BI-FORK: GENERATIVE MODELING OF HIGH-DIMENSIONAL BIFURCATING SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Bifurcations are ubiquitous in physical systems, from structural buckling to fluid and climate dynamics, yet they remain largely unexplored in deep learning. At a symmetry-breaking bifurcation, a single input admits multiple equally valid solutions, violating the one-to-one assumption underlying most learned physical surrogates. We introduce Bi-FORK, a generative framework for learning these one-to-many solution maps in high-dimensional systems. Bi-FORK generates complete trajectories through latent flow matching, preserving space and time coherence, and uses repulsion-guided sampling to recover distinct solution branches in a single amortized pass. We evaluate Bi-FORK on buckling beams, mechanical metamaterials, and Allen–Cahn phase separation, spanning continuous, discrete, and field-valued bifurcations with discretizations up to 260,000 points. Bi-FORK recovers the multimodal solution structure while scaling several orders of magnitude beyond prior approaches, opening generative modeling to high-dimensional bifurcating physical systems.

1 INTRODUCTION

Bifurcations are fundamental to nonlinear physical systems and occur across structural mechanics, fluid dynamics, climate dynamics, and phase-transition phenomena (Ruelle & Takens, 1971; Steffen et al., 2018; Rietkerk et al., 2021). A bifurcation is a qualitative change in the number or stability of solutions as a control parameter varies. When a bifurcation breaks a symmetry, it produces a one-to-many mapping from control parameter to system response: multiple, equally valid solutions rather than one. For example, a compressed column is stable up to a critical load, then bifurcates into symmetric buckled branches (Bažant & Cedolin, 2003). Classical numerical methods, including finite-element and finite-volume methods, can compute these solutions accurately. However, for high-dimensional systems, recovering the full bifurcation structure requires stability and mode analyses together with repeated continuation or deflation solves (Brow & Gearhart, 1971; Farrell et al., 2015), which is computationally expensive. Therefore, machine learning (ML) methods are an attractive alternative as they amortize the cost of building the training set over the many fast evaluations of unseen system conditions (Abramson et al., 2024; Alkin et al., 2024; 2025; Giacomini & Díez, 2026; Tang et al., 2026).

However, because the system is one-to-many, training discriminative ML models averages over the solution space and, as a consequence, produces nonphysical predictions (Rupprecht et al., 2017; Kochkov et al., 2024; Price et al., 2025). This has motivated the development of probabilistic methods based on GANs, diffusion models, and more recently flow matching, typically framed around uncertainty or the loss of fine-scale structure (Gao et al., 2023; Yoon et al., 2025; Calzolari & Liu, 2025). Most probabilistic physical surrogates model epistemic or unresolved uncertainty; symmetry-breaking bifurcations instead produce genuinely multiple, equally valid solutions structured by the system's symmetry. Only a few recent approaches treat the bifurcation structure directly as the cause of solution multiplicity (Fabiani et al., 2021; Pichi et al., 2023; Zou et al., 2025; Hendriks et al., 2025b; Guilhoto et al., 2026; Romero & Ortiz, 2026). Nevertheless, these approaches recover systems of at most a few hundred degrees of freedom with small, known, discrete symmetry groups.

The main challenge. Scaling to larger systems is challenging because recovering a bifurcating distribution requires every part of the system to *agree* on which branch it is realizing in both space *and* time. A beam buckling to the left at one end and right at the other is an invalid solution, as is a beam

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

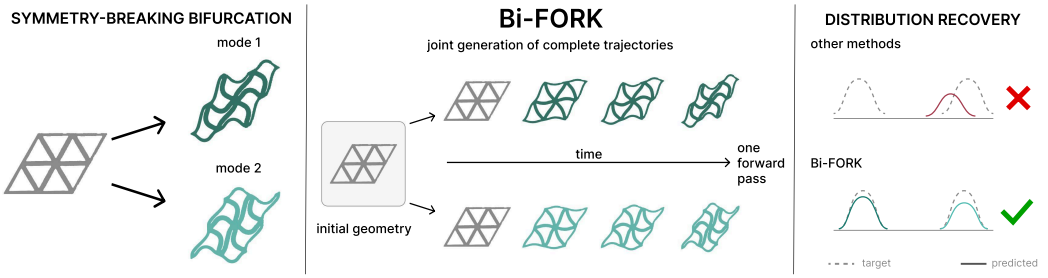


Figure 1: Learning bifurcating solution distributions. Left: Symmetry breaking results in multiple valid trajectories for the same control parameter and initial state. Center: Latent flow matching jointly samples trajectories and resolves distinct branches in one forward pass. Right: Unlike mode-averaging baselines, Bi-FORK recovers the target distribution.

that first buckles to the left and then suddenly switches to buckling to the right halfway through a trajectory. Moreover, bifurcations are typically rare events in trajectories. Training a frame-by-frame autoregressive model to produce trajectories would face significant challenge of capturing this key aspect of the system's behavior while being underrepresented in the training data. Predicting the full trajectory allows the model to instead represent the branch choice globally in latent space, for all steps and elements at the same time. Mesh-based GNN surrogates communicate through local message passing, making such global coordination increasingly difficult as systems grow due to oversmoothing and oversquashing (Oono & Suzuki, 2020; Chen et al., 2020; Cai & Wang, 2020; Alon & Yahav, 2021).

Evaluation is similarly nontrivial. Pointwise metrics such as rank histograms and CRPS (Grönquist et al., 2021; Gao et al., 2023; He et al., 2025; Mahesh et al., 2025; Larsson et al., 2026) assess pointwise calibration but cannot determine whether samples are coherent trajectories, while best-of-K scores ignore most of the predicted distribution. Generic two-sample distances can detect mode collapse but become difficult to estimate in high dimensions and do not exploit the symmetry structure of bifurcating solutions (Hamill, 2001; Bischoff et al., 2024).

We address these challenges with Bi-FORK: Bifurcation Flow-matching Over Repulsion-guided Kinematics, a latent flow-matching framework for symmetry-breaking bifurcations (see Figure 1). Bi-FORK generates full trajectories jointly, keeping branch choices consistent across space and time rather than learning them through local transitions. Because full-resolution execution is intractable, flow matching operates in a compact latent space, decoupling global coordination from discretization. Although compression can average modes, branch identity dominates trajectory variance and is therefore preserved by the reconstruction objective without supervision. At inference, particle guidance (Unterthiner et al., 2018; Corso et al., 2024) repels concurrent samples, acting as a parallelized analog of classical deflation (Brow & Gearhart, 1971; Farrell et al., 2015) to recover distinct solutions. We assess the resulting distribution by matching predictions to their nearest ground truth and, per input, measuring orbit coverage under the relevant symmetry group via the Jensen-Shannon divergence, thereby detecting regime collapse and quantifying coverage of the underlying distribution.

We evaluate the approach on three symmetry-breaking bifurcation problems: buckling beams, buckling mechanical metamaterials, and phase separation described by Allen-Cahn. These systems cover two- and three-dimensional grid- and mesh-based problems, various symmetry groups (both discrete and continuous), trajectories of 32 to 200 timesteps, and discretizations from a handful of points to 260,000. To our knowledge, this is the first demonstration of solution distribution recovery for one-to-many problems at that scale, and we show that our method outperforms existing approaches in both accuracy and efficiency.

Our contributions are: (i) A latent flow-matching framework that models whole trajectories, making branch coherence structural, and scales several orders of magnitude beyond prior work. (ii) A particle-guidance sampling mechanism recovering multiple distinct solutions in one amortized pass, as a parallel analog of deflation. (iii) Diagnostics based on symmetry-orbit matching and per-input Jensen-Shannon divergence that detect regime collapse and extend distributional verification to continuous and infinite symmetry groups.

2 A BLUEPRINT FOR BIFURCATING REGIMES

2.1 BIFURCATION SYSTEMS

Recall that a *bifurcation* is the loss of uniqueness of a stable equilibrium as a control parameter crosses a critical value, with buckling as the canonical mechanical example (Koiter, 1945; Antman, 2005; Hendriks et al., 2025b).

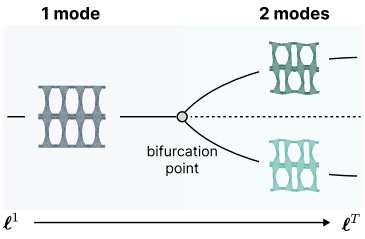


Figure 2: Exemplary pitchfork bifurcation for the buckling metamaterials use case with increasing control protocol $\boldsymbol{\ell}$

Setting. We represent trajectories as either point clouds or scalar fields on a regular grid. In both cases, $\mathbf{M}^t \in \mathbb{R}^{N\times C}$ denotes the modeled response at N spatial locations and timestep $t \in \{1,\dots,T\}$, and $\mathbf{M}^{1:T} \in \mathbb{R}^{T\times N\times C}$ forms the complete response trajectory. For point clouds, the N locations are material points with reference coordinates $\mathbf{X}^1 \in \mathbb{R}^{N\times D}$, where $D \in \{2,3\}$. The response $\mathbf{M}^t$ is their displacement or microfluctuation field, so $C = D$; deformed coordinates $\mathbf{X}^t$ are recovered from $\mathbf{X}^1$ and $\mathbf{M}^t$ using the prescribed loading and dataset-specific kinematics. For regular-grid data, the N locations are fixed grid sites and $\mathbf{M}^t$ is the scalar field itself, so $C = 1$.

Datasets. We consider three physical systems: (i) *Beam3D* models a slender beam that is compressed from one end until it buckles in arbitrary directions. (ii) *Mechanical Metamaterials* captures the buckling behavior of 2D porous materials with different periodic patterns, leading to distinct stable shapes. (iii) *Allen-Cahn* describes how a mixture separates into different phases over time, with a shared homogeneous initial state that evolves into several possible stable configurations.

Modes as a symmetry orbit. A problem instance is a *condition* $\mathbf{c} = (\mathbf{X}^1, \boldsymbol{\ell})$: a reference geometry together with a control protocol $\boldsymbol{\ell} = (\ell^1,\dots,\ell^T)$ that is fixed in advance and not influenced by the system's response. The set of all feasible conditions we denote by $\mathcal{C}$. For the mechanical metamaterials datasets, for example, $\boldsymbol{\ell}$ is a quasi-static loading schedule generated by a single terminal load ($\mathbf{F}^t$ interpolating toward $\mathbf{F}^T$), and the index t enumerates states and load increments simultaneously: $\mathbf{M}^t$ is the equilibrium microfluctuation response to the input $\boldsymbol{\ell}^t$ (see Figure 2).

For Allen-Cahn, the protocol is constant, $\boldsymbol{\ell}^t \equiv (\epsilon, \mu)$, where ϵ is a length scale and μ determines local phase structure. The systems are *multistable*: under a fixed condition, several physically distinct outcomes—*modes*—are admissible. We denote their mode set by $\mathcal{M}_\mathbf{c}$[1], and write $\mathcal{M}_\mathbf{c}^t$ for the set of their responses at timestep t. Symmetry organizes this set: with

$$G_\mathbf{c} = \{\, g \mid g \cdot \mathbf{c} = \mathbf{c} \,\} \tag{1}$$

the symmetry group of the condition, any response $\mathbf{M}$ yields further solutions $g \cdot \mathbf{M}$, so the mode set contains full orbits,

$$G_\mathbf{c} \cdot \mathbf{M}_\mathbf{c}^t \subseteq \mathcal{M}_\mathbf{c}^t, \qquad H_\mathbf{c}^t = \{\, g \in G_\mathbf{c} \mid g \cdot \mathbf{M}_\mathbf{c}^t = \mathbf{M}_\mathbf{c}^t \,\}, \tag{2}$$

where $\mathbf{M}_\mathbf{c}^t$ is a representative response and $H_\mathbf{c}^t$ its isotropy subgroup; each orbit is isomorphic to $G_\mathbf{c}/H_\mathbf{c}^t$ with size $[G_\mathbf{c} : H_\mathbf{c}^t]$. For Beam3D and the mechanical metamaterials, the inclusion is an equality (a single orbit); for Allen-Cahn, the mode set is a union of orbits, one per morphology. Table 1 instantiates $G_\mathbf{c}$, $H_\mathbf{c}$, and $\mathcal{M}_\mathbf{c}$ per dataset; note e.g. the beam, where $G_\mathbf{c} = \mathrm{O}(2)$ and the buckled state retains one reflection, giving the circular orbit $\mathrm{O}(2)/\mathbb{Z}_2 \cong \mathrm{SO}(2)$.

Bifurcation time. We define the bifurcation time as the first step at which the solution loses symmetry,

$$t_b = \min\{t \in \{2,\dots,T\} \mid H_\mathbf{c}^t \subsetneq H_\mathbf{c}^{t-1}\}. \tag{3}$$

In practice, we estimate t_b as the first step at which trajectories under the same condition separate beyond a small threshold. This also accommodates Allen-Cahn, where all trajectories share the same

[1]Here 'Mode' means a distinct admissible outcome, *not* an eigenmode: two symmetry-related metamaterial solutions stem from one buckling eigenmode but count as two modes. The usage agrees with statistics—outcomes follow $p(\cdot \mid \mathbf{c})$, and for finite mode sets each branch is an isolated local maximum of this distribution; for the beam the distribution is uniform along the orbit, which is then its maximizer set.

Table 1: Symmetry structure per dataset: group $G_{\mathbf{c}}$ of the condition, isotropy $H_{\mathbf{c}}$ of an output state, resulting mode set $\mathcal{M}_{\mathbf{c}} \cong G_{\mathbf{c}}/H_{\mathbf{c}}$ (union of orbits for Allen-Cahn).

Dataset	$G_{\mathbf{c}}$	$H_{\mathbf{c}}$	$\mathcal{M}_{\mathbf{c}}$	$\lvert\mathcal{M}_{\mathbf{c}}\rvert$
Beam3D	$\mathrm{O}(2)$	$\mathbb{Z}_2$	circle of directions ϕ	∞
Metamaterials	wallpaper subgroup fixing $\boldsymbol{\ell}$	pattern stabilizer	translated patterns	$1, 2, 4$
Allen-Cahn	domain syms $\times\{u \mapsto \pm u\}$	morphology symmetries	orbits of morphologies	varies

homogeneous initial state but macroscopic modes emerge only later. The choice of representation, the practical estimator, and secondary bifurcations are discussed in Appendix B. Probabilistically, t_b marks the onset of distinct branches in the distribution of outcomes conditioned on the context $\mathbf{c}$ at timestep t, connecting physical symmetry breaking to the need for distribution-valued prediction.

Three regimes of multimodality. Our datasets span discrete ($|\mathcal{M}_{\mathbf{c}}| \in \{1, 2, 4\}$, mechanical metamaterials of Hendriks et al., 2025a), continuous ($|\mathcal{M}_{\mathbf{c}}| = \infty$, Beam3D), and field-valued (Allen-Cahn) mode sets; see Appendix C.

