Empirical Bayes Flow Matching for Continuous Cryo-EM Heterogeneity

Daniel Aibinder
University of Haifa
daibinde@campus.haifa.ac.il

Azmi Haider
University of Haifa
ahaide03@campus.haifa.ac.il

Dan Rosenbaum
University of Haifa
danro@cs.haifa.ac.il

Abstract

We address the problem of learning generative priors over high-dimensional latent variables from indirect, noisy observations, with a focus on continuous heterogeneity in cryo-electron microscopy (cryo-EM). In this setting, 3D atomic structures $x \in \mathbb{R}^{3N_a}$ are never directly observed and must be inferred from 2D projection images y generated by a complex, non-linear forward operator. We propose Empirical Bayes Flow Matching (EB-FM), a method that learns a flow matching prior $p_\theta(x)$ directly from observations using a Monte Carlo stochastic approximation Expectation-Maximization framework. EB-FM alternates between an E-step that performs approximate posterior sampling from $p_\theta(x \mid y)$ via guided flow-based generation while maintaining persistent latent estimates through stochastic approximation, and an M-step that updates the flow model using these stabilized latent variables as training targets. This approach eliminates the need for pretraining on clean data and naturally accommodates non-linear forward models. We validate our method on controlled inverse problems using two-moons and MNIST data, and demonstrate its effectiveness on simulated cryo-EM data with known poses, showing that EB-FM can recover continuous distributions of protein conformations directly in atomic-coordinate space.

1 Introduction

In the natural sciences, quantities of interest are typically not directly observed but inferred from indirect measurements. This inverse-modelling task is central to experimental science, and its success depends on how faithfully we can represent the distribution of the underlying quantities of interest.

Inferring protein structure using cryogenic electron microscopy (cryo-EM) is a prime example of this task [1, 2, 3]. Proteins carry out their biological functions as dynamic molecules, adopting multiple conformational states that mediate essential cellular processes such as transport and signalling [4, 5]. Cryo-EM is uniquely positioned to observe this structural heterogeneity: a biological sample is flash-frozen, preserving the ensemble of conformations present in solution, and an electron beam then produces thousands of noisy 2D projection images of the 3D structures, each in a random, unknown orientation [1, 2, 3, 6]. In contrast to models like AlphaFold [7, 8], which primarily target high confidence static structures, cryo-EM data directly reflects the full distribution of conformations that a protein occupies, and recovering this distribution is essential for understanding function [9]. The task is therefore not to reconstruct a single 3D structure, but to infer a distribution $p(x)$ over 3D atomic configurations from indirect observations. This can be framed as an Empirical Bayes problem [10, 11, 12].

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

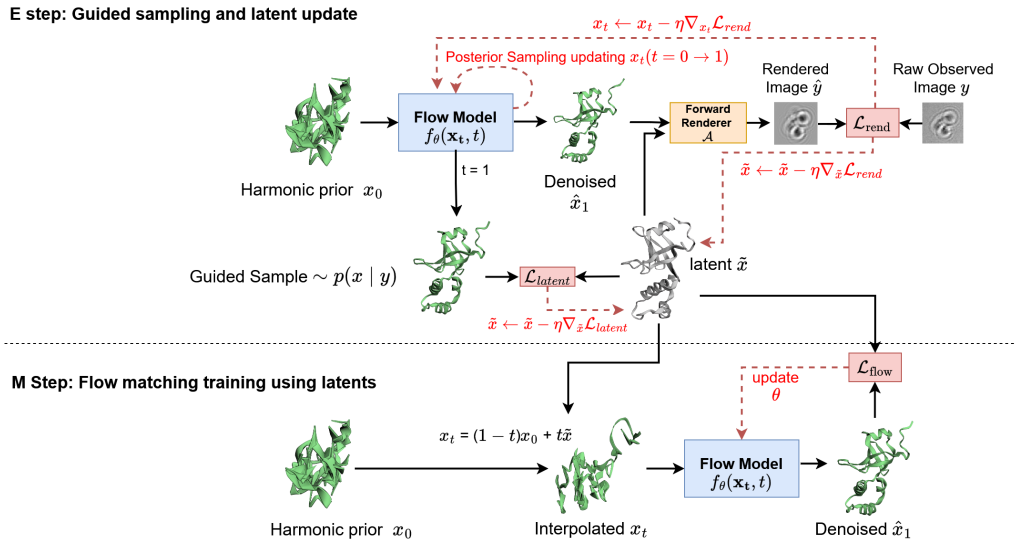

Figure 1: Overview of EB-FM, illustrated for the cryo-EM setting. Each observation y_i is associated with a persistent latent $\tilde{x}_i$ that tracks the posterior mean. In the E-step, the current flow matching prior $p_\theta(x)$ generates a posterior sample $\hat{x}_i$ via gradient-based guidance through the differentiable forward operator $\mathcal{A}$, and $\tilde{x}_i$ is updated by EMA. In the M-step, the prior is trained by conditional flow matching on the persistent latents $\{\tilde{x}_i\}$ in place of clean data.

Classical cryo-EM reconstruction pipelines [13, 14] target single consensus structures and handle heterogeneity only through discrete clustering. Recent deep-learning methods (cryoDRGN [15], 3DFlex [16], cryoFIRE [17], cryoSPHERE [18], mixture-of-neural-fields [19]) learn continuous latent representations of conformational variability, but all share a structural limitation: they operate in a two-stage regime in which the 2D projections are first used to reconstruct 3D density / voxel grids, and atomic models are fit to those densities only as a separate downstream step. This decoupling of the generative model from the underlying atomic physics prevents the use of strong structural priors: the density stage has no access to atomic-coordinate priors, and the atom-fitting stage no longer has access to the raw observations. This matters most when the observations are under-determined and have low signal-to-noise ratios, precisely the regime where a strong atomic prior should help the most. The exception in the literature is [20], which operates directly in atomic-coordinate space and couples an atomic-coordinate prior to the raw image observations using a variational autoencoder. We follow this direction, incorporating modern generative-modelling approaches for the prior.

Diffusion [21, 22, 23] and flow matching [24, 25, 26] models have become the dominant framework for learning complex high-dimensional distributions, with striking successes in molecular and structural biology [27, 28, 29, 30, 8]. Recent work has demonstrated that pre-trained diffusion priors can be successfully leveraged to solve inverse problems directly in protein space. Notably, Levy et al. [31] and CryoBoltz [32] successfully guide pre-trained diffusion models to fit heterogeneous cryo-EM density maps at inference time. However, utilizing a pre-trained prior is not feasible for *de novo* structural discovery, where no clean atomic structures of the target distribution are available and the training data consists only of indirect, noisy projections $\{y_i\}$. The central question is therefore how to train a modern generative prior when the data is itself the output of the inverse problem we are trying to solve.

We propose Empirical Bayes Flow Matching (EB-FM), a method that learns a flow matching prior over atomic structures from indirect observations, by iteratively inverting the forward model during training. Like [20], we operate in atomic-coordinate space, but we depart from its VAE framing in two important ways. First, we replace the variational-autoencoder prior with a flow matching model, which can capture complex multi-modal structural distributions. Second, instead of an amortised encoder, we maintain persistent per-observation latents $\{\tilde{x}_i\}$ that are inferred via optimization during the model training. This is formulated as a Monte Carlo stochastic approximation Expectation-

Maximization algorithm [33]. We optimize the marginal likelihood of the observations by alternating between an E-step, where each latent $\tilde{x}_i$ is refined via deterministic approximate posterior sampling using the current prior and the observation y_i, and an M-step, in which the flow matching prior is retrained on the stabilized latent ensemble $\{\tilde{x}_i\}$.

Several lines of work train generative priors directly from corrupted observations rather than clean data, including AmbientDiffusion [34] for structured measurement corruption. Most relevant to our setting, DiEM [35] proposes a similar empirical Bayes scheme with a diffusion prior, but under two restrictions that prevent its application in our setting. First, DiEM is formulated exclusively for linear Gaussian forward models $p(y \mid x) = \mathcal{N}(y \mid Ax, \Sigma_y)$, for which the intermediate-state marginal likelihood admits a closed form via moment matching. Our cryo-EM forward model is structurally non-linear as the 3D atomic coordinates enter the projected image through a sum of radial-basis exponentials. Second, DiEM's E-step relies on solving a large linear system via a Conjugate-Gradient method at every discretization step. This requires evaluating Vector-Jacobian Products through the generative network, which becomes computationally prohibitive in the case of high dimensional observations. EB-FM bypasses these limitations, targeting non-linear differentiable forward models with a tractable E-step.

We first validate EB-FM on controlled benchmarks using 2D two-moons data and MNIST, where we can ablate design choices. We then apply it to the cryo-EM task of inferring heterogeneous protein structure, with data generated by molecular-dynamics and electron-microscopy simulators [6, 36]. We adopt known poses as a simplifying assumption that isolates the EB-FM contribution from the joint pose-estimation problem. While unknown poses and mismatch between simulated and real data remain substantial open challenges, EB-FM provides the methodological foundation on which those extensions can be built. Our contributions can be summarized as follows:

1. We develop empirical Bayes flow matching (EB-FM), a method for learning flow matching generative priors from indirect observations by iteratively inverting a possibly non-linear forward model during training.
2. On controlled benchmarks we validate our design choices that stabilize the E-step and reduce the posterior sampling variance: flow matching with gradient based guidance and persistent latents, which reduce variance across iterations.
3. We apply EB-FM on simulated cryo-EM data and demonstrate the recovery of 3D conformational distributions from 2D projections directly in atomic-coordinate space, under known poses.

Code is available at: `https://daiyral.github.io/Empirical-Bayes-Flow-Matching-for-Continuous-Cryo-EM-Heterogeneity/`

2 Preliminaries

2.1 Flow Matching

Continuous-time generative models learn to transport a simple, tractable base distribution p_0 (e.g., a standard normal) to a complex data distribution p_1 via an ordinary differential equation (ODE). Let $x_t \in \mathbb{R}^D$ denote the state of the system at time $t \in [0, 1]$. The Probability Flow ODE is defined by a time-dependent vector field v_t formulated as: $dx_t = v_t(x_t)dt$.

