GenMatter: Perceiving Physical Objects with Generative Matter Models

Eric Li[1,*] Arijit Dasgupta[1] Yoni Friedman[1] Mathieu Huot[2]
Vikash Mansinghka[2] Thomas O'Connell[2] William T. Freeman[1] Joshua B. Tenenbaum[1,2]
[1]MIT CSAIL [2]MIT BCS
[*]Corresponding author: esli@mit.edu Project page: esli999.github.io/genmatter

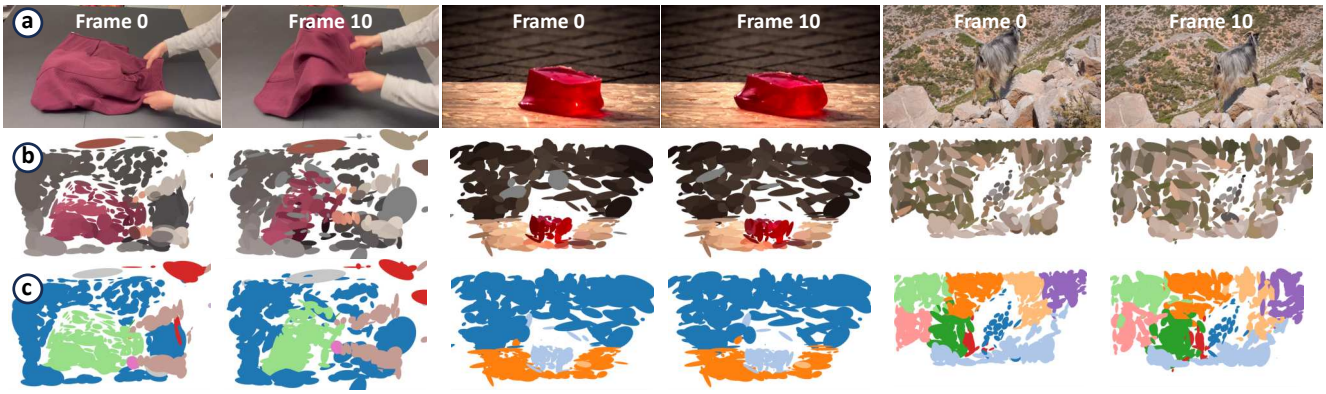


Figure 1. **GenMatter** is a generative model of moving matter. Conditioned on motion and appearance features extracted from RGB video, inference inverts this hierarchical generative model to group observations into *particles* (small Gaussians representing local regions of matter), themselves grouped into *clusters* (coherently and independently moveable physical entities). A hardware-accelerated inference algorithm based on parallelized block Gibbs sampling recovers stable particle motion and groupings. **(a)** RGB video input. **(b)** Inferred 3D matter particles shown as colored ellipses, each colored by the average color of its assigned data points. **(c)** The same particles colored by cluster assignment, revealing independently moving objects.

Abstract

Human visual perception offers valuable insights for understanding computational principles of motion-based scene interpretation. Humans robustly detect and segment moving entities that constitute independently moveable chunks of matter, whether observing sparse moving dots, textured surfaces, or naturalistic scenes. In contrast, existing computer vision systems lack a unified approach that works across these diverse settings. Inspired by principles of human perception, we propose a generative model that hierarchically groups low-level motion cues and high-level appearance features into particles (small Gaussians representing local matter), and groups particles into clusters capturing coherently and independently moveable physical entities. We develop a hardware-accelerated inference algorithm based on parallelized block Gibbs sampling to recover stable particle motion and groupings. Our model operates on different kinds of inputs (random dots, stylized textures, or naturalistic RGB video), enabling it to work across settings where biological vision succeeds but existing computer vision approaches do not. We validate this unified framework across three domains: on 2D random dot kinematograms, our approach captures human object perception including graded uncertainty across ambiguous conditions; on a Gestalt-inspired dataset of camouflaged rotating objects, our approach recovers correct 3D structure from motion and thereby accurate 2D object segmentation; and on naturalistic RGB videos, our model tracks the moving 3D matter that makes up deforming objects, enabling robust object-level scene understanding. This work thus establishes a general framework for motion-based perception grounded in principles of human vision.

1. Introduction

Human vision segments moving objects across diverse settings: random dot kinematograms with minimal form

cues [48], camouflaged textured objects perceivable only through motion [21], and naturalistic scenes [42]. No existing computer vision system has this generality. This gap raises a scientific question about the computational principles that enable such broad perceptual capabilities.

We introduce **GenMatter**, a generative model that segments moving matter across the diverse settings where biological vision succeeds. The model hierarchically groups low-level motion and appearance features into particles (local regions of matter represented as Gaussians), then groups particles into clusters representing coherently and independently moveable entities. The same Bayesian inference algorithm operates across diverse inputs by performing motion feature extraction, which is effective even in abstract textured scenes, and can be supplemented with shape and appearance features for naturalistic video. By jointly inferring particle trajectories and their organization into Spelke objects [51], discovering cluster assignments and cluster-level rigid transformations, our approach identifies which particles belong to which moving entities while inducing soft rigidity priors within each entity. Unlike prior methods that impose hand-crafted regularization constraints and avoid explicit object-level grouping, GenMatter captures entities undergoing significant deformation. Figure 2 illustrates the pipeline from RGB frames to inferred particles and clusters.

We evaluate GenMatter across three diverse settings where biological vision robustly detects and groups independently moving matter, yet no single computer vision system succeeds across all three:

- In random dot kinematogram (RDK) stimuli, where dots follow rigid motion or flicker randomly and correspondences are ambiguous, we test whether motion patterns alone suffice to infer object identity. GenMatter reproduces human perceptual groupings across all difficulty levels and captures graded uncertainty that aligns with intersubject variability.
- We evaluate GenMatter on a new dataset of rotating objects with camouflaged textures inspired by Gestalt grouping principles. Human observers readily perceive 3D structure from motion in these stimuli. GenMatter recovers correct 3D structure, with groupings of independently moving matter emerging from probabilistic inference.
- On naturalistic RGB videos, GenMatter maintains accurate tracking of the moving 3D matter that makes up deforming objects, matching the performance of supervised trackers without task-specific training.

Our evaluation strategy has two complementary objectives. First, we compare GenMatter against standard computer vision approaches across all three settings, demonstrating that both biological vision and GenMatter possess generality that existing computer vision systems lack. Second, within each setting, we provide baselines and ablations to demonstrate specific technical advantages of our structured probabilistic approach: capturing graded perceptual uncertainty, integrating noisy motion signals over time, and generalizing without task-specific training (where supervised baselines may match our performance but lack our cross-domain generality).

We emphasize that GenMatter provides a unifying framework that describes a class of models for motion-based perception. While the class of models defined by our approach requires pretrained feature extractors, the core inference engine requires no task or domain-specific training and fits within a few kilobytes of source code. By recasting perceptual grouping as online probabilistic inference, the structured prior defined by GenMatter's generative model achieves performance comparable to data-driven learning across all studied perception tasks, spanning abstract dot stimuli and naturalistic videos.

2. Related Work

Analysis-by-synthesis The analysis-by-synthesis approach posits perception as inference in a structured generative model. Purely model-based systems using probabilistic graphics programs have achieved success in CAPTCHA-breaking, pose estimation, and scene understanding [24, 25, 35, 40], while hybrid approaches combining learned components with structured priors have improved accuracy and efficiency [20, 47, 55, 66]. Programming systems supporting inference automation have enabled competitive performance for object detection and tracking [8, 36].

Generative models of human perception Perception as Bayesian inference in structured generative models captures key aspects of human visual processing, including motion perception, shape recognition, and facial identification [10, 19, 26, 64]. This framework extends to physical scene understanding, modeling how humans infer object properties and interactions [6, 50, 56, 59, 61, 65]. However, computational complexity forces such models to impose restrictive assumptions on texture and geometry. Recent work shows that biologically-inspired motion energy features improve motion segmentation robustness and enable zero-shot generalization to random dot stimuli [52, 53]. GenMatter provides a tractable probabilistic model achieving human-like robustness to noise and ambiguity while operating on naturalistic visual input.

Tracking any point The problem of tracking any point was introduced as an intermediate approach between sparse feature tracking and dense optical flow, aiming to capture both long-range and dense motion trajectories [49]. More recent systems have improved tracking performance, producing denser and more robust trajectories across time [17, 27, 31, 32, 60]. Extensions to 3D tracking have proceeded along two main directions: some approaches incorporate monocular depth estimation directly into the model [33], while others infer depth implicitly from learned features over point

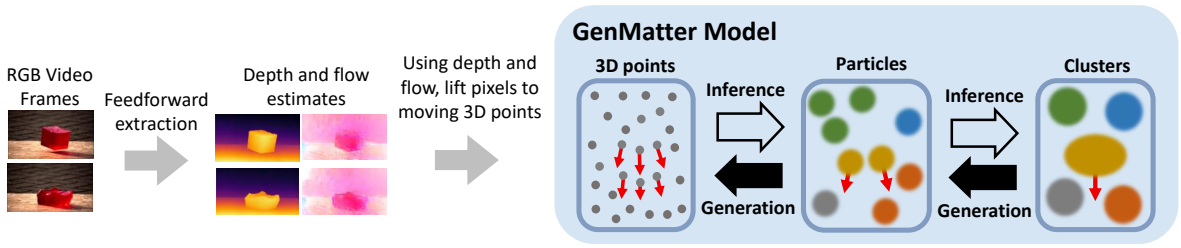


Figure 2. **GenMatter inference pipeline.** RGB video is preprocessed (gray arrows) to extract dense depth and optical flow, lifting each pixel to a 3D point tagged with its velocity. The blue box depicts the GenMatter generative model (Sec. 3), which represents a scene as a hierarchy of clusters and particles that emit moving 3D points. Black arrows indicate the generative direction (clusters generate particles, which generate 3D points). Hollow arrows indicate the inference algorithm (Sec. 4), which conditions on observed 3D points to infer particle and cluster parameters. Red arrows depict motion at the point, particle, and cluster layers.

trajectories [15]. Further work has used as-rigid-as-possible regularization to improve the quality of point tracking [62]. We note that while systems that track any point let users track pre-specified points, they do not offer a way to infer a good point representation. These systems do not pick particle representations adaptively from the scene, nor leverage semantic information for tracking and grouping.

Motion-based grouping and segmentation Contemporary approaches to discovering object structure from motion rely predominantly on learned representations. Motion segmentation methods combine optical flow with appearance priors through diverse training schemes [16, 57, 63], while foundation models like SAM2 [46] achieve promptable segmentation via large-scale supervision, capturing both whole objects and parts without semantic constraints (the stuff-versus-things distinction [1]). Unsupervised methods that learn articulated 3D structure from video [4, 5, 43, 58] typically require pretraining, multi-view input, or focus on rigid categories. Methods using internet-scale priors [37, 38] infer static 3D structure but do not aggregate information through multiple frames of a scene. In contrast, GenMatter infers dynamic 3D matter representations of deformable objects via a structured probabilistic prior, providing richer scene understanding than 2D segmentation.

3. Generative Matter Model

We introduce the Generative Matter Model (**GenMatter**), a two-level hierarchical generative model for structured motion of deformable matter, defined procedurally in Algorithm 1. *Clusters* represent coherent groups, each parameterized by a Gaussian over space and a rigid-body transformation. *Particles* are local Gaussians drawn from clusters that encode spatially localized data points. Data points, each a position-velocity observation constructed from depth and optical flow (Figure 2), are sampled from a mixture over particles. While cluster transformations encode rigid motion, the particle velocity covariance $\mathbf{\Sigma}_\ell^{\mathcal{V}}$ gives slack to model intra-cluster motion, enabling the model to naturally accommodate both rigid and deformable objects.

Hierarchical structure The algorithm begins by sampling mixture weights $\boldsymbol{\pi}^{\mathcal{H}} \sim \mathrm{Dir}(\boldsymbol{\alpha})$ and $\boldsymbol{\pi}^{\mathcal{B}} \sim \mathrm{Dir}(\boldsymbol{\beta})$ for clusters and particles. Each cluster k is parameterized by a spatial distribution $(\boldsymbol{\mu}_k^{\mathcal{H}}, \mathbf{\Sigma}_k^{\mathcal{H}})$ and rigid transformation $(\mathbf{R}_k, \mathbf{t}_k)$ drawn from discretized priors suited to small frame-to-frame motions. Each particle ℓ is assigned to cluster $k = z_\ell^{\mathcal{H}}$ and samples its spatial mean from $\mathcal{N}(\boldsymbol{\mu}_k^{\mathcal{H}}, \mathbf{\Sigma}_k^{\mathcal{H}})$, with covariance $\mathbf{\Sigma}_\ell^{\mathcal{B}}$ to explain local matter. Observed points $\mathbf{x}_n$, each a position-velocity pair derived from depth and optical flow as illustrated in Figure 2, are drawn from particle $\ell = z_n^{\mathcal{B}}$ via $\mathcal{N}(\boldsymbol{\mu}_\ell^{\mathcal{B}}, \mathbf{\Sigma}_\ell^{\mathcal{B}})$. Optionally, to incorporate image features $\mathbf{f}_n$, we define augmented data points $\tilde{\mathbf{x}}_n = [\mathbf{x}_n; \mathbf{f}_n]$ with $\tilde{\mathbf{x}}_n \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_\ell, \tilde{\mathbf{\Sigma}}_\ell)$. $\mathbf{f}_n$ is assumed to be zero-mean, and $\tilde{\mathbf{\Sigma}}_\ell$ is block-diagonal, so the spatial and feature dimensions are assumed to be independent.

Velocity model The per-particle cluster-induced velocity $\bar{\mathbf{v}}_\ell = \mathbf{t}_k + (\mathbf{R}_k - \mathbf{I})(\boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}})$ captures expected motion from the parent cluster's rigid transformation, where $(\mathbf{R}_k - \mathbf{I})$ provides a first-order approximation of rotation about $\boldsymbol{\mu}_k^{\mathcal{H}}$. The particle velocity mean $\mathbf{v}_\ell \sim \mathcal{N}(\bar{\mathbf{v}}_\ell, \sigma_V^2 \mathbf{I})$ introduces isotropic noise to allow particles to deviate from rigidity, while the covariance $\mathbf{\Sigma}_\ell^{\mathcal{V}} \sim \mathcal{W}^{-1}(\mathbf{\Psi}^{\mathcal{V}}, \nu^{\mathcal{V}})$ allows data point velocities $\mathbf{v}_n \sim \mathcal{N}(\mathbf{v}_\ell, \mathbf{\Sigma}_\ell^{\mathcal{V}})$ to deviate from particle motion.