2.2 Learning Task

Given a condition $\mathbf{c}$, the task is to learn the conditional distribution over all feasible trajectories,

$$p\big(\mathbf{M}^{1:T} \mid \mathbf{c}\big)\,, \tag{4}$$

where the support of the target distribution is the mode set $\mathcal{M}_{\mathbf{c}}$. The training datasets differ in their coverage of this support: the mechanical metamaterials enumerate all modes per condition, whereas Beam3D and Allen-Cahn provide samples from the mode set. We assume unknown perturbations that push the system toward one solution within an orbit have no preferred direction. Under this assumption, the condition's symmetry implies that modes within an orbit are equally likely, so for single-orbit datasets the target is uniform over $\mathcal{M}_{\mathbf{c}}$. Unlike standard trajectory prediction, the mapping from $\mathbf{c}$ to the system evolution is one-to-many: multiple physically valid trajectories arise under the same condition. During inference, the model receives $\mathbf{c}$ and generates one or more trajectory samples from a learned distribution $p_{\theta_{\text{flow}}}(\mathbf{M}^{1:T} \mid \mathbf{c})$.

2.3 Modeling

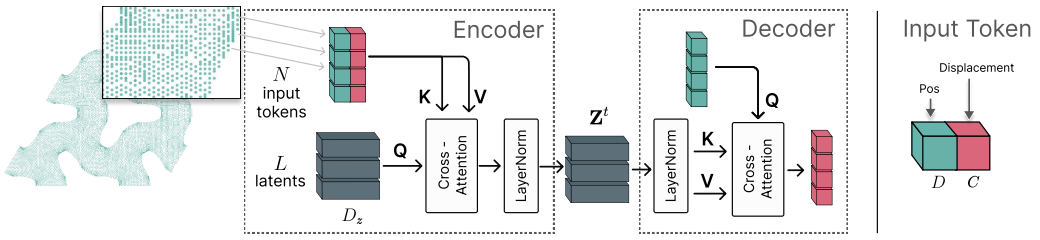


Figure 3: Encoder-decoder structure for point cloud datasets. Input tokens (reference positions $\mathbf{X}^1$ and displacements) are encoded into latent representation by cross-attending to L learned latent query tokens. The decoder reconstructs the displacements conditioned on the reference positions.

Autoencoder. We adopt a Perceiver-style cross-attention architecture (Jaegle et al., 2021; Alkin et al., 2024; Sestak et al., 2025) for all point cloud datasets. For each timestep t, the encoder maps the reference configuration $\mathbf{X}^1$ and response field $\mathbf{M}^t$ to L latent tokens of dimension D_z, i.e., $\mathbf{Z}^t \in \mathbb{R}^{L \times D_z}$, using cross-attention with learned latent queries, followed by self-attention on the latent set. Importantly, the decoder reconstructs $\mathbf{M}^t$ by querying these tokens at the fixed *reference* positions $\mathbf{X}^1$. Thus, the latent representation $\mathbf{Z}^t$ encodes all timestep-dependent states, while spatial reconstruction is always anchored to the same reference geometry $\mathbf{X}^1$. This enables the independent analysis and prediction of latent trajectories $\{\mathbf{Z}^t\}_{t=1}^T$, which can then be decoded at the reference positions to recover $\mathbf{M}^t$. For regular grid datasets, we use a 3D Vision Transformer (ViT) (Dosovitskiy et al., 2021; Prabhakar et al., 2024), detailed in the respective section (Section 3.3).

Latent flow matching. For the approximator, we draw inspiration from the flow-based model framework in Sestak et al. (2025). Given T_{obs} known states $\mathbf{Z}^{1:T_{\text{obs}}}$ and the condition $\mathbf{c}$, it generates the remaining states $\mathbf{Z}^{T_{\text{obs}}+1:T}$. The flow-based model then learns a path between noise and data. Let $\mathbf{Z}_1 := \mathbf{Z}^{1:T} \in \mathbb{R}^{T\times L\times D_z}$ be the true latent trajectory and $\boldsymbol{\xi} \sim \mathcal{N}(0, I)$ be noise. For every $\tau \in [0, 1]$, a probability path is specified by

$$\mathbf{Z}_\tau = \tau\, \mathbf{Z}_1 + (1-\tau)\, \boldsymbol{\xi}, \tag{5}$$

such that pure noise is obtained with $\tau = 0$, and the true trajectory at $\tau = 1$. We train a network with parameters θ_{flow} to reverse the applied corruption: given the noised trajectory, the flow time τ, observed states and the control protocol, the clean trajectory is predicted by

$$f_{\theta_{\text{flow}}} : (\mathbf{Z}_\tau, \tau, \mathbf{Z}^{1:T_{\text{obs}}}, \mathbf{c}) \mapsto \widehat{\mathbf{Z}}_1 \in \mathbb{R}^{T\times L\times D_z}. \tag{6}$$

We adapt the standard flow matching approach through optimal transport-based coupling of in-batch pairs to minimize the path length between the noise and data distributions (Tong et al., 2024). Additionally, we require the model to predict the bifurcation point, represented by a binary mask $\widehat{\mathbf{b}} \in \{0, 1\}^T$. At inference, $\widehat{\mathbf{Z}}_1$ is recovered by solving the probability-flow ODE associated with Eq. 5, starting from pure noise.

All timestep-dependent information for the prediction is expressed by $\mathbf{Z}_\tau$ and its conditioning signals. The formulation is model agnostic to the specific network architecture and training procedure used for $f_{\theta_{\text{flow}}}$ (see Appendix D.1 for details).

Inference using particle guidance. Due to the *multimodal* nature of the problem, the control protocol and initial state do not determine which mode the system settles into, e.g., the orbit coordinate (e.g. the buckling direction ϕ) is left free by $\mathbf{c}$. While the generative model predicts a single valid mode rather than averaging across them, sampling alone does not control the distribution over modes.

We address this at inference with an approach inspired by particle guidance (Corso et al., 2024). Rather than drawing K trajectories independently, we sample K candidate trajectories $\mathbf{Z}_\tau^{(1)}, \ldots, \mathbf{Z}_\tau^{(K)}$ in a shared batch and integrate the probability-flow ODE for all of them jointly, leaving $f_{\theta_{\text{flow}}}$ unmodified from training. To maintain diversity, we add an inter-sample repulsion term to the velocity field at each step (Figure 4).

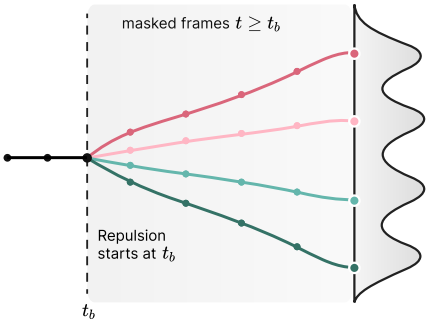


Figure 4: Particle guidance during sampling. Trajectories evolve identically up to bifurcation point t_b. From t_b, the repulsion term acts on the masked frames $t \geq t_b$, pushing the K trajectories apart, yielding a multimodal distribution.

Each $\widehat{\mathbf{Z}}_1^{(k)}$ and its ordinary velocity field are computed independently via Eq. 6, as at training time. We additionally compute a repulsive term restricted to bifurcation-relevant frames, elected by the shared predicted mask $\widehat{\mathbf{b}} \in \{0, 1\}^T$, where $\widehat{b}^t = 1$ indicates a predicted post-bifurcation frame. For each trajectory, the predicted values at these frames are masked and flattened into a single point; pairwise distances between trajectories' points are weighted by a Gaussian RBF kernel with bandwidth set to the median pairwise distance. Each trajectory's displacement $\text{force}^{(k)}$ is the resulting kernel-weighted average of unit vectors pointing away from its neighbors, so trajectories that are similar at the bifurcation frames are pushed apart while already-distinct trajectories barely interact. The repulsive displacement modifies the velocity field $\widehat{v}_{\theta_{\text{flow}}} = (\widehat{\mathbf{Z}}_1 - \mathbf{Z}_\tau)/(1-\tau)$ of the probability-flow ODE for sample k. Schematically,

$$\widehat{v}^{(k)} = w_{\text{vel}}\, \widehat{v}_{\theta_{\text{flow}}}\big(\mathbf{Z}_\tau^{(k)}, \tau\big) + w_{\text{rep}}\, \widehat{\mathbf{b}} \odot \text{force}^{(k)}, \tag{7}$$

where w_{vel} weights the model velocity, w_{rep} controls repulsion strength, and $\widehat{\mathbf{b}}$ selects the predicted post-bifurcation frames. The mask is broadcast over latent tokens and channels. This simplified expression omits the time-dependent scaling and normalization which we describe in Appendix D.5.

As a result, all K trajectories are integrated jointly rather than independently: at likely branch points, samples that would otherwise collapse onto the same mode are pushed apart, so the sample set spreads across distinct branches instead of concentrating on one.

2.4 METRICS

Evaluating generative models of bifurcating dynamical systems requires metrics that jointly capture trajectory-level precision and the distributional fidelity of predicted mode probabilities. Standard metrics such as MSE are inadequate in this setting, as they neither reflect the multimodal structure of the output space nor penalize mode collapse. We adopt two complementary metrics, adapted to systems with $|\mathcal{M}_{\mathbf{c}}| \in \{1, 2, 4\}$ or $|\mathcal{M}_{\mathbf{c}}| = \infty$ responses depending on the condition $\mathbf{c}$.

Mode Conditioned MAE (MCon-MAE) quantifies trajectory-level precision. For each condition $\mathbf{c}$, we draw K samples from the model and assign each to its nearest ground-truth mode by Euclidean distance. MCon-MAE is the resulting reconstruction error, averaged over samples and conditions:

$$\text{MCon-MAE} = \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c} \in \mathcal{C}} \frac{1}{K} \sum_{k=1}^{K} \text{mean}\left(\left| \widehat{\mathbf{X}}_{\mathbf{c}}^{(k)} - \mathbf{X}_{\mathbf{c}, m_\star^{(k)}} \right| \right) \tag{8}$$

with mode assignment $m_\star^{(k)} = \arg\min_{m \in \mathcal{M}_{\mathbf{c}}} \left\| \widehat{\mathbf{X}}_{\mathbf{c}}^{(k)} - \mathbf{X}_{\mathbf{c},m} \right\|_F^2$, where $\mathbf{X}_{\mathbf{c},m}$ highlights the dependence on the condition $\mathbf{c}$ and mode m. We further report the *Jensen-Shannon divergence (JSD)* between the predicted and ground-truth mode distributions, estimated from K samples drawn at inference for each condition. This measures the extent to which a model represents each outcome with the correct relative frequency, rather than favoring a subset of modes.

3 EXPERIMENTS

3.1 BEAM3D

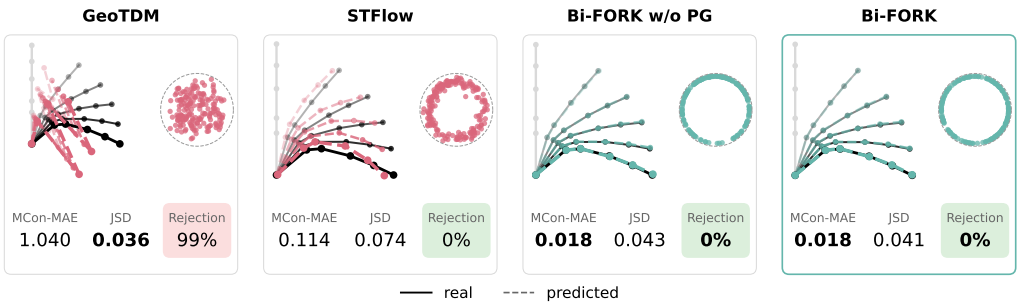


Figure 5: Beam3D result comparison. Each panel shows an evaluated method, with an illustrative sample (colored) plotted over the actual trajectory (black). Their endpoints are shown against the full population and the ideal uniform dashed circle. Metrics are aggregated over all Beam3D test conditions using five random seeds.

Bifurcating System and Learning Task. The 3D buckling beam dataset represents multiple slender beams with a geometry $\mathbf{X}^1$, which varies between data points, under axial compression. Each beam is fixed at one end with an incrementally increasing downward displacement d at the other. The input condition is therefore $\mathbf{c} = (\mathbf{X}^1, \ell)$. Below a critical threshold, the beam stays in axial compression. Beyond the threshold, the rotational symmetry breaks: the axisymmetric state remains an equilibrium but is no longer stable, and the beam buckles in an azimuthal direction $\phi \in [0, 2\pi]$, yielding a continuous $SO(2)$ orbit of solutions. Given the condition $\mathbf{c}$, the model is required to predict a representative set of K trajectories $\{\widehat{\mathbf{X}}^{(k)}\}_{k=1}^{K}$, sampled from this continuous orbit.

Modeling. To model the bifurcating system, we use the architecture described in Section 2.3. In addition to tip displacement, we input per-node rotational stiffness and element-level attributes, including stiffness and length. At inference, we use particle guidance to distribute the samples more evenly over the admissible post-buckling manifold $\mathcal{M}_{\mathbf{c}}$. We additionally evaluate Bi-FORK without particle guidance; we denote this variant Bi-FORK w/o PG. We sample 180 times to estimate the distribution accuracy. Details on the chosen architecture parameters, such as latent dimension or hidden size, are provided in Appendix D.

Metrics. In addition to the metrics described in Section 2.4, we report the rejection rate (%): the fraction of generated trials whose mean Euclidean displacement from their matched mode exceeds a threshold. MCon-MAE and JSD are computed over all generated trials, including those that would have been rejected by the threshold.

Results. Bifurcating dynamical systems are naturally formulated as interacting particle systems, motivating models that operate on sets of arbitrary size without retraining. We compare against two such baselines: GeoTDM (Han et al., 2024) and STFlow (Bennema ten Brinke et al., 2026). Figure 5 reports the MCon-MAE, JSD and rejection rate on the held-out test dataset. Among the models, GeoTDM achieves the lowest JSD due to its built-in equivariance, but rejects 99% of its samples, with the worst MCon-MAE. STFlow reduces the trajectory rejection rate while maintaining low error and JSD values. Notably, Bi-FORK substantially improves MCon-MAE with a 0% trajectory rejection rate and achieves a highly competitive JSD, actually closing the gap with the equivariant GeoTDM model. Visual inspection of the end-point distributions (Figure 5) confirms the quantitative findings: GeoTDM's samples cluster off the target circle manifold, and STFlow samples are scattered noisily around it. Both Bi-FORK w/o PG and Bi-FORK trace the target manifold accurately. Table 11 summarizes performance across five random seeds, reporting the mean and standard deviation.

We also compare inference runtime, shown in Table 15. Bi-FORK offers significant runtime advantages over graph-based baselines. It generates 180 samples for a single condition in 3.95 seconds, making it approximately 18× faster than STFlow and 1,966× faster than GeoTDM.