Flow Matching [24, 25, 26] provides a scalable framework to learn this vector field by regressing a neural network $v_\theta(x_t, t)$ against conditional vector fields $v_t(x_t \mid x_1)$, defined given a data point $x_1 \sim p_1(x_1)$. The model parameters θ are optimized using the Conditional Flow Matching (CFM) objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,x_1,x_t}\left[\|v_\theta(x_t, t) - v_t(x_t \mid x_1)\|^2\right] \tag{1}$$

Evaluating this objective requires drawing clean samples x_1 from the target data distribution p_1. In the context of *de novo* discovery of protein structure, clean 3D conformations are not given directly, but through a forward operator $y = \mathcal{A}(x)$, rendering the standard CFM objective intractable.

Given a generative prior $p_\theta(x)$ trained on clean data, sampling from the posterior $p_\theta(x \mid y) \propto p(y \mid x)\, p_\theta(x)$ requires augmenting the prior's reverse dynamics with the likelihood gradient $\nabla_{x_t} \log p(y \mid x_t)$, which has no closed form when the forward operator is non-linear. Diffusion Posterior Sampling (DPS) [37] approximates this with a point-estimate of the network's x-prediction

$\hat{x}_1(x_t, t)$, yielding a guidance term proportional to $\nabla_{x_t} \|y - \mathcal{A}(\hat{x}_1(x_t, t))\|^2$ that is back-propagated through the (possibly non-linear) forward operator $\mathcal{A}$. The same recipe extends naturally to flow matching [38, 39], and is the basis for our guided E-step in Section 3.1. Score-based priors were first applied to inverse problems in medical imaging [40], and posterior inference has since also been formulated variationally as stochastic optimization over a parametric posterior [41], an alternative to the sampling-based route we adopt.

For linear forward models, exact guidance becomes tractable via moment matching, and DiEM [35] uses this to construct an empirical-Bayes method via a Conjugate-Gradient solver at every integration step. In non-linear settings with high dimensional observations such as in cryo-EM, exact moment matching is intractable, motivating DPS-style point-estimate guidance.

2.2 The Cryo-EM Forward Model

In cryo-EM, the observations are a set of noisy 2D projection images $\{y_i\}_{i=1}^N$. Let $x \in \mathbb{R}^{3N_a}$ denote the continuous 3D structure parametrized by the spatial coordinates of N_a atoms. We assume the electron potential surrounding each atom can be approximated by a mixture of Gaussian radial basis functions. Under the simplifying assumption of known poses $R_i \in \mathrm{SO}(3)$, the formation of an image y_i evaluates the electron potential of the rotated atoms $R_i x$. Specifically, the pixel intensity $y_i(p)$ at a 2D coordinate $p \in \mathbb{R}^2$ is the line integral of this potential along the viewing axis z:

$$y_i(p) = \int \sum_{j=1}^{N_a} \sum_{k=1}^{K} w_{j,k} \mathcal{N}\left(\begin{bmatrix} p \\ z \end{bmatrix} ; R_i x_j, \sigma_{j,k}^2 I \right) dz + \epsilon_i \tag{2}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2 I)$ models detector noise, and $\{w_{j,k}, \sigma_{j,k}\}$ parameterize the Gaussian mixture components (typically $K = 5$) for the atomic potentials. This coordinate-based forward model defines a structurally non-linear rendering operator, $y_i = \mathcal{A}_i(x) + \epsilon_i$, and consequently a highly non-linear likelihood function $p(y_i \mid x)$.

2.3 Expectation-Maximization

When the true 3D structures $\{x_i\}$ are unobserved, they must be treated as hidden variables. To learn the flow matching generative prior $p_\theta(x)$ directly from the observations $\{y_i\}$, one seeks to maximize the marginal log-likelihood:

$$\log p_\theta(y_i) = \log \int p(y_i \mid x_i) p_\theta(x_i) dx_i \tag{3}$$

The standard Expectation-Maximization (EM) algorithm approaches this by alternating between two steps. The E-step computes the expected log-likelihood under the posterior distribution given the current parameters θ_{old}, using $Q(\theta|\theta_{\text{old}}) = \mathbb{E}_{p_{\theta_{\text{old}}}(x_i|y_i)}[\log p(y_i \mid x_i) + \log p_\theta(x_i)]$. The M-step then updates the parameters by maximizing this auxiliary function: $\theta_{\text{new}} = \arg\max_\theta Q(\theta|\theta_{\text{old}})$.

3 Empirical Bayes Flow Matching

We consider the general inverse problem of recovering an unknown continuous state $x \in \mathcal{X}$ from a noisy observation $y \in \mathcal{Y}$. The physical measurement process is governed by a known, potentially non-linear forward, or *rendering*, operator $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ (e.g. the cryo-EM image acquisition process). Assuming additive Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_y^2 I)$, the observation likelihood is given by:

$$p(y \mid x) = \mathcal{N}(y \mid \mathcal{A}(x), \sigma_y^2 I) \tag{4}$$

In the *de novo* setting, we are given a dataset of N independent observations $\mathcal{D} = \{y_i\}_{i=1}^N$, but no clean samples of the underlying states x_i. Our objective is to learn a flow matching generative prior $p_\theta(x)$ that accurately models the distribution of unobserved states, relying exclusively on the corrupted indirect measurements.

Because x is unobserved, this naturally frames an Expectation-Maximization (EM) objective. The standard EM algorithm alternates between computing the posterior distribution $p_{\theta_{\text{old}}}(x \mid y_i)$ in the E-step, and maximizing the expected complete-data log-likelihood in the M-step. However, standard EM is intractable in our setting, since $p_\theta(x)$ is only defined implicitly via the probability flow ODE of a continuous-time vector field $v_\theta(x_t, t)$, and the forward operator $\mathcal{A}(x)$ is non-linear. This prevents the analytical solution to the E step.

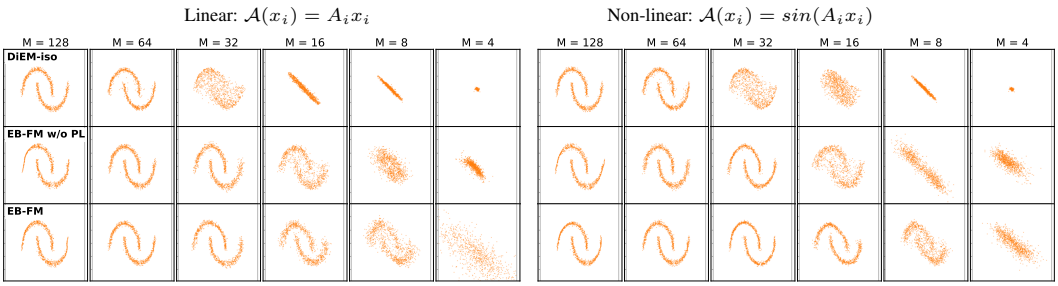

Figure 2: Two-moons benchmark for linear and non-linear forward operators, across decreasing measurement dimensions M. Each panel shows samples from the learned prior projected back to 2D. DiEM-iso (top) is sharp at high M but its linearized CG solver collapses by M=32 under increasing uncertainty. EB-FM without persistent latents (middle) preserves the manifold further into the under-determined regime but loses structure by M=8. EB-FM with persistent latents (bottom) is the only method that recovers the two moons at M=8.

3.1 Stochastic Approximation of the Posterior Mean (E-Step)

To bypass the intractable E-step, we adopt a Monte Carlo Expectation-Maximization framework. Standard MCEM requires drawing a massive number of samples per observation to accurately compute the expected log-likelihood. This is computationally prohibitive for flow matching models that require multi-step sampling guided by back-propagating through the forward operator. Conversely, using only a single posterior sample introduces severe variance, stemming also from the approximate nature of the posterior sampling trajectory itself.

We resolve this by explicitly tracking the posterior mean of each observation. We maintain a set of persistent latents $\{\tilde{x}_i\}_{i=1}^N$ corresponding to each observation y_i. In the E-step of iteration k, we first draw a single, noisy Monte Carlo sample $\hat{x}_i^{(k)}$ from the current posterior $p_{\theta_{k-1}}(x \mid y_i)$ using an approximate guided sampling procedure, based on a point estimate of the predicted clean state, i.e. $\nabla_{x_t} \|\mathbf{y} - \mathcal{A}(\hat{x}_1(x_t, t))\|$ at each sampling time t. Then, rather than using this high-variance sample directly, we update our persistent latent using a stochastic approximation step:

$$\tilde{x}_i^{(k)} = (1 - \gamma_k)\tilde{x}_i^{(k-1)} + \gamma_k \hat{x}_i^{(k)} \tag{5}$$

where γ_k is the step size. In some experiments we also update the persistent latents with the rendering loss directly $\nabla_{\tilde{x}_i} \|\mathbf{y} - \mathcal{A}(\tilde{x}_i)\|$. This moving average approximately tracks the posterior mean, $\mathbb{E}[x \mid y_i]$, reflecting the assumption of local unimodality. In the cryo-EM setting, this means that while the global distribution can be highly complex, the posterior distribution of a conformation given a single specific 2D projection is well approximated by the posterior mean.

3.2 Flow Matching from Inferred States (M-Step)

The M-step trains the flow matching model on the persistent posterior samples $\{\tilde{x}_i^{(k)}\}_{i=1}^N$ in place of the unobserved clean data. The Conditional Flow Matching (CFM) loss on a minibatch $\mathcal{B}$ becomes:

$$\mathcal{L}_{\text{CFM}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E}_{t, x_0} \left[\|v_\theta(x_t, t) - v_t(x_t \mid \tilde{x}_i^{(k)})\|^2 \right] \tag{6}$$

where $x_t = (1 - t)x_0 + t\tilde{x}_i^{(k)}$, using $x_0 \sim p_0(x_0)$, $t \sim \mathcal{U}[0, 1]$ and $v_t(x_t \mid \tilde{x}_i^{(k)}) = \tilde{x}_i^{(k)} - x_0$.