Comparison to ARAP regularization To connect our probabilistic approach to optimization-based techniques more common in the literature, we note that a common way to regularize tracking models is using an *as-rigid-as-possible* (ARAP) assumption [39, 62]. At frame t, an ARAP regularizer promotes locally rigid motion by penalizing changes in pairwise distances between each point $\mathbf{x}_n^t$ and its neighbors within a fixed radius r. In its typical formulation as a loss function, the ARAP is written as:

$$\mathcal{L}_{\text{ARAP}} = \sum_{d(\mathbf{x}_m^t, \mathbf{x}_n^t) < r} w_{m,n} \left\| d(\mathbf{x}_m^{t+1}, \mathbf{x}_n^{t+1}) - d(\mathbf{x}_m^t, \mathbf{x}_n^t) \right\|_1 ,$$

where $d(\cdot, \cdot)$ is a distance metric. This loss encourages locally-rigid motion within all spheres of radius r. $w_{m,n}$ is

Algorithm 1 Generative Particle Model

Input: K, L, N (num. clusters, particles, data points)
Priors: $\boldsymbol{\alpha}, \boldsymbol{\beta}$ (mixture); $\boldsymbol{\mu}^{\mathcal{H}}, \sigma^2_{\mu^{\mathcal{H}}}, \boldsymbol{\Psi}^{\mathcal{H}}, \nu^{\mathcal{H}}$ (cluster);
$\boldsymbol{\Psi}^{\mathcal{B}}, \nu^{\mathcal{B}}$ (particle matter); $\sigma^2_V, \boldsymbol{\Psi}^{\mathcal{V}}, \nu^{\mathcal{V}}$ (velocity)
Sample cluster weights: $\boldsymbol{\pi}^{\mathcal{H}} \sim \text{Dir}(\boldsymbol{\alpha})$
Sample particle weights: $\boldsymbol{\pi}^{\mathcal{B}} \sim \text{Dir}(\boldsymbol{\beta})$
for $k = 1$ to K **do**
  Sample cluster covariance: $\boldsymbol{\Sigma}^{\mathcal{H}}_k \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}^{\mathcal{H}}, \nu^{\mathcal{H}})$
  Sample cluster mean: $\boldsymbol{\mu}^{\mathcal{H}}_k \sim \mathcal{N}(\boldsymbol{\mu}^{\mathcal{H}}, \sigma^2_{\mu^{\mathcal{H}}}\mathbf{I})$
  Sample cluster translation: $\mathbf{t}_k \sim \text{DiscreteNormal}(\mathbf{0}, s^2\mathbf{I})$
  Sample cluster rotation: $\mathbf{R}_k \sim \text{DiscreteVMF}(\kappa^{\text{vmf}}, \theta_{\text{max}})$
end for
for $\ell = 1$ to L **do**
  Sample cluster assignment: $z^{\mathcal{H}}_\ell \sim \text{Cat}(\boldsymbol{\pi}^{\mathcal{H}})$
  Let $k = z^{\mathcal{H}}_\ell$
  Sample particle covariance: $\boldsymbol{\Sigma}^{\mathcal{B}}_\ell \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}^{\mathcal{B}}, \nu^{\mathcal{B}})$
  Sample particle mean: $\boldsymbol{\mu}^{\mathcal{B}}_\ell \sim \mathcal{N}(\boldsymbol{\mu}^{\mathcal{H}}_k, \boldsymbol{\Sigma}^{\mathcal{H}}_k)$
  Compute cluster-induced velocity:
    $\bar{\mathbf{v}}_\ell = \mathbf{t}_k + (\mathbf{R}_k - \mathbf{I})(\boldsymbol{\mu}^{\mathcal{B}}_\ell - \boldsymbol{\mu}^{\mathcal{H}}_k)$
  Sample particle velocity mean: $\mathbf{v}_\ell \sim \mathcal{N}(\bar{\mathbf{v}}_\ell, \sigma^2_V\mathbf{I})$
  Sample particle velocity covariance: $\boldsymbol{\Sigma}^{\mathcal{V}}_\ell \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}^{\mathcal{V}}, \nu^{\mathcal{V}})$
end for
for $n = 1$ to N **do**
  Sample particle assignment: $z^{\mathcal{B}}_n \sim \text{Cat}(\boldsymbol{\pi}^{\mathcal{B}})$
  Let $\ell = z^{\mathcal{B}}_n$
  Sample data point position: $\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\mu}^{\mathcal{B}}_\ell, \boldsymbol{\Sigma}^{\mathcal{B}}_\ell)$
  Sample data point velocity: $\mathbf{v}_n \sim \mathcal{N}(\mathbf{v}_\ell, \boldsymbol{\Sigma}^{\mathcal{V}}_\ell)$
end for

a weighting given by either a predefined pointwise kernel function $k(\mathbf{x}^0_m, \mathbf{x}^0_n)$ or a learned similarity metric $s(m, n)$ defined on pairs of particle indices.

Connection to ARAP via small-variance asymptotics We can relate certain aspects of GenMatter's particle motion model to ARAP by deriving its small-variance asymptotic limit [13, 34]. Taking $\epsilon/\eta \to 0$ (where ϵ and η are datapoint-to-particle and particle-to-cluster noise scales) yields a K-means-like objective that alternates between computing centroids and solving for optimal rigid transforms via Procrustes alignment. Letting $\mathbf{x}'_n = \mathbf{R}_k(\mathbf{x}_n - \boldsymbol{\mu}^{\mathcal{H}}_k) + \boldsymbol{\mu}^{\mathcal{H}}_k + \mathbf{t}_k$ denote the predicted position after rigid transformation, the resulting objective is to minimize:

$$\mathcal{L}(z^{\mathcal{B}}_n, z^{\mathcal{H}}_\ell) = \sum_n \|\mathbf{x}_n + \mathbf{v}_n - \mathbf{x}'_n\|^2_2 ,$$

where $\ell = z^{\mathcal{B}}_n$. A key distinction is that while ARAP applies rigidity penalties based on fixed distance cutoffs r, our approach jointly infers object-centric groupings and rigid motion through hierarchical clustering. By coupling rigidity with probabilistic inference over cluster assignments, the posterior reveals which particles belong to which independently moving entities. However, discrete optimization over assignments is intractable, motivating a hierarchical Bayesian model and probabilistic inference.

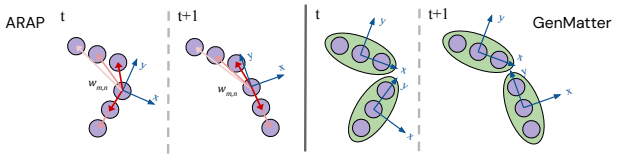


Figure 3. ARAP vs. GenMatter on a two-object scene. ARAP incurs penalties from fixed distance cutoff r, while GenMatter infers cluster assignments (green), modeling discontinuous motion.

4. Inference

We perform inference via blocked Gibbs sampling [29] that exploits the hierarchical conditional independence structure: variables at each level (data points, particles, clusters) can be updated in parallel given other levels. We leverage conjugate updates (Normal-Inverse-Wishart, Normal-Normal) where possible [22, 41]. For rigid transforms, we discretize the SE(3) space and enumerate candidates via parallel likelihood evaluation. Our vectorized implementation in the GenJAX probabilistic programming framework [7, 9, 12] runs efficiently on a single NVIDIA L4 GPU (24GB). We maintain a single-sample posterior approximation, sufficient for high-quality inference in practice. Full derivations are in Appendix C.

Assignment updates Datapoints are assigned to particles via a categorical Gibbs conditional combining spatial and velocity likelihoods:

$$p(z^{\mathcal{B}}_n = \ell \mid \cdot) \propto \pi^{\mathcal{B}}_\ell \cdot \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}^{\mathcal{B}}_\ell, \boldsymbol{\Sigma}^{\mathcal{B}}_\ell) \cdot \mathcal{N}(\mathbf{v}_n \mid \mathbf{v}_\ell, \boldsymbol{\Sigma}^{\mathcal{V}}_\ell).$$

Particles are assigned to clusters based on rigid motion fit:

$$p(z^{\mathcal{H}}_\ell = k \mid \cdot) \propto \pi^{\mathcal{H}}_k \cdot \mathcal{N}(\boldsymbol{\mu}^{\mathcal{B}}_\ell \mid \boldsymbol{\mu}^{\mathcal{H}}_k, \boldsymbol{\Sigma}^{\mathcal{H}}_k) \cdot \mathcal{N}(\mathbf{v}_\ell \mid \bar{\mathbf{v}}_{\ell,k}, \sigma^2_V\mathbf{I}),$$

where $\bar{\mathbf{v}}_{\ell,k} = \mathbf{t}_k + (\mathbf{R}_k - \mathbf{I})(\boldsymbol{\mu}^{\mathcal{B}}_\ell - \boldsymbol{\mu}^{\mathcal{H}}_k)$ is the velocity derived from cluster k's rigid transformation. Mixture weights are updated via Dirichlet-Categorical conjugacy.

Parameter updates Covariances are updated via Normal-Inverse-Wishart conjugacy using scatter matrices of assigned points. Velocity means $\mathbf{v}_\ell$ are sampled from Gaussian Gibbs conditionals that integrate a rigid motion prior from the parent cluster with observed velocities from assigned datapoints. Particle spatial means $\boldsymbol{\mu}^{\mathcal{B}}_\ell$ are sampled from a Gaussian Gibbs conditional factorized into three terms: (1) a spatial likelihood from the assigned cluster, (2) position likelihoods from assigned datapoints, and (3) a velocity likelihood from rigid motion.

This velocity likelihood arises because the predicted velocity $\bar{\mathbf{v}}_{\ell,k} = \mathbf{t}_k + (\mathbf{R}_k - \mathbf{I})(\boldsymbol{\mu}^{\mathcal{B}}_\ell - \boldsymbol{\mu}^{\mathcal{H}}_k)$ depends linearly on the particle position $\boldsymbol{\mu}^{\mathcal{B}}_\ell$. Since both the prior on $\boldsymbol{\mu}^{\mathcal{B}}_\ell$ and the velocity likelihood $\mathcal{N}(\mathbf{v}_\ell \mid \bar{\mathbf{v}}_{\ell,k}, \sigma^2_V\mathbf{I})$ are Gaussian with linear dependencies, the Gibbs conditional over $\boldsymbol{\mu}^{\mathcal{B}}_\ell$ remains Gaussian by conjugacy [23]. Cluster means similarly incorporate priors, particle positions, and velocities. Rigid

transforms $(\mathbf{R}_k, \mathbf{t}_k)$ are sampled from discretized SE(3), yielding a conjugate Dirichlet-Categorical update.

Initialization We initialize the MCMC chain at frame 0 using hierarchical K-Means clustering, which provides an efficient approximation to burn-in for the Gibbs sampler. We use K-Means++ clustering [3] to initialize particle positions, then run a second K-Means pass to initialize cluster centers. Mixture weights, velocity means, and covariances are set from empirical frequencies and sample statistics. Rigid transforms are initialized via Kabsch alignment [30]. Spatial hyperparameters (Inverse-Wishart scale matrices, Gaussian mean priors) encode interpretable priors on cluster and particle size, set from image resolution (see Appendix C.2.1). Motion hyperparameters are set once per dataset from optical flow statistics. All other hyperparameters remain constant across all videos and datasets.

Multi-frame tracking The two-frame generative model extends to video tracking via sequential MCMC. At frame t, particle means are propagated using inferred velocities: $\tilde{\boldsymbol{\mu}}_\ell^{\mathcal{B},t} = \boldsymbol{\mu}_\ell^{\mathcal{B},t-1} + \mathbf{v}_\ell^{t-1}$. Since we observe only unordered point clouds $\{\mathbf{x}_n^t\}$ without tracked correspondences, we first assign data points to particles via spatial proximity since $p(z_n^{\mathcal{B},t} = \ell \mid \mathbf{x}_n^t) \propto \pi_\ell^{\mathcal{B},t} \cdot \mathcal{N}(\mathbf{x}_n^t \mid \tilde{\boldsymbol{\mu}}_\ell^{\mathcal{B},t}, \boldsymbol{\Sigma}_\ell^{\mathcal{B},t})$. After updating particle means from these spatial assignments, subsequent Gibbs sweeps update variables in bottom-up order (data points $\rightarrow$ particles $\rightarrow$ clusters), with assignment distributions now incorporating both position and velocity observations. This ensures cluster inference remains grounded in current observations. Critically, we re-infer cluster assignments and transformations at each frame rather than propagating them, as propagated clusters may incorrectly span articulated parts. This design, inspired by filtering in particle MCMC [2, 18], maintains a tractable posterior while enabling stable tracking.

5. Experiments

We evaluate GenMatter across three settings: 2D random dot kinematograms (RDKs), camouflaged 3D Gestalt stimuli, and naturalistic RGB videos. Figure 2 gives an overview of the complete inference system. We compare against standard computer vision methods and provide within-setting ablations to assess the generality of our probabilistic approach. All CIs are 95% bootstrap intervals (50,000 samples).

5.1. Human Object Judgments from Motion

Stimuli and task We created 9 unique rigid-body physics scenes using PyMunk [11] and generated 3 RDKs per scene using [48], varying probe dot locations and timings to yield 27 total stimuli. Each stimulus contains object-bound and background dots that either follow rigid motion or flicker randomly, and plays forward then backward to highlight apparent motion. Human participants ($n = 150$, 50 per condition) viewed 11 videos each (2 familiarization, 9 experimental) and made binary same-object judgments for red and green probe dots. Participants were recruited from Prolific [45] (median duration: 4 minutes; mean age: 37.2; 85 female, 65 male), compensated at local minimum wage, and screened for English fluency, normal color vision, and normal/corrected visual acuity. The study was IRB-approved with fully anonymized data.

Inference GenMatter employs a 2D inference pipeline with motion vectors estimated via RANSAC affine transform fitting, ensuring the model operates on image-computable features as humans do when viewing the stimuli. A k-nearest neighbor decision policy applied to posterior samples produces binary judgments. GenMatter is run on 50 random seeds per stimulus to match human sample size.

Results GenMatter achieves high correlation with human judgments ($r^2 = 0.86$, $t(25) = 12.4$, $p < 0.001$). Figure 4a shows variation in human responses across stimuli. Figure 4b visualizes GenMatter's internal representation for a single stimulus. The true scene is overlaid with random dots from the kinematogram and their estimated motion vectors. Dots are colored by inferred cluster assignment, with black indicating outliers. These results support latent particle and cluster inference as a computational account of human object perception under motion uncertainty. Figure 4c illustrates how some stimuli yield near-unanimous correct responses while others exhibit lower accuracy. GenMatter closely reproduces this graded perceptual uncertainty. The model's ability to reproduce human uncertainty in ambiguous scenes and achieve high accuracy when the scene affords clear grouping establishes it as a valid computational model of human perception across diverse viewing conditions.

Ablations and baseline To assess the necessity of hierarchical structure, we evaluate two ablated variants that remove cluster-level variables from the generative model. The fixed particles variant initializes $K = 5$ particles with fixed covariance. The adaptive particles variant initializes $K = 500$ particles with covariances updated online. Both ablations eliminate the cluster layer, performing inference solely at the particle level. Both variants exhibit substantially degraded correlation with human judgments ($r^2 = 0.35$ and 0.41, respectively) compared to the full hierarchical model ($r^2 = 0.86$). This demonstrates that the cluster level provides essential structure for capturing human percepts of moving objects. FlowSAM [63], which also relies on motion cues, achieves near-zero correlation with human judgments ($r^2 = 0.04$). This is because random dot kinematograms are a challenging setting where sparse motion extraction is critical, highlighting a fundamental generality gap between human perception and current computer vision systems.

5.2. Structure from Motion in Gestalt Stimuli

Stimuli and task We evaluate GenMatter on a challenging 3D structure-from-motion task where texture provides

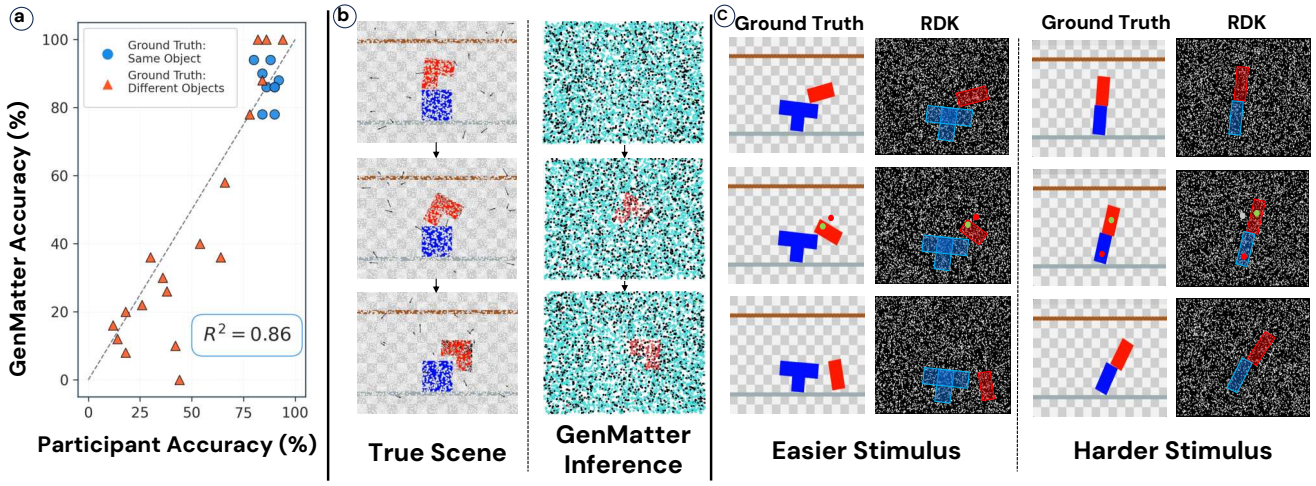


Figure 4. **GenMatter closely tracks human perceptual judgments on random dot kinematograms. (a)** GenMatter accuracy (%) vs. participant accuracy (%) across 27 stimuli ($r^2 = 0.86$). Each blue circle represents a same-object stimulus and each orange triangle a different-object stimulus. **(b)** An internal view of the inferred posterior: red points belong to the moving object, blue points to the background, and black points flickered out of the scene. **(c)** Fast rotation makes object-background separation easy: both GenMatter (86%) and humans (90%) correctly judge the probes as on different objects (left). Slow sliding motion is very challenging: both GenMatter (88%) and humans (84%) incorrectly judge the probes as on the same object (right).

