HETEROGAUSS: HETEROGENEITY-AWARE GAUSSIAN REPRESENTATION FOR CRYO-EM RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-particle cryo-EM captures biomolecular structures with continuous conformational and discrete compositional variability, yet each particle image is an extremely noisy 2D projection of one unknown state. Recovering heterogeneous ensembles requires sharing information across particles while preserving structural correspondence, allowing molecular elements to move, appear or disappear, and adapt spatial capacity to unresolved heterogeneous regions. Existing voxel, implicit-field, and deformation-based models typically satisfy only part of these requirements, either lacking explicit cross-state correspondence, relying on an established reference, or retaining fixed spatial support throughout reconstruction. We propose HeteroGauss, a shared, state-conditioned anisotropic Gaussian representation for heterogeneous cryo-EM reconstruction. Mechanistically, each Gaussian preserves identity across molecular states while its position, scale, orientation, and amplitude vary with the inferred latent state, enabling explicit modeling of both conformational and compositional variability. Learnable spatial anchors coordinate local deformation, while a heterogeneity-aware adaptive topology mechanism reallocates primitives using reconstruction gradients and state-dependent motion. Technically, a batched differentiable CUDA renderer enables efficient optimization of particle-specific Gaussian structures. Experiments on CryoBench and EMPIAR-10180 show that HeteroGauss reconstructs continuous and compositional heterogeneity with strong accuracy and retains substantially more reproducible high-frequency detail on experimental data. Codes are attached for evaluation. Project Webpage.

1 INTRODUCTION

Heterogeneous single-particle cryogenic electron microscopy (cryo-EM) seeks to recover a molecular ensemble from large collections of low-dose 2D particle projections. Since each particle image records only one noisy view of one molecular state and contains little signal for direct structural interpretation, high-resolution reconstruction requires feature alignment and information sharing across thousands to millions of such images. However, the concerned particles may differ in conformation or molecular composition through motion, binding, or assembly. Averaging them indiscriminately erases biologically meaningful differences; separating them too finely leaves insufficient signal to resolve each state. Recovering this variation therefore requires sharing structural information without collapsing distinct molecular states, a central challenge highlighted by recent community evaluations Astore et al. (2025); Kinman et al. (2026).

In practice, how the demanded information is shared depends on the representation. Classical 3D classification pools particles into discrete volumes, approximating continuous motion with separate averages. Linear voxel models Punjani & Fleet (2021); Gilles & Singer (2025) and neural implicit fields Zhong et al. (2021) share information more broadly but do not explicitly track the same structural elements. Deformation models Punjani & Fleet (2023); Ducrocq et al. (2025) and structure-prior models Li et al. (2024) instead establish correspondence through a shared reference. Such transport supports coherent motion, but accommodates binding or dissociation less naturally because molecular content may change as well as move.

A representation for both forms of heterogeneity must therefore preserve correspondence across motion Luiten et al. (2024) while allowing individual density contributions to change with state. It

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

must also allocate spatial detail where it is needed: stable cores, flexible interfaces, and transient components differ in complexity, and their reconstruction requirements are not known in advance. Its explicit spatial support should therefore be refined during optimization Kerbl et al. (2023). This calls for a shared structure with locally adaptable geometry, variable molecular content, and spatial support that can evolve during learning.

Building on Gaussian mixture models for molecular variability Chen & Ludtke (2021); Chen et al. (2023), we propose HeteroGauss, which represents an ensemble through a shared bank of anisotropic Gaussians with state-dependent geometry and amplitude. On the one hand, Gaussian identity provides correspondence across states, while changes in position, shape, and orientation describe local deformation. On the other hand, amplitude modulation allows each element's density contribution to weaken or disappear as molecular content changes. These properties connect directly to cryo-EM observations through analytical Gaussian projection, yielding an explicit representation of both conformational and compositional variation.

To learn this variation from noisy images, HeteroGauss infers a latent molecular state for each particle and conditions Gaussian updates through learnable spatial anchors. The anchors coordinate nearby elements while permitting local changes in geometry and amplitude. Reconstruction gradients and state-dependent motion guide refinement of the shared Gaussian bank, directing capacity toward regions that need additional detail. A differentiable batched CUDA renderer efficiently projects the resulting particle-specific structures for optimization. Together, these mechanisms and techniques make the shared representation locally adaptable and practical to train at particle scale.

Our contributions are:

- HeteroGauss represents heterogeneous cryo-EM structures with state-conditioned anisotropic Gaussian fields, combining shared structural correspondence with local geometric and compositional variability.
- Adaptive topology guided by reconstruction gradients and state-dependent motion allocates Gaussian primitives according to spatial complexity.
- A differentiable batched CUDA renderer supports scalable reconstruction by projecting particle-specific Gaussian structures in a single call.
- Experiments on CryoBench and EMPIAR-10180 demonstrate strong reconstruction quality across diverse forms of molecular heterogeneity.

2 Related Work

2.1 Discrete and Statistical Volume Models

Likelihood-based 3D classification represents heterogeneity as a finite mixture of independently reconstructed volumes. Its interpretable classes suit well-separated assembly states, but approximate continuous motion with a few averages; adding classes divides an already weak particle signal among more reconstructions Astore et al. (2025); Kinman et al. (2026). Statistical volume models instead learn low-dimensional variation in voxel space. 3D Variability Analysis (3DVA) estimates principal components of structural change Punjani & Fleet (2021), while RECOVAR combines regularized covariance estimation with kernel regression to reconstruct volumes along a learned representation Gilles & Singer (2025). Consensus-space analysis can further compare independently learned landscapes Herreros et al. (2025b). These models share information on a fixed voxel grid, but may require several coupled components for large rotations or nonlinear paths.

2.2 Neural Density Models

Neural generative models decode density from a latent molecular state. CryoDRGN queries a coordinate network from each encoded particle Zhong et al. (2021). Subsequent models jointly infer pose Levy et al. (2025), reconstruct in real space Huang et al. (2024); Herreros et al. (2025a), or use neural radiance fields Qu et al. (2025b). Hydra represents heterogeneous density with a mixture of neural fields Levy et al. (2024), while CryoUNI learns a pretrained particle encoder Dai et al. (2026). Related generative formulations extend heterogeneous reconstruction to cryo-electron tomography Powell & Davis (2024); Rangan et al. (2024). This flexibility captures nonlinear conformational and

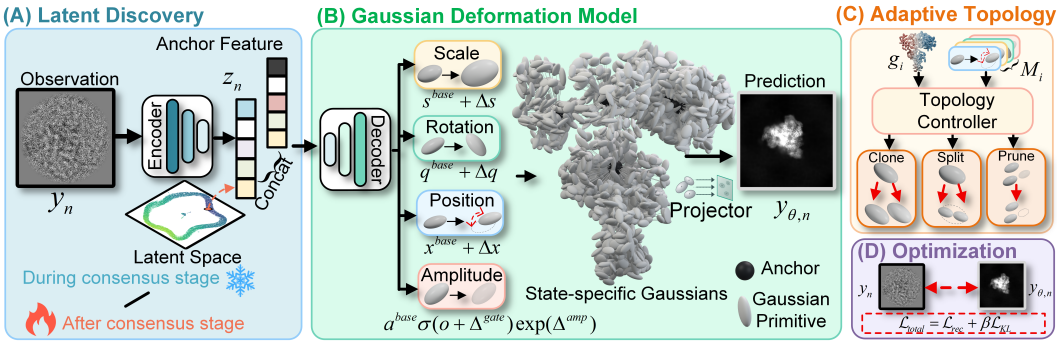

Figure 1: Overview of HeteroGauss. (A) A variational encoder infers a latent molecular state from each particle image. (B) The Gaussian Deformation Model maps the state to Gaussian parameters and predicts the image. (C) Adaptive Topology clones, splits, or prunes Gaussians using gradient and motion statistics. (D) Optimization combines frequency-weighted reconstruction and KL losses.

compositional variation, but implicit fields do not explicitly track structural elements. Correspondence must emerge from the image loss, while capacity remains in the coordinate network rather than being allocated directly to molecular regions.

2.3 Motion-Based Models

Motion-based methods make correspondence explicit by deforming a shared reference. 3DFlex transports a canonical density map Punjani & Fleet (2023), while cryoSPHERE learns structured deformations of a shared representation Ducrocq et al. (2025). Others impose physics-informed geometric regularization Prins et al. (2024) or incorporate atomic and protein-graph priors Li et al. (2024); Krook et al. (2026). By mapping different conformations back to a common reference, these methods let particles from multiple states jointly constrain the same structural regions, improving the recovery of flexible detail. However, transporting existing density accommodates content changes less naturally than motion, and the reference predetermines where density can be represented.

2.4 Gaussian Representations and Rendering

Gaussian representations provide explicit density elements whose positions, shapes, and amplitudes can vary across molecular states. E2gmm captures conformational and compositional variation Chen & Ludtke (2021); deep high-resolution GMMs integrate molecular models into reconstruction Chen et al. (2023), with subsequent work recovering molecular-model series Chen (2025). CryoSPIRE instead discovers molecular parts with a coarse GMM and organizes isotropic Gaussians into a part-based hierarchy Shekarforoush et al. (2026). Building on differentiable 3D Gaussian Splatting Kerbl et al. (2023), GEM Qu et al. (2025a) and CryoSplat Chen & Ling (2026) adapt Gaussian rendering to cryo-EM, yet in the homogeneous scene.

3 Method

Heterogeneous reconstruction must explain how particle images arise from different molecular states while pooling their evidence into a common structural representation. We address these coupled requirements by learning a latent variable for each particle and using it to condition a shared Gaussian density field. As shown in Fig. 1, a variational encoder infers molecular states from the images, the Gaussian deformation model maps each state to a corresponding 3D density through state-dependent geometry and amplitudes, and a differentiable projection connects both components to the observations. To make this connection explicit, we first establish the image-formation model, then explain how the shared representation is learned, deformed, and refined during training.

3.1 Cryo-EM Image Formation and Latent-State Inference

Let $y_n \in \mathbb{R}^{D\times D}$ denote the n-th particle image, with estimated orientation $R_n \in SO(3)$, in-plane translation $t_n \in \mathbb{R}^2$, and contrast transfer function (CTF) C_n, all of which are held fixed during

reconstruction. The unknown molecular state is initially represented by a latent variable $z_n \in \mathbb{R}^L$, which indexes a 3D density $\rho(\cdot; z_n)$ and encodes conformational and compositional variation. Following the linear cryo-EM image-formation model Sigworth (2015), the density is sequentially rotated by R_n, integrated along the beam, and shifted by t_n, before that microscope filtering and noise produce the image:

$$\begin{aligned} p_n &= \mathcal{P}_{R_n,t_n}[\rho(\cdot; z_n)], \\ y_n &= h_n * p_n + \epsilon_n. \end{aligned} \tag{1}$$

Here, $\mathcal{P}_{R_n,t_n}$ is the pose-dependent projection, h_n is the point-spread function corresponding to C_n, $*$ denotes convolution, and ϵ_n represents additive observation noise in the real-space particle image.