Each outer iteration runs one E-step epoch (updating all latents) followed by an M-step epoch (updating the model weights). See Alg. 1. The full method is illustrated for cryo-EM in Figure 1.

3.3 Controlled benchmarks

To isolate the effects of our design choices, we first evaluate EB-FM on a controlled benchmark using the two-moons dataset. We sample 2D points $z_i \in \mathbb{R}^2$ and lift them into a high-dimensional space $x_i \in \mathbb{R}^D$ ($D = 128$) via a fixed random matrix P. For each data sample, we draw a fixed



Figure 3: Unconditional samples from the EB-FM prior. 784-dimensional MNIST observed only through random Gaussian projections. Left: M=100 measurements per image. Right: M=50. The model never sees clean digits during training, yet recovers digit structure from the indirect observations.

random measurement matrix $A_i \in \mathbb{R}^{M \times D}$ and generate noisy observations: $y_i = \mathcal{A}(x_i) + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_M)$. We test two forward operators: a linear model $\mathcal{A}(x_i) = A_i x_i$ and a non-linear model $\mathcal{A}(x_i) = \sin(A_i x_i)$. To evaluate performance across varying degrees of problem difficulty, we sweep the measurement dimension $M \in \{128, 64, 32, 16, 8, 4\}$, representing a transition from a fully-determined system to a highly under-determined regime ($M/D = 0.03$).

We compare three methods, all sharing an identical multi-layer perceptron (MLP) architecture for the generative prior and an identical outer Expectation-Maximization loop structure: (1) **DiEM-iso**: a tractable version of DiEM [35], using a diffusion prior and a conjugate gradient (CG) solver on the linearized forward model, bypassing the need to back-propagate through the network at each CG step by assuming isotropic covariance of the prior. We implement this using the same scheduling as our method, running one E-step epoch and one M-step epoch at each outer iteration, as we find it performs better. (2) **EB-FM w/o PL**: Our flow matching method without persistent latents, i.e., latents are fully replaced at each outer EM iteration. (3) **EB-FM**: Our full proposed method, updating the latents via an exponential moving average (EMA) across outer iterations. For more details about the implementations see App. A.

Results. As shown in Figure 2, all methods successfully recover the distribution in the fully-determined regime ($M = 128$). However, as M decreases and the posterior uncertainty grows, DiEM-iso's performance degrades, leading to gradual loss of structure at $M = 32$ and mode collapse at smaller M. Conversely, our gradient-guided EB-FM framework survives well into the under-determined regime, however without persistent latents (EB-FM w/o PL), the structure is lost at $M = 8$. With the addition of persistent latents, our model (EB-FM) successfully retains the underlying two-moons manifold geometry further into the under-determined regime, becoming the only method capable of resolving non-collapsed distributions at $M = 8$.

In addition to the two-moons data, we also test our method on MNIST data where the observation is a random projection from $D = 784$ pixels to $M = 100$ and $M = 50$ observations. Figure 3 shows resulting samples from the learned EB-FM prior, demonstrating that it captures the digit structure of the data even when during training the model has only access to low dimensional observations. Comparisons with baselines and more details are provided in App. B.

4 Cryo-EM heterogeneity in atomic-coordinate space

We now apply EB-FM to the motivating problem of recovering the distribution of 3D protein conformations from noisy 2D cryo-EM projections, directly in atomic-coordinate space. We describe our experimental pipeline and the results on four generated datasets, one per protein, with EB-FM trained independently on each. More information about the data and method implementation is provided in App. C.

Datasets and Simulation We evaluate our method on four proteins from the **ATLAS** database [42]: `1a62_A` (130 amino acids), `1egw_C` (79 aa), `3rjs_A` (89 aa), and `7r79_C` (78 aa). We use each protein to generate a separate dataset to train our model. These proteins are small to medium-sized structures commonly used in molecular dynamics studies and exhibit varying degrees of conformational flexibility. For all experiments, we use *backbone atoms only* to reduce complexity while preserving the core structural signal. Each protein's MD ensemble contains 10,000 conformations.

To generate observations, we simulate the cryo-electron microscopy (cryo-EM) image formation process using a high-quality TEM simulator [36]. The simulator includes realistic physics such as

the contrast transfer function (CTF), detector effects, solvation, and noise models. For each protein conformation, we generate a single projection image by applying a random rotation as seen in Fig. 6. The simulation follows standard practice, with parameters summarized in Table 5.

Model The generative prior $p_\theta(x)$ is a Transformer operating on backbone atoms, where the 3D coordinate of each atom is treated as a token. The forward operator within our model is simpler than the simulator used to generate the data, and is implemented with CryoJAX [43], a differentiable cryo-EM renderer in JAX, using a fixed single Gaussian per atom.

Training Latents are initialized from the first trajectory frame of the corresponding Protein Data Bank (PDB) protein. When such a structure is unavailable, an initialization could be obtained using methods such as AlphaFold. The forward loss used for posterior sampling and latent optimization is $\mathcal{L}_{\text{forward}} = \alpha\mathcal{L}_{\text{render}} + \beta\mathcal{L}_{\text{backbone}}$, where $\mathcal{L}_{\text{render}} = \|\mathcal{A}(\tilde{x}_i) - y_i\|_1$ compares the observed image y_i to the image produced by our differentiable cryo-EM renderer $\mathcal{A}$ from the current latent structure, and $\mathcal{L}_{\text{backbone}}$ enforces structural consistency by penalizing squared deviations from pairwise interatomic distances of the initialization structure.

We begin with a *warmup phase* that optimizes each $\tilde{x}_i$ by gradient descent against $\mathcal{L}_{\text{forward}}$. After warmup we run the EB-FM EM loop. In each E-step, for every observation we generate 5 candidates with 3 unguided integration steps, select the candidate with the lowest $\mathcal{L}_{\text{render}}$, and apply 10 guided refinement steps using both rendering and backbone losses to produce a single posterior sample.

We then update $\tilde{x}_i$ with one Adam step on the squared distance to the new posterior sample plus $\mathcal{L}_{\text{forward}}$, the auxiliary losses used for guidance. With a plain gradient step and without $\mathcal{L}_{\text{forward}}$, this reduces to the EMA update of Eq 5.

The M-step updates the flow model on the current $\{\tilde{x}_i\}$. We pre-train the flow for 50 epochs ($\approx 2K$ updates), and then we alternate one E-step over all observations with one epoch of M-step updates per outer iteration. Following [30] we use a *harmonic prior* as the base distribution p_0, and align it to the latent target $\tilde{x}_i$ based on the root mean squared deviation (RMSD).

Evaluation To evaluate EB-FM in the cryo-EM setting, we compare the predicted ensemble of protein conformations with the ground truth molecular dynamics (MD) ensemble using metrics that capture both flexibility and distributional accuracy. To assess flexibility, we measure the average pairwise root mean squared deviation (RMSD) within each ensemble, which reflects the overall structural variability, and compare it to the MD simulation. We further compute the root mean square fluctuation (RMSF) per residue, which quantifies local atomic deviations across the ensemble, and compare it to the ground truth, along with the Pearson correlation between the two RMSF profiles to evaluate whether the model correctly identifies regions of relative flexibility and rigidity along the protein. Beyond flexibility, we assess distributional accuracy by comparing the two ensembles as distributions rather than individual structures. Specifically, we approximate the Earth Mover's Distance using RMSD as the ground metric (EMD-RMSD) [20]: we sample $N = 10{,}000$ structures from each ensemble and compute an $N \times N$ matrix of pairwise RMSDs after rigid alignment. We then solve an optimal one-to-one assignment using the Hungarian algorithm, and report the average RMSD over the matched pairs as the final EMD-RMSD score. Finally, we report coverage and fidelity in terms of RMSD. Coverage RMSD computes, for each ground truth structure, the minimum RMSD to any predicted sample, evaluating how well the predicted ensemble spans the true conformational space, while fidelity RMSD computes, for each predicted structure, the minimum RMSD to the ground truth, reflecting the accuracy and realism of the generated conformations.

4.1 Results

The samples shown in Figure 4 demonstrate that EB-FM produces realistic and physically consistent conformations that closely match the ground truth structures across different states of each protein. The reported RMSD values are typically in the range of 0.51–1.30 Å for the more rigid proteins, such as 3rjs_A and 1a62_A, while the more flexible proteins, such as 7r79_C and 1egw_C, exhibit higher RMSD values reaching 3.0 Å. This reflects the increased difficulty of modeling broader conformational distributions from a single noisy 2D observation. The generated samples preserve the global fold and backbone geometry without introducing major artifacts, although small deviations are visible in flexible loop regions. We provide animations of samples from the prior aligned to ground truth trajectories, in the supplementary material.