Table 1. **Summary statistics across 140 Gestalt videos.** GenMatter scores higher on mean per-pixel accuracy and Jaccard index. GenMatter is also more consistent across stimuli. Values reported as mean [95% bootstrap CI].

Method	Accuracy	Jaccard
SegAnyMo	0.33 [0.28, 0.37]	0.26 [0.22, 0.31]
FlowSAM	0.87 [0.85, 0.88]	0.67 [0.63, 0.70]
GenMatter	**0.94 [0.93, 0.94]**	**0.72 [0.70, 0.74]**

little information about scene geometry. We created 140 short videos of 3D objects rotating against backgrounds with matched textures. The dataset comprises 20 distinct object geometries, each rendered with 7 different texture patterns matched to their backgrounds. Each 6-frame video sequence is accompanied by ground-truth binary segmentation masks for evaluation. In these camouflaged stimuli, static frames provide minimal segmentation information, yet human observers readily perceive 3D structure from motion [21]. This design tests whether models rely on per-frame summary statistics or track spatially-localized features across frames.

Baselines We compare GenMatter against two video segmentation methods. SegAnyMo [28] uses point tracking and monocular depth through a learned motion encoder, producing dynamic object masks by grouping tracked points with SAM2. FlowSAM [63] finetunes SAM for optical flow-based video segmentation. GenMatter operates on optical flow and depth. SegAnyMo leverages point tracking (a richer motion representation) and depth, while FlowSAM uses only optical flow. Despite having comparable inputs, GenMatter relies on probabilistic inference rather than learned feed-forward circuits.

Inference Our inputs to GenMatter come from optical flow and monocular depth, via RAFT and VideoDepthAnything [14, 54]. All methods are evaluated at 96×96 resolution. Each video provides 5 flow frames. GenMatter obtains 500 posterior samples per frame via Gibbs sampling to approximate the posterior distribution, from which we extract the maximum a posteriori (MAP) estimate for evaluation.

Evaluation protocol Following the probe-point methodology established in the RDK experiments, we evaluate segmentation by sampling query points rather than densely evaluating all pixels. Dense evaluation on all ground-truth pixels requires establishing correspondence between predicted and ground-truth segments, which can implicitly assume a fixed number of objects. Since motion-based discovery aims to identify moving objects without prior knowledge of object count or identity, we adopt a probe-point sampling approach, visualized in Figure 5. For each frame, we sample 100 probe points uniformly from the ground-truth object region $\mathcal{O}$. For each probe location $\mathbf{p}_i$, we identify the predicted segment $S(\mathbf{p}_i)$ containing it and compute pixel-wise accuracy of $S(\mathbf{p}_i)$ against ground truth. Aggregating over probes yields a Monte Carlo estimate of the probabilistic segmentation, given by $p_{\text{pred}}(\text{object} \mid \mathbf{x}) \approx \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[\mathbf{x} \in S(\mathbf{p}_i)]$ where $\mathbf{p}_i \sim \text{Uniform}(\mathcal{O})$. For each probe point $\mathbf{p}_i$, we compute the Jaccard index $J(S(\mathbf{p}_i), \mathcal{O})$ between its predicted segment $S(\mathbf{p}_i)$ and the ground truth object region $\mathcal{O}$ using the stan-

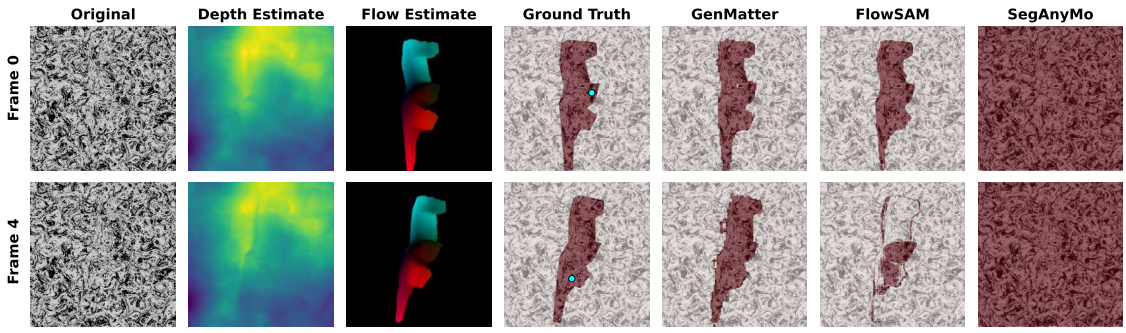


Figure 5. **Qualitative comparison on camouflaged stimuli.** Probe point segmentation on `scene_16`, `texture_01`. The depth estimate is uninformative, and the flow estimate shows that on-axis rotation causes opposing motion at top vs. bottom (blue vs. red). GenMatter correctly segments the moving object, while FlowSAM segments the initial frame correctly but degrades over time. SegAnyMo fails to detect any object in the scene.

dard IoU formulation. Frame-level accuracy is computed as $\mathbb{E}_{\mathbf{p}\sim\text{Uniform}(\mathcal{O})}[J(S(\mathbf{p}), \mathcal{O})]$, approximated via Monte Carlo averaging over probe samples. Per-video accuracy is then obtained by averaging across all frames in each sequence.

Results GenMatter outperforms both supervised baselines, detailed in Table 1, achieving higher Jaccard (J) on 111/140 videos against FlowSAM and 133/140 against SegAnyMo ($p < 1 \times 10^{-6}$, paired t-test). GenMatter's advantage is consistent across all texture patterns. Figure 5 visualizes these performance differences through segmentation overlays, showing that GenMatter assigns concentrated probability mass to contiguous matter regions while FlowSAM produces predictions that degrade over time. On the example shown, SegAnyMo is unable to detect the object ($J = 0.15$), while FlowSAM achieves moderate performance ($J = 0.70$), both falling substantially below GenMatter ($J = 0.92$). Thus, GenMatter's projected 3D representation can outperform learned 2D segmentation on motion-dominated grouping. Monocular depth estimates are sometimes unreliable on camouflaged textures. Ablating depth as a model input entirely still yields accuracy $= 0.89$, slightly above FlowSAM (0.87), confirming that optical flow drives performance on this benchmark.

5.3. 3D Particle Representations from RGB Video

Task and motivation We evaluate whether probabilistic inference in the GenMatter model enables robust particle-based matter representations on videos in TAP-Vid-DAVIS [17]. While GenMatter infers full 3D particle representations, we project to 2D tracks for comparison against CoTracker3 [31], a supervised baseline with the same input-output specification: RGB video $\rightarrow$ particle tracks.

Model In this setting, we condition GenMatter on monocular depth [14], optical flow [54], and DINO [44] features. Each particle maintains a learned appearance vector in DINO space, evolving through Gibbs updates conditioned on cluster assignments. Shared cluster assignments couple appearance to spatial structure, while cluster-level rigid transformations constrain particle motions. Particles and clusters are initialized at frame 0 using SAM2 [46] masks to provide initial proposals for spatial grouping, which are then propagated forward via approximate Bayesian filtering. We use GenMatter with 500 particles with 9 clusters.

Baselines and metrics We compare against CoTracker3 [31], a transformer-based point tracker supervised with simulation data. For consistency across methods, we also initialize CoTracker3 with 500 points. We note that CoTracker3 tracks individual pixel locations, while GenMatter's particles represent Gaussian regions of matter with spatial extent, and observed data is probabilistically attributed to these particles. We evaluate our model on a projection to 2D tracking rather than 3D particle ground truth due to the scarcity of deformable naturalistic datasets with 3D ground truth annotations. TAP-Vid-DAVIS provides high-fidelity 2D segmentation masks, enabling comparison through projection of GenMatter's 3D particles to 2D. To enable comparison between these fundamentally different representations, we adopt matter-weighted Jaccard $J_m = \sum_i w_i f_i / (\sum_i w_i f_i + \sum_j w_j (1 - f_j))$, where $w_i = n_i \pi_i$ weights particle i by spatial extent (n_i pixels) and mixture probability (π_i), and f_i is fractional overlap with ground truth. This metric accounts for graded uncertainty in GenMatter's probabilistic matter representation. It reduces to standard Jaccard when particles have uniform weights, which we assume for CoTracker3.

Results GenMatter achieves matter-weighted Jaccard of 0.79, matching CoTracker3 (0.78) without task-specific pretraining, shown in Table 2. Unlike learned trackers, GenMatter's hierarchical structure enables explicit integration of SAM2 segmentation proposals into spatial clustering. However, ground-truth initialization degrades GenMatter performance to 0.77. This degradation occurs because ground-truth initialization at frame 0 imposes hard geometric constraints that do not always align with noisy flow and depth estimates.

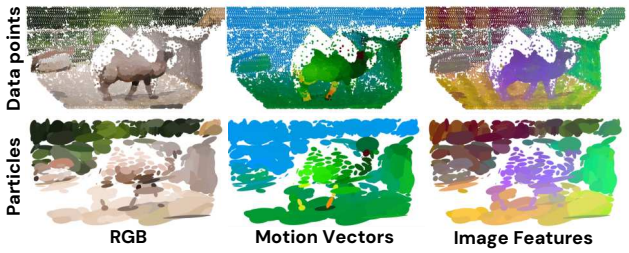


Figure 6. **Per-point particle assignment visualization.** Each data point is colored by its assigned particle's RGB color (left), motion direction (middle), and appearance features (right). Gaussian particles are shown in the second row. Distinct patterns across motion and appearance demonstrate that GenMatter integrates complementary information sources for faithful matter representation.

Table 2. **Tracking performance on TAP-Vid DAVIS.** GenMatter matches CoTracker3 without task-specific pre-training. Using GT segmentation mask instead of SAM for initialization decreases GenMatter performance but does not affect CoTracker3. Values reported as mean [95% bootstrap CI].

Metric	CoTracker3	GenMatter	GenMatter (abl.)
J_m (SAM)	0.78 [0.69, 0.87]	**0.79 [0.73, 0.84]**	0.69 [0.61, 0.77]
J_m (GT)	**0.78 [0.69, 0.87]**	0.77 [0.73, 0.84]	0.68 [0.58, 0.73]

Additionally, ground-truth foreground masks provide only a single binary segmentation without spatial decomposition of background regions, limiting initial clustering of non-object areas. In contrast, SAM2's multi-segment decomposition helps GenMatter model both object and non-object regions effectively. Ablating cluster-level variables degrades performance substantially (J_m from 0.79 to 0.69), confirming that hierarchical structure is essential. Ablating depth as a model input similarly degrades performance ($J_m = 0.69$ [0.61, 0.77] with SAM init, 0.66 [0.58, 0.73] with GT init), confirming that 3D structure contributes meaningfully beyond motion alone. Figure 6 visualizes the particle assignments in an example video. Each pixel's assigned particle is shown by RGB color (left), velocity direction (middle, normalized and color-coded), and DINO features (right, first 3 PCA dimensions). This demonstrates how GenMatter's hierarchical inference integrates position, motion, and appearance into a unified particle representation, enabling structured scene decomposition without task-specific training.

Compute-accuracy tradeoffs GenMatter's hierarchy enables runtime computational control through data point subsampling at each frame. Since data points occupy the lowest hierarchical level, subsampling them directly reduces the number of latent variables without changing model architecture, in contrast to learned approaches with fixed computation circuits. We evaluate subsampling rates from 1/8 to 1/512 of available data points. Across all rates from 1/8 to 1/128, GenMatter incurs no statistically significant performance loss ($J_m = 0.76$–0.79) relative to full resolution ($J_m = 0.76$ [0.71, 0.82]), while running up to $12\times$ faster (9.8 FPS at 1/128 vs. 0.80 FPS). Accuracy saturates beyond moderate subsampling because upsampled features provide no additional information once sampling density exceeds the resolution of the lowest-resolution feature (DINO). Performance degrades substantially only at extreme subsampling ($J_m = 0.56$ [0.46, 0.65] at 1/512), where too few observations remain to support reliable inference. This range of operating speeds and accuracies demonstrates flexibility unavailable to end-to-end learned architectures.

6. Discussion

Limitations Our model currently represents physical matter without explicit dynamics. A natural extension would be incorporating physics-based dynamics, enabling a joint matter and dynamics model akin to a game engine. Such a model would provide forward prediction of object positions and motion, which would be particularly beneficial for tracking through complete occlusion where GenMatter's current performance degrades due to limited mechanisms for reinitializing or reidentifying completely hidden matter. Our implementation uses a fixed particle count L, which limits adaptation to scale changes and objects entering or exiting the scene. This is addressable through dynamic particle allocation strategies from Bayesian nonparametrics.

We provide a general set of evaluation methods across diverse visual perception tasks, though more thorough benchmarking would better assess the quality of both our model and the baselines. The RDK and 3D Gestalt experiments use binary response formats that match typical human experimental protocols, but these collapse posterior distributions into discrete judgments. Extending evaluation to capture per-participant posteriors would enable richer characterization of scene belief states. For particle tracking, we use 2D masks as a proxy for 3D tracking given the current lack of large-scale datasets with ground-truth 3D annotations for deformable objects in naturalistic settings. Developing such datasets through real-world annotation and synthetic rendering would enable more rigorous benchmarks for evaluating 3D matter representations.

Broader implications GenMatter demonstrates that hierarchical probabilistic inference of particle-based representations provides a unified computational account of motion-based segmentation across minimal dot stimuli, camouflaged objects, and naturalistic scenes. This generality arises from structured priors encoding rigid motion and spatial grouping, rather than hand-crafted regularizers, suggesting that structured probabilistic inference provides essential inductive biases for compositional scene understanding. In doing so, GenMatter bridges scientific models of human visual perception and engineered computer vision systems.

Acknowledgements

This work was supported in part by the Department of the Air Force Artificial Intelligence Accelerator (Cooperative Agreement FA8750-19-2-1000), NSF Award 2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions), Navy-ONR MURI N00002610, Navy-ONR MURI N00014-22-1-2740, CoCoSys from the Georgia Institute of Technology (Award 2023-JU-3131), the MIT Siegel Family Quest for Intelligence, and the Probabilistic Computing Foundation.

References

[1] Edward H Adelson. On seeing stuff: the perception of materials by humans and machines. In *Human vision and electronic imaging VI*, pages 1–12. spie, 2001. 3

[2] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010. 5

[3] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006. 5

[4] Junyeob Baek, Yi-Fu Wu, Gautam Singh, and Sungjin Ahn. Dreamweaver: Learning compositional world models from pixels. In *The Thirteenth International Conference on Learning Representations*. 3

[5] Zhipeng Bao, Pavel Tokmakov, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Object discovery from motion-guided tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22972–22981, 2023. 3

[6] Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110 (45):18327–18332, 2013. 2

[7] McCoy Becker, Mathieu Huot, Sam Ritchie, and Colin Smith. GenJAX: Probabilistic Programming with Gen, built on top of JAX. 4

[8] McCoy R Becker, Alexander K Lew, Xiaoyan Wang, Matin Ghavami, Mathieu Huot, Martin C Rinard, and Vikash K Mansinghka. Probabilistic programming with programmable variational inference. *Proceedings of the ACM on Programming Languages*, 8(PLDI):2123–2147, 2024. 2

[9] McCoy R Becker, Alexander K Lew, Xiaoyan Wang, Matin Ghavami, Mathieu Huot, Martin C Rinard, and Vikash K Mansinghka. Probabilistic programming with programmable variational inference. *Proceedings of the ACM on Programming Languages*, 8(PLDI):2123–2147, 2024. 4

[10] Johannes Bill, Hrag Pailian, Samuel J Gershman, and Jan Drugowitsch. Hierarchical structure is employed by humans during visual motion perception. *Proceedings of the National Academy of Sciences*, 117(39):24581–24589, 2020. 2

[11] Victor Blomqvist. Pymunk, 2024. An easy-to-use pythonic rigid body 2D physics library. 5

[12] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. 4

[13] Tamara Broderick, Brian Kulis, and Michael Jordan. Mad-bayes: Map-based asymptotic derivations from bayes. In *International Conference on Machine Learning*, pages 226–234. PMLR, 2013. 4

[14] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. *arXiv preprint arXiv:2501.12375*, 2025. 6, 7

[15] Seokju Cho, Jiahui Huang, Seungryong Kim, and Joon-Young Lee. Seurat: From moving points to depth. *arXiv preprint arXiv:2504.14687*, 2025. 3

[16] Achal Dave, Pavel Tokmakov, and Deva Ramanan. Towards segmenting anything that moves. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[17] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023. 2, 7

[18] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10:197–208, 2000. 5

[19] Goker Erdogan and Robert A Jacobs. Visual shape perception as bayesian inference of 3d object-centered shape representations. *Psychological review*, 124(6):740, 2017. 2