Although Eq. (1) accounts for pose and microscope effects, explaining structural differences also requires inferring which molecular state generated each image. Since state labels are unavailable, we infer these latent states through reconstruction with a variational encoder Zhong et al. (2021):

$$\begin{aligned} q_\phi(z_n \mid y_n) &= \mathcal{N}\big(\mu_n, \mathrm{diag}(\sigma_n^2)\big), \\ z_n &= \mu_n + \sigma_n \odot \varepsilon_n, \qquad \varepsilon_n \sim \mathcal{N}(0, I). \end{aligned} \tag{2}$$

where the encoder ϕ predicts posterior mean μ_n and standard deviation σ_n from the translation-corrected, CTF-phase-flipped particle image, and $\odot$ denotes elementwise multiplication. Reparameterized sampling lets reconstruction gradients pass through z_n to the encoder, while a standard normal prior regularizes the posterior. Because the sampled variable conditions a shared decoder whose projection must explain y_n, latent inference and density reconstruction learn together to distinguish molecular variation while pooling structural evidence across particles.

3.2 State-Conditioned Gaussian Representation

To express the variation without losing correspondence across states, the decoder uses a shared bank of P Gaussian primitives. Each primitive retains its identity with the center $x_i(z_n) \in \mathbb{R}^3$ controlling location, covariance $\Sigma_i(z_n) \in \mathbb{R}^{3\times3}$ controlling extent and orientation, and amplitude $a_i(z_n) > 0$ controlling integrated density. At spatial coordinate $x \in \mathbb{R}^3$, the normalized kernel and resulting density are given as follows.

$$G_i(x; z_n) = \frac{\exp\big[-\frac{1}{2}(x - x_i(z_n))^\top \Sigma_i(z_n)^{-1}(x - x_i(z_n))\big]}{(2\pi)^{3/2}|\Sigma_i(z_n)|^{1/2}}, \tag{3}$$

$$\rho_\theta(x; z_n) = \sum_{i=1}^{P} a_i(z_n) G_i(x; z_n), \tag{4}$$

where θ parameterizes the bank and its state-dependent updates. Unlike CryoSPIRE, which uses isotropic Gaussians Shekarforoush et al. (2026), we employ anisotropic primitives because local density can extend diversely along different directions. Each primitive can therefore elongate and rotate with the structure, allowing directional deformation to be represented through changes in shape as well as displacement. Specifically, the positive scale vector $s_i(z_n) \in \mathbb{R}^3_{>0}$ specifies the Gaussian's extent along its three principal axes, while the unit quaternion $q_i(z_n) \in \mathbb{R}^4$ represents their orientation in 3D space. Together, these parameters determine the covariance:

$$\Sigma_i(z_n) = Q(q_i(z_n))\,\mathrm{diag}\big(s_i(z_n)^2\big)\,Q(q_i(z_n))^\top, \tag{5}$$

where Q converts the unit quaternion into a rotation matrix.

The state-dependent Gaussian parameters introduced above are obtained by applying updates conditioned on z_n to a shared set of base parameters, as illustrated in Fig. 1B. We first learn these base parameters $\{x_i^{\mathrm{base}}, s_i^{\mathrm{base}}, q_i^{\mathrm{base}}, a_i^{\mathrm{base}}\}_{i=1}^P$ through consensus reconstruction, with the encoder frozen and state-dependent updates disabled. Once this shared structure is established, the decoder maps z_n to updates that determine the centers $x_i(z_n)$, scales $s_i(z_n)$, orientations $q_i(z_n)$, and amplitudes $a_i(z_n)$ for the corresponding molecular state. Technically, the base parameters and decoder remain jointly learnable during heterogeneous reconstruction.

To coordinate these state-dependent updates across neighboring primitives, we map z_n to features on $M \ll P$ learnable spatial anchors, from which each Gaussian interpolates a local feature to predict its parameter updates. Centers blend nearby anchor rotations and translations with a Gaussian-specific residual, while scale and orientation updates accommodate local deformation. This coordinates regional motion without requiring a molecular-part segmentation, with the update rules detailed in Appendix A.2.

The geometric updates above allow each Gaussian to deform across molecular states while preserving its integrated density at a fixed amplitude. To also represent components that appear or disappear as molecular composition changes, the same local features are used to modulate the base amplitude:

$$a_i(z_n) = a_i^{\text{base}} \exp(\Delta_i^{\text{amp}}(z_n)) \,\text{sigmoid}\big(o + \Delta_i^{\text{gate}}(z_n)\big)\,. \tag{6}$$

Here, $a_i^{\text{base}} > 0$ is the consensus-initialized amplitude, the exponential factor adjusts density strength, and the sigmoid gate suppresses absent contributions toward zero. This multiplicative form preserves positive density contributions while allowing their strength to vary relative to the shared base amplitude. The positive bias o initializes the gate near one, allowing heterogeneous reconstruction to start close to the consensus. Together, geometry and amplitude express conformational and compositional variation on the same shared bank.

3.3 Gaussian Projection and Optimization

To predict a two-dimensional particle image from the decoded three-dimensional density, we project the Gaussian primitives under the particle's estimated pose. Specifically, R_n rotates each Gaussian into the camera coordinate system, transforming both its center and covariance. Integration along the viewing direction then produces a two-dimensional Gaussian on the detector plane, whose position is adjusted by the in-plane translation t_n. With $J = [I_2\ 0] \in \mathbb{R}^{2\times 3}$ selecting the two detector-plane coordinate components, the projected center and covariance are given as

$$\mu_{ni} = JR_n x_i(z_n) + t_n, \qquad \Lambda_{ni} = JR_n \Sigma_i(z_n) R_n^\top J^\top. \tag{7}$$

The projected center μ_{ni} and covariance Λ_{ni} define a two-dimensional kernel

$$\widetilde{G}_{ni}(u; z_n) = \frac{\exp\big[-\frac{1}{2}(u-\mu_{ni})^\top \Lambda_{ni}^{-1}(u-\mu_{ni})\big]}{2\pi|\Lambda_{ni}|^{1/2}}, \tag{8}$$

where $u \in \mathbb{R}^2$ is a detector-plane coordinate. Although the projected shape and peak density depend on the viewing direction, projection preserves each primitive's integrated density, equal to its amplitude $a_i(z_n)$, for a fixed molecular state.

Summing the projected contributions and convolving with the microscope point-spread function gives the noise-free image predicted by the model, in direct correspondence with Eq. (1):

$$\begin{aligned} p_{\theta,n}(u) &= \sum_{i=1}^{P} a_i(z_n)\widetilde{G}_{ni}(u; z_n), \\ y_{\theta,n} &= h_n * p_{\theta,n}. \end{aligned} \tag{9}$$

Appendix A.4 explains how this orthographic, additive projection differs from the perspective projection and alpha compositing of conventional 3D Gaussian Splatting.

Given the predicted $y_{\theta,n}$, we can now compare it with y_n to jointly learn the latent states and Gaussian representation. We evaluate this discrepancy in the Hartley domain, where frequency-dependent weighting allows us to emphasize fine structural detail. Specifically, letting $\mathcal{H}$ denote the two-dimensional discrete Hartley transform, we elaborate the loss $\mathcal{L}_{\text{total}}$ for optimization as follows.

$$\begin{aligned} \mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{KL}}, \\ \mathcal{L}_{\text{rec}} &= \frac{1}{|\Omega|}\sum_{k\in\Omega} w_k \left|\mathcal{H}[y_n](k) - \mathcal{H}[y_{\theta,n}](k)\right|^2, \\ \mathcal{L}_{\text{KL}} &= \frac{1}{|\Omega|} D_{\text{KL}}(q_\phi(z_n \mid y_n) \,\|\, \mathcal{N}(0, I))\,. \end{aligned} \tag{10}$$

Here, the circular mask Ω imposes a common radial frequency cutoff, while the unit-mean radial weights w_k emphasize higher frequencies to promote fine-detail recovery. The coefficient β controls regularization toward the standard normal prior. We average this objective over each mini-batch and gradually introduce frequency weighting and KL regularization, with the training stages and schedules detailed in Appendix A.6.

3.4 Adaptive Topology and Batched CUDA Rendering

Parameter updates adapt current Gaussian primitives, but a fixed spatial allocation may leave some regions under-resolved while retaining primitives that contribute little. We therefore adapt the shared bank through cloning, splitting, and pruning, guided by reconstruction and state-dependent statistics. For densification, we track moving averages of the center-gradient magnitude g_i and positional variation M_i. The latter measures variation across sampled latent states through the root-mean-square deviation of a primitive's center from its batch mean. Primitives with g_i above a threshold become densification candidates, and when their number exceeds the update budget, we rank them by

$$D_i = g_i + \lambda_m \frac{M_i}{\max(\max_j M_j, \epsilon)}, \tag{11}$$

where $\lambda_m \geq 0$ controls the contribution of normalized positional variation to the ranking, and $\epsilon > 0$ ensures numerical stability. Selected primitives are cloned or split according to their spatial scale, while pruning removes primitives with jointly weak amplitude, gradient, and positional variation. These operations adjust spatial capacity while keeping the Gaussian bank shared across states, with complete criteria detailed in Appendix A.3.

As the bank grows during topology adaptation, rendering each particle's distinct Gaussian set in a separate call repeats kernel launches and limits mini-batch throughput. To reduce this overhead, we design a CUDA renderer that processes B particle-specific Gaussian sets in one call, using particle-aware tile indexing to preserve each image's state and pose, as well as a batched backward pass to return the corresponding gradients. This batches both projection and gradient computation while retaining the additive cryo-EM image-formation model (Appendix A.5).

Remarks. CryoSPIRE Shekarforoush et al. (2026) and HeteroGauss both use state-dependent Gaussians and anchor-based coordination, but HeteroGauss extends this foundation with greater geometric flexibility and adaptive spatial capacity. Whereas CryoSPIRE relies on part discovery to construct an isotropic Gaussian hierarchy, our spatial anchors coordinate deformation within a shared bank without explicit part segmentation. State-dependent anisotropic covariances further enable each primitive to change its directional extent and orientation, directly capturing local shape variations that isotropic primitives cannot express individually. Coupled with heterogeneity-aware topology adaptation, this allows HeteroGauss to refine both the geometry of existing primitives and their spatial allocation as reconstruction demands evolve.