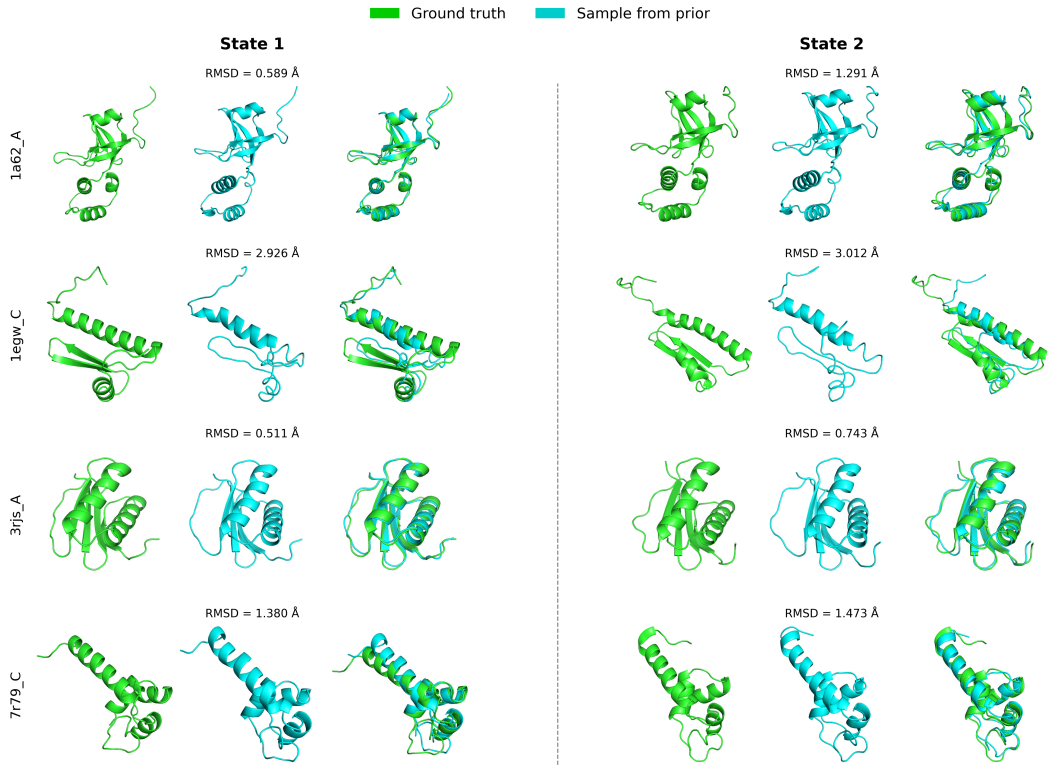

Figure 4: Unconditional EB-FM samples on four ATLAS proteins, compared to ground-truth MD conformations. For each protein we show two ground-truth states selected to span the conformational change in the ensemble. Green: ground-truth. Cyan: the closest unconditional sample from the learned prior. Each pair is also shown superimposed with the corresponding RMSD. Animations of aligned trajectories are provided in the supplementary material.

To further analyze reconstruction quality, Figure 5 (right) compares three quantities: the RMSD of the latent variables' initialization (which is a single representative structure), the RMSD of the final optimized latents after the full EB-FM procedure, and the RMSD of unconditional samples drawn directly from the trained flow matching prior. The optimized latents substantially improve over initialization across all proteins, indicating that the E-step posterior refinement successfully moves structures toward the true conformational ensemble. Importantly, the unconditional EB-FM samples achieve RMSD values close to those of the optimized latents, showing that the learned prior captures the underlying distribution. The significance of the empirical Bayes approach is also highlighted by the training dynamics in Figure 5 (left): Optimization based only on the rendering and backbone losses results in catastrophic divergence to degenerate structures (red curves), whereas the parallel training of the prior (EB-FM) facilitates the flow of information across observations gradually capturing the distribution of underlying conformations (blue curves).

Table 1 compares EB-FM with three complementary approaches for generating protein conformations. We compare against the VAE implemented by Rosenbaum et al [20], which learns an autoencoder-based latent representation of protein structures and generates conformations from the learned latent space. We also compare against AlphaFold3 with MSA subsampling (AF3 MSA sub), which generates different conformations by randomly subsampling the multiple sequence alignment before each prediction [44, 45]. Finally, we compare against cryoDRGN [15] combined with guided-AlphaFold3 (cryoDRGN-guided-AF3). In this approach, cryoDRGN reconstructs heterogeneous 3D density maps from the cryo-EM particle images, which are then used to guide AlphaFold3 during inference following the method of Maddipatla et al.[46]. Thus, while both this approach and EB-FM use the cryo-EM observations to recover conformational heterogeneity, EB-FM directly learns the distribution over atomic coordinates from the observations rather than first reconstructing density maps and subsequently guiding a pretrained structure predictor.

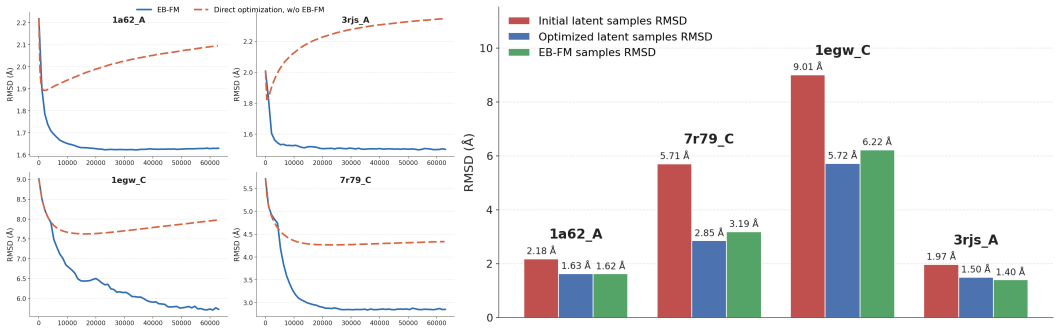

Figure 5: Left: Latent RMSD during optimization. EB-FM vs. direct latent optimization using only rendering and backbone losses (without learning flow matching prior). Right: Final RMSD comparison. Red: latent initialization using a reference structure; Blue: optimized latents after EB-FM training; Green: unconditional samples drawn from the learned flow matching prior.

Table 1: Evaluation of generated conformational distributions across four proteins. We compare EB-FM with the VAE implemented by Rosenbaum et al [20], AlphaFold3 with MSA subsampling (AF3 MSA sub), and cryoDRGN combined with guided-AlphaFold3 (cryoDRGN-guided-AF3). **Distributional accuracy**: EMD-RMSD, Coverage RMSD, and Fidelity RMSD. **Flexibility**: pairwise RMSD, per-residue RMSF, and Pearson correlation r between predicted and ground-truth RMSF profiles. Ground-truth (GT) values are shown for the flexibility metrics. Lower is better for the distributional accuracy metrics, and higher is better for RMSF r. For pairwise RMSD and RMSF, values closer to GT are better ($\rightarrow$ GT).

Protein	Metric	GT	EB-FM	VAE	AF3 MSA	cD+gAF3
1a62_A	EMD RMSD ↓	–	**1.62**	1.80	1.83	1.80
	Coverage RMSD ↓	–	**1.37**	1.65	1.56	1.58
	Fidelity RMSD ↓	–	1.27	1.46	1.18	**1.01**
	Pairwise RMSD → GT	2.40	**1.58**	1.20	1.23	0.77
	RMSF → GT	1.08	**0.77**	0.60	0.59	0.51
	RMSF r ↑	–	0.96	**0.97**	0.57	0.69
7r79_C	EMD RMSD ↓	–	**3.19**	3.96	5.19	4.76
	Coverage RMSD ↓	–	**2.35**	3.11	3.22	3.05
	Fidelity RMSD ↓	–	2.35	**2.06**	4.47	3.72
	Pairwise RMSD → GT	3.21	1.48	2.00	4.73	**3.46**
	RMSF → GT	1.89	0.91	1.31	3.15	**2.35**
	RMSF r ↑	–	**0.91**	0.85	0.64	0.61
1egw_C	EMD RMSD ↓	–	6.22	**5.56**	8.30	7.87
	Coverage RMSD ↓	–	**4.44**	5.17	6.62	6.68
	Fidelity RMSD ↓	–	4.19	4.72	**2.34**	3.02
	Pairwise RMSD → GT	6.45	**4.29**	3.06	1.98	2.85
	RMSF → GT	3.11	**2.55**	1.55	0.74	0.83
	RMSF r ↑	–	**0.93**	0.92	0.88	0.70
3rjs_A	EMD RMSD ↓	–	**1.40**	2.05	1.56	1.41
	Coverage RMSD ↓	–	1.22	1.83	**1.11**	1.16
	Fidelity RMSD ↓	–	0.99	1.59	1.02	**0.93**
	Pairwise RMSD → GT	1.79	1.10	0.93	**1.17**	0.94
	RMSF → GT	0.68	**0.49**	**0.49**	0.36	0.45
	RMSF r ↑	–	**0.95**	0.89	**0.95**	0.90

In terms of EMD-RMSD, EB-FM outperforms cryoDRGN-guided-AF3 on all four proteins, with nearly identical performance on 3rjs_A. For Coverage RMSD, EB-FM performs better on three of the four proteins, while for Fidelity RMSD it performs better on one. Compared with the VAE, EB-FM also achieves better results on the majority of metrics across the four proteins, although the VAE performs better on several metrics, particularly for 7r79_C, and achieves a lower EMD-RMSD

for 1egw_C. These results suggest that learning a flow matching prior directly from observations can provide an effective representation of complex conformational distributions.

The flexibility metrics further highlight the advantages of EB-FM. Its RMSF Pearson correlation ranges from 0.91 to 0.96 across the four proteins, exceeding that of cryoDRGN-guided-AF3 in every case and indicating that EB-FM accurately identifies the relative locations of flexible regions. In terms of absolute flexibility, measured by pairwise RMSD and RMSF, EB-FM produces ensembles that are closer to the ground-truth magnitude than cryoDRGN-guided-AF3 on three of the four proteins. However, EB-FM still underestimates the absolute magnitude of conformational motion relative to the ground-truth MD ensembles across all four proteins, as reflected in lower pairwise RMSD and RMSF values. Thus, although EB-FM captures which regions are flexible, it does not fully recover the magnitude of motion in those regions.

The comparison with AF3 MSA subsampling further illustrates the difference between generating diversity by perturbing sequence-derived information and learning conformational heterogeneity directly from experimental observations. Although AlphaFold3 benefits from extensive structural knowledge acquired during training, its generated ensembles may not fully capture the conformational distributions supported by the cryo-EM observations. In contrast, EB-FM learns the conformational distribution directly from the observations while accounting for observation uncertainty.

Additional ablation experiments can be seen in Table 4 which evaluate the robustness of EB-FM to CTF and pose errors, randomized initialization of the first frame, and removal of the backbone loss.