[20] SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in neural information processing systems*, 29, 2016. 2

[21] Yoni Friedman, Thomas O'Connell, Daniel Bear, Jiajun Wu, Judy Fan, Josh Tenenbaum, and Dan Yamins. Benchmarking human mid-level scene understanding. *Journal of Vision*, 23 (9):5798–5798, 2023. 2, 6

[22] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995. 4

[23] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC Press, 2013. 4

[24] Nishad Gothoskar, Marco Cusumano-Towner, Ben Zinberg, Matin Ghavamizadeh, Falk Pollok, Austin Garrett, Josh Tenenbaum, Dan Gutfreund, and Vikash Mansinghka. 3dp3: 3d scene perception via probabilistic programming. *Advances in Neural Information Processing Systems*, 34:9600–9612, 2021. 2

[25] Nishad Gothoskar, Matin Ghavami, Eric Li, Aidan Curtis, Michael Noseworthy, Karen Chung, Brian Patton, William T Freeman, Joshua B Tenenbaum, Mirko Klukas, and Vikash K Mansinghka. Bayes3d: fast learning and inference in struc-

tured generative models of 3d objects and scenes. *arXiv preprint arXiv:2312.08715*, 2023. 2

[26] Mark Hamilton, Simon Stent, Vasha DuTell, Anne Harrington, Jennifer Corbett, Ruth Rosenholtz, and William T Freeman. Seeing faces in things: A model and dataset for pareidolia. In *European Conference on Computer Vision*, pages 377–395. Springer, 2024. 2

[27] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pages 59–75. Springer, 2022. 2

[28] Nan Huang, Wenzhao Zheng, Chenfeng Xu, Kurt Keutzer, Shanghang Zhang, Angjoo Kanazawa, and Qianqian Wang. Segment any motion in videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3406–3416, 2025. 6

[29] Sonia Jain and Radford M Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of computational and Graphical Statistics*, 13(1):158–182, 2004. 4

[30] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5): 922–923, 1976. 5

[31] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 2, 7

[32] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pages 18–35. Springer, 2024. 2

[33] Skanda Koppula, Ignacio Rocco, Yi Yang, Joe Heyward, João Carreira, Andrew Zisserman, Gabriel Brostow, and Carl Doersch. Tapvid-3d: A benchmark for tracking any point in 3d. *arXiv preprint arXiv:2407.05921*, 2024. 2

[34] Brian Kulis and Michael I. Jordan. Revisiting k-means: new algorithms via bayesian nonparametrics. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, page 1131–1138, Madison, WI, USA, 2012. Omnipress. 4

[35] Tejas D Kulkarni, Pushmeet Kohli, Joshua B Tenenbaum, and Vikash Mansinghka. Picture: A probabilistic programming language for scene perception. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4390–4399, 2015. 2

[36] Alexander K Lew, George Matheos, Tan Zhi-Xuan, Matin Ghavamizadeh, Nishad Gothoskar, Stuart Russell, and Vikash K Mansinghka. Smcp3: Sequential monte carlo with probabilistic program proposals. In *International conference on artificial intelligence and statistics*, pages 7061–7088. PMLR, 2023. 2

[37] Ci Li, Yi Yang, Zehang Weng, Elin Hernlund, Silvia Zuffi, and Hedvig Kjellström. Dessie: Disentanglement for articulated 3d horse shape and pose estimation from images. In *Proceedings of the Asian Conference on Computer Vision*, pages 764–783, 2024. 3

[38] Zizhang Li, Dor Litvak, Ruining Li, Yunzhi Zhang, Tomas Jakab, Christian Rupprecht, Shangzhe Wu, Andrea Vedaldi, and Jiajun Wu. Learning the 3d fauna of the web. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9752–9762, 2024. 3

[39] 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)*, pages 800–809. IEEE, 2024. 3

[40] Vikash K Mansinghka, Tejas D Kulkarni, Yura N Perov, and Josh Tenenbaum. Approximate bayesian image interpretation using generative probabilistic graphics programs. *Advances in neural information processing systems*, 26, 2013. 2

[41] Kevin P Murphy. Conjugate bayesian analysis of the gaussian distribution. *def*, 1(2σ2):16, 2007. 4

[42] Shin'ya Nishida, Takahiro Kawabe, Masataka Sawayama, and Taiki Fukiage. Motion perception: From detection to interpretation. *Annual review of vision science*, 4(1):501–523, 2018. 2

[43] Atsuhiro Noguchi, Umar Iqbal, Jonathan Tremblay, Tatsuya Harada, and Orazio Gallo. Watch it move: Unsupervised discovery of 3d joints for re-posing of articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3677–3687, 2022. 3

[44] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 7

[45] Prolific. Prolific participant recruitment platform. https://www.prolific.com, 2024. Version used: May 2024. London, UK. 5

[46] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 7

[47] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11488–11499, 2021. 2

[48] Sophia Robert, Leslie G Ungerleider, and Maryam Vaziri-Pashkam. Disentangling object category representations driven by dynamic and static visual input. *Journal of Neuroscience*, 43(4):621–634, 2023. 2, 5

[49] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *International journal of computer vision*, 80:72–91, 2008. 2

[50] Kevin Smith, Lingjie Mei, Shunyu Yao, Jiajun Wu, Elizabeth Spelke, Josh Tenenbaum, and Tomer Ullman. Modeling expectation violation in intuitive physics with coarse probabilistic object representations. *Advances in neural information processing systems*, 32, 2019. 2

[51] Elizabeth S Spelke. Principles of object perception. *Cognitive science*, 14(1):29–56, 1990. 2

[52] Zitang Sun, Yen-Ju Chen, Yung-Hao Yang, Yuan Li, and Shin'ya Nishida. Machine learning modelling for multi-order

human visual motion processing. *Nature Machine Intelligence*, 7(7):1037–1052, 2025. 2

[53] Matthias Tangemann, Matthias Kümmerer, and Matthias Bethge. Object segmentation from common fate: Motion energy processing enables human-like zero-shot generalization to random dot stimuli. *Advances in Neural Information Processing Systems*, 37:137135–137160, 2024. 2

[54] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 6, 7

[55] Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezchikov, Josh Tenenbaum, Frédo Durand, Bill Freeman, and Vincent Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *Advances in Neural Information Processing Systems*, 36: 12349–12362, 2023. 2

[56] Tomer D Ullman, Elizabeth Spelke, Peter Battaglia, and Joshua B Tenenbaum. Mind games: Game engines as an architecture for intuitive physics. *Trends in cognitive sciences*, 21(9):649–665, 2017. 2

[57] Rahul Venkatesh, Klemen Kotar, Lilian Naing Chen, Seungwoo Kim, Luca Thomas Wheeler, Jared Watrous, Ashley Xu, Gia Ancone, Wanhee Lee, Honglin Chen, et al. Discovering and using spelke segments. *arXiv preprint arXiv:2507.16038*, 2025. 3

[58] Diwen Wan, Yuxiang Wang, Ruijie Lu, and Gang Zeng. Template-free articulated gaussian splatting for real-time reposable dynamic view synthesis. *arXiv preprint arXiv:2412.05570*, 2024. 3

[59] Haoliang Wang, Khaled Jedoui, Rahul Venkatesh, Felix Jedidja Binder, Josh Tenenbaum, Judith E Fan, Daniel Yamins, and Kevin A Smith. Probabilistic simulation supports generalizable intuitive physics. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2024. 2

[60] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19795–19806, 2023. 2

[61] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28, 2015. 2

[62] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20406–20417, 2024. 3

[63] Junyu Xie, Charig Yang, Weidi Xie, and Andrew Zisserman. Moving object segmentation: All you need is sam (and flow). In *Proceedings of the Asian conference on computer vision*, pages 162–178, 2024. 3, 5, 6

[64] Ilker Yildirim, Mario Belledonne, Winrich Freiwald, and Josh Tenenbaum. Efficient inverse graphics in biological face processing. *Science advances*, 6(10):eaax5979, 2020. 2

[65] Ilker Yildirim, Max H Siegel, Amir A Soltani, Shraman Ray Chaudhuri, and Joshua B Tenenbaum. Perception of 3d shape integrates intuitive physics and analysis-by-synthesis. *Nature Human Behaviour*, 8(2):320–335, 2024. 2

[66] Guangyao Zhou, Nishad Gothoskar, Lirui Wang, Joshua B Tenenbaum, Dan Gutfreund, Miguel Lázaro-Gredilla, Dileep George, and Vikash K Mansinghka. 3d neural embedding likelihood: Probabilistic inverse graphics for robust 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21625–21636, 2023. 2

A. Overview of Supplementary Materials

This supplementary material provides technical details for the main paper. We provide the following supplementary derivations, experimental details, and quantitative results in this document:

- Section B: Derivation of clustering algorithm from small-variance asymptotics
- Section C: Full derivation of each step in the GenMatter Gibbs sampler
- Section D: Feature-augmented variant and inference algorithm modifications
- Section E: Human RDK psychophysics experiment
- Section F: Gestalt structure-from-motion experiment
- Section G: Details for all 3D RGB experiments, including first frame visualizations, quantitative results on deformable RGB videos, and technical details about the TAP-Vid-DAVIS benchmark. These videos are meant to capture our model's ability to explain deformable matter in a wide variety of settings.

B. Clustering Algorithm from Small-Variance Asymptotics

We present the technical details of the SVA clustering algorithm and recover a rigid group-centric loss function along with an iterative procedure that minimizes it.

Deriving GenMatter as a Clustering Algorithm

$\boldsymbol{\mu}_\ell^{\mathcal{B}}$ update: In the model, $\boldsymbol{\mu}_\ell^{\mathcal{B}} \sim \mathcal{N}(\boldsymbol{\mu}_k^{\mathcal{H}}, \boldsymbol{\Sigma}_k^{\mathcal{H}})$ and $\mathbf{x}_n^t \sim \mathcal{N}(\boldsymbol{\mu}_\ell^{\mathcal{B}}, \boldsymbol{\Sigma}_\ell^{\mathcal{B}})$. Assume $\boldsymbol{\Sigma}_\ell^{\mathcal{B}} = \epsilon\mathbf{I}$ and $\boldsymbol{\Sigma}_k^{\mathcal{H}} = \eta\mathbf{I}$, where $\epsilon/\eta \to 0$. The negative log-conditional of $\boldsymbol{\mu}_\ell^{\mathcal{B}}$ is:

$$\mathcal{L}(\boldsymbol{\mu}_\ell^{\mathcal{B}}) \propto \frac{1}{\epsilon} \sum_{n \in \mathcal{B}_\ell} \left\| \mathbf{x}_n^t - \boldsymbol{\mu}_\ell^{\mathcal{B}} \right\|^2 + \frac{1}{\eta} \left\| \boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}} \right\|^2$$

As $\epsilon/\eta \to 0$, the first term dominates the posterior, so the minimizer of the loss is:

$$\boldsymbol{\mu}_\ell^{\mathcal{B}} = \arg\min_{\boldsymbol{\mu}} \sum_{n \in \mathcal{B}_\ell} \|\mathbf{x}_n^t - \boldsymbol{\mu}\|^2 = \frac{1}{|\mathcal{B}_\ell|} \sum_{n \in \mathcal{B}_\ell} \mathbf{x}_n^t.$$

$\boldsymbol{\mu}_k^{\mathcal{H}}$ update: Assuming $\boldsymbol{\Sigma}_k^{\mathcal{H}} = \eta\mathbf{I}$ and $\epsilon/\sigma_{\mu^{\mathcal{H}}}^2 \to 0$, the negative log-conditional of $\boldsymbol{\mu}_k^{\mathcal{H}}$ can be approximated by:

$$\mathcal{L}(\boldsymbol{\mu}_k^{\mathcal{H}}) \propto \sum_{\ell \in \mathcal{H}_k} \left\| \boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}} \right\|^2$$

where the minimizer is:

$$\boldsymbol{\mu}_k^{\mathcal{H}} = \arg\min_{\mu} \sum_{n \in \mathcal{B}_\ell} \|\boldsymbol{\mu}_\ell^{\mathcal{B}} - \mu\|^2 = \frac{1}{|\mathcal{H}_k|} \sum_{\ell \in \mathcal{H}_k} \boldsymbol{\mu}_\ell^{\mathcal{B}}$$

$\mathbf{R}_k, \mathbf{t}_k$ update: We restrict n in this step to only index points that are assigned to cluster k. Taking the limit of all noise terms $\sigma \to 0$ collapses out the dependence on $\boldsymbol{\mu}_\ell^{\mathcal{B}}$ and gives a deterministic motion model that only depends on the relative position of $\mathbf{x}_n$ with respect to $\boldsymbol{\mu}_k^{\mathcal{H}}$. Noting that we also collapse out $\boldsymbol{\Sigma}_\ell^{\mathcal{V}}$ and σ_V^2, algebraic manipulation gives that the negative log-conditional of $\mathbf{R}_k, \mathbf{t}_k$ is, with $\mathbf{p}_n = \mathbf{x}_n - \boldsymbol{\mu}_k^{\mathcal{H}}$:

$$\mathcal{L}_k(\mathbf{R}_k, \mathbf{t}_k) = \sum_n \left\| \mathbf{x}_n + \mathbf{v}_n - \left(\mathbf{R}_k \mathbf{p}_n + \boldsymbol{\mu}_k^{\mathcal{H}} + \mathbf{t}_k \right) \right\|^2$$

Letting $\mathbf{q}_n = \mathbf{x}_n + \mathbf{v}_n - \boldsymbol{\mu}_k^{\mathcal{H}}$, the loss becomes:

$$\mathcal{L}_k(\mathbf{R}_k, \mathbf{t}_k) = \sum_n \|\mathbf{q}_n - (\mathbf{R}_k \mathbf{p}_n + \mathbf{t}_k)\|^2 .$$

This expression corresponds to the orthogonal Procrustes problem, which has a standard solution. We first define $\bar{\mathbf{p}} = \frac{1}{N}\sum_n \mathbf{p}_n$ and $\bar{\mathbf{q}} = \frac{1}{N}\sum_n \mathbf{q}_n$ and compute $\tilde{\mathbf{p}}_n = \mathbf{p}_n - \bar{\mathbf{p}}$ and $\tilde{\mathbf{q}}_n = \mathbf{q}_n - \bar{\mathbf{q}}$. We then compute the cross-covariance matrix $\mathbf{S}_k = \sum_n \tilde{\mathbf{q}}_n \tilde{\mathbf{p}}_n^\top$ and its SVD $\mathbf{S}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^\top$. The optimal rotation is $\mathbf{R}_k = \mathbf{U}_k \mathbf{V}_k^\top$ and the optimal translation is $\mathbf{t}_k = \bar{\mathbf{q}} - \mathbf{R}_k \bar{\mathbf{p}}$.

$z_n^{\mathcal{B}}$, $z_\ell^{\mathcal{H}}$ update: Small variance analysis gives the same form of the objective function as the previous step. The negative log-conditional for $z_n^{\mathcal{B}}$, $z_\ell^{\mathcal{H}}$ then becomes, with $\mathbf{p}_n = \mathbf{x}_n - \boldsymbol{\mu}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}}$:

$$\mathcal{L}(z_n^{\mathcal{B}}, z_\ell^{\mathcal{H}}) = \sum_n \left\| \mathbf{x}_n + \mathbf{v}_n - \left(\mathbf{R}_{z_\ell^{\mathcal{H}}} \mathbf{p}_n + \boldsymbol{\mu}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}} + \mathbf{t}_{z_\ell^{\mathcal{H}}} \right) \right\|^2$$

This is a discrete combinatorial optimization problem that involves searching through particle and cluster assignments.

C. Blocked Gibbs Sampling

We describe the Gibbs sampling approach in greater detail than in the main text. We first independently describe each blocked Gibbs step in Appendix C.1. Then, we describe the procedure of these steps used for initialization in Appendix C.2 and tracking in Appendix C.3.