4 Experiments and Results

Due to space constraints, the datasets, baselines, and evaluation metrics are given in Appendix B.1.

4.1 Reconstruction on CryoBench

HeteroGauss achieves state-of-the-art reconstruction accuracy on all three evaluated CryoBench datasets, ranking first in both mean and median AU-FSC (Table 1). We attribute this performance across conformational and compositional heterogeneity to a shared representation that preserves structural correspondence while allowing geometry and density contributions to vary with state. To examine this explanation, we analyze where the reconstruction gains arise and whether the learned states and amplitudes reflect the underlying molecular variation.

The advantage is clearest when reconstruction must preserve structure throughout motion. On IgG-1D, HeteroGauss raises mean AU-FSC from the strongest baseline's 0.386 to 0.398, while on the more articulated IgG-RL the gain increases from 0.363 to 0.380. The larger improvement on IgG-RL is consistent with the value of coordinating position and orientation changes within a shared representation. This advantage is visible in the flexible regions of Fig. 2, where HeteroGauss preserves a defined hinge and a connected moving Fab while competing reconstructions exhibit blurred connections or detached density. The quantitative and structural results therefore agree that HeteroGauss better preserves molecular continuity as the motion becomes less constrained.

The same formulation also achieves the best mean and median AU-FSC on Ribosembly, reaching 0.430 and 0.432, respectively, compared with 0.429 and 0.430 for the strongest baseline. Here,

Table 1: Per-conformation AU-FSC on CryoBench, summarized by the mean with standard deviation and the median. Higher is better for the metric, with a maximum of 0.5. Best results are shown in bold and second-best results are underlined.

	IgG-1D		IgG-RL		Ribosembly	
Methods	Mean (std)	Med	Mean (std)	Med	Mean (std)	Med
3D Classification	0.295(0.014)	0.292	0.308(0.010)	0.308	0.278(0.070)	0.276
CryoDRGN	0.369(0.001)	0.369	0.354 (0.008)	0.353	0.408 (0.023)	0.409
3DFlex	0.320(0.009)	0.316	0.340(0.006)	0.339	–	–
3DVA	0.351(0.003)	0.352	0.341(0.006)	0.341	0.376(0.039)	0.375
RECOVAR	<u>0.386</u>(0.005)	<u>0.388</u>	<u>0.363</u>(0.011)	<u>0.364</u>	<u>0.429</u>(0.018)	<u>0.430</u>
CryoHype	0.351(0.002)	0.351	0.328(0.008)	0.329	0.403(0.019)	0.405
HeteroGauss	**0.398**(0.001)	**0.398**	**0.380**(0.010)	**0.380**	**0.430**(0.014)	**0.432**

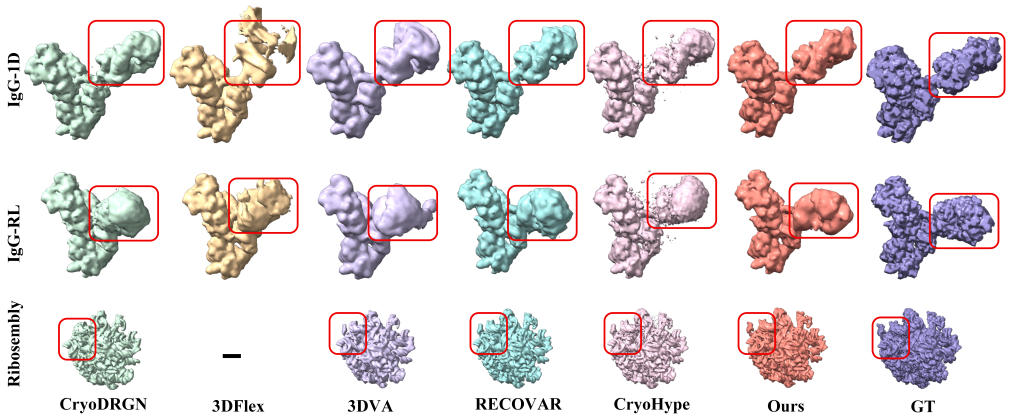

Figure 2: Representative reconstructions on IgG-1D, IgG-RL, and Ribosembly. Red boxes highlight heterogeneous regions; the dash denotes the unavailable 3DFlex result on Ribosembly.

state-dependent amplitudes recover assembly-dependent density changes while retaining the shared ribosomal core, and the supplementary analysis confirms that these changes localize to the appropriate molecular regions (Appendix B.4). Together with the motion benchmarks, this result supports our representation's ability to capture both structural deformation and changing molecular content within a single framework.

To assess whether latent-state inference resolves the underlying conformational variation, Fig. 3 pairs the learned embeddings with four decoded structures from each dataset. The IgG-1D embedding follows a near-circular arrangement ordered by Fab angle, whereas IgG-RL forms a less regular landscape reflecting its more complex articulated motion. Across the sampled states, the moving Fab changes configuration while the antibody core remains stable, showing that the encoder distinguishes conformational states whose decoded structures exhibit coherent molecular motion.

4.2 Experimental Reconstruction on EMPIAR-10180

Having established reconstruction accuracy against known structures, we next evaluate whether HeteroGauss recovers detailed structure from the experimental spliceosome ensemble. Because matched ground truth is unavailable, we first compare molecular continuity, then verify the recovered detail through independent half-map agreement and examine how resolution varies across stable and flexible regions. Under the same filtering and density threshold, Fig. 4A shows that HeteroGauss recovers connected internal features while preserving the overall spliceosome architecture. These features are less well resolved in competing maps, which exhibit smoothing, fragmentation, or granular density, suggesting that HeteroGauss better retains fine structure within the heterogeneous assembly.

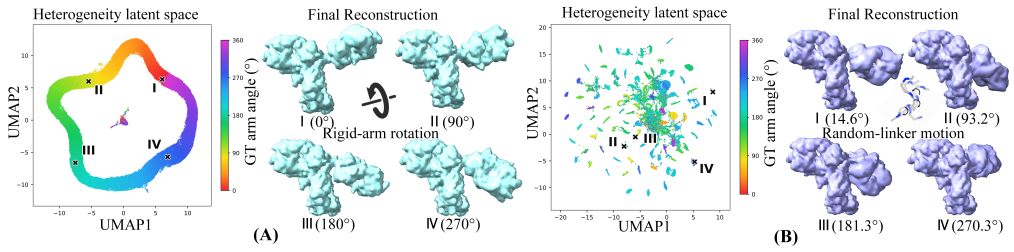

Figure 3: Latent embeddings and decoded structures for (A) IgG-1D and (B) IgG-RL. Colors indicate ground-truth arm angle; I–IV identify the decoded examples.

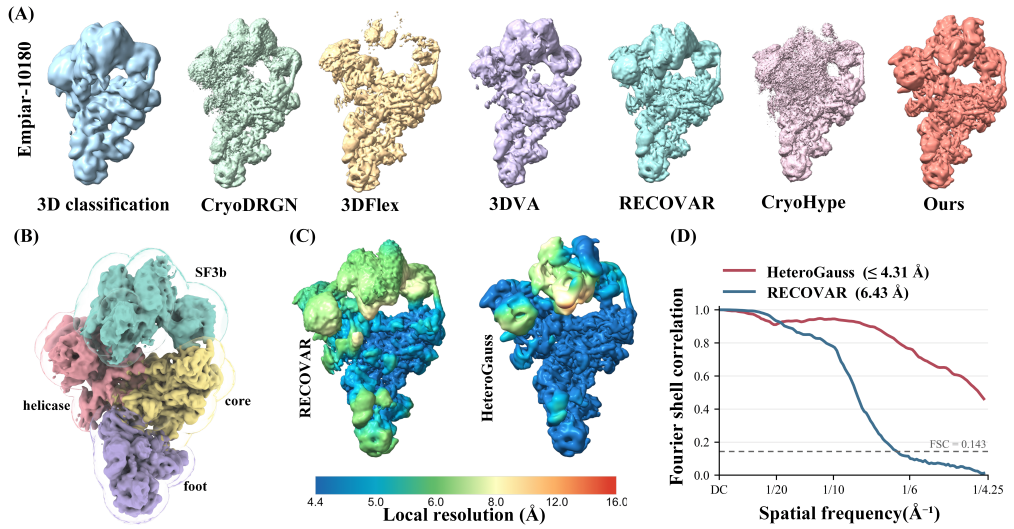

Figure 4: Reconstruction and resolution analysis on EMPIAR-10180. (A) Reconstructed volumes. (B) Annotated reference. (C) Local-resolution maps of RECOVAR and HeteroGauss. (D) Masked gold-standard FSC.

To verify that this finer structure is supported independently by the data, we compare the gold-standard FSC curves in Fig. 4D. Although HeteroGauss and RECOVAR have similar agreement at low frequency, HeteroGauss retains substantially higher correlation as spatial frequency increases. RECOVAR reaches the FSC$= 0.143$ threshold at 6.43 Å, whereas HeteroGauss remains above 0.45 at the highest sampled frequency, supporting a resolution of ≤ 4.31 Å. This separation confirms that the improved structural definition in panel A is accompanied by more reproducible high-frequency detail. Having established this global gain, we examine the local-resolution maps in panel C to determine where the additional detail is retained. The domain annotations in panel B provide the anatomical reference for this analysis, distinguishing the core and foot from SF3b and the helicase, whose concerted motion was reported by multi-body refinement Nakane et al. (2018). This motivates closer examination of SF3b and the helicase, where retaining detail is particularly demanding because of their mobility. Compared with RECOVAR, HeteroGauss improves local resolution across all four domains, recovering finer detail not only in the core and foot but also in the flexible SF3b and helicase regions. Together, the structural comparison, global FSC, and local-resolution analysis demonstrate that HeteroGauss's reconstruction advantage extends across both stable and mobile regions of the heterogeneous assembly.

4.3 Ablation Studies

To establish how the proposed design supports these reconstruction gains, we compare variants that remove anchor-rigid transport, disable topology adaptation, replace positive amplitude modulation with signed additive updates, or remove frequency weighting, while keeping the remaining training

Table 2: Component analysis using AU-FSC, reported as mean with standard deviation and median. Each row changes one property of the full model under the same protocol.

