1 Task-specific Implementation Details

Monomer Backbone Generation. We implement FrameFlow using its official checkpoint available at `https://github.com/microsoft/protein-frame-flow`. To adapt the model to our dataset, we perform one day of post-training on two NVIDIA H100-80GB GPUs. To improve computational efficiency, we adopt a length-aware batching strategy that groups proteins of the same sequence length, thereby avoiding unnecessary padding. Training is conducted using the Adam optimizer with a learning rate of 1e-4, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, consistent with the original configuration. The EMA teacher uses $\mu = 0.999$ and is initialized from the pretrained checkpoint. The terminal sample $\hat{z}_1$ used by $\mathcal{L}_{\mathrm{cal}}$ is produced by a K-step student rollout, of which only the last step carries gradients.

Peptide De Novo Design. We fine-tune PepFlow using the official checkpoint provided at `https://github.com/Ced3-han/PepFlowww`. Specifically, we select the *w/Bb+Seq+Ang* variant, which incorporates CRFM loss, backbone reconstruction loss, and torsion angle loss to improve both global and local peptide conformations. We fine-tune the model for 5K iterations using a single NVIDIA H100-80GB GPU. Optimization is performed using Adam with a learning rate of 1e-5, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We employ a batch size of 32 and apply gradient clipping to mitigate the risk of exploding gradients, ensuring stable training dynamics. The consistency and calibration terms are optimized jointly from the first iteration with $\beta = 1$ in Eq. 9; there is no warm-up phase in which only the consistency loss is active. The terminal sample is produced by a K-step rollout, only the last step of which is differentiated. The binding pocket is defined as the residues in the target protein that have heavy atoms within a 10 Å radius of any heavy atom in the peptide. PepMerge provides 8,207 non-homologous peptide–protein complexes for training and validation and a held-out evaluation set of 158 complexes grouped into 10 clusters by mmseqs2 [57]; all numbers reported in Tab. 2 and Tab. 3 are computed on the 158-complex evaluation set.

As for comparison, RFdiffusion [65] uses pretrained weights from RoseTTAFold, and sequences are subsequently recovered using ProteinMPNN. ProteinGenerator [43] is an extension of RFdiffusion that enables joint generation of sequences and structures. PepGLAD [31] utilizes equivariant latent diffusion networks to design full-atom peptides with high fidelity to geometric and physical constraints. PepHAR [36] is a hot-spot-driven autoregressive generative model. DiffAb [47] and PPFlow [39] are applied to the peptide setting using their public implementations.

Antibody Design. We conduct UniMoMo using the official checkpoint at `https://github.com/kxz18/UniMoMo`. We use the version trained on all binder types, including small molecules, peptides, and antibodies. Following its instruction, we fine-tuned UniMoMo on 8 GPUs with 80G memory each for around 2 days.

For baselines, we benchmark against state-of-the-art antibody-design models: **MEAN** [29], which employs a multi-channel equivariant GNN for iterative CDR structure refinement; **dyMEAN** [30], which extends MEAN to full-atom modeling with flexible antigen poses; **DiffAb** [47], a diffusion-based model jointly generating amino-acid types, coordinates, and orientations; and **GeoAB** [40], which incorporates a heterogeneous residue encoder and energy-informed geometric constraints. These baselines cover the dominant paradigms in sequence–structure co-design for antibodies, and all are evaluated under the conditional CDR-H3 redesign protocol described in Sec. 3.

E.2 Force Field Simulation Details

Force Field Computation. To integrate physical consistency into our models, we use OpenMM [16] to compute atomic forces and potential energy. Prior to simulation, we preprocess protein structures using PDBFixer (`https://github.com/openmm/pdbfixer`) to correct issues such as missing atoms and residues, ensuring structurally complete and simulation-ready inputs. Force field parameters are loaded from amber14-all.xml and amber14/tip3pfb.xml, representing the Amber14 force field [12] and the TIP3P-FB water model, respectively.