C.1. Gibbs Update Steps

There are twelve variables of interest, separated at different hierarchical levels as shown:

1. Cluster-level variables:

$$\{\boldsymbol{\mu}_k^{\mathcal{H}},\ \boldsymbol{\Sigma}_k^{\mathcal{H}},\ \mathbf{R}_k,\ \mathbf{t}_k,\ \pi_k^{\mathcal{H}}\}_{k=1}^K$$

2. Particle-level variables:

$$\{\boldsymbol{\mu}_\ell^{\mathcal{B}},\ \boldsymbol{\Sigma}_\ell^{\mathcal{B}},\ \mathbf{v}_\ell,\ \boldsymbol{\Sigma}_\ell^{\mathcal{V}},\ z_\ell^{\mathcal{H}},\ \pi_\ell^{\mathcal{B}}\}_{\ell=1}^L$$

Algorithm 2 Clustering Algorithm for GenMatter via Small-Variance Asymptotics

Input:
  Number of clusters and particles K, L
  Data point positions $\{\mathbf{x}_n, \mathbf{v}_n\}_{n=1}^N$,
Initialize: Assign data points to particles $z_n^{\mathcal{B}}$, particles to clusters $z_\ell^{\mathcal{H}}$
repeat
  for each particle $\ell = 1, \ldots, L$ **do**
    Compute particle mean: $\boldsymbol{\mu}_\ell^{\mathcal{B}} \leftarrow \frac{1}{|\mathcal{B}_\ell|} \sum_{n: z_n^{\mathcal{B}} = \ell} \mathbf{x}_n$
  end for
  for each cluster $k = 1, \ldots, K$ **do**
    Compute cluster mean: $\boldsymbol{\mu}_k^{\mathcal{H}} \leftarrow \frac{1}{|\mathcal{H}_k|} \sum_{\ell: z_\ell^{\mathcal{H}} = k} \boldsymbol{\mu}_\ell^{\mathcal{B}}$
    Collect point pairs $(\mathbf{x}_n, \mathbf{v}_n)$ assigned to cluster k
    Center points: $\mathbf{p}_n \leftarrow \mathbf{x}_n - \boldsymbol{\mu}_k^{\mathcal{H}}$, $\mathbf{q}_n \leftarrow \mathbf{x}_n + \mathbf{v}_n - \boldsymbol{\mu}_k^{\mathcal{H}}$
    Compute cross-covariance: $\mathbf{S}_k \leftarrow \sum_n \mathbf{q}_n \mathbf{p}_n^\top$
    Compute SVD: $\mathbf{S}_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k^\top$
    Set rotation: $\mathbf{R}_k \leftarrow \mathbf{U}_k \mathbf{V}_k^\top$
    Set translation: $\mathbf{t}_k = \bar{\mathbf{q}} - \mathbf{R}_k \bar{\mathbf{p}}$
  end for
  for each data point $n = 1, \ldots, N$ **do**
    Compute motion loss: $\mathcal{L}_n(z_n^{\mathcal{B}}, z_\ell^{\mathcal{H}}) \leftarrow \left\| \mathbf{x}_n + \mathbf{v}_n - \left(\mathbf{R}_{z_\ell^{\mathcal{H}}} \left(\mathbf{x}_n - \boldsymbol{\mu}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}} \right) + \boldsymbol{\mu}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}} + \mathbf{t}_{z_\ell^{\mathcal{H}}} \right) \right\|^2$
    Update $z_n^{\mathcal{B}}$, $z_\ell^{\mathcal{H}}$ to minimize $\sum_n \mathcal{L}_n(z_n^{\mathcal{B}}, z_\ell^{\mathcal{H}})$
  end for
until assignments converge or objective does not decrease

3. Data point-level variables:

$$\{z_n^{\mathcal{B}}\}_{n=1}^N$$

For each of these variables, we independently describe each of the Gibbs updates.

C.1.1. Data point-to-Particle Assignments ($z_{1:N}^{\mathcal{B}}$)

We update each data point's particle assignment $z_n^{\mathcal{B}}$ for $n = 1, \ldots, N$, using the conditional:

$$p(z_n^{\mathcal{B}} = \ell \mid \mathbf{x}_n, \mathbf{v}_n, \text{rest}) \propto \pi^{\mathcal{B}}(\ell) \cdot \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_\ell^{\mathcal{B}}, \boldsymbol{\Sigma}_\ell^{\mathcal{B}}) \cdot \mathcal{N}(\mathbf{v}_n \mid \mathbf{v}_\ell, \boldsymbol{\Sigma}_\ell^{\mathcal{V}})$$

The prior is given by categorical weights $\pi^{\mathcal{B}}$; the likelihood is a product of two Gaussians over position $\mathbf{x}_n$ and velocity $\mathbf{v}_n$. We compute unnormalized log-probabilities $\tilde{p}_{n,\ell}$ for each particle:

$$\tilde{p}_{n,\ell} = \log \pi^{\mathcal{B}}(\ell) + \log \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_\ell^{\mathcal{B}}, \boldsymbol{\Sigma}_\ell^{\mathcal{B}}) + \log \mathcal{N}(\mathbf{v}_n \mid \mathbf{v}_\ell, \boldsymbol{\Sigma}_\ell^{\mathcal{V}})$$

and normalize to obtain the categorical:

$$p(z_n^{\mathcal{B}} = \ell) = \frac{\exp(\tilde{p}_{n,\ell})}{\sum_{\ell'=1}^L \exp(\tilde{p}_{n,\ell'})}$$

from which we sample:

$$z_n^{\mathcal{B}} \sim \text{Categorical}(p(z_n^{\mathcal{B}} = 1), \ldots, p(z_n^{\mathcal{B}} = L))$$

All data points are jointly reassigned in a blocked manner, each selecting the particle that best explains its position and motion, weighted by the prior over particles.

C.1.2. Particle Mixture Weights $\boldsymbol{\pi}^{\mathcal{B}}$

We update the particle mixture weights $\boldsymbol{\pi}^{\mathcal{B}}$ conditioned on data point-to-particle assignments $\{z_n^{\mathcal{B}}\}$. By Dirichlet–Categorical conjugacy, the conditional distribution becomes:

$$\boldsymbol{\pi}^{\mathcal{B}} \mid \{z_n^{\mathcal{B}}\} \sim \text{Dir}(\beta_1 + M_1, \ldots, \beta_L + M_L)$$

where $M_\ell = \#\{n : z_n^{\mathcal{B}} = \ell\}$ counts how many data points are currently assigned to each particle ℓ. This step re-weights the prior particle proportions according to updated data point assignments.

C.1.3. Particle Spatial Means $\boldsymbol{\mu}_\ell^{\mathcal{B}}$

We update each particle center $\boldsymbol{\mu}_\ell^{\mathcal{B}}$ from its Gaussian conditional, combining: (1) a spatial prior from its assigned cluster, (2) position likelihoods from assigned data points, and (3) a velocity constraint derived from rigid motion.

Let $\mathbf{A}_\ell = \mathbf{R}_{z_\ell^{\mathcal{H}}} - \mathbf{I}$ and $\mathbf{b}_\ell = \mathbf{t}_{z_\ell^{\mathcal{H}}} - \mathbf{A}_\ell \boldsymbol{\mu}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}}$. Then:

$$\mathbf{v}_\ell \sim \mathcal{N}(\mathbf{A}_\ell \boldsymbol{\mu}_\ell^{\mathcal{B}} + \mathbf{b}_\ell, \sigma_V^2 \mathbf{I})$$

The conditional distribution is a Gaussian-Gaussian conjugate of the form:

$$\boldsymbol{\mu}_\ell^{\mathcal{B}} \,|\boldsymbol{\mu}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}}, \boldsymbol{\Sigma}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}}, \mathbf{v}_\ell, \mathbf{t}_{z_\ell^{\mathcal{H}}}, \mathbf{R}_{z_\ell^{\mathcal{H}}}, \sigma_V^2, \\ \{\mathbf{x}_n : z_n^{\mathcal{B}} = \ell\}, \boldsymbol{\Sigma}_\ell^{\mathcal{B}} \sim \mathcal{N}(\mathbf{P}_\ell^{-1}\mathbf{m}_\ell, \mathbf{P}_\ell^{-1})$$

with precision and mean:

$$\mathbf{P}_\ell = (\boldsymbol{\Sigma}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}})^{-1} + N_\ell(\boldsymbol{\Sigma}_\ell^{\mathcal{B}})^{-1} + \frac{1}{\sigma_V^2}\mathbf{A}_\ell^\top \mathbf{A}_\ell$$
$$\mathbf{m}_\ell = (\boldsymbol{\Sigma}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}})^{-1}\boldsymbol{\mu}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}} + (\boldsymbol{\Sigma}_\ell^{\mathcal{B}})^{-1}\mathbf{S}_\ell \\ + \frac{1}{\sigma_V^2}\mathbf{A}_\ell^\top(\mathbf{v}_\ell - \mathbf{b}_\ell),$$

where N_ℓ is the number of data points assigned to particle ℓ, and $\mathbf{S}_\ell = \sum_{n:z_n^{\mathcal{B}}=\ell} \mathbf{x}_n$ is the sum of their positions.

C.1.4. Particle Spatial Covariances $\boldsymbol{\Sigma}_\ell^{\mathcal{B}}$

We update each particle's spatial covariance matrix $\boldsymbol{\Sigma}_\ell^{\mathcal{B}}$ using Normal–Inverse-Wishart conjugacy. Let $N_\ell = \#\{n : z_n^{\mathcal{B}} = \ell\}$ be the number of data points assigned to particle ℓ, and define the scatter matrix:

$$\mathbf{S}_\ell = \sum_{n:z_n^{\mathcal{B}}=\ell} (\mathbf{x}_n - \boldsymbol{\mu}_\ell^{\mathcal{B}})(\mathbf{x}_n - \boldsymbol{\mu}_\ell^{\mathcal{B}})^\top$$

Given an Inverse-Wishart prior $\mathcal{W}^{-1}(\boldsymbol{\Psi}^{\mathcal{B}}, \nu^{\mathcal{B}})$, the conditional distribution is:

$$\boldsymbol{\Sigma}_\ell^{\mathcal{B}} \mid \boldsymbol{\mu}_\ell^{\mathcal{B}}, \{\mathbf{x}_n : z_n^{\mathcal{B}} = \ell\} \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}'_\ell = \boldsymbol{\Psi}^{\mathcal{B}} + \mathbf{S}_\ell, \nu^{\mathcal{B}} + N_\ell)$$

This update adjusts each particle's spatial uncertainty based on the observed spread of its assigned data points.

C.1.5. Particle Velocity Means $\mathbf{v}_\ell$

We update each particle velocity anchor $\mathbf{v}_\ell$ via a Gaussian conditional distribution combining: (1) a rigid motion prior from its assigned cluster, and (2) velocity observations from assigned data points. Let $\bar{\mathbf{v}}_\ell = \mathbf{t}_{z_\ell^{\mathcal{H}}} + (\mathbf{R}_{z_\ell^{\mathcal{H}}} - \mathbf{I})(\boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_{z_\ell^{\mathcal{H}}}^{\mathcal{H}})$ be the prior mean.

Given the set $\{\mathbf{v}_n : z_n^{\mathcal{B}} = \ell\}$ and count $N_\ell = \#\{n : z_n^{\mathcal{B}} = \ell\}$, the conditional is a Gaussian-Gaussian conjugate update:

$$\mathbf{v}_\ell \mid \bar{\mathbf{v}}_\ell, \sigma_V^2, \boldsymbol{\Sigma}_\ell^{\mathcal{V}}, \{\mathbf{v}_n : z_n^{\mathcal{B}} = \ell\} \sim \mathcal{N}(\boldsymbol{\mu}_\ell^v, \boldsymbol{\Sigma}_\ell^v)$$

with:

$$(\boldsymbol{\Sigma}_\ell^v)^{-1} = \frac{1}{\sigma_V^2}\mathbf{I} + N_\ell(\boldsymbol{\Sigma}_\ell^{\mathcal{V}})^{-1}$$
$$\boldsymbol{\mu}_\ell^v = \boldsymbol{\Sigma}_\ell^v \left(\frac{1}{\sigma_V^2}\bar{\mathbf{v}}_\ell + (\boldsymbol{\Sigma}_\ell^{\mathcal{V}})^{-1} \sum_{n:z_n^{\mathcal{B}}=\ell} \mathbf{v}_n \right)$$

This update accounts for the velocity prediction from the cluster's rigid transform along with the empirical data point velocities, with each contribution weighted by its respective uncertainty.

C.1.6. Particle Velocity Covariances $\boldsymbol{\Sigma}_\ell^{\mathcal{V}}$

Each particle's velocity covariance $\boldsymbol{\Sigma}_\ell^{\mathcal{V}}$ is inferred using Normal–Inverse-Wishart conjugacy. Let $N_\ell = \#\{n : z_n^{\mathcal{B}} = \ell\}$ be the number of data points assigned to particle ℓ, and define the velocity scatter:

$$\mathbf{T}_\ell = \sum_{n:z_n^{\mathcal{B}}=\ell} (\mathbf{v}_n - \mathbf{v}_\ell)(\mathbf{v}_n - \mathbf{v}_\ell)^\top$$

Given prior $\mathcal{W}^{-1}(\boldsymbol{\Psi}^{\mathcal{V}}, \nu^{\mathcal{V}})$, the conditional distribution is:

$$\boldsymbol{\Sigma}_\ell^{\mathcal{V}} \mid \mathbf{v}_\ell, \{\mathbf{v}_n : z_n^{\mathcal{B}} = \ell\} \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}'_\ell = \boldsymbol{\Psi}^{\mathcal{V}} + \mathbf{T}_\ell,\ \nu^{\mathcal{V}} + N_\ell)$$

This update reflects the velocity noise structure within each particle, accounting for spread in assigned data point velocities.

C.1.7. Particle-to-Cluster Assignments ($z_{1:L}^{\mathcal{H}}$)

We update each particle's cluster assignment $z_\ell^{\mathcal{H}}$ for $\ell = 1, \ldots, L$, using the conditional:

$$p(z_\ell^{\mathcal{H}} = k \mid \boldsymbol{\mu}_\ell^{\mathcal{B}}, \mathbf{v}_\ell, \text{rest}) \propto \pi^{\mathcal{H}}(k) \cdot \mathcal{N}(\boldsymbol{\mu}_\ell^{\mathcal{B}} \mid \boldsymbol{\mu}_k^{\mathcal{H}}, \boldsymbol{\Sigma}_k^{\mathcal{H}}) \\ \cdot \mathcal{N}\big(\mathbf{v}_\ell \mid \mathbf{t}_k + (\mathbf{R}_k - \mathbf{I}) \\ \times (\boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}}), \sigma_V^2 \mathbf{I}\big)$$

The prior is given by categorical weights $\pi^{\mathcal{H}}$; the likelihood combines a spatial Gaussian over the particle's position $\boldsymbol{\mu}_\ell^{\mathcal{B}}$ and a velocity Gaussian that accounts for rigid-body motion induced by the cluster's rotation $\mathbf{R}_k$ and translation $\mathbf{t}_k$. We compute unnormalized log-probabilities $\tilde{p}_{\ell,k}$ for each cluster:

$$\tilde{p}_{\ell,k} = \log \pi^{\mathcal{H}}(k) + \log \mathcal{N}(\boldsymbol{\mu}_\ell^{\mathcal{B}} \mid \boldsymbol{\mu}_k^{\mathcal{H}}, \boldsymbol{\Sigma}_k^{\mathcal{H}}) \\ + \log \mathcal{N}\big(\mathbf{v}_\ell \mid \mathbf{t}_k + (\mathbf{R}_k - \mathbf{I})(\boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}}), \sigma_V^2 \mathbf{I}\big)$$

and normalize to obtain the categorical:

$$p(z_\ell^{\mathcal{H}} = k) = \frac{\exp(\tilde{p}_{\ell,k})}{\sum_{k'=1}^K \exp(\tilde{p}_{\ell,k'})}$$

from which we sample:

$$z_\ell^{\mathcal{H}} \sim \text{Categorical}(p(z_\ell^{\mathcal{H}} = 1), \ldots, p(z_\ell^{\mathcal{H}} = K))$$

This constitutes a blocked Gibbs step, where all particle-to-cluster assignments are jointly updated. Each particle selects the cluster whose spatial and rigid motion parameters best explain its position and velocity.

C.1.8. Cluster Mixture Weights $\boldsymbol{\pi}^{\mathcal{H}}$

We update the cluster mixture weights $\boldsymbol{\pi}^{\mathcal{H}}$ given particle-to-cluster assignments $\{z_\ell^{\mathcal{H}}\}$. Using Dirichlet–Categorical conjugacy, the conditional is:

$$\boldsymbol{\pi}^{\mathcal{H}} \mid \{z_\ell^{\mathcal{H}}\} \sim \mathrm{Dir}(\alpha_1 + N_1, \ldots, \alpha_K + N_K)$$

where $N_k = \#\{\ell : z_\ell^{\mathcal{H}} = k\}$ is the number of particles assigned to cluster k. This step updates the prior cluster proportions based on current assignment counts.