	IgG-1D		IgG-RL		Ribosembly	
Variant	Mean (std)	Median	Mean (std)	Median	Mean (std)	Median
Full HeteroGauss	**0.398(0.001)**	**0.398**	**0.380(0.010)**	**0.380**	**0.430(0.014)**	**0.432**
w/o anchor-rigid transport	0.393(0.003)	0.394	0.367(0.009)	0.366	0.427(0.015)	0.429
additive amplitude update	0.391(0.002)	0.392	0.367(0.012)	0.367	0.426(0.014)	0.427
w/o adaptive topology	0.379(0.003)	0.380	0.355(0.010)	0.354	0.399(0.013)	0.400
w/o frequency weighting	0.389(0.003)	0.389	0.368(0.009)	0.368	0.423(0.015)	0.425

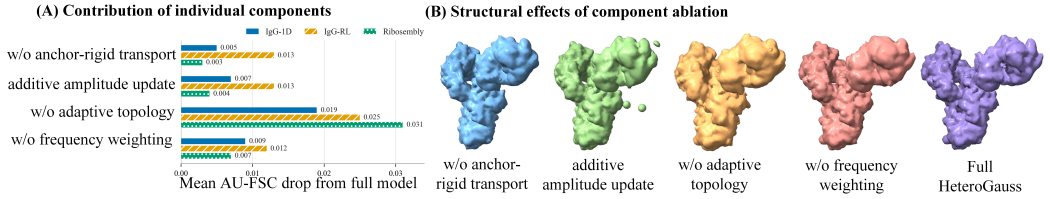

Figure 5: Component ablation. (A) Decrease in mean AU-FSC relative to the full model across three datasets. (B) Representative IgG-1D reconstructions for each variant.

and evaluation settings fixed. Table 2 and Fig. 5 show that each change reduces accuracy across all three datasets, with effects that reflect the role of the corresponding component.

Without anchor-rigid transport, the largest loss occurs on IgG-RL, where coordinated changes in position and orientation are essential, while the effect is smaller on Ribosembly with limited geometric motion. Disabling topology adaptation produces the largest degradation on every dataset and visibly broadens structural features, supporting the need to refine spatial capacity during reconstruction. Signed additive amplitudes introduce detached density and reduce accuracy, whereas removing frequency weighting weakens local detail. Additionally, experiments on the number of Gaussian primitives can be found in Appendix B.2, while Appendix C documents unsuccessful design variants encountered during development. Together, these ablations and failure cases informed the final architecture and support the complementary roles of its components.

5 Conclusion

We presented HeteroGauss, a heterogeneous cryo-EM reconstruction framework built on a shared, state-conditioned anisotropic Gaussian representation. By preserving correspondence across molecular states while allowing geometry and amplitude to vary, HeteroGauss captures both conformational and compositional heterogeneity within a single representation. Learnable spatial anchors and heterogeneity-aware topology adaptation coordinate local deformation and refine spatial capacity, while batched differentiable rendering enables efficient optimization. HeteroGauss achieves state-of-the-art reconstruction accuracy on the evaluated CryoBench datasets and recovers reproducible high-frequency detail on EMPIAR-10180. Our work demonstrates that an explicit, adaptive structural representation can share information across noisy particle images while resolving the molecular differences within a heterogeneous ensemble. **Limitation and future work.** The current formulation uses pre-estimated particle poses, which remain fixed during training, and therefore does not yet perform *ab initio* Levy et al. (2025) heterogeneous reconstruction. Within this setting, HeteroGauss provides a shared representation of conformational and compositional variation with adaptive spatial capacity and efficient differentiable projection. These capabilities offer a foundation for jointly learning particle poses and molecular structure. Future work will dive deeper into joint optimization of poses, latent variables, and Gaussian parameters, with the goal of extending the framework to *ab initio* reconstruction from heterogeneous particle images.

AI USE STATEMENT

We acknowledge the assistance of GPT-5.6 with grammatical correction and language polishing. The research design, experimental analysis, and central conclusions of this work were not generated by LLMs.

REPRODUCIBILITY STATEMENT

To support reproduction of the reported results, the supplementary material provides the code used in this work, while the appendix details the model implementation, training protocol, and experimental settings. To complement these technical resources and the figures presented in the paper, we provide additional visualizations on our anonymous project webpage, enabling a more detailed visual assessment of the reconstructed structures.

REFERENCES

Miro A Astore, Geoffrey Woollard, David Silva-Sánchez, Wenda Zhou, Mykhailo Kopylov, Khanh Dao Duc, Roy R Lederman, Yilai Li, Yi Zhou, Jing Yuan, et al. The inaugural flatiron institute cryo-em conformational heterogeneity challenge. *bioRxiv*, 2025.

Muyuan Chen. Building molecular model series from heterogeneous cryoem structures using gaussian mixture models and deep neural networks. *Communications Biology*, 8(1):798, 2025.

Muyuan Chen and Steven J Ludtke. Deep learning-based mixed-dimensional gaussian mixture model for characterizing variability in cryo-em. *Nature methods*, 18(8):930–936, 2021.

Muyuan Chen, Bogdan Toader, and Roy Lederman. Integrating molecular models into cryoem heterogeneity analysis using scalable high-resolution deep gaussian mixture models. *Journal of molecular biology*, 435 (9):168014, 2023.

Suyi Chen and Haibin Ling. Cryosplat: Gaussian splatting for cryo-em homogeneous reconstruction. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 57985–58007, 2026.

Haizhao Dai, Qihe Chen, Lingqi Li, Yingjun Shen, Zhenyang Xu, Minzhang Li, Yufan Xie, Jingjing Zheng, Zhijie Liu, Liping Sun, et al. Cryo-em as latent structural landscape microscopy. *bioRxiv*, pp. 2026–04, 2026.

Gabriel Claude Jean Ducrocq, Lukas Grunewald, Sebastian Westenhoff, and Fredrik Lindsten. cryosphere: Single-particle heterogeneous reconstruction from cryo em. In *International Conference on Learning Representations*, volume 2025, pp. 16774–16787, 2025.

Marc Aurele Gilles and Amit Singer. Cryo-em heterogeneity analysis using regularized covariance estimation and kernel regression. *Proceedings of the National Academy of Sciences*, 122(9):e2419140122, 2025.

David Herreros, Carlos Perez Mata, Chari Noddings, Deli Irene, James Krieger, David A Agard, Ming-Daw Tsai, Carlos Oscar Sanchez Sorzano, and Jose Maria Carazo. Real-space heterogeneous reconstruction, refinement, and disentanglement of cryoem conformational states with hetsiren. *Nature communications*, 16 (1):3751, 2025a.

David Herreros, Carlos Perez Mata, Carlos Oscar Sanchez Sorzano, and Jose Maria Carazo. Merging conformational landscapes in a single consensus space with flexconsensus algorithm. *Nature methods*, 22(10): 2118–2126, 2025b.

Yue Huang, Chengguang Zhu, Xiaokang Yang, and Manhua Liu. High-resolution real-space reconstruction of cryo-em structures using a neural field network. *Nature Machine Intelligence*, 6(8):892–903, 2024.

Minkyu Jeon, Rishwanth Raghu, Miro Astore, Geoffrey Woollard, Ryan Feathers, Alkin Kaz, Sonya M Hanson, Pilar Cossio, and Ellen D Zhong. Cryobench: Diverse and challenging datasets for the heterogeneity problem in cryo-em. *Advances in Neural Information Processing Systems*, 37:89468–89512, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Laurel F. Kinman, Andrew V. Grassetti, Maria V. Carreira, and Joseph H. Davis. Heterogeneous reconstruction algorithms for cryoem achieve limited particle classification accuracy on real benchmark datasets. *bioRxiv*, 2026.

Jonathan Krook, Axel Janson, Joakim Andén, Melanie Weber, and Ozan Öktem. Protein graph neural networks for heterogeneous cryo-em reconstruction. *arXiv preprint arXiv:2602.21915*, 2026.

Axel Levy, Rishwanth Raghu, David Shustin, Adele R Peng, Huan Li, Oliver B Clarke, Gordon Wetzstein, and Ellen D Zhong. Mixture of neural fields for heterogeneous reconstruction in cryo-em. *Advances in Neural Information Processing Systems*, 37:56988–57017, 2024.

Axel Levy, Rishwanth Raghu, J Ryan Feathers, Michal Grzadkowski, Frédéric Poitevin, Jake D Johnston, Francesca Vallese, Oliver Biggs Clarke, Gordon Wetzstein, and Ellen D Zhong. Cryodrgn-ai: neural ab initio reconstruction of challenging cryo-em and cryo-et datasets. *Nature Methods*, 22(7):1486–1494, 2025.

Yilai Li, Yi Zhou, Jing Yuan, Fei Ye, and Quanquan Gu. Cryostar: leveraging structural priors and constraints for cryo-em heterogeneous reconstruction. *Nature Methods*, 21(12):2318–2326, 2024.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pp. 800–809. IEEE, 2024.

Takanori Nakane, Dari Kimanius, Erik Lindahl, and Sjors HW Scheres. Characterisation of molecular motions in cryo-em single-particle data by multi-body refinement in relion. *elife*, 7:e36861, 2018.

Clemens Plaschka, Pei-Chun Lin, and Kiyoshi Nagai. Structure of a pre-catalytic spliceosome. *Nature*, 546 (7660):617–621, 2017.

Barrett M Powell and Joseph H Davis. Learning structural heterogeneity from cryo-electron sub-tomograms with tomodrgn. *Nature Methods*, 21(8):1525–1536, 2024.

Victor Prins, Willem Diepeveen, Erik J Bekkers, and Ozan Öktem. Physics-informed geometric regularization of heterogeneous reconstructions in cryo-em. In *ICLR 2024 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2024.

Ali Punjani and David J Fleet. 3d variability analysis: Resolving continuous flexibility and discrete heterogeneity from single particle cryo-em. *Journal of structural biology*, 213(2):107702, 2021.

Ali Punjani and David J Fleet. 3dflex: determining structure and motion of flexible proteins from cryo-em. *Nature methods*, 20(6):860–870, 2023.

Huaizhi Qu, Xiao Wang, Gengwei Zhang, Jie Peng, and Tianlong Chen. Gem: 3d gaussian splatting for efficient and accurate cryo-em reconstruction. *arXiv preprint arXiv:2509.25075*, 2025a.

Huaizhi Qu, Xiao Wang, Yuanyuan Zhang, Sheng Wang, William Stafford Noble, and Tianlong Chen. Cryonerf: reconstruction of homogeneous and heterogeneous cryo-em structures using neural radiance field. *bioRxiv*, pp. 2025–01, 2025b.

Ramya Rangan, Ryan Feathers, Sagar Khavnekar, Adam Lerer, Jake D Johnston, Ron Kelley, Martin Obr, Abhay Kotecha, and Ellen D Zhong. Cryodrgn-et: deep reconstructing generative networks for visualizing dynamic biomolecules inside cells. *Nature Methods*, 21(8):1537–1545, 2024.