Nonbonded interactions use Particle Mesh Ewald (PME) for long-range electrostatics (nonbondedMethod=PME) with a 1 nm real-space cutoff, and hydrogen-involving bonds are constrained (HBonds). We emphasize that *no molecular dynamics is propagated*: the pipeline performs a local energy minimization followed by a single force evaluation. An integrator object is instantiated only because the OpenMM Simulation API requires one, and its parameters never affect the reported forces. See Remark C.1 for the consequence of the periodic box for the exactness of Theorem 2.2.

OpenMM returns forces in $\mathrm{kJ} \cdot \mathrm{mol}^{-1} \cdot \mathrm{nm}^{-1}$, and structural imperfections (*e.g.*, steric clashes or bond strain) can produce very large magnitudes. To prevent numerical instability in downstream learning, we clip the per-atom magnitudes and divide by the maximum force observed in that structure (Eq. 5). The resulting supervision signal $\widetilde{\mathcal{F}}$ is *dimensionless*, with per-atom magnitudes in $[0, 1]$; the physical units are absorbed into $\lambda_{\mathcal{F}}$, which therefore has units of length in the Cartesian target of Eq. 3.

Relaxation. The Relax operator of Eq. 2 performs energy minimization to obtain a locally relaxed configuration. This minimization is used *only* to define the force-correction target and is treated

as a non-differentiable preprocessing step; gradients are not propagated through it. The tolerance criterion halts minimization once the maximum force on any atom falls below $1.0\,\mathrm{kJ}\cdot\mathrm{mol}^{-1}\cdot\mathrm{nm}^{-1}$, effectively resolving severe steric clashes, with a cap of 500 iterations to bound the cost in strained regions. We observe that, on average, 100–200 iterations suffice, keeping the minimization step efficient.

Time Complexity Analysis. We benchmarked our force-only OPENMM pipeline (preprocessing $\rightarrow$ minimization $\rightarrow$ force evaluation) on an NVIDIA H100 GPU and observed an average per-structure runtime of approximately 1.5 seconds. This cost is paid once per sampled structure per training iteration, so the total OPENMM overhead of a calibration run is $1.5\,\mathrm{s}\times N\times B$ for N iterations at batch size B: for the peptide setting ($N=5\mathrm{K}$, $B=32$) this is $\approx 1.6\times 10^5$ evaluations. All of this cost is incurred during calibration; inference is unaffected, as Tab. 3 shows. In practice, it can be reduced by (1) reusing a single OPENMM Context/System and swapping in new coordinates, (2) lowering the minimization tolerance or capping maxiterations (*e.g.*, to 100 steps), (3) enabling mixed-precision mode on the CUDA platform, and (4) refreshing the force target only every few iterations rather than at every step.

E.3 Adaptation of Comparable Methods to Flow Models

To enable fair comparisons, we adapt key ideas from prior energy-guided methods to the flow-matching framework. These adaptations are the baselines of Tab. 3 and serve to isolate the contribution of our force field calibration approach: all share the PepFlow architecture, the same OPENMM force field, and the same evaluation pipeline, differing only in how the physical signal enters.

Energy-Conditioned Flow (ConfDiff-style). Following ConfDiff [63], we train an auxiliary energy predictor network $E_\psi(x_t,t)$ that estimates the potential energy at each timestep. The flow model is then conditioned on the predicted energy:

$$v_\theta(x_t,t,E_\psi(x_t,t)), \tag{14}$$

where E_ψ is jointly trained with v_θ using a regression loss on ground-truth energies computed via OPENMM. This baseline tests whether explicit energy conditioning can match direct force field calibration.

Inference-Time Force Guidance (DiffForce-style). Inspired by DiffForce [33], we apply force field guidance only during inference without modifying the model parameters. At each ODE integration step, the velocity is augmented with normalized forces:

$$v_{\text{guided}}(x_t,t)=v_\theta(x_t,t)+\lambda_{\text{infer}}\cdot\widetilde{\mathcal{F}}(x_t), \tag{15}$$

where λ_{infer} is tuned on a validation set. This baseline isolates the benefit of training-time calibration versus post-hoc guidance, and accounts for the $30\times$ inference overhead reported in Tab. 3, since the force field must be evaluated at every solver step.