C.1.9. Cluster Spatial Means $\boldsymbol{\mu}_k^{\mathcal{H}}$

We update each cluster center $\boldsymbol{\mu}_k^{\mathcal{H}}$ via a Gaussian conditional that integrates: (1) a Gaussian prior centered at $\mu^{\mathcal{H}}$, (2) assigned particle centers $\boldsymbol{\mu}_\ell^{\mathcal{B}}$, and (3) observed particle velocities corrected by the cluster's affine transform.

Let $\mathbf{A}_k = \mathbf{I} - \mathbf{R}_k$ and $\mathbf{b}_\ell = \mathbf{t}_k - \mathbf{A}_k \boldsymbol{\mu}_\ell^{\mathcal{B}}$. Then the velocity residual is:

$$\mathbf{r}_\ell = \mathbf{v}_\ell - \mathbf{b}_\ell$$

Given the sum of assigned particle means $\mathbf{S}_k = \sum_{\ell : z_\ell^{\mathcal{H}} = k} \boldsymbol{\mu}_\ell^{\mathcal{B}}$, the velocity residual sum $\mathbf{R}_k = \sum_{\ell : z_\ell^{\mathcal{H}} = k} \mathbf{r}_\ell$, and the count $N_k = \#\{\ell : z_\ell^{\mathcal{H}} = k\}$ of particles assigned to cluster k, the conditional is:

$$\begin{aligned}\boldsymbol{\mu}_k^{\mathcal{H}} \mid & \mu^{\mathcal{H}}, \sigma_H^2, \boldsymbol{\Sigma}_k^{\mathcal{H}}, \mathbf{t}_k, \mathbf{R}_k, \sigma_V^2, \mathbf{R}_k, \\ & \{\boldsymbol{\mu}_\ell^{\mathcal{B}}, \mathbf{v}_\ell : z_\ell^{\mathcal{H}} = k\} \sim \mathcal{N}(P_k^{-1}\mathbf{m}_k, P_k^{-1})\end{aligned}$$

with:

$$\begin{aligned}P_k &= \frac{1}{\sigma_H^2}\mathbf{I} + N_k \left(\boldsymbol{\Sigma}_k^{\mathcal{H}\,-1} + \frac{1}{\sigma_V^2}\mathbf{A}_k^\top \mathbf{A}_k \right) \\ \mathbf{m}_k &= \frac{1}{\sigma_H^2}\mu^{\mathcal{H}} + \boldsymbol{\Sigma}_k^{\mathcal{H}\,-1}\mathbf{S}_k + \frac{1}{\sigma_V^2}\mathbf{A}_k^\top \mathbf{R}_k\end{aligned}$$

This update integrates global priors, spatial evidence from assigned particles, and velocity-based constraints under rigid motion. We parallelize this step by batching cluster-level quantities over K and particle-level inputs over L, with per-cluster residual aggregation. The final blocked multivariate normal update samples new cluster means in parallel from their respective posteriors.

C.1.10. Cluster Spatial Covariances $\boldsymbol{\Sigma}_k^{\mathcal{H}}$

We infer each cluster's spatial covariance $\boldsymbol{\Sigma}_k^{\mathcal{H}}$ using a Normal–Inverse-Wishart update conditioned on its assigned particles. Let $L_k = \#\{\ell : z_\ell^{\mathcal{H}} = k\}$ be the number of particles assigned to cluster k, and define the cluster-centered scatter:

$$\mathbf{S}_k = \sum_{\ell : z_\ell^{\mathcal{H}} = k} (\boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}})(\boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}})^\top$$

Given the Inverse-Wishart prior $\mathcal{W}^{-1}(\boldsymbol{\Psi}^{\mathcal{H}}, \nu^{\mathcal{H}})$, the conditional becomes, with $\boldsymbol{\Psi}'_k = \boldsymbol{\Psi}^{\mathcal{H}} + \mathbf{S}_k$ and $\nu'_k = \nu^{\mathcal{H}} + L_k$:

$$\boldsymbol{\Sigma}_k^{\mathcal{H}} \mid \boldsymbol{\mu}_k^{\mathcal{H}}, \{\boldsymbol{\mu}_\ell^{\mathcal{B}} : z_\ell^{\mathcal{H}} = k\} \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}'_k, \nu'_k)$$

This posterior captures the spatial extent of each cluster based on the spread of its assigned particle centers.

C.1.11. Cluster Rotation $\mathbf{R}_k$

We update each cluster's rotation matrix $\mathbf{R}_k$ by evaluating a discrete set of candidate rotations $\{\mathbf{R}^{(j)}\}_{j=1}^{M_r}$ drawn from a spherical cap (e.g., von Mises–Fisher). For each candidate, we compute a probability based on how well the induced rigid motion explains observed particle velocities. Let $\bar{\mathbf{v}}_\ell^{(j)} = \mathbf{t}_k + (\mathbf{R}^{(j)} - \mathbf{I})(\boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}})$ be the expected velocity for particle ℓ under candidate j. Then:

$$\log \tilde{q}_j = \sum_{\ell : z_\ell^{\mathcal{H}} = k} \log \mathcal{N}(\mathbf{v}_\ell \mid \bar{\mathbf{v}}_\ell^{(j)}, \sigma_V^2 \mathbf{I})$$

Adding the prior log-probabilities $\log p(\mathbf{R}^{(j)})$, we normalize the log-scores to obtain:

$$q_j = \frac{\exp(\log \tilde{q}_j + \log p(\mathbf{R}^{(j)}))}{\sum_{j'=1}^{M_r} \exp(\log \tilde{q}_{j'} + \log p(\mathbf{R}^{(j')}))}$$

from which we sample:

$$\mathbf{R}_k \sim \mathrm{Categorical}(\{q_j\}_{j=1}^{M_r})$$

This update selects the rotation that best aligns relative particle positions with their observed velocities, conditioned on the current cluster translation $\mathbf{t}_k$, velocity noise σ_V^2, cluster means ($\boldsymbol{\mu}_k^{\mathcal{H}}$) and assigned particle means ($\{\boldsymbol{\mu}_\ell^{\mathcal{B}} : z_\ell^{\mathcal{H}} = k\}$).

C.1.12. Cluster Translation Velocities $\mathbf{t}_k$

We update each cluster's translation velocity $\mathbf{t}_k$ by evaluating a discrete set of candidate translations $\{\mathbf{t}^{(m)}\}_{m=1}^{M_t}$ sampled from an isotropic Gaussian prior $\mathcal{N}(\mathbf{0}, s^2\mathbf{I})$. Each candidate is scored based on how well it explains the observed particle velocities under the current rotation $\mathbf{R}_k$. Let $\bar{\mathbf{v}}_\ell^{(m)} = \mathbf{t}^{(m)} + (\mathbf{R}_k - \mathbf{I})(\boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}})$ be the expected velocity for particle ℓ under candidate m. Then:

$$\log \tilde{p}_m = \sum_{\ell : z_\ell^{\mathcal{H}} = k} \log \mathcal{N}(\mathbf{v}_\ell \mid \bar{\mathbf{v}}_\ell^{(m)}, \sigma_V^2 \mathbf{I})$$

We add prior log-probabilities and normalize to form a categorical:

$$p_m = \frac{\exp(\log \tilde{p}_m + \log p(\mathbf{t}^{(m)}))}{\sum_{m'=1}^{M_t} \exp(\log \tilde{p}_{m'} + \log p(\mathbf{t}^{(m')}))}$$

from which we sample:

$$\mathbf{t}_k \sim \mathrm{Categorical}(\{p_m\}_{m=1}^{M_t})$$

This update selects the translation that best explains the observed particle velocities, conditioned on current cluster rotation $\mathbf{R}_k$, velocity noise σ_V^2, cluster center $\boldsymbol{\mu}_k^{\mathcal{H}}$, and assigned particle means $\{\boldsymbol{\mu}_\ell^{\mathcal{B}} : z_\ell^{\mathcal{H}} = k\}$.

C.2. Initialization Procedure

It is well known that MCMC chains are sensitive to the initialization and should be initialized at a high density region. In both the 2D and 3D variants of GenMatter, we use K-Means clustering and a data-driven approach to initialize the MCMC chain for the initial frame ($T = 0$).

C.2.1. K-Means and Data-driven Initialization at T=0

Given the number of particles (L), we use K-means via a K-Means++ initialization to initialize the particle spatial positions ($\boldsymbol{\mu}_\ell^{\mathcal{B}}$). We then use an additional K-means step to initialize the cluster spatial positions ($\boldsymbol{\mu}_k^{\mathcal{H}}$) by treating the particle spatial positions as data points to cluster.

This K-means initialization provides initial values for assignments at both layers ($z_n^{\mathcal{B}}$, $z_\ell^{\mathcal{H}}$). We then use these assignments to initialize the mixture weights at both layers ($\pi^{\mathcal{B}}$, $\pi^{\mathcal{H}}$) by computing the empirical frequencies of each cluster and normalizing: $\pi_\ell^{\mathcal{B}} = \frac{M_\ell}{N}$ and $\pi_k^{\mathcal{H}} = \frac{N_k}{L}$, where M_ℓ is the number of datapoints assigned to particle ℓ and N_k is the number of particles assigned to cluster k. We initialize the velocity mean of each particle $\mathbf{v}_\ell$ by averaging the observed velocities of the datapoints assigned to it:

$$\mathbf{v}_\ell = \frac{1}{M_\ell} \sum_{n: z_n^{\mathcal{B}} = \ell} \mathbf{v}_n.$$

To initialize the covariance matrices, we compute the sample covariance of the relevant residuals for each component:

1. **Particle Spatial Covariance:**

$$\boldsymbol{\Sigma}_\ell^{\mathcal{B}} = \frac{1}{M_\ell - 1} \sum_{n: z_n^{\mathcal{B}} = \ell} (\mathbf{x}_n - \boldsymbol{\mu}_\ell^{\mathcal{B}})(\mathbf{x}_n - \boldsymbol{\mu}_\ell^{\mathcal{B}})^\top.$$

2. **Particle Velocity Covariance:**

$$\boldsymbol{\Sigma}_\ell^{\mathcal{V}} = \frac{1}{M_\ell - 1} \sum_{n: z_n^{\mathcal{B}} = \ell} (\mathbf{v}_n - \mathbf{v}_\ell)(\mathbf{v}_n - \mathbf{v}_\ell)^\top.$$

3. **Cluster Spatial Covariance:**

$$\boldsymbol{\Sigma}_k^{\mathcal{H}} = \frac{1}{N_k - 1} \sum_{\ell: z_\ell^{\mathcal{H}} = k} (\boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}})(\boldsymbol{\mu}_\ell^{\mathcal{B}} - \boldsymbol{\mu}_k^{\mathcal{H}})^\top.$$

To initialize each cluster's rigid transform $(\mathbf{R}_k, \mathbf{t}_k)$, we apply the Kabsch algorithm to align assigned particle positions with their next-frame displacements. For cluster k, we collect all datapoints $\mathbf{x}_n$ assigned to particles ℓ with $z_\ell^{\mathcal{H}} = k$ and define their estimated displacements $\mathbf{x}'_n = \mathbf{x}_n + \mathbf{v}_n$. Let $\mathcal{X}_k = \{\mathbf{x}_n\}$ and $\mathcal{X}'_k = \{\mathbf{x}'_n\}$ be the source and target sets.

We compute centroids $\bar{\mathbf{x}}_k = \frac{1}{|\mathcal{X}_k|} \sum \mathbf{x}_n$, $\bar{\mathbf{x}}'_k = \frac{1}{|\mathcal{X}'_k|} \sum \mathbf{x}'_n$, and form centered sets $\tilde{\mathbf{x}}_n = \mathbf{x}_n - \bar{\mathbf{x}}_k$, $\tilde{\mathbf{x}}'_n = \mathbf{x}'_n - \bar{\mathbf{x}}'_k$. The cross-covariance matrix is:

$$\mathbf{H}_k = \sum_n \tilde{\mathbf{x}}_n \tilde{\mathbf{x}}_n'^\top$$

We compute the singular value decomposition $\mathbf{H}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^\top$, and define the optimal rotation as:

$$\mathbf{R}_k = \mathbf{V}_k \mathbf{D}_k \mathbf{U}_k^\top$$

where $\mathbf{D}_k$ is defined as:

$$\mathbf{D}_k = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & \det(\mathbf{V}_k \mathbf{U}_k^\top) \end{bmatrix} & \text{(2D model)} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V}_k \mathbf{U}_k^\top) \end{bmatrix} & \text{(3D model)} \end{cases}$$

The corresponding translation is:

$$\mathbf{t}_k = \bar{\mathbf{x}}'_k - \mathbf{R}_k \bar{\mathbf{x}}_k$$

This provides an initialization of cluster motion consistent with the observed displacements of assigned particles. The update is applied independently for each cluster $k = 1, \ldots, K$.

C.2.2. Data-Dependent Hyperparameters

We initialize model hyperparameters directly from empirical statistics computed on the initial frame ($T = 0$). The global cluster location prior $\mu^{\mathcal{H}}$ is set to the median datapoint position, while the prior spatial scale $\boldsymbol{\Psi}^{\mathcal{B}}, \boldsymbol{\Psi}^{\mathcal{H}}, \boldsymbol{\Psi}^{\mathcal{V}}$ are initialized using the median initialized particle and cluster covariances length scales.

The degrees of freedom $\nu^{\mathcal{B}}, \nu^{\mathcal{H}}, \nu^{\mathcal{V}}$ are initialized proportionally to the number of datapoints assigned, weighted by particle or cluster weights:

$$\begin{aligned} \nu^{\mathcal{B}} &= \left\lfloor \text{median}(w_\ell^{\mathcal{B}} \cdot N) \right\rfloor, \\ \nu^{\mathcal{H}} &= \left\lfloor \text{median}(w_k^{\mathcal{H}} \cdot N) \right\rfloor, \\ \nu^{\mathcal{V}} &= \left\lfloor \text{median}(w_\ell^{\mathcal{B}} \cdot N) \right\rfloor \end{aligned}$$

where $w_\ell^{\mathcal{B}}$ and $w_k^{\mathcal{H}}$ are the normalized empirical weights of each particle and cluster.

C.3. Tracking Gibbs Procedure

To perform inference over video sequences, we extend our generative particle model into the sequential filtering regime using a structured Markov Chain Monte Carlo (MCMC) procedure. Specifically, we implement a blocked Gibbs sampler that leverages the causal ordering of the variables from the previous frame to initialize each frame and performs bottom-up inference to refine all data point-, particle-, and cluster-level variables. Our approach maintains a tractable posterior approximation at each timestep by propagating forward a subset of latent variables and resampling the remaining ones conditioned on new observations. This sequential per-frame MCMC design supports inference in dynamic scenes where data associations must be re-inferred at every timestep.

At each timestep t, we target the posterior over latent variables given the observed data point positions $\mathbf{x}^t_{1:N}$ and velocities $\mathbf{v}^t_{1:N}$:

$$p(\boldsymbol{\mu}^t_{\mathcal{H}}, \boldsymbol{\Sigma}^t_{\mathcal{H}}, \mathbf{R}^t_{\mathcal{H}}, \mathbf{t}^t_{\mathcal{H}}, \boldsymbol{\mu}^t_{\mathcal{B}}, \mathbf{v}^t_{\mathcal{B}}, \boldsymbol{\Sigma}^t_{\mathcal{V}}, z^t_{1:N}, z^t_{1:L}, \boldsymbol{\pi}^t_{\mathcal{B}}, \boldsymbol{\pi}^t_{\mathcal{H}} \mid \mathbf{x}^t_{1:N}, \mathbf{v}^t_{1:N})$$

where $\boldsymbol{\Sigma}_{\mathcal{B}}$ (particle spatial covariances) are held fixed throughout tracking in both 2D and 3D experiments to preserve the spatial extent of the deformable visual matter represented by each particle, and particle-to-cluster assignments $z^t_{1:L}$ are held fixed only in the 3D case to keep consistency with the initial scene segmentation.

Particle Propagation and Initialization Each frame begins by propagating the inferred particle means using their previously inferred velocity vectors:

$$\tilde{\boldsymbol{\mu}}^{\mathcal{B},t}_{\ell} = \boldsymbol{\mu}^{\mathcal{B},t-1}_{\ell} + \mathbf{v}^{t-1}_{\ell}$$

This serves as an initialization for the particle positions in the next frame.