Shayan Shekarforoush, David Lindell, Marcus Brubaker, and David Fleet. Reconstructing heterogeneous biomolecules via hierarchical gaussian mixtures and part discovery. *Advances in Neural Information Processing Systems*, 38:75568–75587, 2026.

Fred J Sigworth. Principles of cryo-em single-particle image processing. *Journal of Electron Microscopy*, 65 (1):57–67, 2015.

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.

APPENDIX

The appendix connects the implementation of HeteroGauss to its experimental evaluation and the design choices examined during development. Appendix A explains how latent states update the shared Gaussian representation and how the resulting structures are projected and trained, providing the methodological details needed for reproduction. Building on this description, Appendix B first establishes the datasets, baselines, and evaluation protocol used in the main experiments, then presents supplementary analyses of Gaussian capacity, rendering efficiency, and the relationship between amplitude changes and molecular composition. Finally, Appendix C examines two unsuccessful variants to clarify the optimization choices behind the final model and the limits of these observations.

A TECHNICAL DETAILS

This section supplies the implementation details behind the method in Secs. 3.2–3.4. We begin with the complete training pipeline and then develop each operation it invokes, from decoding and adapting the shared Gaussian bank to projecting it for particle images. We close with the schedules and architecture settings needed to reproduce the reported experiments.

A.1 OVERALL TRAINING PIPELINE

Algorithm 1 implements the three-stage procedure described in Sec. 3.2. To make the preprocessing explicit, let $y_n^{\mathrm{c}} = \mathrm{CenterImage}(y_n, t_n)$ denote the translation-corrected observation; its corresponding projection is rendered with zero in-plane translation, while R_n and C_n remain fixed. The encoder receives $\mathrm{sign}(C_n) \odot \mathcal{H}[y_n^{\mathrm{c}}]$, whereas the reconstruction target remains $\mathcal{H}[y_n^{\mathrm{c}}]$, so CTF phase flipping affects inference without changing the image-formation loss. For the real, even CTF used here, multiplication by C_n in Hartley space implements the convolution in Eq. (9).

In the algorithm, particle-indexed operations run in parallel over $n = 1, \ldots, B$, with $Z = [z_1, \ldots, z_B]^\top$ collecting the sampled states and Θ collecting their decoded Gaussian parameters. The parameters ϕ and θ belong to the encoder and shared Gaussian decoder, respectively, while $\mathcal{S} = (g, S, A, M)$ stores topology statistics and $\mathcal{O}$ stores the AdamW optimizer state. Training progress t is measured in epochs, with boundaries E_c and E_a marking the ends of consensus reconstruction and topology adaptation, respectively, and ℓ_t denoting the scheduled learning rates. When $\mathrm{AdaptBank}$ changes the shared bank, Rebind transfers optimizer state for retained parameters; the topology-freezing condition $\mathrm{AtCapacity}$ prevents further changes after the configured limit is reached.

A.2 ANCHOR-CONDITIONED STATE DECODER

Given a molecular state z, the decoder constructs the state-dependent parameters in Sec. 3.2 by deforming the consensus-initialized Gaussian bank. To coordinate these changes spatially, a latent network produces tokens $u_m(z) \in \mathbb{R}^K$ on learnable anchors $\{c_m\}_{m=1}^M$ in the canonical coordinate system, after which each Gaussian interpolates the tokens of its k nearest anchors $\mathcal{N}_k(i)$:

$$w_{im} = \frac{\mathrm{e}^{-\|x_i^{\mathrm{base}} - c_m\|^2/\tau}}{\sum_{j \in \mathcal{N}_k(i)} \mathrm{e}^{-\|x_i^{\mathrm{base}} - c_j\|^2/\tau}}, \tag{12}$$

$$\bar{u}_i(z) = \sum_{m \in \mathcal{N}_k(i)} w_{im} u_m(z), \tag{13}$$

where τ is the mean squared distance over the selected Gaussian–anchor pairs, detached from the gradient computation and lower-bounded by 10^{-6}. Concatenating the interpolated token with a positional encoding γ gives the local feature

$$f_i(z) = \left[\gamma(x_i^{\mathrm{base}}), \bar{u}_i(z)\right], \tag{14}$$

which shared prediction heads use to produce Gaussian-specific updates to position, scale, orientation, and amplitude.

Algorithm 1 HeteroGauss training

Require: Dataset $\mathcal{D} = \{y_n, R_n, t_n, C_n\}$; stage boundaries E_c, E_a; epochs E
Ensure: Learned parameters (θ, ϕ)

$(\theta, \phi) \leftarrow \text{InitializeModel}()$; $\mathcal{S} \leftarrow \mathbf{0}$; $\mathcal{O} \leftarrow \text{InitAdamW}(\theta, \phi)$
for $e = 0, \dots, E-1$ **do**
  for $\{y_n, R_n, t_n, C_n\}_{n=1}^{B} \in \text{MiniBatches}(\mathcal{D})$ **do**
    $(t_-, t, B) \leftarrow \text{BatchProgress}()$; $(\beta, w, \ell_t) \leftarrow \text{Schedule}(t)$
    $y_n^{\mathrm{c}} \leftarrow \text{CenterImage}(y_n, t_n)$; $\mathcal{K} \leftarrow 0$
    if $t < E_c$ **then**
      $Z \leftarrow \mathbf{0}_{B \times L}$; $\Theta \leftarrow \text{Broadcast}(\text{CanonicalBank}(\theta), B)$
    else
      $(\mu_n, \log \sigma_n^2) \leftarrow \text{Encoder}_\phi(\text{PhaseFlip}(y_n^{\mathrm{c}}, C_n))$
      $z_n \leftarrow \mu_n + \exp(\frac{1}{2} \log \sigma_n^2) \odot \varepsilon_n, \quad \varepsilon_n \sim \mathcal{N}(0, I)$
      $Z \leftarrow [z_1, \dots, z_B]^\top$; $U \leftarrow \text{AnchorTokens}_\theta(Z)$; $F \leftarrow \text{Interpolate}_\theta(U)$
      $\Theta \leftarrow \text{DecodeGaussians}_\theta(F, Z)$
      $\mathcal{K} \leftarrow \frac{1}{B} \sum_n D_{\mathrm{KL}}(q_\phi(z_n \mid y_n^{\mathrm{c}}) \| \mathcal{N}(0, I))$
    end if
    $(\{p_{\theta,n}\}_{n=1}^{B}, \mathcal{C}) \leftarrow \text{BatchedProject}(\Theta, \{R_n\}_{n=1}^{B})$ ▷ Algorithm 2
    $r_n \leftarrow \mathcal{H}[y_n^{\mathrm{c}}] - C_n \odot \mathcal{H}[p_{\theta,n}]$
    $\mathcal{L}_{\mathrm{rec}} \leftarrow \frac{1}{B|\Omega|} \sum_{n=1}^{B} \sum_{k \in \Omega} w_k |r_n(k)|^2$; $\mathcal{L}_{\mathrm{KL}} \leftarrow \frac{\mathcal{K}}{|\Omega|}$
    $\mathcal{L}_{\mathrm{total}} \leftarrow \mathcal{L}_{\mathrm{rec}} + \beta \mathcal{L}_{\mathrm{KL}}$
    $\text{ZeroGrad}(\mathcal{O})$; $\nabla \mathcal{L}_{\mathrm{total}} \leftarrow \text{Backward}(\mathcal{L}_{\mathrm{total}})$
    if not $\text{AtCapacity}(\theta)$ **then**
      $\mathcal{S} \leftarrow \text{UpdateStats}(\mathcal{S}, \Theta, \nabla_x \mathcal{L}_{\mathrm{total}})$ ▷ Eq. (20)
    end if
    $(\theta, \phi, \mathcal{O}) \leftarrow \text{AdamWStep}(\theta, \phi, \mathcal{O}, \nabla \mathcal{L}_{\mathrm{total}}; \ell_t)$
    if $E_c \le t < E_a$ **and** $\text{TopologyDue}(t_-, t)$ **and not** $\text{AtCapacity}(\theta)$ **then**
      $(\theta, \mathcal{S}, h) \leftarrow \text{AdaptBank}(\theta, \mathcal{S})$ ▷ Eqs. (11), (21)
      if h **then**
        $\mathcal{O} \leftarrow \text{Rebind}(\theta, \phi, \mathcal{O})$
      end if
    end if
  end for
end for
return (θ, ϕ)

The anchor features also coordinate regional motion through a translation $v_m(z)$ and an axis–angle rotation $\omega_m(z)$ for each anchor. Blending these local rigid transformations gives a common motion field, to which a Gaussian-specific residual adds finer displacement:

$$T_m(x_i^{\text{base}}; z) = c_m + \mathcal{R}(\omega_m(z))(x_i^{\text{base}} - c_m) + v_m(z), \tag{15}$$

$$x_i(z) = \sum_{m \in \mathcal{N}_k(i)} w_{im} T_m(x_i^{\text{base}}; z) + \lambda_x \delta_i^x(z), \tag{16}$$

where $\delta_i^x(z)$ is predicted from $f_i(z)$ and λ_x controls its contribution. To distinguish the physical base geometry in the main text from its stored parameters, let $\widetilde{s}_i^{\text{base}}$ and $\widetilde{q}_i^{\text{base}}$ denote the scale and quaternion parameters before activation and normalization. The remaining state-dependent updates are then

$$s_i(z) = \text{clip}_{[10^{-3},1]}\big(\text{softplus}(\widetilde{s}_i^{\text{base}} + \delta_i^s(z))\big), \tag{17}$$

$$q_i(z) = \text{normalize}(\widetilde{q}_i^{\text{base}} + \delta_i^q(z)), \tag{18}$$

$$a_i(z) = a_i^{\text{base}} \exp(\Delta_i^{\text{amp}}(z)) \, \text{sigmoid}\big(o + \Delta_i^{\text{gate}}(z)\big). \tag{19}$$

Both amplitude heads use $f_i(z)$, with $\Delta_i^{\text{amp}}(z) = \lambda_a \tanh(h_{\text{amp}}(f_i(z)))$ bounding the exponential multiplier within $[\exp(-\lambda_a), \exp(\lambda_a)]$ and $\Delta_i^{\text{gate}}(z) = h_{\text{gate}}(f_i(z))$ controlling a gate in $(0, 1)$.