Idempotent Flow (IDFlow-style). Inspired by IDFlow [79], we train the flow model with a reconstruction-based energy objective that encourages idempotent predictions. Specifically, we add an auxiliary loss that minimizes the reconstruction error after self-refinement:

$$\mathcal{L}_{\text{idem}}=\mathbb{E}_{x_t,\hat{x}_1}\left[\|f_\theta(f_\theta(x_t,t),t)-x_1\|^2\right], \tag{16}$$

where $\hat{x}_1=f_\theta(x_t,t)$ is the initial prediction and the network is trained to refine its own outputs. This baseline assesses whether iterative refinement alone can match the benefits of direct force-field supervision.

Consistency-Only Flow. To assess the contribution of physical calibration independent of few-step sampling, we train a model with only the consistency losses $\mathcal{L}_{\text{cfc}}$, omitting the force field calibration term $\mathcal{L}_{\text{cal}}$ (i.e. $\beta=0$ in Eq. 9). This ablation demonstrates whether improved sampling efficiency alone accounts for our gains, or whether physical grounding is essential. Tab. 3 shows that consistency alone recovers roughly 40% of the validity gain, confirming that the two components contribute separately.

F Discrete Flow on Amino Acid Sequences

Introduction. Although flow models excel at generating continuous spatial signals such as images [21, 52], their application to discrete sequential data has proven challenging. Two primary strategies have been proposed to address this limitation: one embeds discrete data into a continuous space and applies traditional continuous flow or diffusion frameworks [37], while the other directly operates within categorical state spaces by designing transformations over discrete states [11]. In this work, we explore both approaches and find that discrete flow demonstrates a significant advantage over the continuous embedding approach for discrete data.

Building on the foundations of prior works [11, 18, 32, 35], we formulate flow in the discrete setting using a Continuous-Time discrete Markov Chain (CTMC) paradigm. Here, the categorical variable a_t^i transitions between states in the amino acid type space $\mathcal{A}$ over continuous time $t \in [0, 1]$. The evolution of probabilities is governed by the *probability velocity* u_t^{aa}, analogous to the velocity field in continuous normalizing flows (CNFs). For each category in $\mathcal{A}(\|\mathcal{A}\| = 20)$, u_t^{aa} quantifies the rate of change in the probability distribution of a_t^i, and the state is updated iteratively using an Euler step:

$$a_{t+\frac{1}{N}}^i \sim \delta_{a_t^i}(\cdot) + \frac{1}{N} u_t^{\text{aa}}\left(a^i, \cdot\right)$$

As $N \to \infty$, u_t^{aa} must satisfy the constraints:

$$\sum_{a^i \in \mathcal{A}} u_t^{\text{aa}}\left(a^i, z\right) = 0 \quad \text{and} \quad u_t^{\text{aa}}\left(a^i, z\right) \geq 0 \quad \text{for } \forall i \in [n], a^i \neq z$$

This allows u_t^{aa} to be expressed as a marginalization of its conditional counterpart $u_t^{\text{aa}}\left(a^i, z \mid a_0^i, a_1^i\right)$, enabling the generation of a smooth probability path $p_t\left(a^i\right)$. The marginal velocity can thus be written as:

$$u_t^{\text{aa}}\left(a^i, z\right) = \sum_{a_0^i, a_1^i \in \mathcal{A}} u_t^{\text{aa}}\left(a^i, z \mid a_0^i, a_1^i\right) p_t\left(a_0^i, a_1^i \mid z\right).$$