Across the four proteins, Fidelity RMSD reaches as low as 0.99 Å, while Coverage RMSD is typically in the 1.2–2.3 Å range for three proteins, indicating that the generated samples remain close to valid ground-truth conformations while covering part of the true ensemble. The main limitations appear for the most flexible protein, 1egw_C, where the higher EMD-RMSD and larger coverage and fidelity errors reflect the difficulty of recovering a broad, highly variable distribution from single noisy projections. Taken together, these results show that EB-FM learns a meaningful atomic-coordinate prior that recovers accurate structures, preserves the main flexible regions of the ensemble, and demonstrates the feasibility of learning conformational distributions directly from cryo-EM observations.

5 Discussion

In this work we introduced Empirical Bayes Flow Matching (EB-FM), a method for training a flow matching prior from indirect noisy observations, designed to make empirical Bayes efficient enough to handle high dimensional observations and non-linear forward models. The standard cryo-EM pipeline first reconstructs a 3D density from 2D images and only then fits an atomic model, with no opportunity to use atomic-coordinate priors at the most under-determined stage of the problem. We showed that (1) combining single-sample DPS-style guidance with persistent latents updated by stochastic approximation gives an efficient empirical Bayes scheme, and (2) this enables bypassing the two-stage cryo-EM pipeline and training a strong atomic-coordinate prior *de novo* directly from 2D projections. On two-moons and MNIST we validate the role of our design choices, and on simulated cryo-EM the learned prior produces realistic backbone conformations and recovers the flexibility profile of the ensemble.

Limitations and future work: Our cryo-EM experiments assume known poses, while in real data the pose is an unknown parameter of the forward model. Posterior sampling with unknown forward parameters has been studied in the diffusion prior setting [47]. A natural next step would be to adopt these approaches within our E-step. A second challenge, visible in our diversity metrics, is that the predicted ensembles underestimate the magnitude of conformational change. We expect this to become more limiting with unknown poses, where the local posterior $p(x|y_i)$ is more complex and multimodal and cannot be well summarized by its mean. One possible solution is to maintain multiple persistent latents per observation. Finally, the method still remains computationally expensive at the current scale. We believe that stronger GPUs and better parallelization will make this approach practical for larger cryo-EM datasets and full atomic structures that contain side-chains rather than backbone atoms only. Another promising direction is to initialize the flow matching prior from a model pretrained on known structures, such as CryoBoltz [32], combining a strong starting prior with the ability of EB-FM to refine it directly on the observed data.

Acknowledgments and Disclosure of Funding

We thank Rachel Kolodny for her guidance on protein structure and conformational heterogeneity, and for her help in choosing the data, proteins, evaluations and experiments. This research was supported by the NVIDIA Academic Grant Program, using NVIDIA H100 GPUs on NVIDIA Brev.

References

[1] Joachim Frank. Single-particle imaging of macromolecules by cryo-electron microscopy. *Annual review of biophysics and biomolecular structure*, 31(1):303–319, 2002.

[2] Edward H Egelman. The current revolution in cryo-em. *Biophysical journal*, 110(5):1008–1012, 2016.

[3] Eva Nogales and Sjors HW Scheres. Cryo-em: a unique tool for the visualization of macromolecular complexity. *Molecular cell*, 58(4):677–689, 2015.

[4] Hans Frauenfelder, Stephen G Sligar, and Peter G Wolynes. The energy landscapes and motions of proteins. *Science*, 254(5038):1598–1603, 1991.

[5] Hesam N Motlagh, James O Wrabl, Jing Li, and Vincent J Hilser. The ensemble nature of allostery. *Nature*, 508(7496):331–339, 2014.

[6] Miloš Vulović, Raimond BG Ravelli, Lucas J van Vliet, Abraham J Koster, Ivan Lazić, Uwe Lücken, Hans Rullgård, Ozan Öktem, and Bernd Rieger. Image formation modeling in cryo-electron microscopy. *Journal of structural biology*, 183(1):19–32, 2013.

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

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

[9] Nabin Giri, Raj S. Roy, and Jianlin Cheng. Deep learning for reconstructing protein structures from cryo-em density maps: Recent advances and future directions. *Current Opinion in Structural Biology*, 79:102536, 2023.

[10] Herbert Robbins. The empirical bayes approach to statistical decision problems. *The Annals of Mathematical Statistics*, 35(1):1–20, 1964.

[11] George Casella. An Introduction to Empirical Bayes Data Analysis. 39(2):83–87, 1985.

[12] Bradley Efron. Two modeling strategies for empirical Bayes estimation. *Statistical science : a review journal of the Institute of Mathematical Statistics*, 29(2):285–301, 2014.

[13] Sjors HW Scheres. Relion: implementation of a bayesian approach to cryo-em structure determination. *Journal of structural biology*, 180(3):519–530, 2012.

[14] Ali Punjani, John L Rubinstein, David J Fleet, and Marcus A Brubaker. cryosparc: algorithms for rapid unsupervised cryo-em structure determination. *Nature methods*, 14(3):290–296, 2017.

[15] Ellen D Zhong, Tristan Bepler, Bonnie Berger, and Joseph H Davis. Cryodrgn: reconstruction of heterogeneous cryo-em structures using neural networks. *Nature methods*, 18(2):176–185, 2021.

[16] Ali Punjani and David J. Fleet. 3DFlex: determining structure and motion of flexible proteins from cryo-EM. *Nature Methods*, 20:860–870, 2023.

[17] Axel Levy, Gordon Wetzstein, Julien N. P. Martel, Frédéric Poitevin, and Ellen D. Zhong. Amortized inference for heterogeneous reconstruction in cryo-EM. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

[18] Gabriel Ducrocq, Lukas Grunewald, Sebastian Westenhoff, and Fredrik Lindsten. cryosphere: Single-particle heterogeneous reconstruction from cryo em, 2025.

[19] Axel Levy, Rishwanth Raghu, David Shustin, Adele Rui-Yang Peng, Huan Li, Oliver Biggs Clarke, Gordon Wetzstein, and Ellen D. Zhong. Mixture of neural fields for heterogeneous reconstruction in cryo-em, 2024.

[20] Dan Rosenbaum, Marta Garnelo, Michal Zielinski, Charlie Beattie, Ellen Clancy, Andrea Huber, Pushmeet Kohli, Andrew W. Senior, John Jumper, Carl Doersch, S. M. Ali Eslami, Olaf Ronneberger, and Jonas Adler. Inferring a continuous distribution of atom coordinates from cryo-em images using vaes, 2021.

[21] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *PMLR*, pages 2256–2265, 2015.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[23] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. Outstanding Paper Award.

[24] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

[25] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

[26] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

[27] Bowen Jing, Ezra Erives, Peter Pao-Huang, Gabriele Corso, Bonnie Berger, and Tommi Jaakkola. Eigenfold: Generative protein structure prediction with diffusion models. *arXiv preprint arXiv:2304.02198*, 2023.

[28] John B. Ingraham, Max Baranov, Zak Costello, Karl W. Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M. Lord, Christopher Ng-Thow-Hing, Erik R. Van Vlack, Shan Tie, Vincent Xue, Sarah C. Cowles, Alan Leung, João V. Rodrigues, Claudio L. Morales-Perez, Alex M. Ayoub, Robin Green, Katherine Puentes, Frank Oplinger, Nishant V. Panwar, Fritz Obermeyer, Adam R. Root, Andrew L. Beam, Frank J. Poelwijk, and Gevorg Grigoryan. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

[29] Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[30] Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles, 2024.

[31] Axel Levy, Eric R Chan, Sara Fridovich-Keil, Frédéric Poitevin, Ellen D Zhong, and Gordon Wetzstein. Solving inverse problems in protein space using diffusion-based priors. *arXiv preprint arXiv:2406.04239*, 2024.

[32] Rishwanth Raghu, Axel Levy, Gordon Wetzstein, and Ellen D. Zhong. Multiscale guidance of protein structure prediction with heterogeneous cryo-em data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[33] Bernard Delyon, Marc Lavielle, and Eric Moulines. Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):94–128, 1999.

[34] Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alexandros G. Dimakis, and Adam Klivans. Ambient diffusion: Learning clean distributions from corrupted data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[35] François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

[36] Hans Rullgård, L-G Öfverstedt, Sergey Masich, Bertil Daneholt, and Ozan Öktem. Simulation of transmission electron microscope images of biological specimens. *Journal of microscopy*, 243(3):234–256, 2011.

[37] Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023.

[38] Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. FlowDPS: Flow-driven posterior sampling for inverse problems. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[39] Ashwini Pokle, Matthew J. Muckley, Ricky T. Q. Chen, and Brian Karrer. Training-free linear image inverses via flows. *arXiv preprint arXiv:2310.04432*, 2023.

[40] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations (ICLR)*, 2022.

[41] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. *arXiv preprint arXiv:2305.04391*, 2023.

[42] Yann Vander Meersche, Gabriel Cretin, Aria Gheeraert, Jean-Christophe Gelly, and Tatiana Galochkina. Atlas: protein flexibility description from atomistic molecular dynamics simulations. *Nucleic acids research*, 52(D1):D384–D392, 2024.

[43] Michael J. O'Brien, David Silva-Sánchez, Geoffrey Woollard, Kwanghwi Je, Sonya M. Hanson, Daniel J. Needleman, Pilar Cossio, Erik Henning Thiede, and Miro A. Astore. CryoJAX: a cryo-electron microscopy image-simulation library in JAX. *Acta Crystallographica Section D*, 82(3), Mar 2026.

[44] Diego Del Alamo, Davide Sala, Hassane S Mchaourab, and Jens Meiler. Sampling alternative conformational states of transporters and receptors with alphafold2. *elife*, 11:e75751, 2022.

[45] Richard A Stein and Hassane S Mchaourab. Speach_af: Sampling protein ensembles and conformational heterogeneity with alphafold2. *PLoS computational biology*, 18(8):e1010483, 2022.

[46] Advaith Maddipatla, Nadav Sellam Bojan, Meital Bojan, Volodymyr Masalitin, Sanketh Vedula, Paul Schanda, Ailie Marx, and Alex M Bronstein. Experiment-guided alphafold3 resolves measurement-consistent protein ensembles. *Nature Biotechnology*, pages 1–9, 2026.