3.2 Mechanical Metamaterials

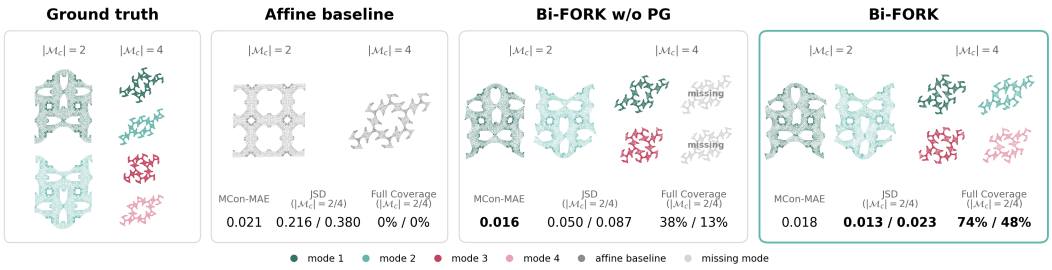


Figure 6: Mechanical metamaterials result comparison. Ground truth modes and generated samples are shown for representative conditions with $|\mathcal{M}_{\mathbf{c}}| = 2$ and $|\mathcal{M}_{\mathbf{c}}| = 4$ buckling modes. Colors indicate the mapped ground-truth mode; missing modes are shown in gray. Affine baseline results in mean prediction. Reported MCon-MAE is averaged over the full test set, while JSD and full coverage are reported for $|\mathcal{M}_{\mathbf{c}}| \in \{2, 4\}$. Metrics reported over five random seeds on the test dataset.

Bifurcating System and Learning Task. The dataset of Hendriks et al. (2025a) contains buckling trajectories of periodic porous metamaterials spanning all 17 wallpaper groups. Depending on geometry and loading, symmetry breaking produces 1, 2, or 4 translation-related solutions, i.e., $|\mathcal{M}_{\mathbf{c}}| \in \{1, 2, 4\}$: one mode when no observable symmetry breaking occurs, and two or four modes after isotropy reduction, with four modes potentially arising after a secondary bifurcation.

The condition is therefore $\mathbf{c} = (\mathbf{X}^1, \boldsymbol{\ell})$ with $\boldsymbol{\ell}^t = \mathbf{F}^t$, with $\mathbf{F}^t$ the applied macroscopic deformation gradient at time t. Since $\mathbf{F}^t$ is known, the predicted $\{\widehat{\mathbf{M}}^t\}_{t=1}^T$ are the microfluctuation fields instead of the full deformation field. The deformed coordinates are then reconstructed as $\widehat{\mathbf{X}}^t = \mathbf{X}^1(\mathbf{F}^t)^\top + \widehat{\mathbf{M}}^t$. Unlike the beam, the mode distribution here is discrete: the model must predict the correct number of modes for a given condition, because it should generate each physically plausible branch in $\mathcal{M}_{\mathbf{c}}$.

Modeling. Following the architecture in Section 2.3, we use the first frame as the conditioning frame for each geometry. Additionally, we add node type information and the crystallographic group. Inference is executed similarly to the Beam3D dataset, with one difference for Bi-FORK: when no bifurcation time point is detected ($|\mathcal{M}_{\mathbf{c}}| = 1$), the sampling algorithm discards the repulsive term in the diffusion process. In this setting, we sample 16 times to estimate the potential distribution for each condition.

Metrics. We employ the metrics from Section 2.4. Additionally, we report coverage (%), defined as the percentage of conditions for which every ground-truth mode is represented by a valid generated sample. We report JSD and coverage only for cases with $|\mathcal{M}_{\mathbf{c}}| \in \{2, 4\}$. MCon-MAE and JSD are computed over all generated trials, including those that would have been rejected by the threshold.

Results. GeoTDM and STFlow were not viable baselines for this problem setting because both rely on a fully connected graph, which is infeasible with thousands of nodes. On Beam3D ($\leqslant$ 11 nodes for $T = 200$), both models showed no improvement and, being GNN-based, were already slow to train and sample from, requiring up to three orders of magnitude more inference time than Bi-FORK (Table 15). With roughly $160\times$ more nodes per timestep and all timesteps processed jointly (Section 1), both models become computationally infeasible. We therefore compare against the affine baseline, which propagates $\mathbf{X}^1$ using only the deformation gradient. Figure 6 shows that Bi-FORK w/o PG already improves over the affine baseline in accuracy and in distributional coverage: the per-mode JSD shows that it recovers multiple modes rather than collapsing to a mean prediction. With the full potential of our method, we further improve coverage, retrieving all four modes in roughly 50% of the sampled test cases. Table 12 provides the quantitative summary, including standard deviation. Table 2 shows the diversity improvement for the plotted samples with two and four modes, producing a nearly uniform distribution with Bi-FORK.

Table 2: Mode distributions of the two illustrated geometries in Figure 6 with $|\mathcal{M}_{\mathbf{c}}| \in \{2, 4\}$. Each row aggregates five seeds, with $K = 16$ samples per seed.

Sample	Method	K	Mode 1	Mode 2	Mode 3	Mode 4
2 modes	Bi-FORK w/o PG	16	80.0%	20.0%	–	–
	Bi-FORK	16	56.2%	43.8%	–	–
4 modes	Bi-FORK w/o PG	16	18.2%	0.0%	72.7%	9.1%
	Bi-FORK	16	23.2%	15.4%	36.3%	25.1%

Finally, we ablate the bifurcation head on Beam3D and mechanical metamaterials. The results in Table 14 show that the learned bifurcation masks are predicted accurately on both datasets.

3.3 Allen-Cahn

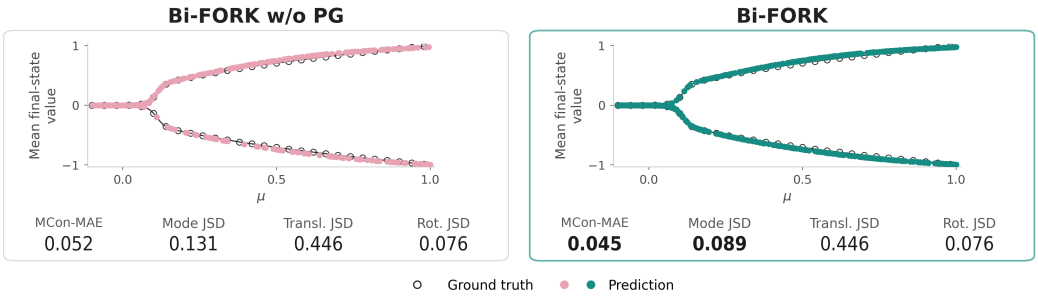


Figure 7: Allen–Cahn result comparison. Ground truth (black open circles) and predictions (teal and pink circles) are shown across a parameter sweep $\mu \in [-0.1, 1.0]$ at fixed $\epsilon = 0.1$. The vertical axis shows the final-frame mean value. We evaluate metrics on the test dataset and average over five random seeds; lower values indicate better performance.

Bifurcating System and Learning Task. The 3D Allen-Cahn dataset describes the evolution of a phase-field variable governed by the Allen-Cahn equation, which models phase separation in binary mixtures on a periodic domain. Because all trajectories share the same initial homogeneous state, the condition is only $\mathbf{c} = (\epsilon, \mu)$, which is fixed over time. This initial state is invariant to $G_{\mathbf{c}}$, which contains phase exchange $u \mapsto -u$, periodic translations and the cubic point group. The initial state can evolve into many patterns, which can be symmetry-related through $G_{\mathbf{c}}$ or morphologically distinct. The number of morphologically distinct modes rises as ϵ^{-3}: shrinking ϵ opens more unstable directions.

Modeling. Given Allen-Cahn's grid structure, the first stage uses a ViT instead of the Perceiver-style encoder. The periodic scalar field is compressed into a compact spatial-latent-token grid. Implementation details are in Appendix D. The second stage follows Section 2.3. As input conditions, we only use the physical parameters ϵ and μ; since the full trajectory is generated starting at $t = 1$, no additional conditioning frame is needed. The output is the full scalar-field rollout. We apply symmetry-

aware augmentations (periodic translations and sign flips) that match the domain's periodicity and point-group structure. At inference, the sampling follows the previously described pipeline, with one exception: we apply particle guidance at every valid frame by setting $\widehat{b}^t = 1$ throughout the sampling process. Therefore, we do not train a bifurcation classification head for this problem setting.

Metrics. We extend the metrics described in Section 2.4 to the Allen-Cahn solution space. A raw distance between a generated field and any single reference is uninformative when the reference could equally be shifted, reflected, rotated, or sign-inverted and still represent the same solution. We therefore first match each generated trial to the closest branch among a predefined set of ground-truth solutions using an FFT-based descriptor invariant to these symmetries, then align it to that branch via an exhaustive search over the point group. Full implementation details are given in Appendix D.8.

Estimating the JSD on the joint distribution over (branch, rotoreflection, shift) is infeasible, as the number of possible combinations exceeds the number of available samples. We therefore split the JSD metrics into three separate components: **mode-coverage JSD** over the predefined solutions (including their sign-inverted counterparts); a **translation JSD** that similarly fixes a branch and tests whether the recovered periodic shift is uniform; and a **rotoreflection JSD** that fixes a single matched branch and tests whether the recovered rotations and reflections relative to it are uniform. MCon-MAE compares the full predicted trajectory with its relaxed counterpart, obtained by a Gauss–Newton projection onto the Allen–Cahn PDE. It thus measures the distance to a physically valid solution rather than to precomputed references.

Results. Figure 7 summarizes the results, and Table 13 reports the mean and standard deviation over five seeds. Bi-FORK reduces the MCon-MAE from 0.052 (without particle guidance) to 0.045, showing that the generated trajectories require smaller corrections to satisfy the PDE. It also lowers the mode JSD from 0.131 to 0.089, reflecting improved coverage of the solution branches. The translation and rotoreflection JSDs are identical for both variants. To illustrate mode coverage, we fix $\epsilon = 0.1$ and vary only μ. Without particle guidance, the model samples mostly from the positive branch and misses several solutions on the negative branch. Bi-FORK samples both branches with similar density over the full range of μ. Figure 8 shows samples from both methods next to the ground truth. Even when the samples differ, they showcase valid solutions.

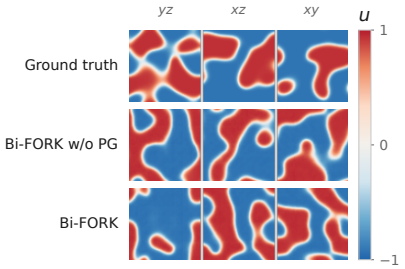


Figure 8: Visual comparison of ground truth vs predicted samples for $\mu = 0.86$ and $\epsilon = 0.094$.

4 Discussion

Limitations. Our experiments show that Bi-FORK applies to a diverse set of bifurcating systems; however, some limitations remain. First, particle guidance can trade accuracy for diversity: the repulsion acts on all samples, including those already in the same valid mode, and can push them off the solution manifold. Second, the Allen–Cahn evaluation is necessarily incomplete. The full distribution over morphologies is unavailable, so the mode-coverage JSD relies on four simulated realizations per condition and their sign flips. In addition, separate JSDs for modes, translations, and rotoreflections do not guarantee coverage of the joint distribution. Third, although symmetry is central to the problem, the model learns it from data and augmentation rather than enforcing it architecturally.

Conclusion. We introduce Bi-FORK, a framework for learning symmetry-breaking bifurcations. It maps solution multiplicity via symmetry orbits and bifurcation time. The task is framed as learning a conditional distribution over complete trajectories using latent flow matching and particle guidance. We evaluate models using symmetry-aware metrics that detect mode collapse. Across buckling beams, mechanical metamaterials, and Allen-Cahn phase separation, Bi-FORK recovers multimodal solutions at discretizations up to 260,000 points, several orders of magnitude beyond prior work. Because no existing stochastic method scales to this size, we evaluate the largest systems against a deterministic affine baseline and our own ablations. Ultimately, this opens generative modeling to large-scale bifurcating physical systems.

AI USE STATEMENT

In this work, we used generative AI tools for three tasks requiring disclosure: implementing methods, cleaning and reformatting datasets, and formulating mathematical claims, where AI served only as a source of inspiration. We used Codex to implement the baseline methods and supporting evaluation code in Section 3; two authors reviewed, verified, and tested this code, and we implemented our proposed method ourselves. We used Codex and Claude to clean and reformat the datasets and checked the processed data against the original sources. Codex and Claude also offered ideas for the mathematical claims in Section 2, but the final claims were formulated and verified by the authors. We have not used generative AI tools to generate synthetic datasets; to develop the problem setting, problem framing, or proposed architecture; to propose or refine hypotheses; to design or give feedback on experiments; or to interpret results. Translation and qualitative or thematic data analysis are not applicable to this work.

Additionally, we used generative AI tools to identify relevant literature, to polish the manuscript for readability, and to make cosmetic improvements to figures. We did not use them for other recommended-disclosure tasks such as drafting text, suggesting hyperparameters, identifying research gaps, formatting references, or proposing a title. We have reviewed all AI-assisted work: suggested papers were read by the authors before being cited, text edits were checked to ensure that the technical content and claims were unchanged, and figure edits were limited to styling, with the underlying numbers never modified by AI. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide a detailed description of the datasets, including the evaluation splits used, in Appendix C. Appendix D also elaborates on the model architecture, training pipeline, loss functions, and hyperparameters for each dataset. Additionally we provide a detailed overview on the evaluation pipeline for each dataset in Appendix D.7 for Beam3D and mechanical metamaterials, and in Appendix D.8 for the Allen-Cahn dataset. We will provide the code of the underlying experiments upon publication of the paper.

REFERENCES

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

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

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. *Advances in Neural Information Processing Systems*, 37:25152–25194, 2024.

Benedikt Alkin, Maurits Bleeker, Richard Kurle, Tobias Kronlachner, Reinhard Sonnleitner, Matthias Dorfer, and Johannes Brandstetter. Ab-upt: Scaling neural cfd surrogates for high-fidelity automotive aerodynamics simulations via anchored-branched universal physics transformers. *Transactions on Machine Learning Research*, 2025.

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *International Conference on Learning Representations*, 2021.

Stuart S. Antman. *Nonlinear Problems of Elasticity*, volume 107 of *Applied Mathematical Sciences*. Springer, New York, 2 edition, 2005.

Zdeněk P Bažant and Luigi Cedolin. *Stability of structures: elastic, inelastic, fracture, and damage theories*. Courier Corporation, 2003.

Kiet Bennema ten Brinke, Koen Minartz, and Vlado Menkovski. Stflow: Data-coupled flow matching for geometric trajectory simulation. In *International conference on machine learning*. PMLR, 2026.

Sebastian Bischoff, Alana Darcher, Michael Deistler, Richard Gao, Franziska Gerken, Manuel Gloeckler, Lisa Haxel, Jaivardhan Kapoor, Janne K Lappalainen, Jakob H Macke, et al. A practical guide to sample-based statistical distances for evaluating generative models in science. *Transactions on Machine Learning Research*, 2024.