To parameterize conditional probability paths, we represent $p_t\left(a^i \mid a_0^i, a_1^i\right)$ as a convex combination of m conditional probabilities:

$$p_t\left(a^i \mid a_0^i, a_1^i\right) = \sum_{j=1}^{m} \kappa_t^j p^j\left(a^i \mid a_0^i, a_1^i\right)$$

where $\kappa_t^j \geq 0$ and $\sum_j \kappa_t^j = 1$. A straightforward yet effective example is the linear interpolation:

$$p_t\left(a^i \mid a_0^i, a_1^i\right) = \kappa_t \delta_{a_1^i}\left(a^i\right) + \left(1 - \kappa_t\right) \delta_{a_0^i}\left(a^i\right),$$

where κ_t transitions monotonically from 0 to 1 over t. This implies $p_0\left(a^i \mid a_0^i, a_1^i\right) = \delta_{a_0^i}\left(a^i\right)$ and $p_1\left(a^i \mid a_0^i, a_1^i\right) = \delta_{a_1^i}\left(a^i\right)$. The resulting marginal probability velocity is:

$$u_t^{\text{aa}}\left(a^i, z\right) = \frac{\dot{\kappa}_t}{1 - \kappa_t}\left[p_{1|t}\left(a^i \mid z\right) - \delta_z\left(a^i\right)\right]$$

where $p_{1|t}\left(a^i \mid z\right) = \sum_{a_0^i, a_1^i} \delta_{a_1^i}\left(a^i\right) p_t\left(a_0^i, a_1^i \mid z\right)$ serves as the probability denoiser. The model v_θ^{aa} is trained to approximate this denoiser $p_{1|t}$ by minimizing the cross-entropy (CE) loss:

$$\mathcal{L}_{\text{aa}}^\theta = -\,\mathbb{E}_{i \in [n], t \sim \mathcal{U}(0,1)}\left[\log p_{1|t}\left(a_1^i \mid a_t^i\right)\right]$$

Finally, instead of interpolating towards a_1^i from a uniform categorical prior, we adopt a masked state M for enhanced flexibility [11]. The conditional path becomes:

$$p_{t|1}^{\text{mask}}\left(a_t^i \mid a_0^i, a_1^i\right) = \text{Cat}\left(t\delta_{a_1^i}(\cdot) + (1-t)\delta_{\text{M}}(\cdot)\right)$$

Interaction with FFC. While the reconstruction module is fully differentiable with respect to continuous structural variables (positions, orientations, and torsion angles), amino-acid identities are treated as discrete and non-differentiable during force-field calibration. Consequently, no force-field gradients are backpropagated to the residue-identity logits. Instead, physical supervision is applied

exclusively to continuous modalities, and its influence on sequence generation is propagated indirectly through shared representations and cross-modal consistency with the teacher flow. Empirically, we find that *the primary gains from FFC stem from improved geometric fidelity, reduced steric clashes, improved packing, and lower physical energies*, while sequence-level metrics exhibit smaller but consistent improvements. These sequence gains reflect enhanced geometry–sequence compatibility induced by calibrated structures, rather than direct optimization of residue identities by the force field alone. This is the reason the AAR improvements in Tab. 4 are modest relative to the RMSD improvements, and we do not claim otherwise.

G Differentiable Reconstruction Module

In order to integrate physical supervision, such as force fields, into multi-modality protein generation, it is essential to reconstruct full-atom Cartesian coordinates from latent representations. We define a reconstruction module $\zeta(\mathbf{x}^i_{\mathrm{C}_\alpha}, O^i, \boldsymbol{\chi}^i, a^i)$ that maps the predicted latent variables for each residue i, its C_α position, backbone orientation, torsion angles, and amino acid type, to the 3D coordinates of all heavy atoms. This section describes the procedure and proves its differentiability.

G.1 Structure of the Reconstruction Module

The reconstruction module consists of three key components:

Backbone Atom Placement. Each residue's local frame is constructed with C_α at the origin and orientation $O^i \in \mathrm{SO}(3)$ encoding the local backbone geometry. Using pre-tabulated ideal coordinates (e.g., from PDB or rotamer libraries), the positions of the N, C, and O atoms in this local frame are defined as fixed vectors. These local coordinates are then transformed into global coordinates using:

$$y^j_{\text{global}} = O^i\, y^j_{\text{local}} + \mathbf{x}^i_{\mathrm{C}_\alpha}, \quad \text{for all } j \in \{\mathrm{N}, \mathrm{C}, \mathrm{O}\}.$$