[47] Naoki Murata, Koichi Saito, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. GibbsDDRM: A Partially Collapsed Gibbs Sampler for Solving Blind Inverse Problems with Denoising Diffusion Restoration. In *Int. Conf. on Machine Learning*.

Algorithm 1 Empirical Bayes Flow Matching

Require: Observations $\{y_i\}_{i=1}^N$, forward operator $\mathcal{A}$, flow model f_θ
Require: Persistent latents $\{\tilde{x}_i\}_{i=1}^N$, base distribution p_0
Require: Outer iterations K, sampling steps N_s, flow-training steps N_f
Require: EMA step size γ, guidance strength η

for $k = 1$ to K **do**
  E-step: guided posterior samples
  for $i = 1$ to N **do**
    $x_t \sim p_0$
    for $n = 0$ to $N_s - 1$ **do**
      $t \leftarrow n/N_s, \quad s \leftarrow t + 1/N_s$
      $\hat{x}_1 \leftarrow f_\theta(x_t, t)$
      $\mathcal{L}_{\text{render}} \leftarrow \|\mathcal{A}(\hat{x}_1) - y_i\|_2^2$
      $v \leftarrow (\hat{x}_1 - x_t)/(1-t)$
      $x_t \leftarrow x_t + (s-t)v - \eta \nabla_{x_t} \mathcal{L}_{\text{render}}$
    end for
    $\tilde{x}_i \leftarrow (1-\gamma)\tilde{x}_i + \gamma x_t$
  end for
  M-step: train flow prior on latents
  for $m = 1$ to N_f **do**
    Sample minibatch B
    $x_1 \leftarrow \{\tilde{x}_i : i \in B\}$
    Sample $x_0 \sim p_0, \quad t \sim \mathcal{U}(0,1)$
    $x_t \leftarrow (1-t)x_0 + t x_1$
    $\hat{x}_1 \leftarrow f_\theta(x_t, t)$
    $\mathcal{L}_{\text{FM}} \leftarrow \left\| \frac{\hat{x}_1 - x_t}{1-t} - \frac{x_1 - x_t}{1-t} \right\|_2^2$
    Update θ using $\nabla_\theta \mathcal{L}_{\text{FM}}$
  end for
end for
return f_θ, $\{\tilde{x}_i\}_{i=1}^N$

A Two-Moons Experimental Details

Dataset and problem setup. We use the two-moons distribution using the function `sklearn.datasets.make_moons` ($n_{\text{train}} = 10{,}000$, noise 0.05), standardized to zero mean and unit variance per coordinate. Each 2D sample $z_i \in \mathbb{R}^2$ is lifted to $x_i = Pz_i \in \mathbb{R}^D$ using a fixed column-orthonormal embedding $P \in \mathbb{R}^{D\times 2}$ drawn once as the orthogonal factor of a QR decomposition of an i.i.d. Gaussian matrix; we use $D = 128$. For each sample we draw an independent measurement matrix $A_i \in \mathbb{R}^{M\times D}$ with i.i.d. $\mathcal{N}(0, 1/D)$ entries, fixed for the lifetime of i, and form the observation $y_i = \mathcal{A}(x_i) + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma_y^2 I_M)$ and $\sigma_y = 10^{-3}$. We evaluate two forward operators, $\mathcal{A}(x) = A_i x$ and $\mathcal{A}(x) = \sin(A_i x)$, and sweep $M \in \{128, 64, 32, 16, 8, 4\}$.

Flow model architecture. All three methods share the same x-prediction multilayer perceptron: 5 hidden ReLU layers of width 256, with a sinusoidal time embedding (dim 128) projected and concatenated to x_t. EB-FM and EB-FM w/o PL train this network with the flow matching v-loss; DiEM-iso trains it with the analogous v-loss for VP-DDPM under a linear β schedule with $T = 1000$.

EB-FM guided sampling. The E-step uses explicit Euler integration with $N = 20$ steps. At each step we predict $\hat{x}_1 = f_\theta(x_t, t)$, compute the rendering-loss gradient $g = \nabla_{x_t} \|A_i \hat{x}_1 - y_i\|_2^2$ via back-propagation through the network, and combine the prior velocity and gradient correction in a single update,

$$x_{t+\Delta t} = x_t + \Delta t\, v - \eta\, g, \qquad v = (\hat{x}_1 - x_t)/(1-t).$$

The guidance strength η is tuned per cell from $\{0.05, 0.3, 0.5, 1.0, 2.0\}$, with larger values used at smaller M where the rendering signal is weaker.

Table 2: MNIST: quality of the learned EB-FM prior under linear random-projection observations (M=50 and M=100 measurements per 28×28 image). Metrics are computed between 10K unconditional samples from the prior and 10K held-out images: EMD (Hungarian assignment), Fidelity (each generated → nearest GT), Coverage (each GT → nearest generated), each in PSNR and SSIM. Higher is better. EB-FM with persistent latents outperforms the diffusion baseline and the EB-FM ablation without persistent latents.

		PSNR ↑			SSIM ↑		
Obs.	**Method**	**EMD**	**Fidelity**	**Coverage**	**EMD**	**Fidelity**	**Coverage**
50	Diffusion	13.75	14.08	14.49	0.59	0.60	0.62
	EB-FM w/o PL	13.75	14.33	14.48	0.56	0.58	0.62
	EB-FM	15.67	16.20	16.35	0.70	0.72	0.74
100	Diffusion	14.72	15.36	15.42	0.68	0.70	0.71
	EB-FM w/o PL	14.39	15.04	15.55	0.68	0.70	0.76
	EB-FM	15.37	16.06	16.13	0.71	0.73	0.74

Persistent latents. For EB-FM, the per-observation latent is updated each E-step by $\tilde{x}_i \leftarrow \alpha\,\tilde{x}_i + (1-\alpha)\,\hat{x}_i$ with $\alpha \in \{0.5, 0.9\}$ tuned per cell with larger α at well-determined cells and smaller α at low M/D. The EB-FM w/o PL ablation replaces the latent fully each iteration ($\alpha = 0$).

DiEM-iso baseline. We use a tractable variant of DiEM [35] with the isotropic-covariance approximation, which avoids back-propagation through the network inside the CG solve. We use ancestral DDPM with $N = 50$ steps; the $M \times M$ system is solved by direct linear solve when $M < 16$ and by 5-iteration conjugate gradient otherwise.

Outer EM schedule. All three methods share the same training schedule: a 50-step rendering-only warm-up of the latents, a prior bootstrap of $50{,}000$ steps on the warm-up latents, and 100 outer EM iterations. Each outer iteration runs one E-step (one round of guided sampling per observation) followed by one M-step (100 Adam inner steps with batch 256, $\beta_2 = 0.999$, learning rate 3×10^{-4}).

Evaluation. For each cell we generate 1000 unconditional samples from the trained prior projected to 2D via $P^\top$, as shown in Figure 2.

Hardware. All two-moons experiments were run on a single NVIDIA RTX 4090 GPU. Each cell takes 50–120 seconds depending on M and method.

B MNIST Experiment

In order to quantify the quality of the prior learned on MNIST, we compute PSNR and SSIM metrics using the following: (1) Earth Mover's Distance (EMD) PSNR/SSIM, matching 10K generated samples with 10K held-out ground truth samples using the Hungarian algorithm, and averaging the PSNR/SSIM over all matched pairs; (2) Fidelity PSNR/SSIM, where each generated sample is compared to the closest GT samples; and (3) Coverage PSNR/SSIM, where each GT sample is compared to the closest generated sample. Table 2 compares these metrics between EB-FM, EB-FM w/o PL, and a Diffusion baseline, all using the same network architecture, posterior guidance mechanism and outer EM loop. EB-FM outperforms the baselines.

Dataset and preprocessing. We evaluate our method on the MNIST dataset consisting of grayscale handwritten digit images of resolution 28×28. All images are normalized to the range $[-1, 1]$. Unless otherwise stated, experiments use $10{,}000$ training images.

Problem setup. Each image $\mathbf{x} \in \mathbb{R}^{1\times 28\times 28}$ is observed through a linear measurement operator

$$\mathbf{y} = A^\top \mathbf{x} + \epsilon,$$

where $A \in \mathbb{R}^{784\times M}$ is a random Gaussian sensing matrix and $\epsilon \sim \mathcal{N}(0, \sigma_y^2 I)$ is additive Gaussian noise. We use $M = 100/50$ measurements and observation noise standard deviation $\sigma_y = 10^{-3}$.

For reproducibility, each training image is assigned a deterministic sensing matrix generated using the image index as the random seed. Columns of the sensing matrix are ℓ_2 normalized.

Flow model architecture. The EB-FM prior is implemented using a lightweight U-Net-style convolutional architecture operating directly in image space. The network receives a noisy/intermediate sample $\mathbf{x}_t$ together with time t, and predicts a clean endpoint estimate $\hat{\mathbf{x}}_1$.

Time conditioning is implemented using Gaussian Fourier features followed by a two-layer MLP. The architecture consists of residual convolutional blocks with Group Normalization and SiLU activations, average-pooling downsampling, nearest-neighbor upsampling, and attention-gated skip connections. The base channel dimension is 16.

EB-FM guided sampling. At inference time, reconstruction begins from Gaussian noise $\mathbf{x}_0 \sim \mathcal{N}(0, I)$, and is evolved using Euler integration with $N = 20$ steps. At each step, the flow model predicts a clean estimate $\hat{\mathbf{x}}_1 = f_\theta(\mathbf{x}_t, t)$. Measurement consistency is enforced using the residual $\mathbf{r} = A^\top \hat{\mathbf{x}}_1 - \mathbf{y}$, and guidance objective $\mathcal{L}_{\text{guide}} = \|\mathbf{r}\|_2^2$. The gradient of the guidance loss with respect to the current noisy state $\mathbf{x}_t$ is used to correct the Euler update:

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \Delta t\, \hat{\mathbf{v}}_\theta - \frac{\eta}{\|\mathbf{r}\|_2} \nabla_{\mathbf{x}_t} \mathcal{L}_{\text{guide}}.$$

We use guidance strength $\eta = 1.0$.