First Assignment: Spatial Anchoring Data points are first assigned to particles based on spatial likelihoods alone:

$$p(z^{\mathcal{B},t}_n = \ell \mid \mathbf{x}^t_n) \propto \pi^{\mathcal{B}}_{\ell} \cdot \mathcal{N}(\mathbf{x}^t_n \mid \tilde{\boldsymbol{\mu}}^{\mathcal{B},t}_{\ell}, \boldsymbol{\Sigma}^{\mathcal{B}}_{\ell})$$

This step is crucial because, in the absence of known correspondences across frames, we cannot assume that data point n at time $t-1$ is the same as datapoint n at time t. Instead, we reinterpret each new frame as an unordered set of observations and rely on spatial proximity to propagated particle means to re-establish associations. By using position alone and excluding any top-down beliefs from velocity or cluster structure, this step provides a stable initialization for the rest of the Gibbs updates. Note that this is a partial version of the full assignment step described in Appendix C.1.1, used here to anchor the initial framewise alignment. After assignments, we update the mixture weights $\boldsymbol{\pi}^{\mathcal{B}}$ by sampling from their conjugate Dirichlet distribution (Appendix C.1.2).

Particle Mean Update After data points have been assigned to particles based on spatial proximity, we update each particle's spatial mean to better reflect this assignment. Specifically, we sample the particle mean from its posterior conditioned on the assigned data points and the expected motion induced by its cluster assignment, as detailed in Appendix C.1.3. Since the assignments in the previous Gibbs step compensate for the absence of point-wise correspondences, this update typically results in small adjustments to the propagated means, ensuring that particles remain anchored to observed data while maintaining temporal coherence with the previous frame.

Second Assignment and Particle Refinement A second data point-to-particle assignment uses both spatial and velocity likelihoods as described in Appendix C.1.1:

$$p(z^{\mathcal{B},t}_n = \ell \mid \mathbf{x}^t_n, \mathbf{v}^t_n) \propto \pi^{\mathcal{B}}_{\ell} \cdot \mathcal{N}(\mathbf{x}^t_n \mid \boldsymbol{\mu}^{\mathcal{B}}_{\ell}, \boldsymbol{\Sigma}^{\mathcal{B}}_{\ell}) \cdot \mathcal{N}(\mathbf{v}^t_n \mid \mathbf{v}_{\ell}, \boldsymbol{\Sigma}^{\mathcal{V}}_{\ell})$$

This step helps resolve ambiguous associations by combining spatial proximity with motion information. The mixture weights $\boldsymbol{\pi}^{\mathcal{B}}$ are updated again based on the refined assignments (Appendix C.1.2).

Each particle's velocity mean $\mathbf{v}_{\ell}$ is updated from its posterior as described in Appendix C.1.5, and the velocity covariance $\boldsymbol{\Sigma}^{\mathcal{V}}_{\ell}$ is resampled as shown in Appendix C.1.6. These updates reflect the motion structure inferred from grouped data point velocities.

Cluster-level Updates Each particle is assigned to a cluster using a joint spatial and velocity likelihood as described in Appendix C.1.7, and the cluster mixture weights $\boldsymbol{\pi}^{\mathcal{H}}$ are resampled using the equation in Appendix C.1.8. Conditioned on these assignments, the cluster mean $\boldsymbol{\mu}^{\mathcal{H}}_k$ and spatial covariance $\boldsymbol{\Sigma}^{\mathcal{H}}_k$ are updated from their conditional distributions (Appendix C.1.9 and C.1.10), and the rigid transform $(\mathbf{R}_k, \mathbf{t}_k)$ is inferred by categorical sampling over candidate rotations and translations (Appendix C.1.11 and C.1.12).

In the 3D experiment, particle-to-cluster assignments $z^t_{1:L}$ are held fixed throughout tracking to stabilize the scene representation's semantic content, which provides a reliable prior over object structure. However, cluster parameters including spatial statistics and rigid transforms are still inferred at each frame to update the spatial localization of the structure given in the original segmentation.

D. Feature-augmented Variant

D.1. Model Modification and Initialization

In the feature-augmented variant of our model, we incorporate image features as additional dimensions of the data points. Following the main text, we define augmented data points $\tilde{\mathbf{x}}_n = [\mathbf{x}_n; \mathbf{f}_n]$ where $\mathbf{f}_n$ are feature vectors extracted from the image. We use the first 10 PCA components of DINO features, where the PCA basis is computed by analyzing all per-pixel features across the entire video. Each particle ℓ is associated with a feature mean $\mathbf{f}_{\ell}$, and the sampling process of the data point features $\mathbf{f}_n$ from the particle features is defined as a Gaussian with variance σ^2_F:

$$\mathbf{f}_n \sim \mathcal{N}(\mathbf{f}_{\ell}, \sigma^2_F \mathbf{I})$$

We only fit our per-particle feature parameter during initialization. We perform the steps described in Appendix C.2.1, followed by computing the initial feature mean

of each particle $\mathbf{f}_\ell$ as the average feature vector of its assigned data points:

$$\mathbf{f}_\ell = \frac{1}{M_\ell} \sum_{n: z_n^{\mathcal{B}} = \ell} \mathbf{f}_n$$

where $M_\ell = \#\{n : z_n^{\mathcal{B}} = \ell\}$ is the number of data points assigned to particle ℓ. This feature mean serves as the representative feature vector for each particle throughout inference.

D.2. Data point-to-Particle Assignments with Feature Likelihood

The main modification to the Gibbs sampler involves the data point-to-particle assignment step, which is modified to include feature similarity. We update each data point's particle assignment $z_n^{\mathcal{B}}$ for $n = 1, \ldots, N$, using the conditional distribution:

$$\begin{aligned} p(z_n^{\mathcal{B}} = \ell \mid \mathbf{x}_n, \mathbf{v}_n, \mathbf{f}_n, \text{rest}) \propto \pi^{\mathcal{B}}(\ell) \cdot \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_\ell^{\mathcal{B}}, \boldsymbol{\Sigma}_\ell^{\mathcal{B}}) \\ \cdot \mathcal{N}(\mathbf{v}_n \mid \mathbf{v}_\ell, \boldsymbol{\Sigma}_\ell^{\mathcal{V}}) \\ \cdot \mathcal{N}(\mathbf{f}_n \mid \mathbf{f}_\ell, \sigma_F^2 \mathbf{I}) \end{aligned}$$

The prior is given by categorical weights $\pi^{\mathcal{B}}$, and the likelihood now consists of three independent Gaussian terms: one for position $\mathbf{x}_n$, one for velocity $\mathbf{v}_n$, and one for features $\mathbf{f}_n$. The feature likelihood uses an isotropic covariance $\sigma_F^2 \mathbf{I}$, which assumes that the features are independent and identically distributed.

We compute unnormalized log-probabilities $\tilde{p}_{n,\ell}$ for each particle:

$$\begin{aligned} \tilde{p}_{n,\ell} = \log \pi^{\mathcal{B}}(\ell) + \log \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_\ell^{\mathcal{B}}, \boldsymbol{\Sigma}_\ell^{\mathcal{B}}) \\ + \log \mathcal{N}(\mathbf{v}_n \mid \mathbf{v}_\ell, \boldsymbol{\Sigma}_\ell^{\mathcal{V}}) + \log \mathcal{N}(\mathbf{f}_n \mid \mathbf{f}_\ell, \sigma_F^2 \mathbf{I}) \end{aligned}$$

and normalize to obtain the categorical conditional distribution:

$$p(z_n^{\mathcal{B}} = \ell) = \frac{\exp(\tilde{p}_{n,\ell})}{\sum_{\ell'=1}^{L} \exp(\tilde{p}_{n,\ell'})}$$

from which we sample:

$$z_n^{\mathcal{B}} \sim \text{Categorical}(p(z_n^{\mathcal{B}} = 1), \ldots, p(z_n^{\mathcal{B}} = L))$$

This update is also a blocked update, executed in a computational manner similar to Appendix C.1.1.

E. Human Psychophysics Experiment

A total of 9 RDKs were created, with each RDK having 3 separate time points and locations where we introduce the red and green dot probes to create a total of 27 stimuli. We recruited a total of 150 human participants through the Prolific platform and all participants were paid at least the local minimum wage for an expected completion time of 4 minutes. The study was designed and conducted under an approved institutional review board (IRB) protocol. All demographic data collected were fully anonymized, and no personally identifying information was provided or collected. All participants were filtered for the following conditions:

1. Fluent in English as the study is conducted in English.
2. Explicitly declared to not have color-blindness, as this study requires each participant to distinguish the red and green probes clearly from the rest of the points in the stimuli.
3. Has normal to corrected vision, as this study requires clear vision of the stimuli.

The instructions as viewed on Prolific for this study can be seen in Figure 7. We used Google Forms to conduct the data collection.

The instructions were repeated in the Google Form and each participant saw two familiarization trials with feedback on the correct answer. Figure 8 shows how these familiarization trials looked to the participant.

F. Gestalt 3D Inference

We provide additional implementation details for the Gestalt 3D structure-from-motion experiments described in the main paper.

Data preprocessing We compute RAFT optical flow and VideoDepthAnything monocular depth on the native 1000×1000 frames, then downsample to 96×96 for inference. We lift 2D pixels to 3D using a pinhole camera model with focal length scaled by 2.0 to enhance depth separation, and compute 3D motion vectors from optical flow.

Initialization GenMatter uses $L = 100$ particles and $K = 5$ clusters. We use the initialization procedure in Appendix C.2.1, with the addition of a coarse segmentation mask proposal containing all data points with flow magnitude above the median flow magnitude, as well as data points within a standard deviation of the median depth. These assumptions are loose and apply to any natural image regime where GenMatter could be run, and are used only to accelerate MCMC burn-in (since a proposal does not change the posterior being approximated). We run 50 Gibbs sweeps on frame 0 to initialize all model parameters.

Per-frame Gibbs schedule For tracking frames $t = 1, \ldots, 4$, we apply 20 Gibbs iterations focused on velocity parameters, followed by 500 full Gibbs sweeps. Particle-to-cluster assignments and particle spatial covariances remain fixed throughout tracking, but data point-to-particle assignments are resampled at each frame.

Figure 9 shows an example of the Gestalt structure-from-motion stimuli with different textures. We visualize the first frame of scene 00000 rendered with seven different texture patterns. The Gestalt experiment uses 20 scenes (00000–00019), each rendered with these seven textures (00, 07,

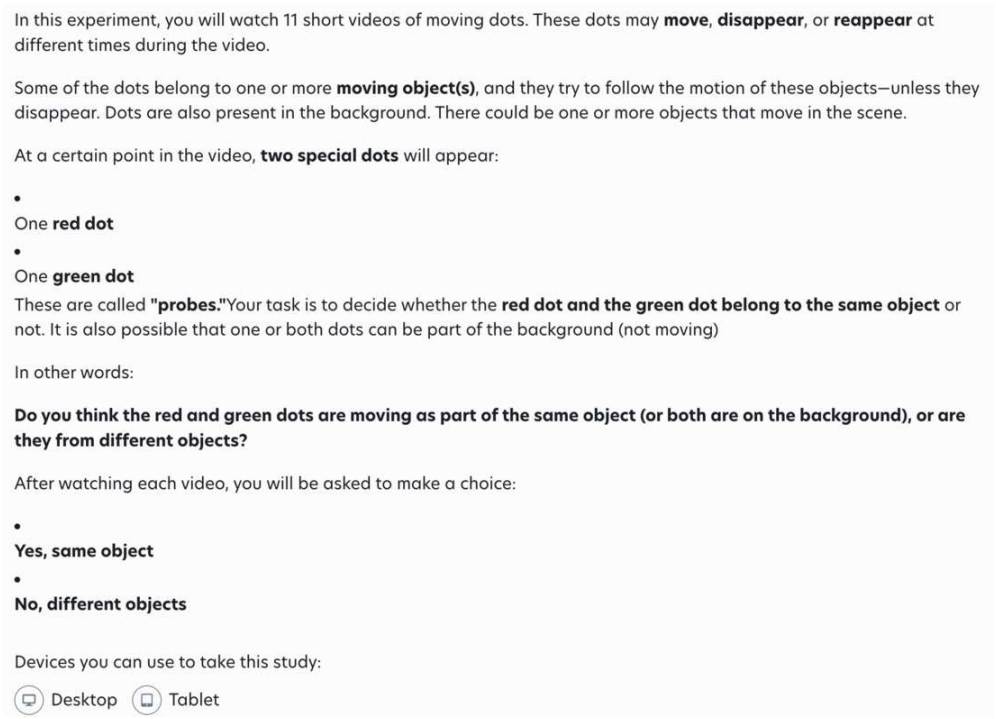


Figure 7. Instructions shown to all participants. This task was allowed to be conducted on either a desktop or a tablet. The 11 videos mentioned refer to the 2 familiarization trials and 9 test trials. Details of compensation is cropped out to preserve anonymity.

13, 16, 21, 22, 25), yielding 140 total stimuli to evaluate structure-from-motion segmentation across diverse visual appearances.

G. RGB 3D Inference

Table 3. **Jaccard Index on Supplementary Videos.** We report the Jaccard index for GenMatter and CoTracker3 on each supplementary video. Best per video is bolded.

Video	GenMatter	CoTracker3
cloth_bag	**0.84**	0.32
gray_jacket	0.91	**0.98**
jello	**0.93**	0.57
manta_ray	0.96	**1.00**
eagle	0.86	**0.89**
ostrich	0.91	**0.97**
purple_jacket	0.81	**0.99**
snake	**0.93**	0.47
whiskey_swirl	0.49	**0.79**
wine_swirl	0.67	**0.97**

G.1. Experimental Details

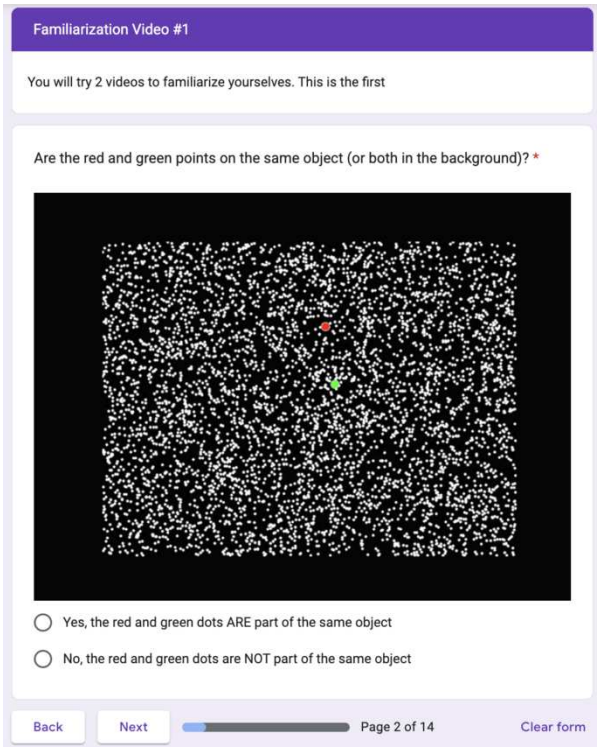
GenMatter Setup We provide additional technical implementation details for our TAP-Vid-DAVIS experiments.

Initialization At frame 0, we perform 30 Gibbs sweeps to initialize all model parameters before sequential tracking begins. Particles are initialized through hierarchical K-means clustering on 3D positions lifted from tracked points using monocular depth estimates. For each particle, DINO features $\mathbf{f}_\ell^{\mathcal{B}}$ are initialized by averaging DINO descriptors over all pixels assigned to that particle. When SAM2 frame-0 masks are available, we adaptively determine the number of clusters K based on the number of components in the mask rather than fixing K. We sample tracked points uniformly across each frame for the whole video.