Setting the residuals to zero yields the activated base scale $s_i^{\text{base}} = \text{clip}_{[10^{-3},1]}(\text{softplus}(\widetilde{s}_i^{\text{base}}))$ and unit base quaternion $q_i^{\text{base}} = \text{normalize}(\widetilde{q}_i^{\text{base}})$ used in the main text. Scale and quaternion residuals are bounded by scaled tanh activations, while base amplitudes are softplus-activated and clipped to $[10^{-8}, 10^3]$ for numerical stability. Because the final layers of the residual and anchor-motion

heads are zero-initialized, heterogeneous reconstruction begins with the consensus geometry and an amplitude multiplier of $\mathrm{sigmoid}(o)$, which is close to one. During consensus reconstruction itself, the decoder bypasses both state-dependent amplitude factors so that $a_i = a_i^{\mathrm{base}}$, consistently with the initialization described in Sec. 3.2.

A.3 Adaptive Gaussian Topology

Whereas the decoder changes the parameters of existing primitives, topology adaptation changes the capacity of the shared bank using reconstruction and state-dependent statistics. For Gaussian i, we maintain exponential moving averages g_i, S_i, A_i, and M_i of its center-gradient magnitude, maximum scale, amplitude, and positional variation, respectively. Given B sampled latent states and the batch loss $\mathcal{L}_{\mathrm{rec}}^{\mathrm{batch}} = B^{-1}\sum_{n=1}^{B}\mathcal{L}_{\mathrm{rec},n}$, these statistics are updated as follows:

$$\begin{aligned}
g_i &\leftarrow \eta g_i + (1-\eta)\frac{1}{B}\sum_{n=1}^{B}\left\|\frac{\partial \mathcal{L}_{\mathrm{rec}}^{\mathrm{batch}}}{\partial x_i(z_n)}\right\|_2, \\
S_i &\leftarrow \eta S_i + (1-\eta)\frac{1}{B}\sum_{n=1}^{B}\|s_i(z_n)\|_\infty, \\
A_i &\leftarrow \eta A_i + (1-\eta)\max_n a_i(z_n), \\
M_i &\leftarrow \eta M_i + (1-\eta)\left[\frac{1}{B}\sum_{n=1}^{B}\|x_i(z_n)-\bar{x}_i\|_2^2\right]^{1/2},
\end{aligned} \tag{20}$$

where η is the averaging momentum and $\bar{x}_i$ is the batch-mean center. Using the batch maximum in A_i helps retain a Gaussian that contributes strongly to only a subset of the sampled states, although this protection depends on those states being represented in recent batches.

The gradient statistic g_i triggers densification, while Eq. (11) ranks densification candidates when their number exceeds the update budget. The scale statistic then distinguishes cloning from splitting, and jointly weak amplitude, gradient, and motion trigger pruning:

$$\mathrm{op}(i) = \begin{cases} \text{clone}, & g_i > \tau_g,\ S_i \le \tau_c, \\ \text{split}, & g_i > \tau_g,\ S_i > \tau_s, \\ \text{prune}, & A_i < \tau_a,\ g_i < \tau_p,\ M_i < \tau_m, \\ \text{keep}, & \text{otherwise}. \end{cases} \tag{21}$$

Cloning and splitting divide the parent base amplitude among the resulting primitives, preserving its sum before numerical clipping. Because the new centers and scales also alter local anchor features, however, conservation of base amplitude does not imply identical state-dependent amplitudes or projections immediately after an update. The operations nevertheless act on a single canonical bank, so all molecular states continue to share the resulting primitives and their identities.

A.4 From Radiance Splatting to Cryo-EM Projection

The projection rules must match the measurement even when the underlying Gaussian representation is shared across applications. Whereas conventional 3D Gaussian Splatting (3DGS) models photographic appearance using perspective projection and opacity compositing Kerbl et al. (2023), cryo-EM records contrast from electrons transmitted through the specimen. We therefore use orthographic Gaussian projection, as in homogeneous CryoSplat reconstruction Chen & Ling (2026), while retaining the state-dependent geometry and amplitudes defined in the main text.

In the linear cryo-EM image-formation model, the weak-phase and projection assumptions relate image contrast to a line integral of electrostatic potential under approximately parallel illumination Sigworth (2015). With microscope magnification absorbed into the spatial calibration, define

$$\rho_n^{\mathrm{view}}(r; z_n) = \rho_\theta\big(R_n^\top(r - J^\top t_n); z_n\big), \qquad p_{\theta,n}(u) = \int_{\mathbb{R}} \rho_n^{\mathrm{view}}((u^\top, \ell)^\top; z_n)\, d\ell. \tag{22}$$

Here ℓ is distance along the beam and $J = [I_2\ 0]$. Consistent with Eq. (9), the corresponding noise-free image is

$$y_{\theta,n} = h_n * p_{\theta,n}, \qquad \mathcal{H}[h_n] = C_n, \tag{23}$$

where $\mathcal{H}$ is the two-dimensional Hartley transform used in the reconstruction loss and fixed physical scale factors are absorbed into the density convention. Because the real, even CTF has a real, even point-spread function under the assumed imaging model, its Hartley and Fourier transfer functions coincide; C_n is the frequency-domain transfer function, not an image to be transformed again. Orthographic projection expresses parallel integration through the specimen, while multiplication by C_n in Hartley space describes the subsequent microscope contrast transfer.

The rigid transformation from molecular to viewing coordinates remains affine in both settings, but the subsequent camera projection differs: 3DGS uses a perspective map whose effect on each Gaussian is approximated locally. For viewing-frame coordinates $r = (X, Y, Z)^\top$, a pinhole camera gives

$$\pi(r) = \begin{bmatrix} f_x X/Z \\ f_y Y/Z \end{bmatrix}, \qquad J_\pi(r) = \begin{bmatrix} f_x/Z & 0 & -f_x X/Z^2 \\ 0 & f_y/Z & -f_y Y/Z^2 \end{bmatrix}. \tag{24}$$

Writing $\Sigma_i^{\mathrm{view}} = R\Sigma_i R^\top$ and denoting the viewing-frame Gaussian center by $\bar{r}_i$, its projected covariance is locally approximated as $J_\pi(\bar{r}_i)\Sigma_i^{\mathrm{view}} J_\pi(\bar{r}_i)^\top$. In our orthographic model, the projection map is simply $\pi_{\mathrm{ortho}}(r) = Jr$, with constant Jacobian J. Consequently,

$$\mu_{ni} = JR_n x_i(z_n) + t_n, \qquad \Lambda_{ni} = JR_n \Sigma_i(z_n) R_n^\top J^\top \tag{25}$$

are exact for Gaussian marginalization under the assumed projection model. Thus, although particle orientation changes the projected covariance of an anisotropic primitive, marginalization requires neither depth-dependent division by Z nor local perspective linearization.

Because integration preserves the total contribution of a normalized Gaussian, marginalizing the three-dimensional kernel yields the factor $(2\pi|\Lambda_{ni}|^{1/2})^{-1}$ in Eq. (8), ensuring that

$$\int_{\mathbb{R}^2} a_i(z_n)\widetilde{G}_{ni}(u; z_n)\, du = a_i(z_n). \tag{26}$$

Without this factor, integrating an exponential footprint would instead give $2\pi a_i(z_n)|\Lambda_{ni}|^{1/2}$, entangling its total contribution with projected size and orientation. Retaining normalization preserves the interpretation of $a_i(z_n)$ as integrated density.

Original 3DGS models view-dependent RGB color with spherical harmonics, $c_i(d) = \sum_{l,m} b_{i,lm} Y_{lm}(d)$, and composites depth-ordered splats as

$$I_{\mathrm{3DGS}}(u, d) = \sum_i c_i(d)\alpha_i(u) \prod_{j<i}(1 - \alpha_j(u)), \tag{27}$$

where d is the viewing direction and $\alpha_i(u)$ combines learned opacity with a projected footprint Kerbl et al. (2023). Because our scalar density representation describes molecular structure rather than directional RGB appearance, it requires no spherical-harmonic color coefficients: for a fixed state, $a_i(z_n)$ is independent of the viewing direction, while the image changes through geometric projection and the particle CTF. Variation between molecular states is instead expressed by $a_i(z_n)$, $x_i(z_n)$, and $\Sigma_i(z_n)$.

Under the linear cryo-EM model, contributions along the beam add rather than occlude one another. We therefore replace color-opacity compositing with

$$p_{\theta,n}(u) = \sum_i a_i(z_n)\widetilde{G}_{ni}(u; z_n), \qquad y_{\theta,n} = h_n * p_{\theta,n}. \tag{28}$$

Consequently, accumulation requires neither a transmittance product nor depth ordering, and the sigmoid factor in Eq. (6) regulates molecular density rather than optical opacity. Tile binning and differentiable rasterization remain useful for evaluating these additive contributions efficiently, motivating the batched implementation below.

A.5 Batched CUDA Renderer

To evaluate the additive projection efficiently, a custom CUDA extension renders the decoded Gaussian bank for all particles in a mini-batch. Following the preprocessing in Algorithm 1, it uses $t_n = 0$ and compares each projection with the translation-corrected observation y_n^{c}. The renderer first accumulates $p_{\theta,n}$ in real space, after which the centered Hartley transform and real, even CTF give

$\mathcal{H}[y_{\theta,n}] = C_n \odot \mathcal{H}[p_{\theta,n}]$. Comparing this prediction with $\mathcal{H}[y_n^c]$ yields the masked, frequency-weighted loss in Eq. (10), using the real-valued frequency representation adopted in CryoDRGN Zhong et al. (2021).

The renderer receives particle-specific centers, scales, quaternions, and amplitudes with shapes $B \times P \times 3$, $B \times P \times 3$, $B \times P \times 4$, and $B \times P \times 1$, together with one view matrix per particle. By flattening the first two dimensions, the preprocessing kernel associates each index r with particle $\lfloor r/P \rfloor$, selects the corresponding view matrix, and computes the projected mean, 2D covariance, and normalized amplitude of that Gaussian.

To preserve the same particle association during tile binning, each Gaussian–tile instance receives a 64-bit key whose upper bits encode the combined particle–tile index and whose lower bits encode the Gaussian index within that particle. Sorting these keys produces a contiguous Gaussian range for every image tile in every batch element. The rasterization grid uses the particle index as its third dimension; each block reads only the range associated with its particle and tile, evaluates the normalized 2D Gaussian footprints, and adds them directly to the output pixel. No depth ordering or alpha transmittance is used.

Using the same particle–tile indexing, the backward pass first propagates pixel gradients to the projected means, conic parameters, and amplitudes, then through the projection and covariance computations to the 3D centers, scales, rotations, and amplitudes. The batched operator therefore exposes the same differentiable inputs as the per-particle implementation while avoiding a separate renderer call for each image. Algorithm 2 collects the forward and backward operations. Here T denotes the number of tiles per image, ℓ identifies a tile, and $\mathcal{I}_{n\ell}$ stores the flattened indices $r = nP + i$ of its Gaussian instances; the rasterizer recovers $i = r \bmod P$ within particle n, and all algorithm indices are zero-based. The backward routine uses $\bar{v} = \partial \mathcal{L}_{\text{total}} / \partial v$ to denote the gradient of any variable v.