Persistent latent optimization (EB-FM). For the full EB-FM method, we maintain one trainable latent image tensor $\mathbf{z}_i \in \mathbb{R}^{1\times 28\times 28}$ per training sample. These latent images are jointly optimized together with the flow prior. The optimization objective combines reconstruction consistency and guided-sampling consistency: $\mathcal{L} = \lambda_{\text{flow}} \|\hat{\mathbf{z}} - \mathbf{z}\|_1 + \lambda_{\text{render}} \|A^\top \mathbf{z} - \mathbf{y}\|_2^2$. We use $\lambda_{\text{flow}} = 1, \qquad \lambda_{\text{render}} = 1.$

EB-FM without persistent latents. In the non-persistent variant (EB-FM w/o PL), latent tensors are not optimized parameters. Instead, latents are initialized from Gaussian noise and replaced directly by the guided reconstruction samples during training.

Diffusion baseline. We compare against a diffusion-based reconstruction baseline using the same backbone architecture and measurement-guidance framework. The diffusion baseline replaces the flow matching objective with a standard diffusion denoising objective while keeping the same forward operator, reconstruction setup, and measurement guidance mechanism.

Optimization details. All models are trained using Adam optimization. The flow prior uses learning rate 10^{-4}, while persistent latent tensors use learning rate 10^{-2}. Training is performed for 2000 epochs with batch size 64 and gradient clipping with maximum norm 1.0.

For EB-FM, the flow prior is periodically refined on the current latent estimates using 5 additional local training epochs after each refinement stage. At the end of training, the flow prior is additionally refined for 100 epochs.

Hardware. All MNIST experiments were run on a single NVIDIA RTX 4090 GPU with 24 GB of memory. The total wall-clock runtime was approximately 37 hours for experiments using the diffusion based algorithm and 13 hours for experiments using the flow matching algorithm.

Table 3: MNIST experimental configuration.

Setting	Diffusion	EB-FM w/o PL	EB-FM
Latent LR	–	–	10^{-2}
Guidance steps	100	20	20
Persistent latents	No	No	Yes

C Cryo-EM Experimental Details

We write $\textsc{FlowSample}(f_\theta, p_0, N)$ for Euler sampling from the flow prior for N steps, starting from $x_0 \sim p_0$. For cryo-EM, $\mathcal{A}$ denotes the differentiable renderer for observation y_i. The operation $\textsc{RMSDAlign}(x, x')$ rigidly aligns x' to x by minimizing RMSD using the Kabsch algorithm; this removes global rotation and translation before computing the flow update. The backbone-continuity loss compares the current pairwise atom distances to the expected atom distances on a fixed set of local neighbor pairs:

$$\mathcal{L}_{\text{backbone}}(x) = \frac{1}{|\mathcal{M}|} \sum_{(i,j)\in\mathcal{M}} \big(D_{ij}(x) - D^0_{ij}\big)^2,$$

where $D(x)$ is the pairwise atom-distance matrix of structure x, D^0 is computed from the initial structure, and

$$\mathcal{M} = \big\{(i,j) : 0.1\,\text{Å} < D^0_{ij} < 6.0\,\text{Å}\big\}$$

is the set of neighbor pairs determined from the initial structure.

Algorithm 2 Cryo-EM E-step

Require: Observation y_i, renderer $\mathcal{A}$, flow model f_θ, current latent $\tilde{x}_i$
Require: Harmonic prior p_0, expected atom distances D^0
Require: Candidate count C, unguided steps N_u, guided steps N_g
Require: Guidance strength η, latent learning rate ρ
Require: Weights $\lambda_{\text{flow}}, \lambda_{\text{render}}, \lambda_{\text{backbone}}$

Candidate selection with unguided sampling

for $c = 1$ to C **do**
  $x^{(c)} \leftarrow \textsc{FlowSample}(f_\theta, p_0, N_u)$
  $\mathcal{L}^{(c)}_{\text{cand}} \leftarrow \|\mathcal{A}(x^{(c)}) - y_i\|_1$
end for
$c^\star \leftarrow \arg\min_{c\in\{1,\dots,C\}} \mathcal{L}^{(c)}_{\text{cand}}$
$x_t \leftarrow x^{(c^\star)}$

Guided sampling

for $n = 0$ to $N_g - 1$ **do**
  $t \leftarrow n/N_g, \quad s \leftarrow t + 1/N_g$
  $\hat{x}_1 \leftarrow f_\theta(x_t, t)$
  $\mathcal{L}_{\text{guide}} \leftarrow \lambda_{\text{render}}\|\mathcal{A}(\hat{x}_1) - y_i\|_1 + \lambda_{\text{backbone}}\|D(\hat{x}_1) - D^0\|_2^2$
  $\hat{x}_1 \leftarrow \hat{x}_1 - \eta\nabla_{x_t}\mathcal{L}_{\text{guide}}$
  $z_t \leftarrow \textsc{RMSDAlign}(\hat{x}_1, x_t)$
  $v \leftarrow (\hat{x}_1 - z_t)/(1-t)$
  $x_t \leftarrow z_t + (s-t)v$
end for
$\hat{x}_i \leftarrow \hat{x}_1$

Latent update

$\mathcal{L}_{\text{latent}} \leftarrow \lambda_{\text{flow}}\|\tilde{x}_i - \hat{x}_i\|_1 + \lambda_{\text{render}}\|\mathcal{A}(\tilde{x}_i) - y_i\|_1 + \lambda_{\text{backbone}}\|D(\tilde{x}_i) - D^0\|_2^2$
Update $\tilde{x}_i$ with Adam using $\nabla_{\tilde{x}_i}\mathcal{L}_{\text{latent}}$
return $\tilde{x}_i$

Flow model architecture. For each protein, we represent the latent 3D conformation using backbone-atom coordinates, treating each backbone atom as a token with three coordinate channels. The flow prior is parameterized by a transformer network with 3 transformer blocks, hidden dimension 256, and 8 attention heads. Each transformer block consists of multi-head self-attention, a two-layer feed-forward network with expansion factor 4 and GELU activation, and layer normalization. The model further includes a learned positional embedding over backbone atoms and a sinusoidal time embedding projected to the model dimension.

Flow matching training objective. The flow model is trained on the current latent structures using a flow matching objective. Given a latent structure x_1, we sample x_0 from the harmonic prior and align it to x_1. We then sample $t \sim \mathcal{U}(0,1)$, construct the interpolated structure $x_t = tx_1 + (1-t)x_0$,

and train the model to predict the corresponding velocity field. The model predicts a clean structure estimate $\hat{x}_1$, which is converted into a velocity prediction by

$$\hat{v}(x_t, t) = \frac{\hat{x}_1 - x_t}{1 - t}.$$

The flow loss is the mean squared error between this predicted velocity and the target velocity. For numerical stability near $t = 1$, the denominator $1 - t$ is lower-bounded by 0.05.

EB-FM guided sampling. Guided sampling first generates 5 candidate samples per observation using 3 unguided flow evaluations. Each candidate is rendered and compared to the observed image, and we select the candidate with the lowest rendering loss. Starting from this selected candidate, we then perform guided sampling using 10 additional flow evaluations with guidance strength 0.5.

Differentiable renderer. During training and guided sampling, we use a differentiable JAX-based renderer. Each structure is first centered by subtracting the midpoint of its atomic bounding box. The centered structure is then represented as a Gaussian mixture volume with one Gaussian centered at each backbone atom. We use a Gaussian variance parameter of $\sigma = 1$ for all experiments. The renderer produces 64×64 images with pixel size $1.2\,\text{Å}$, acceleration voltage $300\,\text{kV}$, defocus $5000\,\text{Å}$, spherical aberration $2.1\,\text{mm}$, and amplitude contrast 0.06 matching the parameters in the TEM simulated dataset. The image formation model includes a contrast transfer function through an astigmatic CTF model, and each rendered image is normalized to zero mean and unit variance.

Persistent latent optimization (EB-FM). The latent structures are initialized from the first frame and optimized directly with Adam using a learning rate of 0.1. We use an $L1$ rendering loss with rendering-loss weight 0.8 and flow-guidance weight 0.8. The backbone-continuity weight is set to 0.8 for $1a62_A$ and 0.3 for the remaining proteins. Each experiment uses $10{,}000$ frames and one image per frame. During the warmup phase, only the latent structures are updated, using the rendering loss and backbone-continuity regularization. The warmup phase is run for 50 epochs for $1a62_A$ and $3rjs_A$, and for 200 epochs for $1egw_C$ and $7r79_C$.

Optimization details. All experiments are trained for 3000 epochs with a main training batch size of 512, corresponding to 20 optimizer steps per epoch and $60{,}000$ latent-structure gradient steps in total. The flow model is trained with a learning rate of 2×10^{-4}. After warmup, the flow model is first trained on the current latent structures for 50 epochs. It is then retrained every epoch using a flow-training batch size of 256, corresponding to 40 gradient steps per flow epoch. In addition, every 300 epochs after warmup, we perform a longer flow retraining stage of 50 epochs, and at the end of training we perform one final 50-epoch flow-training stage. This results in $153{,}600$ total flow-model gradient steps for $1a62_A$ and $3rjs_A$, and $147{,}600$ total flow-model gradient steps for $1egw_C$ and $7r79_C$.

Evaluation protocol. For evaluation, we use unguided sampling from the trained flow model. For each protein, we generate $10{,}000$ structures, matching the number of conformations in the ground-truth MD trajectory. The number of unguided flow sampling steps is selected separately for each protein: 20 steps for $1a62_A$, 7 steps for $7r79_C$, 6 steps for $1egw_C$, and 20 steps for $3rjs_A$. The resulting generated ensemble is then compared against the corresponding ground-truth MD ensemble using the evaluation metrics described in the main text.