Kenneth M Brow and William B Gearhart. Deflation techniques for the calculation of further solutions of a nonlinear system. *Numerische Mathematik*, 16(4):334–342, 1971.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

Giovanni Calzolari and Wei Liu. Accelerating large eddy simulations of urban airflow with generative adversarial networks. *Building and Environment*, 286:113622, 2025.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3438–3445, 2020.

Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle guidance: non-iid diverse sampling with diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 22480–22507, 2024.

Michael G. Crandall and Paul H. Rabinowitz. Bifurcation from simple eigenvalues. *Journal of Functional Analysis*, 8(2):321–340, 1971.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021.

Gianluca Fabiani, Francesco Calabrò, Lucia Russo, and Constantinos Siettos. Numerical solution and bifurcation analysis of nonlinear partial differential equations with extreme learning machines. *Journal of Scientific Computing*, 89(2):44, 2021.

Patrick E Farrell, A Birkisson, and Simon W Funke. Deflation techniques for finding distinct solutions of nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 37(4): A2026–A2045, 2015.

Zhihan Gao, Xingjian Shi, Boran Han, Hao Wang, Xiaoyong Jin, Danielle Maddix, Yi Zhu, Mu Li, and Yuyang Bernie Wang. Prediff: Precipitation nowcasting with latent diffusion models. *Advances in Neural Information Processing Systems*, 36:78621–78656, 2023.

Matteo Giacomini and Pedro Díez. Surrogates for physics-based and data-driven modelling of parametric systems: Review and new perspectives. *Archives of Computational Methods in Engineering*, 33:8519–8551, 2026.

Martin Golubitsky and Ian Stewart. *The Symmetry Perspective: From Equilibrium to Chaos in Phase Space and Physical Space*, volume 200 of *Progress in Mathematics*. Birkhäuser, Basel, 2002.

Martin Golubitsky, Ian Stewart, and David G. Schaeffer. *Singularities and Groups in Bifurcation Theory, Volume II*, volume 69 of *Applied Mathematical Sciences*. Springer, New York, 1988.

Peter Grönquist, Chengyuan Yao, Tal Ben-Nun, Nikoli Dryden, Peter Dueben, Shigang Li, and Torsten Hoefler. Deep learning for post-processing ensemble weather forecasts. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194): 20200092, 2021.

John Guckenheimer and Philip Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences*. Springer, New York, 1983.

Leonardo Ferreira Guilhoto, Akshat Kaushal, and Paris Perdikaris. Multimodal scientific learning beyond diffusions and flows. *arXiv preprint arXiv:2602.00960*, 2026.

Thomas M Hamill. Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, 129(3):550–560, 2001.

Jiaqi Han, Minkai Xu, Aaron Lou, Haotian Ye, and Stefano Ermon. Geometric trajectory diffusion models. *Advances in Neural Information Processing Systems*, 37:25628–25662, 2024.

Shuangshuang He, Hongli Liang, Yuanting Zhang, and Xingyuan Yuan. Skillful high-resolution ensemble precipitation forecasting with an integrated deep learning framework. *arXiv preprint arXiv:2501.02905*, 2025.

Fleur Hendriks, Vlado Menkovski, Martin Doškář, Marc GD Geers, Kevin Verbeek, and Ondřej Rokoš. Wallpaper group-based mechanical metamaterials: Dataset including mechanical responses. *Scientific Data*, 12:1880, 2025a.

Fleur Hendriks, Ondřej Rokoš, Martin Doškář, Marc GD Geers, and Vlado Menkovski. Equivariant flow matching for symmetry-breaking bifurcation problems. *arXiv preprint arXiv:2509.03340*, 2025b.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pp. 4651–4664. PMLR, 2021.

Hansjörg Kielhöfer. *Bifurcation Theory: An Introduction with Applications to Partial Differential Equations*, volume 156 of *Applied Mathematical Sciences*. Springer, New York, 2 edition, 2012.

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *Advances in Neural Information Processing Systems*, 36:59886–59910, 2023.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, et al. Neural general circulation models for weather and climate. *Nature*, 632(8027):1060–1066, 2024.

Warner T. Koiter. *On the Stability of Elastic Equilibrium*. PhD thesis, Delft University of Technology, 1945. English translation NASA TT F-10833 (1967) / AFFDL-TR-70-25 (1970).

Yuri A. Kuznetsov. *Elements of Applied Bifurcation Theory*, volume 112 of *Applied Mathematical Sciences*. Springer, New York, 3 edition, 2004.

Erik Larsson, Joel Oskarsson, Tomas Landelius, and Fredrik Lindsten. Crps-lam: Probabilistic regional weather forecasting with continuous ranked probability score. *arXiv preprint arXiv:2510.09484*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Ankur Mahesh, William Collins, Boris Bonev, Noah Brenowitz, Yair Cohen, Joshua Elms, Peter Harrington, Karthik Kashinath, Thorsten Kurth, Joshua North, et al. Huge ensembles part i: Design of ensemble weather forecasts using spherical fourier neural operators. *Geoscientific Model Development*, 18:5575–5603, 2025.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *International Conference on Learning Representations*, 2020.

Federico Pichi, Francesco Ballarin, Gianluigi Rozza, and Jan S Hesthaven. An artificial neural network approach to bifurcating phenomena in computational fluid dynamics. *Computers & Fluids*, 254:105813, 2023.

Chinmay Prabhakar, Hongwei Li, Jiancheng Yang, Suprosanna Shit, Benedikt Wiestler, and Bjoern Menze. Vit-ae++: improving vision transformer autoencoder for self-supervised medical image representations. In *Medical Imaging with Deep Learning*, pp. 666–679. PMLR, 2024.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, 2025.

Max Rietkerk, Robbin Bastiaansen, Swarnendu Banerjee, Johan van de Koppel, Mara Baudena, and Arjen Doelman. Evasion of tipping in complex systems through spatial pattern formation. *Science*, 374(6564):eabj0359, 2021.

Ignacio Romero and Michael Ortiz. A note on data-driven methods for mechanical problems with non-unique solutions. *Meccanica*, 61(1):26, 2026.

David Ruelle and Floris Takens. On the nature of turbulence. *Les rencontres physiciens-mathématiciens de Strasbourg-RCP25*, 12:1–44, 1971.

Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *2017 IEEE international conference on computer vision (ICCV)*, pp. 3611–3620. IEEE, 2017.

Florian Sestak, Artur P. Toshev, Andreas Fürst, Günter Klambauer, Andreas Mayr, and Johannes Brandstetter. Lam-SLide: Latent space modeling of spatial dynamical systems via linked entities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Rüdiger Seydel. *Practical Bifurcation and Stability Analysis*, volume 5 of *Interdisciplinary Applied Mathematics*. Springer, New York, 3 edition, 2010.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *International Conference on Learning Representations*, 2021.

Will Steffen, Johan Rockström, Katherine Richardson, Timothy M Lenton, Carl Folke, Diana Liverman, Colin P Summerhayes, Anthony D Barnosky, Sarah E Cornell, Michel Crucifix, et al. Trajectories of the earth system in the anthropocene. *Proceedings of the national academy of sciences*, 115(33):8252–8259, 2018.

Haocheng Tang, Liang Shi, Ya-Shi Zhang, Xixian Liu, Jian Tang, and Jiarui Lu. Learning structure, energy, and dynamics: A survey of artificial intelligence for protein dynamics. *arXiv preprint arXiv:2604.25244*, 2026.

J. M. T. Thompson and G. W. Hunt. *A General Theory of Elastic Stability*. Wiley, London, 1973.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Thomas Unterthiner, Bernhard Nessler, Calvin Seward, Günter Klambauer, Martin Heusel, Hubert Ramsauer, and Sepp Hochreiter. Coulomb gans: Provably optimal nash equilibria via potential fields. *International Conference on Learning Representations*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Donggeun Yoon, Minseok Seo, Doyi Kim, Yeji Choi, and Donghyeon Cho. Probabilistic weather forecasting with deterministic guidance-based diffusion model. pp. 108–124, 2025.

Zongren Zou, Zhicheng Wang, and George Em Karniadakis. Learning and discovering multiple solutions using physics-informed neural networks with random initialization and deep ensemble. *Proceedings of the Royal Society A*, 481(2325):20250205, 2025.

APPENDIX CONTENTS

A Notation **15**

B Additional bifurcation context **16**

C Datasets **16**

C.1 Beam3D 16

C.2 Mechanical Metamaterials 17

C.3 Allen-Cahn Dataset 19

D Experimental Details **20**

D.1 Architecture 20

D.2 Implementation Details 21

D.3 Training 21

D.4 Loss Function 22

D.5 Adaptations to particle guidance 22

D.6 Hyperparameters 23

D.7 Evaluation Details: Beam3D and Metamaterials 23

D.8 Evaluation Details: Allen-Cahn 24

E Results **25**

E.1 Quantitative results 25

E.2 Ablations 26

E.3 Qualitative Results 27

A NOTATION

The notation used throughout the paper is summarized in Table 3.

Table 3: Overview of used symbols and notations. Superscripts t index trajectory frames, subscripts τ flow time, and parenthesized superscripts (k) generated samples. Dataset-specific quantities are defined where they are used (Sections 3 and C).

Symbol	Meaning	Space
Data and trajectories		
N, n	number of spatial locations (nodes); node index	$\mathbb{N}$
T, t	number of timesteps; timestep index	$\mathbb{N}$
D, C	spatial dimension; channels of the response	$\{2,3\}$, $\{1,D\}$
$\mathbf{X}^1$, $\mathbf{X}^t$	reference geometry; deformed coordinates at timestep t	$\mathbb{R}^{N\times D}$
$\mathbf{M}^t$	response (signal field) at timestep t	$\mathbb{R}^{N\times C}$
$\mathbf{M}^{1:T}$	response trajectory $(\mathbf{M}^1,\dots,\mathbf{M}^T)$	$\mathbb{R}^{T\times N\times C}$
Conditions, modes, and symmetry		
$\mathbf{c}$, $\mathcal{C}$	condition $(\mathbf{X}^1,\boldsymbol{\ell})$, or (ϵ,μ) for Allen-Cahn; set of conditions	
$\boldsymbol{\ell}$	control protocol (loading schedule) $(\boldsymbol{\ell}^1,\dots,\boldsymbol{\ell}^T)$	
$\mathcal{M}_{\mathbf{c}}$, $\mathcal{M}_{\mathbf{c}}^t$	mode set (admissible trajectories); responses at timestep t	$\subset \mathbb{R}^{T\times N\times C}$
$G_{\mathbf{c}}$	symmetry group of the condition	
$\mathbf{M}_{\mathbf{c}}^t$	representative response at timestep t	$\in \mathcal{M}_{\mathbf{c}}^t$
$H_{\mathbf{c}}^t$	isotropy subgroup of $\mathbf{M}_{\mathbf{c}}^t$	$\leq G_{\mathbf{c}}$
t_b	bifurcation time: first strict decrease of $H_{\mathbf{c}}^t$ (∞ if none)	$\{2,\dots,T\}\cup\{\infty\}$
Distributions		
$p(\mathbf{M}^{1:T}\mid\mathbf{c})$	ground-truth distribution of trajectories, supported on $\mathcal{M}_{\mathbf{c}}$	
$p_{\theta_{\text{flow}}}(\mathbf{M}^{1:T}\mid\mathbf{c})$	learned distribution (latent flow model, decoded)	
Latent representation		
$\mathcal{E}_{\phi_{\text{enc}}}$, $\mathcal{D}_{\phi_{\text{dec}}}$	encoder $(\mathbf{X}^1,\mathbf{M}^t)\mapsto\mathbf{Z}^t$; decoder $(\mathbf{Z}^t,\mathbf{X}^1)\mapsto\widehat{\mathbf{M}}^t$	
$\mathbf{Z}^t$	latent state at timestep t	$\mathbb{R}^{L\times D_z}$
L, D_z	number of latent tokens; token dimension	$\mathbb{N}$
Latent flow matching		
τ	flow time (0: noise, 1: data)	$[0,1]$
$\mathbf{Z}_1$	clean latent trajectory $\mathbf{Z}^{1:T}$	$\mathbb{R}^{T\times L\times D_z}$
$\boldsymbol{\xi}$	Gaussian noise	$\mathcal{N}(0,I)$
$\mathbf{Z}_\tau$	noisy latent trajectory $\tau\mathbf{Z}_1+(1-\tau)\boldsymbol{\xi}$	$\mathbb{R}^{T\times L\times D_z}$
T_{obs}	number of observed (conditioning) frames	$\{0,\dots,T\}$
$f_{\theta_{\text{flow}}}$, θ_{flow}	predictor $(\mathbf{Z}_\tau,\tau,\mathbf{Z}^{1:T_{\text{obs}}},\mathbf{c})\mapsto\widehat{\mathbf{Z}}_1$; its parameters	
$\widehat{\mathbf{Z}}_1$	predicted clean latent trajectory	$\mathbb{R}^{T\times L\times D_z}$
$\widehat{v}_{\theta_{\text{flow}}}$	velocity field of the probability-flow ODE, induced by $\widehat{\mathbf{Z}}_1$	$\mathbb{R}^{T\times L\times D_z}$
Sampling and particle guidance		
K, k	number of jointly generated samples; sample index	$\mathbb{N}$
$\mathbf{Z}_\tau^{(k)}$	latent state of sample k	$\mathbb{R}^{T\times L\times D_z}$
p^t, $\widehat{\mathbf{b}}$	predicted post-bifurcation probability; mask $\widehat{b}^t=\mathbb{1}[p^t\geq 1/2]$	$[0,1]$, $\{0,1\}^T$
$\text{force}^{(k)}$	repulsive force on sample k	$\mathbb{R}^{T\times L\times D_z}$
w_{vel}, w_{rep}	weights of model velocity and repulsion	$\mathbb{R}_{\geq 0}$
$\widehat{v}^{(k)}$	guided velocity of sample k	$\mathbb{R}^{T\times L\times D_z}$
Training and evaluation		
$\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{flow}}$, $\mathcal{L}_{\text{bif}}$	reconstruction, latent-flow, and bifurcation losses	
b^t, r^t	bifurcation label $\mathbb{1}[t\geq t_b]$; valid-frame indicator	$\{0,1\}$
$\widehat{\mathbf{X}}_{\mathbf{c}}^{(k)}$, $\mathbf{X}_{\mathbf{c},m}$	generated trajectory k; ground-truth trajectory of mode m	
$m_\star^{(k)}$	ground-truth mode matched to sample k	
MCon-MAE, JSD	mode-conditioned MAE; Jensen–Shannon divergence	$\mathbb{R}_{\geq 0}$

B ADDITIONAL BIFURCATION CONTEXT

Estimating the bifurcation time. In the datasets, t_b is obtained from simulated trajectories rather than from the isotropy subgroups directly. For two trajectories i, j under the same condition, we compute the pairwise divergence $\mathrm{d}_{ij}^t = \|\mathbf{M}_i^t - \mathbf{M}_j^t\|$ (the load-prescribed part cancels) and take the first timestep at which it exceeds a small threshold $\vartheta > 0$. Before the bifurcation, the equilibrium is unique and $G_{\mathbf{c}}$-invariant, so all trajectories coincide; afterwards, trajectories on distinct branches separate. For the quasi-static datasets, this estimate therefore recovers t_b up to the resolution ϑ. For Allen-Cahn, where small noise perturbs the shared homogeneous initial state, it marks the onset of macroscopic domain formation.