Algorithm 2 Batched CUDA Gaussian projection

Require: $\Theta = \{x_{ni}, s_{ni}, q_{ni}, a_{ni}\}$; rotations R; image tiles $\{\mathcal{T}_\ell\}_{\ell=0}^{T-1}$
Ensure: Projection batch $p = \{p_{\theta,n}\}$; cached intermediates $\mathcal{C}$ for backward propagation

function BatchedProject(Θ, R)
for $r = 0, \ldots, BP-1$ **in parallel do**
  $(n, i) \leftarrow (\lfloor r/P \rfloor, r \bmod P)$
  $\Sigma_{ni} \leftarrow Q(q_{ni})\,\mathrm{diag}(s_{ni}^2) Q(q_{ni})^\top$
  $\mu_{ni} \leftarrow J R_n x_{ni}$; $\Lambda_{ni} \leftarrow J R_n \Sigma_{ni} R_n^\top J^\top$
  $K_{ni} \leftarrow \Lambda_{ni}^{-1}$; $b_{ni} \leftarrow a_{ni} / (2\pi \sqrt{|\Lambda_{ni}|})$
  $\mathcal{V}_r \leftarrow \mathrm{OverlappingTiles}(\mu_{ni}, \Lambda_{ni})$
end for
$o \leftarrow \mathrm{PrefixSum}(\{|\mathcal{V}_r|\}_r)$
for $(r, \ell) : \ell \in \mathcal{V}_r$ **in parallel do**
  $(n, i) \leftarrow (\lfloor r/P \rfloor, r \bmod P)$
  $\mathrm{Emit}(o, r, \ell;\ \mathrm{key} = ((nT + \ell) \ll 32) \,|\, i,\ \mathrm{value} = r)$
end for
$(\mathbf{k}, \mathbf{v}) \leftarrow \mathrm{RadixSort}(\mathrm{keys}, \mathrm{values})$
$\mathcal{I} \leftarrow \mathrm{TileRanges}(\mathbf{k}, \mathbf{v})$; $p \leftarrow \mathbf{0}$
for (n, ℓ) **in parallel blocks do**
  $(\mu, K, b)_{\mathcal{I}_{n\ell}} \leftarrow \mathrm{LoadShared}(\mathcal{I}_{n\ell})$
  for $u \in \mathcal{T}_\ell$ **in parallel threads do**
    $p_{\theta,n}(u) \leftarrow \sum_{r \in \mathcal{I}_{n\ell},\, i = r \bmod P} b_{ni} \exp[-\frac{1}{2}(u - \mu_{ni})^\top K_{ni} (u - \mu_{ni})]$
  end for
end for
$\mathcal{C} \leftarrow (\Theta, R, \mu, K, b, \mathcal{I})$
return $(p, \mathcal{C})$
end function
function BatchedProjectBackward($\bar{p}, \mathcal{C}$)
$(\bar{\mu}, \bar{K}, \bar{b}) \leftarrow \mathrm{TileSumBackward}(\bar{p}, \mathcal{C})$
$(\bar{\Lambda}, \bar{a}) \leftarrow \mathrm{ConicNormBackward}(\bar{K}, \bar{b}, \mathcal{C})$
$(\bar{x}, \bar{s}, \bar{q}) \leftarrow \mathrm{ProjectCovBackward}(\bar{\mu}, \bar{\Lambda}, \mathcal{C})$
return $\bar{\Theta} = (\bar{x}, \bar{s}, \bar{q}, \bar{a})$
end function

A.6 Training Protocol

All datasets follow the three-stage procedure in Algorithm 1, beginning with consensus reconstruction in which the encoder is frozen, state-dependent updates are bypassed, and the shared base parameters are fitted to the particle images. Latent inference and deformation are then enabled together with topology adaptation, allowing the model to learn molecular variation while reallocating spatial capacity. Once adaptation ends, joint optimization of the encoder, decoder, and base parameters continues with a fixed Gaussian bank. CryoBench datasets are trained for 100 epochs, whereas EMPIAR-10180 is trained for 50 epochs. For CryoBench, one consensus epoch and 25 topology-adaptation epochs give $E_c = 1$ and $E_a = 26$; for EMPIAR-10180, one consensus epoch and 20 topology-adaptation epochs give $E_c = 1$ and $E_a = 21$. In either setting, topology adaptation can stop earlier when the configured Gaussian limit is reached.

Frequency weighting is introduced through a linear interpolation between uniform weights and the target radial weights. Let r_k denote the distance of frequency bin k from the transform origin, normalized by the maximum radius of the transform grid. We define

$$\begin{aligned} \bar{w}_k &= \frac{\max(r_k, r_{\min})^\gamma}{\frac{1}{|\Omega|}\sum_{j\in\Omega} \max(r_j, r_{\min})^\gamma}, \\ w_k(t) &= [1 - \alpha(t)] + \alpha(t)\bar{w}_k, \end{aligned} \tag{29}$$

where

$$\alpha(t) = \min\left(1, \frac{t}{T_w}\right). \tag{30}$$

Here, t measures training progress in epochs from the start of consensus reconstruction, T_w is the frequency-weight warm-up duration, and $r_{\min}$ prevents the lowest frequencies from receiving zero weight. We use $\gamma = 0.5$ and $r_{\min} = 0.1$, with $T_w = 1$ epoch for CryoBench and $T_w = 5$ epochs for EMPIAR-10180. Both the initial and target weights have unit mean over Ω, so the interpolation preserves this normalization throughout training.

While consensus reconstruction disables the KL term, state-dependent learning enables it with coefficient

$$\beta(t) = \beta_{\max} \min\left(1, \frac{t}{T_\beta}\right), \tag{31}$$

using the same training-time origin as the frequency-weight schedule. We set $T_\beta = 5$ epochs, with $\beta_{\max} = 0.08$ for CryoBench and 0.32 for EMPIAR-10180. Thus, the warm-up schedules are measured from the start of training rather than restarted after consensus initialization. Appendix A.7 provides the architecture and remaining optimization settings.

A.7 Architecture and Optimization Settings

All datasets use the same state-conditioned model formulation, including the amplitude parameterization in Eq. (6), while capacity, architecture, and optimization settings are documented by dataset in Table 3. The stages follow Appendix A.6, which also specifies the dataset-specific frequency-weight and KL schedules.

Table 3: Architecture and optimization settings for CryoBench and EMPIAR-10180. CryoBench datasets are trained for 100 epochs, whereas EMPIAR-10180 is trained for 50 epochs.

Setting	IgG-1D / IgG-RL	Ribosembly	EMPIAR-10180
Latent dimension	8	8	8
Number of anchors	256	256	256
Anchor neighbors	4	4	4
Anchor-token dimension	64	64	64
Encoder	3 residual layers, width 1024		
Initial/max. Gaussians	8,000/16,000	16,384/32,768	10,000/20,000
Consensus training	1 epoch	1 epoch	1 epoch
Topology adaptation	25 epochs	25 epochs	20 epochs
Total training	100 epochs	100 epochs	50 epochs
Batch size	128	128	128
Optimizer	AdamW with cosine decay		

B Experimental Setup and Additional Experiments

This section first specifies the experimental setup underlying the results in the main text, linking the training and architecture details in Appendix A to the datasets and evaluation protocol. It then extends the main evaluation with analyses of representation capacity, computational efficiency, and compositional variation.

B.1 Experimental Setup

Datasets. CryoBench provides matched ground-truth structures and imaging parameters at SNR 0.01 Jeon et al. (2024), allowing reconstruction accuracy to be evaluated across distinct forms of heterogeneity. Specifically, subset IgG-1D contains 100,000 projections from 100 conformations along a prescribed Fab rotation, whereas subset IgG-RL retains the same numbers of particles and states but introduces flexible-linker motion that couples Fab position and orientation, placing greater demands on coherent geometric deformation. Subset Ribosembly instead comprises 16 assembly intermediates with a shared ribosomal core and progressively added proteins and RNA, requiring a reconstruction to accommodate components that appear or disappear across states while preserving the common structure.

To move beyond simulations, we use EMPIAR-10180, comprising 327,490 experimental particles of the yeast pre-catalytic spliceosome Plaschka et al. (2017). The complex combines a relatively stable core with flexible peripheral domains and coordinated large-scale motions Nakane et al. (2018), so reconstruction must preserve detail across regions with different degrees of mobility. Unlike the prescribed structural ensembles in CryoBench, the distribution and full extent of this experimental variability are not known, and matched reference structures are unavailable for evaluating the recovered states. These conditions make reconstruction and validation more challenging, while providing a direct assessment of the method's practical value on experimental data.

Baselines and metrics. We compare HeteroGauss with 3D Classification, CryoDRGN, 3DFlex, 3DVA, RECOVAR, and CryoHype, using fixed poses for all methods. On CryoBench, we compute per-conformation AU-FSC against matched ground-truth volumes using the official masking and alignment protocol, and report the mean, standard deviation, and median. Because experimental particles lack matched target volumes, we evaluate EMPIAR-10180 through equally filtered map comparisons, masked gold-standard FSC, and local resolution from independent half-map reconstructions.

Implementation. We train HeteroGauss through three stages of consensus reconstruction, topology adaptation, and fixed-topology refinement. All datasets use the same encoder and decoder architecture, with limited differences in capacity and optimization settings. Appendix A.6 describes the training protocol, while Appendix A.7 details the architecture and parameter settings.

B.2 GAUSSIAN-COUNT ABLATION

To determine the default Gaussian budget for each dataset, we vary the initial and maximum bank sizes together and compare reconstruction accuracy with mean training time per epoch in Table 4. All configurations are trained for 100 epochs with the remaining architecture and optimizer settings held fixed, allowing us to assess whether additional capacity improves reconstruction and thereby justify the budgets used in the main experiments.

Table 4: Gaussian-budget ablation on CryoBench. All configurations use 100 training epochs. AU-FSC is summarized across conformations as mean (standard deviation), and time denotes mean wall-clock duration per epoch.