Proteins used. 3rjs_A is a small, compact dynein light-chain protein, representing a relatively constrained system. It has 89 amino acids in an a+b two layer structure. 1a62_A is the bacterial transcription termination factor Rho, with an HeH motif-like domain (alpha-array) and an OB-fold domain (beta-barrel), representing a moderately flexible system. It has 130 amino acids. 7r79_C is a calcium-binding protein with 78 amino acids in an alpha array structure, representing an intermediate-flexibility system with moderate conformational changes. 1egw_C is the human myocyte-specific enhancer factor 2A. It has 79 residues in an alpha+beta three layer structure, and represents a more flexible system with larger structural variability. This selection allows us to evaluate performance across increasing levels of conformational complexity, from near-rigid structures to more dynamic and potentially multi-modal ensembles.

Hardware. This research was supported by the NVIDIA Academic Grant Program using a single NVIDIA H100 GPU with 80 GB of memory for each protein. The total wall-clock runtime varied by protein: 1egw_C required approximately 10 hours, 7r79_C required 11 hours, 3rjs_A required 14 hours, and 1a62_A 23 hours.

D Cryo-EM Ablation

We report an ablation where the backbone continuity loss is omitted, which significantly degrades performance. Additionally we report sensitivity to different initializations and levels of noise found in traditional cryo-EM experimental data where there exist errors in CTF and pose parameters. We run EB-FM with different levels of noise in regards to CTF and poses as seen in Table 4. The results show that randomized initialization helps in 3 out of 4 proteins for most distributional metrics and for the flexibility correlation metric (RMSF r) although not for the absolute flexibility metrics (pairwise RMSD and RMSF). This suggests randomized initialization improves overall distributional accuracy, though it does not by itself increase the recovered flexibility magnitude. The results also show that small noise in pose and CTF values can also lead to better results. We hypothesize that this is caused by reducing the bias of the intentional imperfect forward model that we use.

Table 4: Sensitivity to initialization and hyperparameters. Randomized-init initializes each persistent latent with its own random deformation of the first frame; small-error-all applies a small pose/CTF error to every particle; p=0.1 and p=0.3 apply a pose/CTF error to 10% and 30% of the particles, respectively; w/o backbone-loss removes the backbone continuity loss. Best per protein in bold.

Protein	Method	Distributional accuracy			Flexibility		
		EMD RMSD Å	Coverage RMSD Å	Fidelity RMSD Å	Pairwise RMSD Å	RMSF Å	RMSF r
1a62_A	Ground truth	–	–	–	2.40	1.08	–
	EB-FM	1.62	1.37	1.27	1.58	0.77	0.96
	EB-FM randomized-init	**1.59**	**1.33**	**1.23**	1.43	0.71	**0.97**
	EB-FM small-error-all	1.62	1.39	1.24	1.46	0.77	0.95
	EB-FM p=0.1 error	1.70	1.45	1.37	1.66	0.92	0.93
	EB-FM p=0.3 error	2.06	1.66	1.78	**2.24**	1.40	0.83
	EB-FM w/o backbone-loss	3.00	2.87	2.45	1.66	**1.07**	0.78
7r79_C	Ground truth	–	–	–	3.21	1.89	–
	EB-FM	3.19	2.35	2.35	1.48	0.91	0.91
	EB-FM randomized-init	**2.86**	**2.30**	**2.27**	1.24	0.73	0.94
	EB-FM small-error-all	2.99	2.40	2.37	1.32	0.75	**0.94**
	EB-FM p=0.1 error	3.31	2.48	2.83	2.39	1.60	0.75
	EB-FM p=0.3 error	3.91	2.78	3.23	**2.75**	**1.68**	0.75
	EB-FM w/o backbone-loss	5.43	5.19	3.64	1.27	0.80	0.55
1egw_C	Ground truth	–	–	–	6.45	3.11	–
	EB-FM	6.22	4.44	4.19	**4.29**	**2.55**	0.93
	EB-FM randomized-init	5.56	4.36	4.10	3.54	2.07	**0.96**
	EB-FM small-error-all	**4.73**	**3.87**	**3.57**	3.64	2.13	0.96
	EB-FM p=0.1 error	5.42	4.27	4.25	3.53	2.13	0.95
	EB-FM p=0.3 error	5.27	4.16	4.14	4.00	2.42	0.95
	EB-FM w/o backbone-loss	9.12	8.73	5.05	2.35	1.46	0.57
3rjs_A	Ground truth	–	–	–	1.79	0.68	–
	EB-FM	**1.40**	**1.22**	**0.99**	1.10	0.49	0.95
	EB-FM randomized-init	1.43	1.23	1.02	1.10	0.49	**0.96**
	EB-FM small-error-all	1.44	1.25	1.00	0.97	0.45	0.95
	EB-FM p=0.1 error	1.50	1.27	1.11	1.26	**0.65**	0.95
	EB-FM p=0.3 error	1.73	1.36	1.31	**1.58**	0.84	0.87
	EB-FM w/o backbone-loss	3.25	3.05	2.72	0.97	0.57	0.79

E Cryo-EM Simulation Model

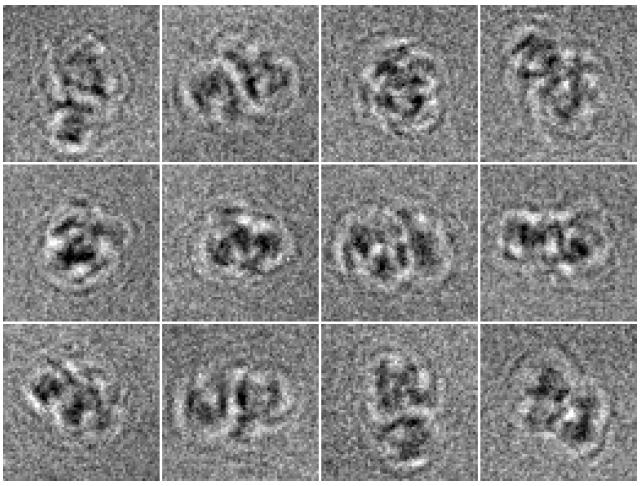

Figure 6: Samples of picked particle images from our Cryo-EM TEM simulated dataset.

We follow the same simulation procedure of Rosenbaum et al [20]. Molecular dynamics ensembles of large, highly dynamic proteins are scarce, hence we used proteins of 78–130 residues. These are far smaller than the particles that can be resolved experimentally, even more so when rendered from backbone atoms only, and at standard exposures their projections carry almost no signal. We therefore simulate at an elevated dose of 100,000 e/Å^2, 1,000 times higher than a reference exposure of 100 e/Å^2, which raises the SNR by a factor of $\sqrt{1000} \approx 32$. Since the backbone-only rendering retains roughly half of the scattering mass, we estimate that the task is comparable in difficulty to imaging proteins about 16 times larger (roughly 1,200–2,100 residues) at 100 e/Å^2.

Table 5: Simulation parameters used for cryo-EM data generation.

Parameter	**Value**
Number of picked particles	10,000
Dose	$100,000\ \mathrm{e}/\mathrm{Å}^2$
Image size	64×64 px
Image pixel size	1.2 Å
Magnification	25,000
Defocus	$5{,}000$ Å
Accelerating voltage	300 keV
Spherical aberration (CS)	2.1 mm
Amplitude contrast ratio	0.06

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 state the main contribution as Empirical Bayes Flow Matching, a method for learning a flow matching prior from indirect observations using a Monte Carlo stochastic-approximation EM procedure. The scope is also stated clearly: the cryo-EM experiments are performed on simulated data with known poses, and the paper frames the cryo-EM results as a proof of concept rather than a complete real-data reconstruction pipeline.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses important limitations, including the use of simulated cryo-EM data, the assumption of known poses, and the remaining challenge of mismatch between simulated and real cryo-EM data. The results section also discusses limitations for highly flexible proteins, where EB-FM underestimates the absolute scale of motion and obtains larger distributional errors

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs. It introduces a methodological framework and derives the training and sampling objectives, but the contribution is empirical and algorithmic rather than theoretical.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

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 provides the main algorithmic details, forward models, datasets, architectures, losses, optimizer settings, training schedules, renderer parameters, sampling procedures, and evaluation metrics needed to reproduce the reported experiments. The appendix contains additional implementation details for more specific details to allow others to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: [Yes]

   Justification: All code, data and evaluations will be released upon paper publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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 the controlled two-moons and MNIST settings, the cryo-EM datasets, the use of ATLAS molecular-dynamics conformations, the TEM simulation pipeline, the CryoJAX differentiable renderer,optimizer, learning rates,batch sizes, number of epochs, number of generated samples, and evaluation metrics. These details are sufficient to understand the experimental setting and interpret the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

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: The paper reports quantitative metrics and compares against baselines, but it does not currently report error bars, confidence intervals, statistical significance tests, or variation across random seeds. The reported metrics are computed over large generated and ground-truth ensembles, and remain stable across multiple evaluation results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: For each experiment, we report the hardware configuration and total wall-clock runtime.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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 work uses standard benchmark data, simulated inverse problems, molecular dynamics trajectories, and simulated cryo-EM observations. It does not involve human subjects, private personal data, surveillance, deception, etc.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work is fundamental research on learning generative priors for inverse problems, with cryo-EM used as a scientific example. Cryo-EM is an established paradigm in structural biology, and the paper focuses on simulated proof-of-concept experiments rather than a deployed system or application with direct societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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 model such as a large language model, image generator for human content, scraped dataset, or other asset with a clear misuse path.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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: [Yes]

    Justification: The paper cites the main assets including the CryoJax library used to render cryo-EM images, the TEM simulator used to create a simulated dataset, the ATLAS dataset containing Molecular dynamics used in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

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 current paper does not state that it releases a new dataset, benchmark, model checkpoint, or software package as a new public asset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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, user studies, annotation by human participants, or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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 human subjects, study participants, crowdsourced workers, or personal data, so IRB approval or equivalent human subjects review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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 research involved in the paper does not involve LLMS as any non standard or important components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.