Non-bifurcating conditions. If no timestep satisfies $H_{\mathbf{c}}^t \subsetneq H_{\mathbf{c}}^{t-1}$, the condition admits a single mode, and we set $t_b := \infty$. The bifurcation labels are then $b^t = \mathbb{1}[t \geq t_b] \equiv 0$, and the estimator above detects no separation. Accordingly, the bifurcation head is trained to predict $\widehat{\mathbf{b}} = \mathbf{0}$ for such conditions, which deactivates the repulsion term in Eq. 7.

Secondary bifurcations. A loading path can exhibit several isotropy drops $t_b^{(1)} < t_b^{(2)} < \cdots$, i.e., hierarchical bifurcations; the four-mode metamaterials arise from two such drops (Figure 10b). Throughout, t_b denotes the first one.

Choice of representative. The definition of t_b in Section 2.1 refers to a representative response $\mathbf{M}_{\mathbf{c}}^t$ at every timestep. Two responses on the same orbit have conjugate isotropy subgroups, $H_{g\cdot\mathbf{M}} = g\, H_{\mathbf{M}}\, g^{-1}$ for $g \in G_{\mathbf{c}}$, and conjugate subgroups are in general not nested. The inclusion $H_{\mathbf{c}}^t \subsetneq H_{\mathbf{c}}^{t-1}$ is therefore to be read along a single trajectory: $\mathbf{M}_{\mathbf{c}}^t$ denotes frame t of one fixed admissible trajectory $\mathbf{M}_{\mathbf{c}}^{1:T} \in \mathcal{M}_{\mathbf{c}}$. With this convention, t_b does not depend on the trajectory chosen. Replacing $\mathbf{M}_{\mathbf{c}}^{1:T}$ by $g \cdot \mathbf{M}_{\mathbf{c}}^{1:T} = (g \cdot \mathbf{M}_{\mathbf{c}}^1, \ldots, g \cdot \mathbf{M}_{\mathbf{c}}^T)$ conjugates every $H_{\mathbf{c}}^t$ by the same g; conjugation by a fixed element preserves inclusions and their strictness, so the first timestep at which $H_{\mathbf{c}}^t \subsetneq H_{\mathbf{c}}^{t-1}$ holds is unchanged. For Allen-Cahn, where $\mathcal{M}_{\mathbf{c}}$ is a union of orbits, this defines t_b per orbit; the estimator above applies unchanged.

Bifurcation, not chaos. Unlike chaotic systems, where nearby trajectories diverge exponentially and long-run outcomes are unpredictable (Ruelle & Takens, 1971; Guckenheimer & Holmes, 1983), multistable systems branch into a *small, structured* set of symmetry-related outcomes. This makes the problem well posed for learning: the target is a multimodal distribution over a few valid branches, not an unpredictable attractor.

Relation to bifurcation theory. For the mechanical datasets, t counts load increments, so the "bifurcation time" is really a bifurcation *load*: the classical setting of equilibrium bifurcation theory, where a solution branch splits once the equilibrium stops being unique—mechanically, where the tangent stiffness matrix becomes singular (Koiter, 1945; Thompson & Hunt, 1973; Antman, 2005; Crandall & Rabinowitz, 1971; Kielhöfer, 2012; Seydel, 2010). In symmetric problems, the new equilibria have strictly less symmetry and appear in symmetry-related copies, which is exactly the isotropy reduction defining t_b in Section 2.1 and the content of equivariant bifurcation theory (Golubitsky et al., 1988; Golubitsky & Stewart, 2002). Allen-Cahn is instead genuinely time-dependent and covered by the dynamical version of the theory (Guckenheimer & Holmes, 1983; Kuznetsov, 2004). We consider the generic supercritical scenario in which branches grow continuously out of the symmetric state; sudden jumps (subcritical or imperfect buckling) are out of scope.

C DATASETS

C.1 BEAM3D

The dataset describes a 3D beam-buckling process, containing simulations with 3 to 11 nodes. In the problem setting, the beam is fixed at the base and aligned along the z-axis in its undeformed state, with a specified downward displacement applied incrementally to the tip. This displacement sequence is the control protocol $\boldsymbol{\ell} = (\ell^1, \ldots, \ell^T)$ in the condition $\mathbf{c} = (\mathbf{X}^1, \boldsymbol{\ell})$. Below a critical displacement threshold, the beam undergoes pure axial compression; beyond it, the system loses stability and deflects in any direction ϕ drawn uniformly from $[0, 2\pi]$ in the x-y-plane. As every

deflection direction is equally probable, the system admits $|\mathcal{M}_{\mathbf{c}}| = \infty$ distinct post-buckling modes under identical loading.

Dataset generation. The beam contains N_{seg} compressible elements, each characterized by a reference length L_i^{beam}, axial stiffness k_i^{ax}, and rotational spring constant k_i^{rot} resisting bending between neighboring elements. The mechanical state is parametrized by axial strains $e_i = \ln(L_i^{\text{def}}/L_i^{\text{beam}})$, where L_i^{def} is the deformed element length, and q_i represents the bending angle from the vertical. The resulting node positions are expressed relative to the reference geometry as the displacement trajectory $\{\mathbf{M}^t\}_{t=1}^T$ used by the model. Equilibrium configurations minimize the strain energy

$$E_{\text{beam}}(\mathbf{q}, \mathbf{e}) = \frac{1}{2}\left[k_1^{\text{rot}} q_1^2 + \sum_{i=2}^{N_{\text{seg}}} k_i^{\text{rot}}(q_i - q_{i-1})^2\right] + \frac{1}{2}\sum_{i=1}^{N_{\text{seg}}} L_i^{\text{beam}} k_i^{\text{ax}} e_i^2 \tag{9}$$

subject to $d = \sum_{i=1}^{N_{\text{seg}}} L_i^{\text{beam}}[1 - \exp(e_i)\cos q_i]$, enforced via a Lagrange multiplier. The dataset contains 997 trajectories of up to $T = 200$ timesteps, simulated to full deflection d, with system parameters N_{seg}, L_i^{beam}, k_i^{rot}, k_i^{ax}, and ϕ sampled across instances. The sampling distributions are summarized in Table 4.

Table 4: Sampling distributions across used parameters.

Parameter	Range	Add. Information
N_{seg}	3–11 nodes (2–10 segments)	per trajectory
L_i^{beam}	log-uniform$[0.5, 2.0]$	per segment
k_i^{ax}	log-uniform$[0.5, 2.0]$	per segment
k_i^{rot}	log-uniform$[0.5, 2.0]$	per node
ϕ	uniform on $[0, 2\pi)$	per trajectory

The dataset is well-suited to our analysis since the buckling direction ϕ is continuous, yielding infinitely many symmetry-related modes under the same loading condition.

Examples. Figure 9 visualizes representative 3D beam trajectories for increasing node counts, showing how the initially vertical beam progressively buckles into different directions over time.

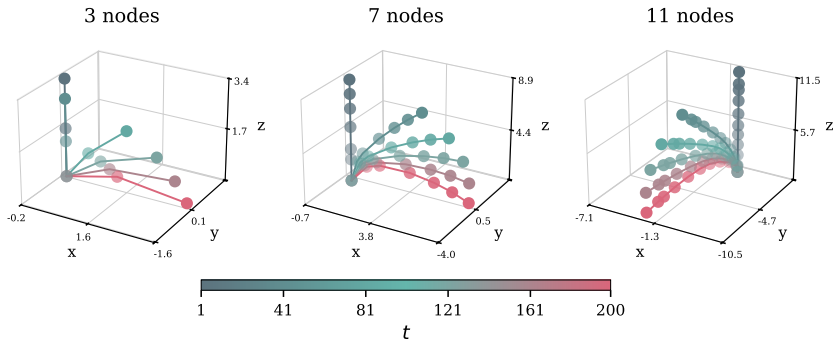


Figure 9: Example trajectories from the 3D beam dataset with different buckling directions ϕ for diverse node counts over timestep t.

Dataset Split. Table 5 illustrates the distribution across different data splits. We consider a single sample per condition in the table, whereas during training the full range of buckling directions, $\phi \in [0, 2\pi]$, is applied across the dataset.

C.2 Mechanical Metamaterials

For the second problem setting, we use the publicly available dataset of Hendriks et al. (2025a)[2]. It represents buckling trajectories of 2D porous elastic metamaterials, covering all 17 wallpaper groups.

[2] https://zenodo.org/records/18172947

Table 5: Summary of 3D beam dataset splits.

Split	Trajectories	Share (%)
Training	697	70
Validation	150	15
Test	150	15
Total	997	100

Each geometry is simulated on a 2×2 representative volume element under 12 loading paths, giving 12,240 trajectories from 1,020 geometries, with mesh sizes between 1,000 - 36,000 per timestep. In our notation, the prescribed loading path is the control protocol $\boldsymbol{\ell}$; for this dataset each increment is a known deformation gradient, $\boldsymbol{\ell}^t = \mathbf{F}^t$. The trajectories contain up to $T = 32$ timesteps, but can terminate earlier if the system runs into contact and reaches a steady state.

For the underlying dataset the deformed configuration at timestep t is obtained by combining the macroscopic deformation with the displacement field $\mathbf{M}^t$:

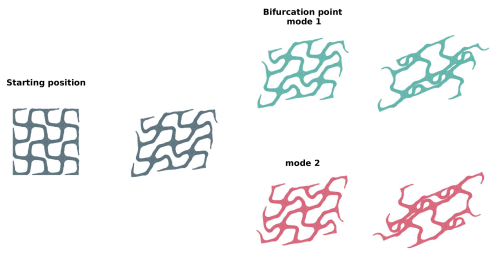
$$\mathbf{X}^t = \mathbf{X}^1 (\mathbf{F}^t)^\top + \mathbf{M}^t. \tag{10}$$

The deformation gradient is constructed by linearly interpolating from the identity matrix $\mathbf{I}$ to a prescribed terminal deformation gradient $\mathbf{F}_T$:

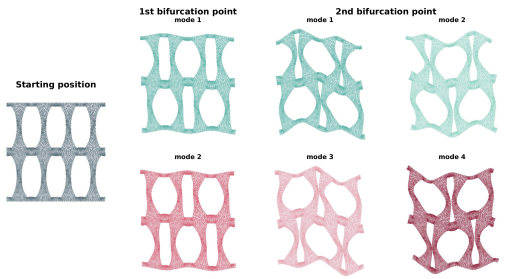
$$\mathbf{F}^t = \mathbf{I} + \frac{t-1}{T-1}(\mathbf{F}_T - \mathbf{I}), \quad t \in \{1, \dots, T\}. \tag{11}$$

The dataset is well-suited to our analysis because geometries can end in $|\mathcal{M}_\mathbf{c}| \in \{1, 2, 4\}$ distinct post-buckling modes under the same condition. Before bifurcation, symmetry-related modes should remain close in latent space; after bifurcation, the model must represent distinct modes without losing their shared trajectory-level structure.

Examples. Figure 10 summarizes two examples of the metamaterials dataset with distinct modes. Figure 10a shows a trajectory with 2 modes. The starting configuration and the last timestep before buckling are displayed on the left in gray. Under progressive loading, the structure branches into two distinct post-buckling configurations: modes 1 and 2, which, in this case, are mirror reflections of one another. Both modes are shown at two successive timesteps, capturing how the transformed pattern becomes more pronounced with increasing load.



(a) Example trajectory with one bifurcation point resulting in two modes.



(b) Example trajectory with two bifurcation points resulting in four modes.

Figure 10: Example mechanical metamaterials trajectories with distinct post-buckling mode structures.

Figure 10b shows a trajectory with *4 modes*. Starting from the undeformed configuration on the left, the structure first reaches a bifurcation point where two modes emerge. By moving toward the terminal deformation gradient $\mathbf{F}_T$, a second bifurcation point is reached, resulting in four distinct modes. This hierarchical branching behavior demonstrates how repeated symmetry-breaking events can generate increasingly varied deformed configurations from a single initial geometry.

Dataset Split To prevent leakage between subsets, we split the data by geometry: each of the 1,020 geometries is assigned to exactly one of the training, validation, or test sets. All 12 loading paths of a geometry are kept together in the same subset, so no test geometry is seen during training. We provide a detailed overview of the training, validation, and test split in Table 6 and their mode distributions. Overall, the validation and test data share a similar number of modes across categories.

Table 6: Summary of dataset splits.

Split	1 mode	2 modes	4 modes	Trajectories	Share (%)
Training	6,529	1,777	262	8,568	70
Validation	1,408	377	51	1,836	15
Test	1,391	354	91	1,836	15
Total	9,328	2,508	404	12,240	100

C.3 Allen-Cahn Dataset

The Allen-Cahn dataset describes phase separation in a three-dimensional periodic domain Ω, see Figure 11. The state is a scalar phase field $u(\mathbf{x}, t) \colon \Omega \times [0, 50] \to \mathbb{R}$ governed by

$$\frac{\partial u}{\partial t} = \epsilon^2 \nabla^2 u - \left(u^3 - \mu u\right). \tag{12}$$

Here, ϵ sets the characteristic width and smoothness of the interfaces, and μ determines the local phase structure. For $\mu > 0$, small perturbations of the mixed state grow and form spatial domains approaching $u = \pm\sqrt{\mu}$. A small subset of the data uses $\mu < 0$, a non-separating regime in which the homogeneous state is stable. We discretize u on an $N_x \times N_y \times N_z$ grid and denote the discrete state at timestep t by $\mathbf{u}^t \in \mathbb{R}^{N_x \times N_y \times N_z}$.

Dataset generation. We generated 1,000 parameter conditions on a 64^3 grid over the periodic domain $[0, 2\pi]^3$. For every condition, ϵ was sampled log-uniformly as $\epsilon = 10^z$ with $z \sim \mathcal{U}(-1.5, -1.0)$, i.e. $\epsilon \in [10^{-1.5}, 10^{-1}] \approx [0.0316, 0.1]$. Of the sampled μ values, 98% were drawn uniformly from $[0.05, 1.0]$ and 2% from $[-0.1, 0]$.