Side-Chain Reconstruction via Torsion Angles. Each side chain is modeled as a tree of atoms with fixed bond lengths and bond angles, and rotatable bonds determined by four torsion angles $\boldsymbol{\chi}^i \in [0, 2\pi)^4$. Starting from a backbone anchor atom (typically C_β), internal coordinates are recursively propagated using torsion-angle-based rotations. For each torsion χ_k, we compute a rotation matrix R_k around the parent bond axis using Rodrigues' formula:

$$R_k = I + \sin(\chi_k)[\hat{u}_k]_\times + (1 - \cos(\chi_k))[\hat{u}_k]^2_\times,$$

where $\hat{u}_k$ is the unit vector along the bond axis and $[\cdot]_\times$ denotes the skew-symmetric matrix. Each downstream atom's position is updated by applying R_k about its parent axis in the local frame. The resulting coordinates are then transformed globally using the same O^i and $\mathbf{x}^i_{\mathrm{C}_\alpha}$.

Residue-Specific Geometry via Amino Acid Type. The amino acid identity $a^i \in \{1, \ldots, 20\}$ is used to index a residue-specific template of side-chain structure. This includes a set of atom names, bond topology, and canonical internal coordinates. During training, a^i is treated as fixed (i.e., no gradients flow through it). This allows us to use discrete conditioning without compromising the reconstruction process's differentiability.

G.2 Differentiability of ζ

All operations in ζ are composed of matrix multiplications, trigonometric functions (e.g., $\sin$, $\cos$), and vector additions—each of which is differentiable almost everywhere. The use of rotation matrices derived from torsion angles ensures that gradients can flow from Cartesian coordinates back to the latent torsional and backbone variables. The amino acid index a^i is fixed during backpropagation (e.g., via stop-gradient), so it introduces no discontinuities in the computational graph.

Specifically, we apply a stop-gradient operation to amino-acid identities during reconstruction. This design choice reflects the fact that classical force fields are defined over atomic coordinates rather than discrete residue types. Preventing gradients from flowing through amino-acid identities avoids ill-posed supervision while preserving stable and physically meaningful calibration of continuous structural variables.

Consequently, the composite function ζ is fully differentiable with respect to $(\mathbf{x}^i_{C_\alpha}, O^i, \boldsymbol{\chi}^i)$, and thus allows end-to-end training of vector field models v_θ under physical constraints. Because ζ is built from rigid transformations applied to residue-local templates, it also satisfies $\zeta(Q \cdot z) = Q\,\zeta(z)$ for any $Q \in \mathrm{SO}(3)$, the equivariance property assumed in Theorem 2.2. In practice, we implement ζ using PyTorch, ensuring compatibility with modern auto-differentiation frameworks. This design allows ζ to be seamlessly integrated into both the consistency training and FFC terms without altering model structure or sampling procedure.

Listing 1: Differentiable residue reconstruction function

```
def reconstruct_residue(x_ca, O, chi, aa_type):
    """
    Inputs:
        x_ca:    (3,) C-alpha coordinate in global frame
        O:       (3, 3) rotation matrix for residue local frame
        chi:     (4,) torsion angles (radians)
        aa_type: int in [0, 19] amino acid type
    Returns:
        coords:  (n_atoms, 3) full-atom 3D coordinates
    """

    # --- Load atom templates for this amino acid ---
    atom_names, ideal_coords, chi_atoms = residue_library[aa_type]
    # ideal_coords: (n_atoms, 3) local frame coordinates
    # chi_atoms: list of 4 tuples (a, b, c, d) defining torsions

    # --- Apply torsion rotations ---
    coords_local = ideal_coords.clone()
    for k, chi_k in enumerate(chi):
        if k >= len(chi_atoms): break
        a, b, c, d = chi_atoms[k]
        axis = coords_local[c] - coords_local[b]
        axis = axis / (axis.norm() + eps)

        downstream = get_downstream_atoms(coords_local, d, parent=b)
        R = axis_angle_rotation(axis, angle=chi_k)
        for idx in downstream:
            vec = coords_local[idx] - coords_local[b]
            coords_local[idx] = coords_local[b] + R @ vec

    # --- Apply global frame transformation ---
    coords_global = (O @ coords_local.T).T + x_ca

    return coords_global
```

H Additional Experiments

Ablation on the Force Weight $\lambda_{\mathcal{F}}$. We investigate the effect of key hyperparameters on the performance of our calibrated flow models, focusing particularly on the force control parameter $\lambda_{\mathcal{F}}$, which modulates the influence of the force field during training. Since the force magnitudes are normalized (Eq. 5), $\lambda_{\mathcal{F}}$ is shared across molecular systems, facilitating direct comparison across tasks.