	IgG-1D		IgG-RL		Ribosembly	
Initial / max. Gaussians	Mean (std)	Time/epoch (min)	Mean (std)	Time/epoch (min)	Mean (std)	Time/epoch (min)
2,048 / 4,096	0.391 (0.003)	1.52	0.362 (0.009)	1.38	0.376 (0.011)	5.39
4,096 / 8,192	0.397 (0.002)	2.62	0.366 (0.009)	2.40	0.403 (0.013)	9.47
8,000 / 16,000	**0.398 (0.001)**	4.46	**0.380 (0.010)**	4.30	0.423 (0.014)	18.77
16,384 / 32,768	0.396 (0.003)	6.63	0.370 (0.009)	6.41	**0.430 (0.014)**	36.98

For IgG-1D and IgG-RL, an initial bank of 8,000 Gaussians with a maximum of 16,000 achieves the highest mean AU-FSC among the tested configurations, at 0.398 and 0.380, respectively. Increasing the budget to 16,384/32,768 reduces these values to 0.396 and 0.370 while raising the corresponding training times from 4.46 and 4.30 to 6.63 and 6.41 minutes per epoch. Thus, under the evaluated training protocol, further expansion increases computational cost without improving reconstruction accuracy, supporting our choice of 8,000/16,000 for both IgG datasets. Although IgG-1D already approaches this accuracy at 4,096/8,192, making the smaller budget an efficient alternative, the selected default uses the same budget for both IgG datasets and attains the highest observed mean accuracy on each.

In contrast, mean AU-FSC on Ribosembly increases consistently with the Gaussian budget, rising from 0.376 at the smallest configuration to 0.430 at the largest. The final expansion from 8,000/16,000 to 16,384/32,768 improves mean AU-FSC from 0.423 to 0.430, although training time nearly doubles from 18.77 to 36.98 minutes per epoch. We therefore prioritize reconstruction accuracy for Ribosembly and adopt an initial bank of 16,384 Gaussians with a maximum of 32,768, accepting the higher computational cost associated with this improvement.

These results motivate dataset-specific defaults based on reconstruction accuracy across the tested budgets: 8,000/16,000 for the IgG datasets, where further expansion adds cost without improving accuracy, and 16,384/32,768 for Ribosembly, where the larger bank yields more accurate reconstructions.

B.3 RENDERING EFFICIENCY

To isolate the computational benefit of batching, we compare the renderer in Appendix A.5 with a per-particle loop on one NVIDIA RTX A6000, using identical Gaussian parameters and poses. Each batch contains 128 images of size 128×128, with 8,000 Gaussians per image in FP32, and timings include synchronized forward and backward passes. After three warm-up iterations, measurement stops once the coefficient of variation remains below 2% for three consecutive five-sample windows, which occurs after ten measured batches for both modes. Before timing, we verify agreement between outputs and input gradients; during the benchmark, we measure peak process-level GPU memory with `nvidia-smi` on an otherwise idle device.

As shown in Table 5, batching increases throughput from 570 to 1,698 images/s ($2.98\times$) while adding 0.05 GiB of peak process memory. This gain preserves numerical agreement: forward outputs are identical in FP32, and the maximum absolute input-gradient discrepancy is 8.94×10^{-8}. Because the benchmark isolates projection and its backward pass, the measured speedup describes renderer throughput rather than end-to-end training speed.

Table 5: Renderer efficiency at batch size 128 with 8,000 Gaussians per 128×128 image. Throughput measures synchronized forward and backward passes; peak memory is process-level GPU usage. Numerical errors compare batched and per-particle outputs and input gradients.

Renderer	Images/s ↑	Memory (GiB) ↓	Output error ↓	Gradient error ↓
Per-particle loop	570.0	**0.62**	–	–
Batched CUDA	**1698.0**	0.67	0.00e+00	8.94e-08

B.4 Recovery of Compositional Heterogeneity

Beyond geometric deformation caused by molecular motion, heterogeneous reconstruction must also represent components that appear or disappear across states. The Ribosembly experiments in the main text evaluate reconstruction accuracy across assembly intermediates; here, we further examine whether Gaussian amplitude changes express these compositional differences and correspond spatially to ground-truth changes. Figure 6 develops this analysis from amplitude variation across assembly states to its location within the molecule and its agreement with ground-truth density changes.

Figure 6A shows the individually normalized amplitudes of the 500 Gaussians with the largest amplitude ranges across the 16 assembly states. As assembly progresses, different groups increase their density contributions at different stages, revealing a pattern consistent with progressive assembly. Panel B then maps these Gaussians back to the molecular structure, coloring each by the state in which its amplitude is highest. Their concentration in peripheral assembly regions, with relatively few in the shared core, connects amplitude variation across states to the molecular locations where composition changes.

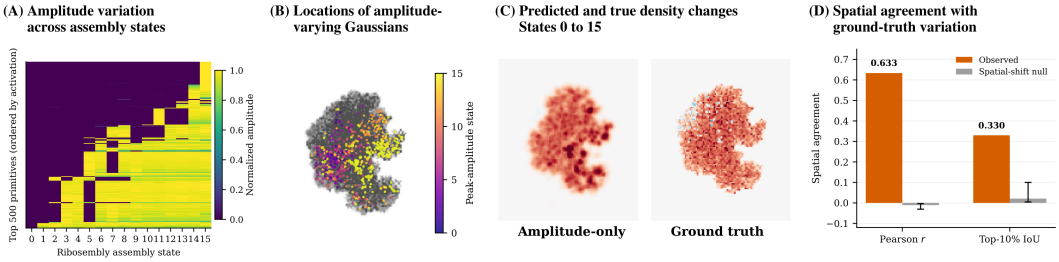

Figure 6: Recovery of Ribosembly compositional heterogeneity through Gaussian amplitude changes. (A) Amplitudes of the 500 most variable Gaussians across 16 assembly states, normalized separately for each Gaussian. (B) Their peak-amplitude positions, colored by the corresponding assembly state. (C) Projected density differences from state 0 to 15, comparing amplitude-only changes at fixed Gaussian centers with ground truth. (D) Quantitative agreement between amplitude ranges and ground-truth density ranges across all states, compared with spatially shifted controls. A–B show the selected 500 Gaussians; C–D use all 15,462 Gaussians.

To further examine this spatial correspondence, Fig. 6C compares density differences generated solely by amplitude changes between states 0 and 15 with the corresponding ground-truth differences. We fix the Gaussian centers and use only the amplitude differences between the two states to construct the predicted map, isolating the information expressed by amplitude modulation. Under this constraint, the prediction still exhibits major regions of density change corresponding to those in the ground-truth map, supporting the use of local amplitude modulation to represent compositional changes during assembly.

Building on this endpoint comparison, Fig. 6D extends the evaluation to all 16 assembly states by comparing the spatial distributions of amplitude ranges and ground-truth density ranges. Within the official Ribosembly mask, the maps achieve a Pearson correlation of 0.633 and an intersection over union of 0.330 for the top 10% of voxels ranked by variation, with the threshold determined separately for each map. As a control, we apply 100 large circular translations to the predicted map, preserving its value distribution and smoothness while disrupting alignment with the molecular structure; the corresponding mean correlation and IoU are -0.011 and 0.021. The higher agree-

ment under correct alignment indicates that amplitude variation localizes the regions of ground-truth compositional change.

Together with the reconstruction results in the main text, these analyses show that the recovery of compositional heterogeneity is reflected both in overall reconstruction accuracy and in the spatial correspondence between amplitude variation and ground-truth changes. The shared Gaussian representation can therefore capture both conformational and compositional heterogeneity through geometric deformation and state-dependent amplitude modulation, respectively.

C Failure Case Analysis

The following experiments examine how amplitude parameterization and reconstruction-loss design affect optimization, complementing the component ablations in Sec. 4.3. We first compare the measured training losses for positive and signed base amplitudes, then examine latent contraction in a real-space-loss trial, distinguishing the observed behavior from possible underlying mechanisms.

C.1 Optimization Instability with Signed Base Amplitudes

To examine the role of positive base amplitudes on IgG-1D, we replace the Softplus activation with Tanh while retaining the remaining training protocol:

$$a_i^{\text{base}} = \begin{cases} \text{clip}_{[10^{-8},10^{3}]}\big(\text{softplus}(\widetilde{a}_i^{\text{base}})\big), & \text{full model,} \\ \tanh(\widetilde{a}_i^{\text{base}}), & \text{signed-amplitude variant,} \end{cases} \tag{32}$$

where $\widetilde{a}_i^{\text{base}}$ is the unconstrained amplitude parameter. Replacing Softplus with Tanh allows positive and negative base amplitudes, unlike the additive-update ablation in Table 2, which modifies the state-dependent amplitude rule. Under the same training protocol as the full model, the Tanh variant repeatedly oscillated instead of converging toward a stable solution and achieved lower final AU-FSC on IgG-1D.

The experimental loss curves in Fig. 7 show this difference directly: after consensus training, the full-model loss continues to decrease, whereas the Tanh variant remains highly oscillatory and plateaus near 0.636. The persistent fluctuations are consistent with the lower reconstruction accuracy and indicate that allowing signed base amplitudes makes optimization less stable in this experiment.



Figure 7: IgG-1D training loss with Softplus versus Tanh base amplitudes, shown at ten samples per epoch without smoothing.

One possible explanation is cancellation between overlapping Gaussians with opposite signs: different positive–negative combinations can produce similar projected densities, leaving individual amplitudes weakly constrained and allowing successive updates to undo one another. Softplus excludes this cancellation mechanism by preserving nonnegative contributions, which is consistent with the more stable full-model loss and supports retaining positive base amplitudes.

C.2 Latent Contraction with a Real-Space Reconstruction Loss

In early experiments, we computed the reconstruction loss by directly comparing predicted projections with observed particle images in real space. During training, however, the inferred latent states became concentrated in a small region, while the decoded density deteriorated and eventually appeared nearly blank. The model consequently failed both to distinguish molecular states and to recover their corresponding structures, preventing effective heterogeneous reconstruction. Because the task requires structure and state to be learned jointly from noisy particle images, this failure prompted us to reconsider how the reconstruction error was evaluated.

To address this problem, we followed the design of CryoDRGN Zhong et al. (2021) and evaluated the reconstruction error in Hartley space. The model still generates a real-space projection from the Gaussian representation, then applies the Hartley transform and CTF to compare the prediction with the frequency-domain representation of the observed particle. This adjustment changes the evaluation of the reconstruction error while preserving the explicit real-space interpretation of the molecular structure and its projection. With this loss formulation, training recovered, restoring latent-state learning and structural reconstruction; we therefore retained the Hartley-space reconstruction objective in the final model.

Having established this reconstruction objective, we further considered the contributions of different spatial frequencies to structural recovery. Low-frequency information primarily describes the overall molecular shape, whereas recovering fine structural detail requires the model to make effective use of high-frequency information. We therefore introduced radial weights into the frequency-domain reconstruction error to increase the relative contribution of high-frequency errors and place greater emphasis on detailed reconstruction. Since the error is already evaluated in frequency space, its contributions can be weighted directly by spatial frequency, allowing this design to integrate naturally into the objective.