All trajectories start from the same homogeneous state $u^0(\mathbf{x}) \equiv 0$. However, because this state is a valid solution but unstable, uniform random noise with an amplitude of 0.1 is added to the initial state in order to perturb it away from the unstable all-zero solution. The equation is integrated for 5,000 steps with $\Delta t = 0.01$, using a semi-implicit Fourier-spectral scheme: the nonlinear reaction term is evaluated explicitly, whereas the diffusion term is solved implicitly by pointwise division in Fourier space. Periodic boundary conditions are applied along all three axes. The initial state and every 100th integration step are retained, giving $T = 51$ snapshots per trajectory over the time interval $[0, 50]$. Consequently, the stored phase-field tensor before splitting has shape $4{,}000 \times 51 \times 64 \times 64 \times 64$.

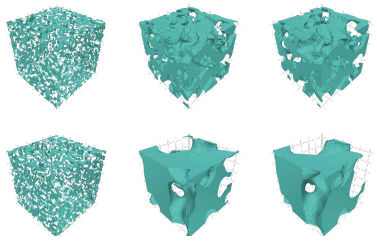


Figure 11: Allen-Cahn example for two distinct ϵ, μ settings.

Modes. For Allen-Cahn the control protocol is constant, $\boldsymbol{\ell}^t \equiv (\epsilon, \mu)$. For each sampled (ϵ, μ) condition, four trajectories were simulated from the shared homogeneous initial state. These trajectories share the same physical parameters and provide samples of different phase-separation morphologies under the same condition $\mathbf{c}$. In addition, the Allen-Cahn equation is invariant under the analytic transformation $u \mapsto -u$. Applying this sign flip to each realization provides its phase-exchanged counterpart, yielding eight candidate modes per condition (four simulated realizations and their four sign-reversed counterparts).

Dataset Split. The data are split at the level of parameter conditions rather than individual trajectories. Thus, all four trajectories belonging to the same (ϵ, μ) pair remain in the same subset, preventing

information leakage between training and evaluation. The conditions were shuffled and divided into 70% training, 15% validation, and 15% test data, as summarized in Table 7.

Table 7: Summary of the Allen-Cahn dataset splits. Each condition contains four simulated trajectories and up to eight modes after analytic sign augmentation.

Split	Conditions	Simulated trajectories	Candidate modes	Share (%)
Training	700	2,800	5,600	70
Validation	150	600	1,200	15
Test	150	600	1,200	15
Total	1,000	4,000	8,000	100

D Experimental Details

D.1 Architecture

Autoencoder. The encoder $\mathcal{E}_{\phi_{\text{enc}}}$ and decoder $\mathcal{D}_{\phi_{\text{dec}}}$ form a Perceiver-style cross-attention autoencoder (Jaegle et al., 2021; Alkin et al., 2024; Sestak et al., 2025) for the 2D mechanical metamaterials and Beam3D datasets. The encoder maps N node points into a fixed, learned latent array of L tokens. The decoder reconstructs per-point deformations via cross-attention from positional queries. All nonlinear activations are GELU (Hendrycks & Gimpel, 2016), and we additionally embed input and query positions via the sine-cosine position embedding from transformers (Vaswani et al., 2017). A schematic of the architecture is provided in Figure 3; dataset-specific autoencoder hyperparameters are listed in Table 8.

The Allen-Cahn Autoencoder uses a 3D Vision Transformer (ViT3D). The encoder patchifies the input voxel grid to $8 \times 8 \times 8$ patches, which are run through self-attention/MLP transformer blocks to get a $4 \times 4 \times 4$ latent representation. The decoder mirrors the encoder where the latent representation is upsampled to the original voxel grid.

We ensure latent stability by normalizing the physical fields and conditioning parameters using dataset-specific statistics, while layer normalization is used throughout the attention-based encoders and latent flow network. This stabilizes the latent representations without introducing an additional variational regularization term.

Flow Matching Model. We employ a two-stage latent flow matching model. In the first stage, an encoder maps each physical state to a compact latent representation

$$\mathbf{Z}^t = \mathcal{E}_{\phi_{\text{enc}}}(\mathbf{X}^1, \mathbf{M}^t), \tag{13}$$

while a decoder reconstructs the original state.

The latent representation of a complete trajectory is

$$\mathbf{Z}_1 := \left(\mathbf{Z}^1, \ldots, \mathbf{Z}^T\right) \in \mathbb{R}^{T \times L \times D_z}. \tag{14}$$

In the second stage, the autoencoder is frozen and a Scalable Interpolant Transformer (SiT) (Ma et al., 2024; Albergo et al., 2025) is trained to model the distribution of latent trajectories. A Gaussian sample $\boldsymbol{\xi} \sim \mathcal{N}(0, I)$ is coupled to each data trajectory. At a randomly sampled flow time $\tau \in [0, 1]$, the interpolated latent state is

$$\mathbf{Z}_\tau = \alpha_\tau \mathbf{Z}_1 + \sigma_\tau \boldsymbol{\xi}, \tag{15}$$

where Beam3D and the mechanical metamaterials use the generalized variance-preserving (GVP) Ma et al. (2024) path with $\alpha_\tau = \sin(\frac{1}{2}\pi\tau)$ and $\sigma_\tau = \cos(\frac{1}{2}\pi\tau)$, and Allen-Cahn uses the linear path with $\alpha_\tau = \tau$ and $\sigma_\tau = 1 - \tau$. Moreover, Eq. 5 in the main text shows the linear path. For a general path, the velocity field of Eq. 7 is $\widehat{v}_{\theta_{\text{flow}}} = \dot{\alpha}_\tau \widehat{\mathbf{Z}}_1 + \dot{\sigma}_\tau \left(\mathbf{Z}_\tau - \alpha_\tau \widehat{\mathbf{Z}}_1\right)/\sigma_\tau$, which reduces to $(\widehat{\mathbf{Z}}_1 - \mathbf{Z}_\tau)/(1 - \tau)$ for the linear path. The configured models directly estimate the clean latent trajectory,

$$\widehat{\mathbf{Z}}_1 = f_{\theta_{\text{flow}}}(\mathbf{Z}_\tau, \tau, \mathbf{Z}^{1:T_{\text{obs}}}, \mathbf{c}). \tag{16}$$

The predicted latent trajectory is mapped back to the physical signal using the frozen decoder:

$$\widehat{\mathbf{M}}^t = \mathcal{D}_{\phi_{\text{dec}}}\left(\widehat{\mathbf{Z}}_1^t, \mathbf{X}^1\right). \tag{17}$$

D.2 Implementation Details

Beam3D For Beam3D, each state contains the three-dimensional displacement field of the beam. The Perceiver encoder represents a state using six latent tokens of dimension 128, producing a fixed-size representation that is independent of the number of mesh nodes. The flow model is conditioned on the prescribed scalar tip displacement and observes the initial trajectory frame in the standard configuration. Because the direction of beam buckling is rotationally symmetric, random azimuthal rotations are applied during training to expose the model to the complete SO(2) orbit of equivalent buckling directions.

Mechanical Metamaterials Each mechanical metamaterial state is represented by a two-dimensional nodal displacement field together with the node positions, node types, and wallpaper-group information. A Perceiver autoencoder compresses each state into 128 latent tokens of dimension 128. The flow model is conditioned on the prescribed 2×2 deformation gradient, represented as a four-dimensional vector. We group solution branches belonging to the same physical condition during training, and restrict optimal-transport coupling to samples within the same group. This encourages consistent matching between alternative bifurcation modes without coupling trajectories generated under different physical conditions.

Allen–Cahn Each Allen–Cahn state is a scalar phase field defined on a periodic 64^3 grid. A three-dimensional vision transformer divides the field into volumetric patches and encodes them into a structured latent grid. The flow model is conditioned on the physical parameters (ϵ, μ). Temporal and spatial rotary embeddings allow the transformer to distinguish both trajectory time and spatial latent-token position. The implementation supports periodic translations, sign inversion, axis reflections, and axis permutations as symmetry-preserving transformations of the Allen–Cahn system.

D.3 Training

Training is divided into two stages: (i) autoencoder training and (ii) latent flow training. The first stage involves training the autoencoder to reconstruct individual physical states. The encoder $\mathcal{E}$ and decoder $\mathcal{D}$ are jointly optimized to reconstruct individual physical states. Invalid or padded spatial elements are excluded from the reconstruction objective. The second stage involves training the latent flow model to learn the distribution of latent trajectories. The pretrained autoencoder is frozen, and only the SiT flow model and physical-parameter conditioner are optimized. Gaussian noise with the same shape as the latent trajectory is sampled and paired with the data using either independent coupling or a Hungarian optimal-transport assignment based on latent-space squared distance. When mode groups are available, the assignment is restricted to trajectories belonging to the same physical condition.

For clean-trajectory prediction, the flow time is sampled according to $\tau \sim \mathcal{U}(\tau_{\min}, 1 - \tau_{\min}), \tau_{\min} = 10^{-3}$.

The interpolated latent trajectory $\mathbf{Z}_\tau$, flow time τ, control protocol, and observed-frame condition are then passed to the model. Invalid physical timesteps are replaced by zero before coupling and are excluded from both attention and loss computation.

The standard conditional models observe the first trajectory frame. The Allen–Cahn transfer configuration uses unconditional generation by setting the number of observed frames to zero.

Optimization. The second-stage models are optimized with AdamW (Loshchilov & Hutter, 2019). The standard learning rate is 10^{-4}, with cosine learning-rate decay for the mechanical metamaterial and standard Allen–Cahn models and a constant learning rate for Beam3D. Gradient norms are clipped at 1, and optimizer updates containing non-finite gradients are skipped. Training is step-based. Checkpoints are selected using the validation loss, with additional checkpoints maintained for generation metrics such as mode coverage.

Data augmentation. No generic spatial augmentation is applied to the mechanical metamaterial trajectories because their geometric, node-type, and wallpaper-group information is explicitly represented. Beam3D trajectories are randomly rotated around the longitudinal beam axis, reflecting the continuous rotational symmetry of the buckling problem. Allen–Cahn augmentation can combine periodic translations, sign inversion, axis reflection, and axis permutation. Each transformation is applied consistently to all frames of a trajectory to preserve its temporal dynamics.

D.4 Loss Function

The first-stage reconstruction model and the second-stage latent flow model are optimized separately.

Reconstruction loss. For a physical signal $\mathbf{M} \in \mathbb{R}^{N\times C}$ and its reconstruction $\widehat{\mathbf{M}}$, the first-stage reconstruction loss is the mean squared error over nodes:

$$\mathcal{L}_{\text{rec}} = \frac{1}{N}\sum_{n=1}^{N} \|\widehat{\mathbf{m}}_n - \mathbf{m}_n\|_2^2. \tag{18}$$

Here, $\mathbf{m}_n$ and $\widehat{\mathbf{m}}_n$ denote the signal and its reconstruction at node n. Padded or invalid nodes, where present, are excluded from both the sum and the count N. For Metamaterials, $\mathbf{m}_n \in \mathbb{R}^2$ is the nodal displacement. For Beam3D, $\mathbf{m}_n \in \mathbb{R}^3$ is the three-dimensional nodal displacement. For Allen-Cahn, $\mathbf{m}_n \in \mathbb{R}$ is the scalar phase-field value.

Latent flow loss The principal second-stage objective is the frame-masked mean-squared error between the predicted and target latent trajectories. With

$$\widehat{\mathbf{Z}}_1 = f_{\theta_{\text{flow}}}\left(\mathbf{Z}_\tau, \tau, \mathbf{Z}^{1:T_{\text{obs}}}, \mathbf{c}\right), \tag{19}$$

the loss is

$$\mathcal{L}_{\text{flow}} = \frac{\displaystyle\sum_{t=1}^{T} r^t \left\|\widehat{\mathbf{Z}}_1^{\,t} - \mathbf{Z}_1^t\right\|_F^2}{\displaystyle LD_z \sum_{t=1}^{T} r^t}. \tag{20}$$

Here, τ denotes flow time and t indexes trajectory frames. The tensors $\widehat{\mathbf{Z}}_1^{\,t}, \mathbf{Z}_1^t \in \mathbb{R}^{L\times D_z}$ are the predicted and target clean latents at frame t, respectively. The indicator $r^t \in \{0,1\}$ specifies whether frame t contributes to the loss. The denominator normalizes the squared error by the total number of latent entries across contributing frames.

Bifurcation classification loss. The Metamaterial and Beam3D models optionally include a temporal attention head that predicts whether each trajectory frame lies before or after bifurcation. The head is trained using masked binary cross-entropy:

$$\mathcal{L}_{\text{bif}} = -\frac{\sum_{t=1}^{T} r^t \left[b^t \log p^t + (1-b^t)\log(1-p^t)\right]}{\sum_{t=1}^{T} r^t}. \tag{21}$$

Here, p^t is the predicted post-bifurcation probability, b^t is the corresponding ground-truth target, and $\widehat{b}^t = \mathbb{1}[p^t \geq 1/2]$ is its binary prediction.

The complete second-stage objective is

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{bif}}\mathcal{L}_{\text{bif}}. \tag{22}$$

The standard Allen–Cahn model does not use the auxiliary classification head and is optimized only using $\mathcal{L}_{\text{flow}}$.

D.5 Adaptations to particle guidance

In their notation, Corso et al. (2024) add the gradient of a joint cross-particle potential to the reverse-diffusion drift. They evaluate this interaction on the noisy state x_τ. We make four adaptations for our latent data-prediction model.

Denoised interaction. We compute repulsion on the estimate $\widehat{\mathbf{Z}}_1^{(k)}$. It is restricted by the shared predicted mask $\widehat{\mathbf{b}}$ in Eq. 7, which is computed once before guidance begins and held fixed throughout guided sampling. The original method computes it directly on x_τ. We scale the resulting displacement by $(\sigma_\tau^2 + \delta)^{-1}$ before adding it to the velocity update in Eq. 7; the constant $\delta > 0$ prevents divergence as $\sigma_\tau \to 0$. The interaction strength therefore follows the model's denoising confidence.

Normalized force. The original force is an unnormalized gradient of a scalar potential. Our force is a kernel-weighted average of unit pairwise directions. We divide by the total kernel mass $\sum_j k_{ij}$. This keeps its scale stable as the number and weight of neighboring particles change.

Self-interaction. For a translation-invariant kernel, the original self term has zero gradient and vanishes. Our self term is excluded from the numerator because its direction is undefined. It remains in the normalizing denominator and therefore damps the interaction.

Frame and time gating. The original potential acts on the full particle state with a generic coefficient. We apply repulsion only to the predicted bifurcation selected by $\widehat{\mathbf{b}} = 1$. It is zero elsewhere. We use the bounded schedule $\gamma(\tau) = \max(\tau^\kappa, \gamma_{\min})$. The schedule strengthens toward the data endpoint and never switches off. Here $\kappa > 0$ is the schedule exponent and $\gamma_{\min} \in (0, 1]$ is its floor; w_{rep} is the global repulsion weight. No additional per-frame weight is used beyond $\widehat{\mathbf{b}}$.

The bandwidth uses a median heuristic over squared, dimension-scaled pairwise distances. This choice follows the synthetic-experiment code released with Corso et al. (2024). Their paper leaves the kernel and bandwidth domain-specific and does not prescribe a general formula. We provide the latent-flow and guidance-schedule hyperparameters used for the reported models in Table 9.