As shown in Fig. 3, increasing $\lambda_{\mathcal{F}}$ from 0.0 to 1.0 improves the valid ratio in the peptide design task, peaking at 52.66%. Beyond this point, performance declines sharply, reaching just 12.55% at $\lambda_{\mathcal{F}} = 10.0$. RMSD remains stable around 2.07 Å at low force weights but increases substantially to 5.68 Å when $\lambda_{\mathcal{F}}$ exceeds 1.0. This trend suggests that moderate force supervision improves structural quality, while excessive regularization destabilizes generation.

In the backbone generation task, designability remains flat near 0.78 for small $\lambda_{\mathcal{F}}$, but reaches a maximum of 0.80 at $\lambda_{\mathcal{F}} = 0.1$ before decreasing sharply. Meanwhile, novelty remains consistent for small force weights but increases significantly at higher values, peaking at 0.80. This behavior illustrates the quality–diversity trade-off: force strength trades structural accuracy against diversity and novelty.

Based on these results, we identify $\lambda_{\mathcal{F}} = 1.0$ as optimal for peptide design and 0.1 for backbone generation. These settings are further fine-tuned using random search in $[0.5, 2.0]$ during final evaluations. The sharpness of the optimum is itself a limitation of the method, as noted in Sec. 4.

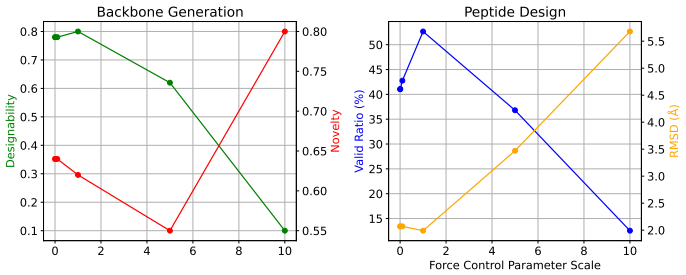


Figure 3: Performance across force control strengths for peptide design and backbone generation.

Consistency-based Flow Tuning. To improve sampling efficiency and structural stability, we explore different strategies for enforcing temporal consistency in the learned vector field $v_\theta(\mathbf{x}_t, t)$. Specifically, we consider two criteria: (i) enforcing time-consistency in the velocity field against the EMA teacher, and (ii) ensuring that the initial vector field aligns with the geodesic displacement between source and target, $v_\theta(\mathbf{x}_0, 0) \approx \log_{\mathbf{x}_0}(\mathbf{x}_1)$.

Prior work in image and video domains has proposed progressive consistency training techniques, where velocity predictions at different timesteps are aligned over time. However, such strategies often rely on large-scale, pretrained diffusion models and massive datasets, which are not available in protein modeling. In our setting, with limited training data, these progressive schedules tend to be unstable or ineffective.

Instead, we directly optimize simplified and data-efficient consistency losses:

(1) Initialization Consistency. We enforce that the velocity field at $t = 0$ matches the geodesic displacement vector between noisy and clean coordinates:

$$\mathcal{L}_{\text{Init}} = \mathbf{L}\left(v_\theta(\mathbf{x}_0, 0), \log_{\mathbf{x}_0}(\mathbf{x}_1)\right).$$

(2) Time-step Displacement Matching. We replace the teacher–student velocity consistency objective in Eq. 7 with a simpler displacement-based supervision that directly regresses the student velocity toward the total geodesic displacement between endpoints. Specifically, we train v_θ to predict the same global displacement at intermediate timesteps t and r:

$$\mathcal{L}_{\text{t-time}} = \mathbf{L}\big(v_\theta(\mathbf{x}_t, t), \log_{\mathbf{x}_0}(\mathbf{x}_1)\big), \quad \mathcal{L}_{\text{r-time}} = \mathbf{L}\big(v_\theta(\mathbf{x}_r, r), \log_{\mathbf{x}_0}(\mathbf{x}_1)\big),$$

where $\log_{\mathbf{x}_0}(\mathbf{x}_1)$ denotes the geodesic displacement vector between the start and end states. We evaluate these losses individually and in combination with the initialization constraint $\mathcal{L}_{\text{Init}}$.