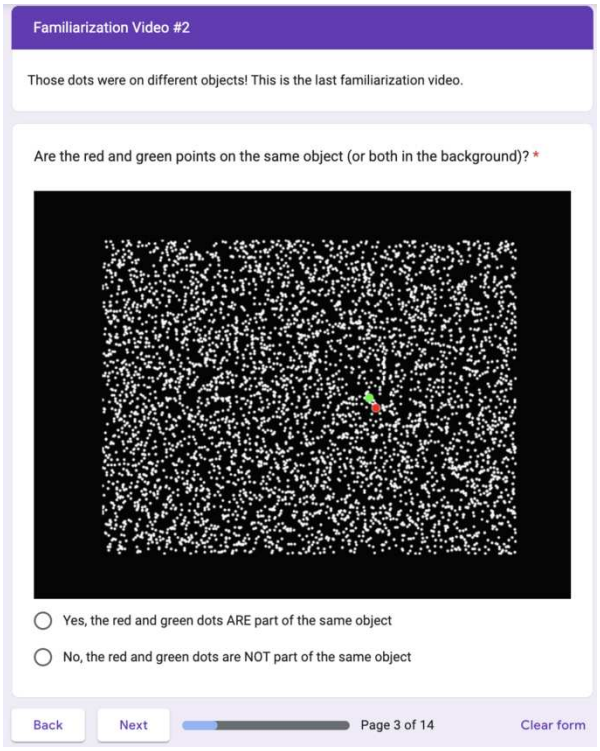
Per-frame Gibbs schedule For each frame $t > 0$, we apply a fixed schedule of blocked Gibbs updates. We apply updates to:

- cluster-level rigid transformations $(\mathbf{R}_k, \mathbf{t}_k)$
- data point-to-particle assignments $z_n^{\mathcal{B}}$, conditioned on position likelihood only (with outliers disabled)
- data point-to-particle assignments $z_n^{\mathcal{B}}$, with full position-velocity-feature likelihood and $p_{\text{outlier}} = 0.1$
- particle spatial means $\boldsymbol{\mu}_\ell^{\mathcal{B}}$ and velocity parameters $(\mathbf{v}_\ell^{\mathcal{B}}, \boldsymbol{\Sigma}_\ell^{\mathcal{V}})$
- particle DINO features $\mathbf{f}_\ell^{\mathcal{B}}$

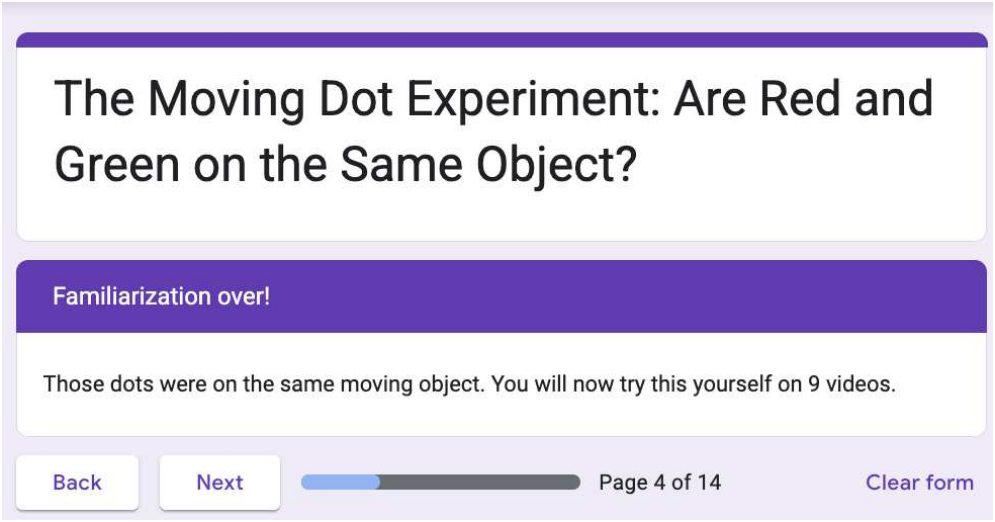
Multiple iterations are performed for spatial and velocity updates to ensure convergence. Mixture weights $\boldsymbol{\pi}^{\mathcal{B}}$ and $\boldsymbol{\pi}^{\mathcal{H}}$ are updated via Dirichlet conditionals after their respective assignment steps.



(a) First familiarization trial



(b) Second familiarization trial with ground truth answer for first familiarization trial revealed.

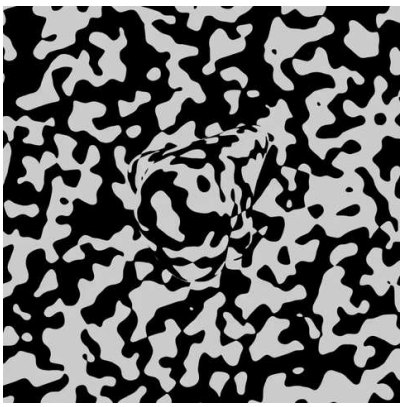


(c) Ground Truth answer for Second Familiarization trial revealed.

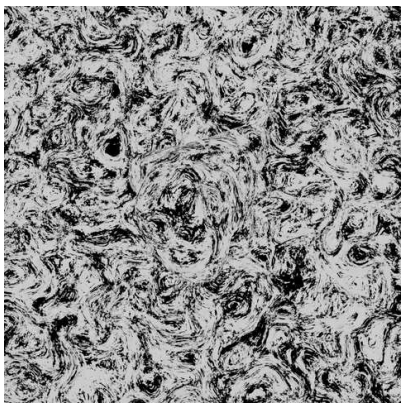
Figure 8. Visual descriptions of the familiarization trials, shown to all 150 participants

Outlier handling During tracking (frames $t > 0$), we enable outliers by including an additional mixture component with weight $p_{\text{outlier}} = 0.1$. The outlier likelihood for a data point with velocity $\mathbf{v}_n$ is modeled as a Gamma distribution on speed $\|\mathbf{v}_n\|$ with shape parameter α and rate parameter β, which accounts for velocity outliers typically arising from unreliable motion estimates at object boundaries.

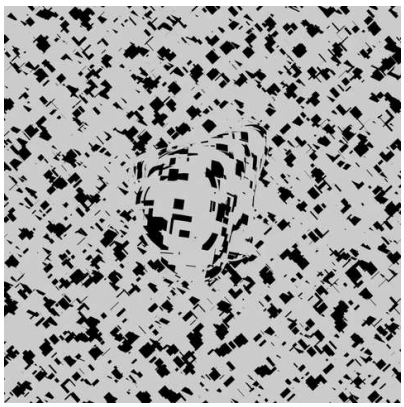
CoTracker3 Setup We run CoTracker3 with its default PyTorch Hub offline mode implementation. We initialize 500 query points in frame 0, matching the particle count used in GenMatter for fair comparison. Query points are randomly sampled uniformly across the first frame. Because we do not use ground-truth segmentation masks during initialization, query points are distributed uniformly across the object and



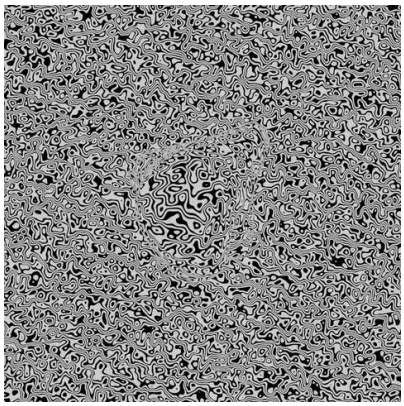
Texture 00



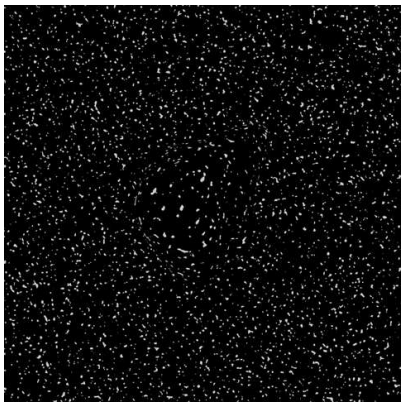
Texture 07



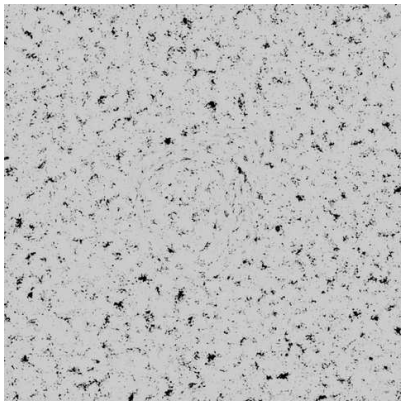
Texture 13



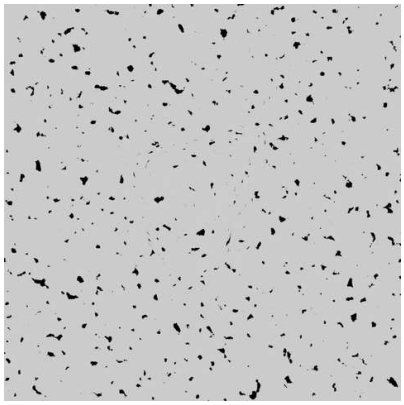
Texture 16



Texture 21



Texture 22



Texture 25

Figure 9. **Example Gestalt Stimuli with Different Textures.** First frame of scene 00000 rendered with seven different texture patterns. The Gestalt experiment uses 20 scenes (00000–00019), each rendered with these seven textures (00, 07, 13, 16, 21, 22, 25), yielding 140 total stimuli to evaluate structure-from-motion segmentation. In the paper, these textures are referenced sequentially as (00, 01, 02, 03, 04, 05, 06).

background regions. As a result, the tracker cannot concentrate query points on the object, making it difficult to share statistical strength across object particles. We evaluate tracking quality using the same particle-based metrics as GenMatter, with the difference that at each frame, we only consider points which CoTracker3 has identified as visible. We first classify particles as object or background based on their frame-0 location relative to the segmentation mask, and we compute per-frame Jaccard by projecting tracked locations onto ground-truth masks at each timestep. The per-

frame Jaccard indices are averaged to obtain the per-video Jaccard index.

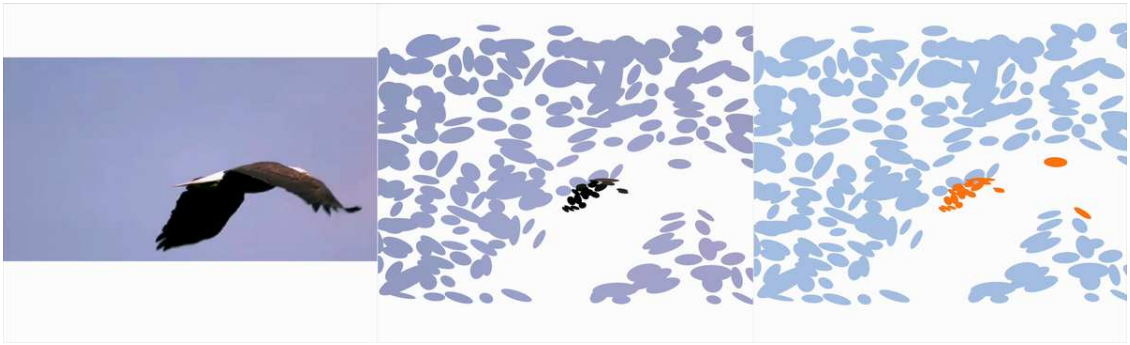
G.2. Supplementary Video Descriptions

We include supplementary videos that visualize the full particle-based inference process across time for the small qualitative deformable dataset introduced in the main paper: `cloth_bag`, `gray_jacket`, `jello`, `manta_ray`, `eagle`, `ostrich`, `purple_jacket`, `snake`, `whiskey_swirl`, and `wine_swirl`. These sequences span a range stuff and things observable in the physical world, including articulated structures, highly deformable solids, and liquids, allowing us to evaluate model performance across the full spectrum of matter interpretable by human vision. The first frame of particle tracking in these videos is shown in Figure 10 , Figure 11, and Figure 12.

Evaluation GenMatter achieves higher average Jaccard (0.83 vs 0.79) on the small qualitative deformable dataset, with strong performance on highly deformable solid matter (`cloth_bag`, `snake`, and `jello` in particular have highly deformable solid matter). It performs weakly on liquid (`wine_swirl` and `whiskey_swirl`) because the appearance of liquid makes it difficult to estimate matter motion, and liquid is particularly unstructured. On the other videos in the set, the performance of both models is similar. This pattern suggests GenMatter's probabilistic particle representation describes highly deformable solid matter better than it describes persistent liquid. The reported Jaccard indices in Table 3 use SAM-generated masks as pseudo-ground-truth, as we do not have ground truth segmentation for these videos. Because GenMatter's initial particle clustering also uses SAM, this evaluation is not as robust as datasets derived from DAVIS. However, visual inspection confirms the SAM-generated masks are accurate, and this evaluation helps us bridge the gap towards more precise evaluation of 3D matter representations.

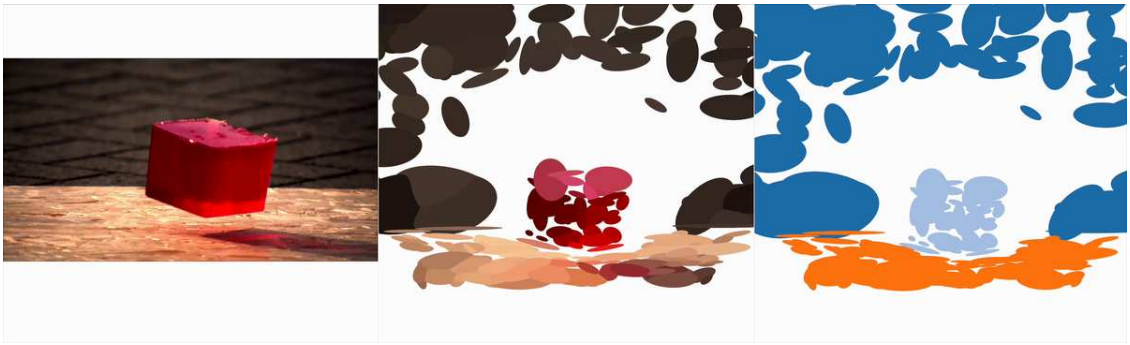
Visualization The supplementary videos apply weight thresholding to particles before rendering. Many particles explain negligible data and have near-zero weights. Our model places little belief in the matter represented by these particles. Removing these low-weight particles improves visual clarity. However, hard thresholding causes flicker at the threshold boundary. Marginal particles will flicker across frames depending on whether their weight exceeds the cut-off. This flicker is a visualization artifact and not model instability.

(a) `cloth_bag`



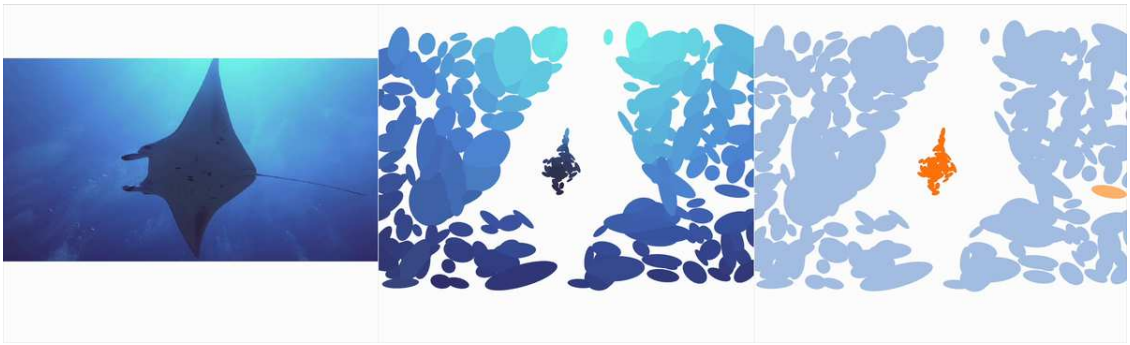
(b) `eagle`

(c) `gray_jacket`



(d) `jello`

Figure 10. First frame visualizations of RGB 3D inference (Part 1). Each image shows the initial particle distribution and segmentation for the respective sequence.



(a) `manta_ray`

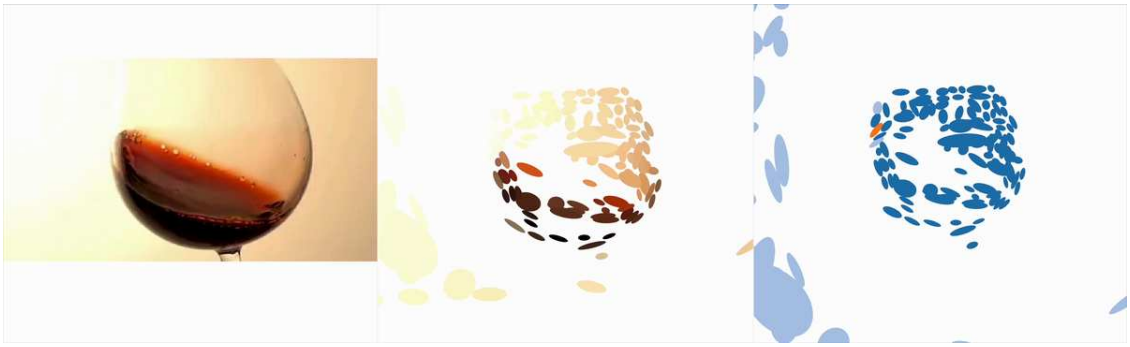
(b) `ostrich`

(c) `purple_jacket`

Figure 11. First frame visualizations of RGB 3D inference (Part 2). Each image shows the initial particle distribution and segmentation for the respective sequence.

(a) `snake`

(b) `whiskey_swirl`



(c) `wine_swirl`

Figure 12. First frame visualizations of RGB 3D inference (Part 3). Each image shows the initial particle distribution and segmentation for the respective sequence.