D.6 HYPERPARAMETERS

Table 8: Hyperparameters of the autoencoder (first stage) used to produce the reported outputs, per dataset.

Parameter	Beam3D	Metamaterials	Allen-Cahn
Architecture	Perceiver AE	Perceiver AE	ViT3D + spatial token bottleneck
Params	1.7×10^6	1.7×10^6	6.1×10^6
Input shape	node states ($N \times C$)	point set (17 groups, 8 node types)	$64 \times 64 \times 64 \times 1$ voxel grid
Patch size	–	–	$8 \times 8 \times 8$ (512 tokens)
Latent shape ($L \times D_z$)	6×128	128×128	$4 \times 4 \times 4$ (64×384)
Encoder blocks (cross/self-attn)	1 / 1	1 / 1	ViT depth 6 (heads 6, embed 192)
Decoder blocks (cross/self-attn)	1 / 1	1 / 1	ViT depth 4 (heads 6, embed 192)
Head dims (enc cross/latent/dec cross/latent)	16/16/16/16	16/16/16/16	–
Heads (enc cross/latent/dec cross/latent)	32/16/32/16	32/16/32/16	6 (enc) / 6 (dec)
Decoder query width	160	160	–
Activation	GELU	GELU	GELU
Dropout (latent / query)	0.0 / 0.05	0.0 / 0.05	0.0 (encoder)
Loss	masked MSE	masked MSE	masked MSE
Optimizer	AdamW	AdamW	AdamW
Learning rate	10^{-4}	10^{-4}	2×10^{-4}
Weight decay	0.0	0.01	0.01
Scheduler	Cosine (eta_min 10^{-6})	Cosine (eta_min 1.5×10^{-5})	none
Gradient norm clipping	1.0	1.0	1.0
Batch size	128	128	128
Precision	bf16-mixed	bf16-mixed	bf16-mixed

D.7 EVALUATION DETAILS: BEAM3D AND METAMATERIALS

We apply a mode-aware evaluation pipeline with task-specific adaptations to the underlying mode set. For discrete-mode settings, we match generated rollouts to their nearest ground-truth mode using a nearest-mode distance. For metamaterials, this uses an absolute distance threshold of 0.1 over the whole trajectory. We compute the mean absolute error over the whole trajectory and sample 16 trajectories per condition, using 50 integration steps. For both Beam3D and Allen-Cahn, the mode space is continuous. Rollouts for the Beam3D dataset are therefore evaluated by binning the buckling azimuth and matching against the nearest orbit member with a relative-distance rejection criterion. We reject a generated trajectory if its relative distance to the matched orbit member exceeds 50% of its ground-truth displacement magnitude. Here we sample 180 trials with fewer integration steps (10 steps, DDIM (Song et al., 2021)), and compare against GeoTDM (1000 steps, DDPM (Ho et al., 2020)) and STFlow (10 steps, Euler integration) using the same number of evaluation samples.

Table 9: Hyperparameters of the latent-space (second-stage) flow model used to produce the reported outputs, per dataset.

Parameter	Beam3D	Metamaterials	Allen-Cahn
Params (non-AE)	7.1×10^6	17.2×10^6	31.8×10^6
in_dim / cond_dim	128 / 128	128 / 128	384 / 192
Hidden size	256	512	512
Depth	5	4	8
Heads	8	16	16
MLP ratio	2.0	2.0	2.0
Bifurcation head	attention, λ_{bif}=0.1	attention, λ_{bif}=0.1	none
Conditioner (in/dim/hidden)	1 / 128 / 512	4 / 128 / 256	2 / 192 / 512
Transport path / target	GVP / data	GVP / data	linear / data
Sampler	DDIM, η=0.0	DDIM, η=0.0 (eval η=1.0)	DDIM, η=0.0 (eval η=1.0)
Loss	flow MSE (frame-masked)	flow MSE (frame-masked)	flow MSE (frame-masked)
Optimizer	AdamW	AdamW	AdamW
Learning rate	10^{-4}	10^{-4}	3×10^{-5}
Weight decay	0.0	0.01	0.01
Scheduler	none	Cosine (eta_min 10^{-6})	Cosine (T_{max}=500k, eta_min 10^{-5})
Batch size	64	16	8
Precision	bf16-mixed	bf16-mixed	fp32
PG settings ($w_{\text{vel}}/w_{\text{rep}}/\kappa$)	1.0 / 1.0 / 1.0	1.0 / 50 / 0.25	1.0 / 75 / 0.025

D.8 Evaluation Details: Allen-Cahn

The Allen-Cahn dataset dynamics are invariant under periodic translations, sign flips and point-group symmetries of the cubic domain, making pointwise comparison to a single reference rollout not meaningful. We evaluate in two steps: (i) each generated sample is matched to a ground-truth branch using canonical alignment, (ii) and three different JSD metrics test distributional uniformity over the possible orbit operations.

Matching. For a generated field $\mathbf{M}$, we compute the time-averaged radial power spectrum $S(\mathbf{M})$ as the magnitude of the 3D FFT at each timestep, binned by integer radial frequency and averaged over time. This value is invariant to translation and the octahedral point group, making it suitable for identifying the best mode independent of orientation and shift. The matched branch is $m_\star^{(k)} = \arg\max_m \cos_\text{sim}\big(S(\mathbf{M}^{(k)}),\, S(\mathbf{R}^{(m)})\big)$, where $\mathbf{R}^{(m)}$ is the selected ground-truth reference for branch m. We count sign-inverted counterparts separately for mode-coverage JSD, but disregard sign for translation and rotoreflection JSD. Given $m_\star^{(k)}$, we recover translation and orientation by an exhaustive search over the 48-element octahedral group G_{oct} (axis reflections and permutations). For each $g \in G_{\text{oct}}$, we apply g to the sample's mean field and phase-correlate it against the matched mode's mean field using the normalized cross-power spectrum. The peak location gives the best periodic shift, and the height gives an alignment score. We only keep the $g_\star$ with the highest peak, resulting in one match result $(m_\star^{(k)}, \boldsymbol{\psi}_\star^{(k)}, g_\star^{(k)})$ per sample, where $\boldsymbol{\psi}_\star^{(k)} \in \mathbb{Z}_{64}^3$ is the optimal periodic grid shift. The approach mirrors the alignment step in Klein et al. (2023). In contrast to that work, and because the symmetry group is small and finite, the exact minimum is found directly rather than approximated.

MCon-MAE. For each generated rollout $\widehat{\mathbf{M}}^{(k)}$, let $\widehat{\mathbf{M}}^{\text{rel},(k)}$ be its Gauss–Newton relaxation onto an Allen-Cahn PDE-consistent trajectory. We compute

$$\text{MCon-MAE} = \frac{1}{K}\sum_{k=1}^{K} \text{mean}\Big(\Big|\widehat{\mathbf{M}}^{\text{rel},(k)} - \widehat{\mathbf{M}}^{(k)}\Big|\Big), \tag{23}$$

This dataset-specific metric compares each raw generation with its own relaxed projection; no saved reference rollout, matched branch, or branch label enters the calculation. It is therefore distinct from the nearest-reference MCon-MAE in Eq. 8.

JSD Metrics. Each JSD metric evaluates one component of the matching approach above over K trials. The results are then aggregated into a categorical distribution, and we compute the Jensen-Shannon divergence against the uniform distribution, as summarized in Table 10.

Sampling. To estimate the distribution over the large number of potential solutions, we sample 100 times for each condition $\mathbf{c}$, using 30 denoising steps.

Table 10: Summary of the Allen-Cahn JSD metrics, including their categories and tests.

Metric	Categories	Tests
Mode Coverage JSD	mapping to $\mathcal{M}_{\mathbf{c}}$ ground truth branches	Are all solution branches reached equally often?
Translation JSD	grid shifts, coarsened to 8 bins per axis ($8^3 = 512$)	Given one ground truth sample, how often are periodic shifts recovered?
Rotoreflection JSD	48 octahedral elements	Given one ground truth sample, is the recovered orientation/reflection uniform?

E RESULTS

E.1 QUANTITATIVE RESULTS

We present a summary of our results for all three datasets in the tables below. We divide the results into two sections: an overview of the JSD and the MCon-MAE results. Section 3 explains additional metrics, which we include for each dataset. For the JSD tables, we also incorporate upper and lower bounds as references. The upper bound represents total mode collapse, while the lower bound reflects the expected JSD when sampling from a random uniform distribution. For all tables, we used five different evaluation seeds.

Beam3D

Table 11: Beam3D prediction performance. Lower MCon-MAE, JSD, and rejection rate indicate better reconstruction accuracy, better agreement with the target mode distribution, and fewer physically invalid rollouts. (mean $\pm$ std over five evaluation seeds)

(a) JSD

Method	JSD
GeoTDM	$\mathbf{0.036 \pm 0.001}$
STFlow	0.074 ± 0.001
Bi-FORK w/o PG	0.043 ± 0.003
Bi-FORK	0.041 ± 0.001
Upper bound*	0.644
Lower bound**	0.040

* assuming total mode collapse, ** expected JSD when sampling from a random uniform distribution

(b) MCon-MAE and rejection rate

Method	MCon-MAE	Rejection
GeoTDM	1.040 ± 0.001	99.2 ± 0.0
STFlow	0.114 ± 0.000	0.5 ± 0.1
Bi-FORK w/o PG	0.018 ± 0.001	0.1 ± 0.1
Bi-FORK	$\mathbf{0.018 \pm 0.000}$	$< \mathbf{0.01}$

Metamaterials For the dataset Bi-FORK achieves JSD values below the lower-bound reference. This reference represents the expected JSD for a finite set of independent samples from the uniform target distribution, rather than a strict lower bound. Particle guidance can create dependence between samples that promotes low-discrepancy sampling, leading to more balanced mode frequencies and JSD values below the reference. Full coverage requires a valid sample for every mode; JSD includes all generated samples.

Table 12: Result comparison of Bi-FORK (including w/o particle guidance) against the affine baseline, evaluated with MCon-MAE, JSD, and coverage for each mode count. (mean ± std over five evaluation seeds)

(a) JSD

Method	JSD		
	$\lvert\mathcal{M}_c\rvert = 1$	$\lvert\mathcal{M}_c\rvert = 2$	$\lvert\mathcal{M}_c\rvert = 4$
Affine Baseline	0.000	0.216	0.380
Bi-FORK w/o PG	0.000 ± 0.000	0.050 ± 0.002	0.087 ± 0.008
Bi-FORK	0.000 ± 0.000	$\mathbf{0.013 \pm 0.001}$	$\mathbf{0.023 \pm 0.002}$
Upper bound*	0	0.216	0.380
Lower bound**	0	0.008	0.027

* assuming total mode collapse, ** expected JSD when sampling from a random uniform distribution

(b) MCon-MAE and full coverage

Method	MCon-MAE	Full Coverage %	
		$\lvert\mathcal{M}_c\rvert = 2$	$\lvert\mathcal{M}_c\rvert = 4$
Affine Baseline	0.021	0	0
Bi-FORK w/o PG	$\mathbf{0.016 \pm 0.000}$	38 ± 0.9	13 ± 3.4
Bi-FORK	0.018 ± 0.000	74 ± 1.0	48 ± 3.7

Allen-Cahn

Table 13: Result comparison of Bi-FORK (including w/o particle guidance), evaluated with relaxed MCon-MAE and three JSD metrics. (mean ± std over five evaluation seeds)

(a) JSD

Method	Mean JSD JSD	Mode-Coverage JSD	Translation JSD	Rotoreflection JSD
Bi-FORK w/o PG	0.217 ± 0.004	0.131 ± 0.010	0.446 ± 0.000	0.076 ± 0.001
Bi-FORK	$\mathbf{0.204 \pm 0.001}$	$\mathbf{0.089 \pm 0.001}$	0.446 ± 0.000	0.076 ± 0.002
Upper bound*	-	0.497	0.686	0.642
Lower bound**	-	0.009	0.446	0.076

* assuming total mode collapse, ** expected JSD when sampling from a random uniform distribution

(b) MCon-MAE

Method	MCon-MAE
Bi-FORK w/o PG	0.0515 ± 0.001
Bi-FORK	$\mathbf{0.0453} \pm 0.000$

E.2 Ablations

Bifurcation Head Prediction Accuracy. To effectively use particle guidance, we predict a bifurcation mask corresponding to the bifurcation time point. We assess the accuracy of the predicted masks on the Beam3D and mechanical metamaterials datasets. The Allen-Cahn method does not use a bifurcation head. Instead, we set the guidance mask to $\widehat{b}^t = 1$ for every valid frame, allowing guidance throughout the trajectory. We compare the results based on accuracy, precision, recall, F1 score, and balanced accuracy, as shown in Table 14.

Table 14: Bifurcation-head classification on the complete test splits. Reporting the mean and standard deviation over five seeds.

Dataset	Accuracy	Precision	Recall	F1	Balanced acc.
Beam3D	0.989 ± 0.000	0.988 ± 0.000	1.000 ± 0.000	0.994 ± 0.000	0.950 ± 0.001
Metamaterials	0.970 ± 0.000	0.940 ± 0.002	0.871 ± 0.002	0.904 ± 0.002	0.930 ± 0.001

Wall Time. We analyze the wall time of each tested method on the Beam3D dataset. For the underlying experiments, we used a B300 SXM6 with 288 GB evaluated on one input condition over four random draws. We summarize the resulting times in Table 15.

Table 15: Wall-clock time taken to generate 180 samples of the Beam3D dataset over four random seeds.

Method	Samples	Conditions	Seconds Taken
GeoTDM	180	1	7766.36
STFlow	180	1	72.32
Bi-FORK w/o PG	180	1	**2.94**
Bi-FORK	180	1	3.95

E.3 Qualitative Results

Beam3D.