Results in Tab. 6 show that displacement-only supervision performs poorly when applied in isolation ($\mathcal{L}_{\text{t-time}} + \mathcal{L}_{\text{r-time}}$ reaches only 0.133 AAR), indicating that collapsing the trajectory into a single global direction discards important temporal structure. Anchoring the flow at $t = 0$ recovers most of the performance, and adding displacement terms on top of $\mathcal{L}_{\text{Init}}$ helps further: among the two-term variants, $\mathcal{L}_{\text{Init}} + \mathcal{L}_{\text{r-time}}$ is the stronger one (0.439 AAR, 1.596 rotational and 4.50 translational error), slightly ahead of $\mathcal{L}_{\text{Init}} + \mathcal{L}_{\text{t-time}}$, and the three-term combination is best overall.

These ablations establish that direct displacement matching alone is insufficient and that an initialization anchor is necessary; they compare variants of the displacement-based family and do not by themselves measure the teacher–student velocity objective of Eq. 7.

Table 6: Comparison of consistency-based flow tuning objectives on protein structure generation.

Loss Setting	Val. AAR (↑)	Val. Rot. (↓)	Val. Trans. (↓)
$\mathcal{L}_{\text{prog.}}$ (progressive only)	0.030	2.486	44.78
$\mathcal{L}_{\text{Init}}$	0.437	1.731	4.52
$\mathcal{L}_{\text{t-time}} + \mathcal{L}_{\text{r-time}}$	0.133	1.880	4.56
$\mathcal{L}_{\text{Init}} + \mathcal{L}_{\text{r-time}}$	0.439	1.596	4.50
$\mathcal{L}_{\text{Init}} + \mathcal{L}_{\text{t-time}}$	0.430	1.627	4.58
$\mathcal{L}_{\text{Init}} + \mathcal{L}_{\text{t-time}} + \mathcal{L}_{\text{r-time}}$	**0.448**	**1.515**	**4.27**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the proposed FFC framework, its use of differentiable force fields and consistency training, and the evaluated protein-design settings. The claims are supported by the method, theory, and experiments.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a "Limitations" section discussing dependence on force-field quality and scaling challenges from differentiable force-field computation.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper states assumptions for the theoretical results, including SO(3)-invariance/equivariance and the idealized calibrated-flow assumption, and provides proofs in the appendix.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes the proposed algorithm, training objectives, datasets, baselines, metrics, and implementation details needed to reproduce the main experimental claims.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper cites official checkpoints and public resources for some baselines.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies datasets, evaluation protocols, metrics, model checkpoints, optimizers, learning rates, batch size, sampling steps, and other implementation details in the main text and appendix.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Most main tables do not report error bars or confidence intervals. A paired t-test is reported for one 1-step peptide-design comparison, but statistical significance is not reported comprehensively for all main experimental claims.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports relevant compute resources for training/fine-tuning, including NVIDIA H100-80GB GPUs and training duration/iterations for the main experimental settings.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses public scientific datasets/models and computational experiments, and we have reviewed the NeurIPS Code of Ethics. We are not aware of any deviation from it.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper discusses scientific benefits for protein design, but it does not explicitly discuss both positive and potential negative societal impacts, such as dual-use risks in molecular or protein design.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk pretrained generative model or scraped dataset requiring special release safeguards.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites existing datasets, models, and software packages, but it does not explicitly list licenses or terms of use for all existing assets.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release a new dataset, benchmark, or standalone asset requiring separate documentation.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development does not involve LLMs as an important, original, or non-standard component.