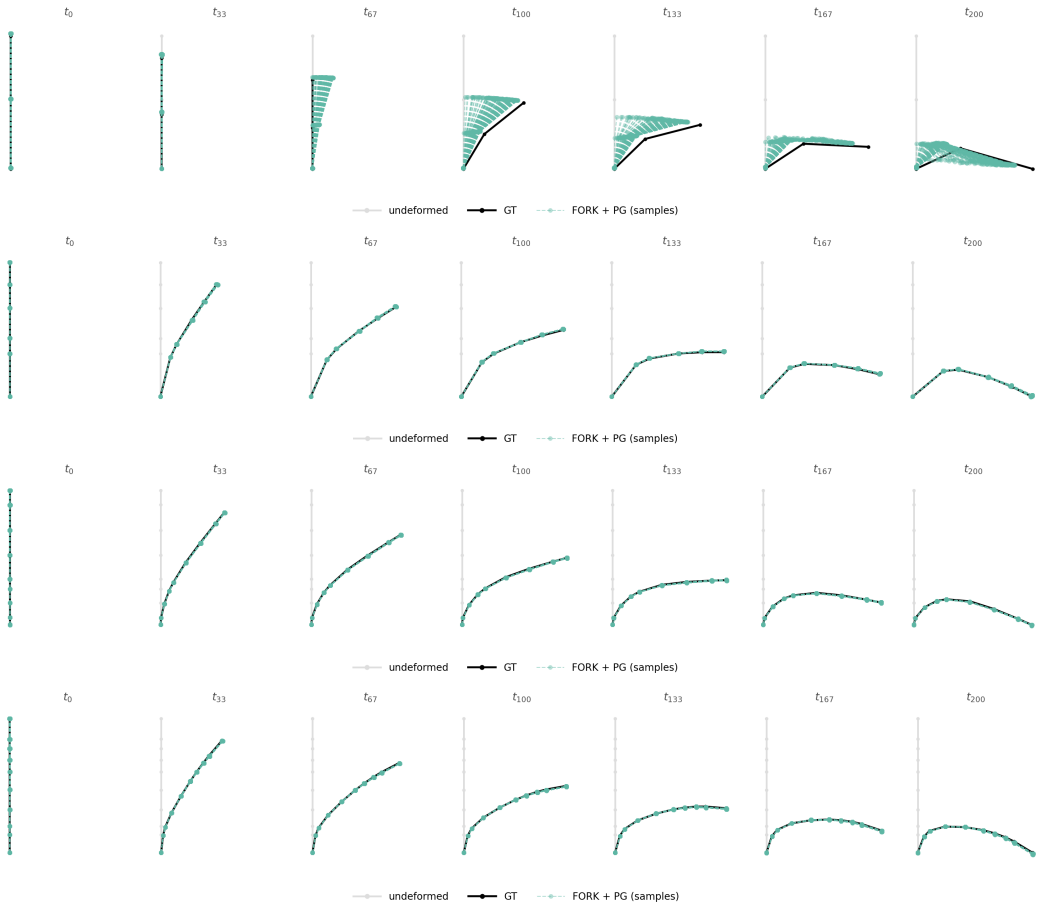


Figure 12: Beam bending process: showing predicted (green) vs actual (black) trajectory evolution for 180 samples. Node count ranging from 3 to 11 nodes

Allen-Cahn

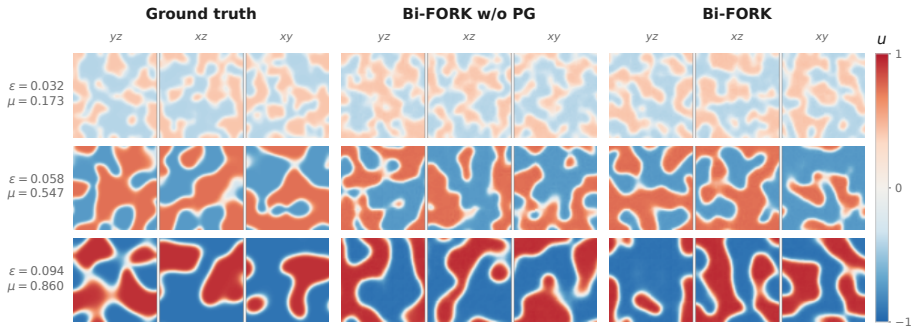


Figure 13: Sample comparison for increasing μ and ϵ, showcasing the diversity of potential solutions compared to the ground truth.

Mechanical Metamaterials.

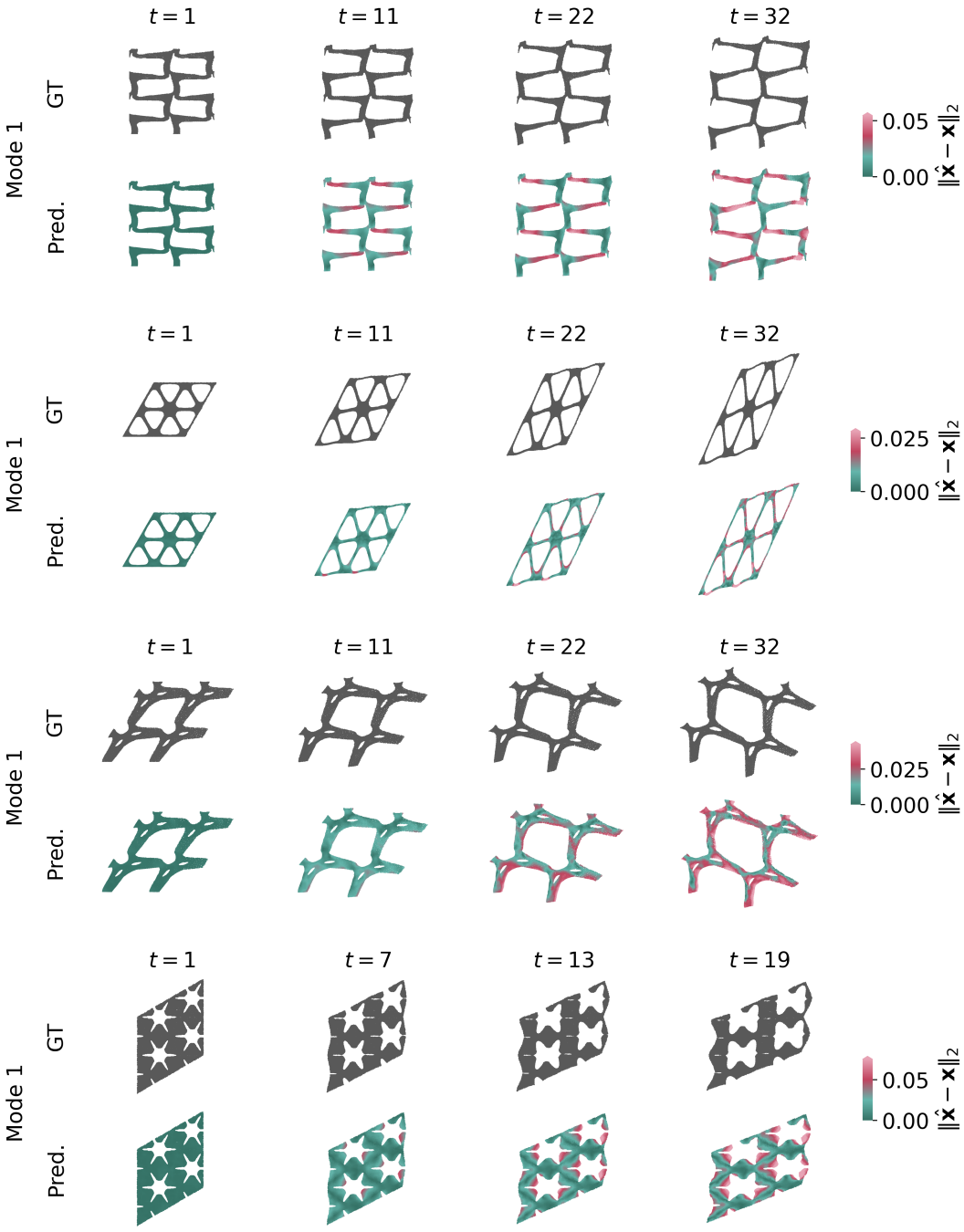


Figure 14: Visualization of **one-mode** samples generated by Bi-FORK with $K = 16$. The ground truth is compared with the closest predicted mode.

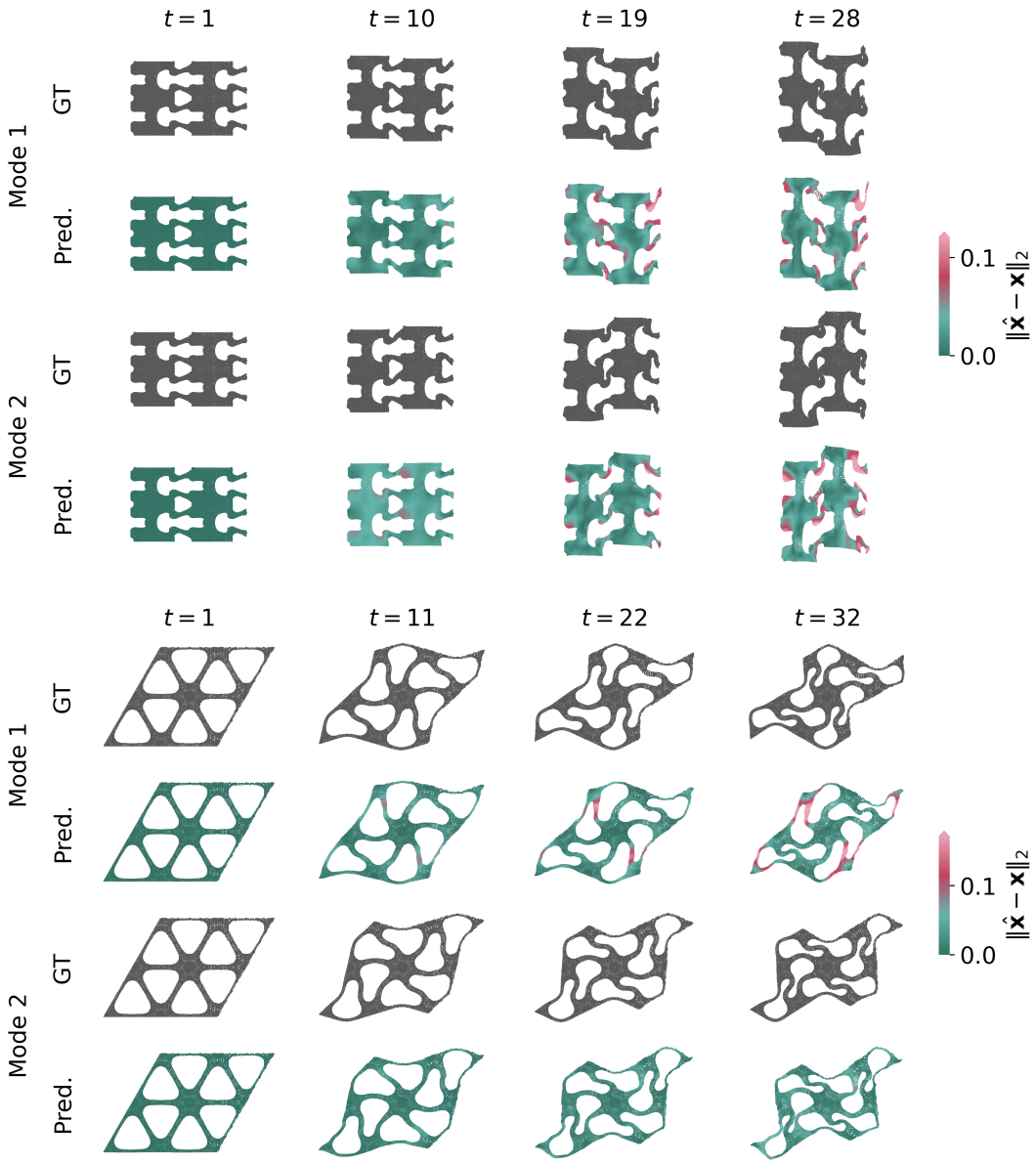


Figure 15: Visualization of **two-mode** samples generated by Bi-FORK with $K = 16$. The ground truth is compared with the closest predicted mode.

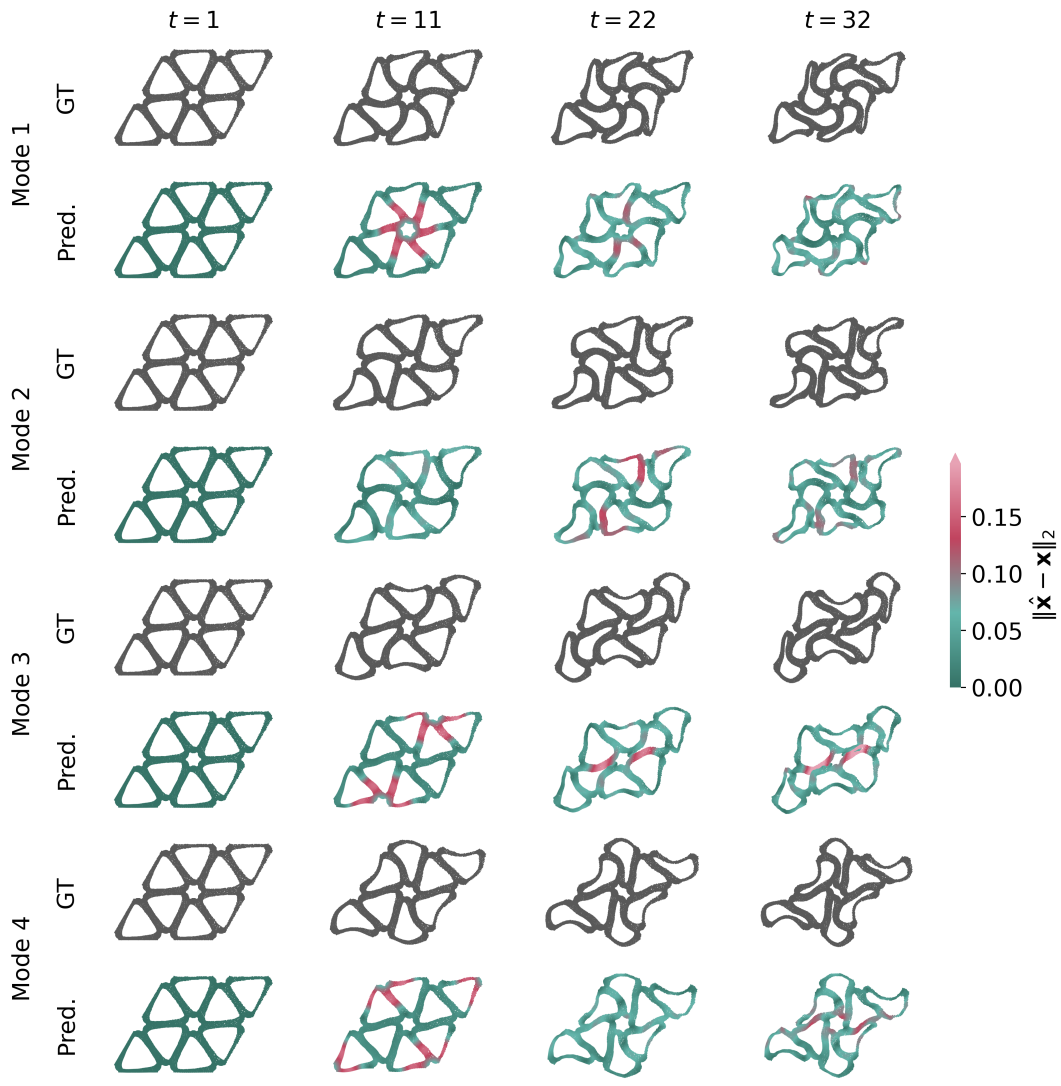


Figure 16: Visualization of a **four-mode** sample generated by Bi-FORK with $K = 16$. The ground truth is compared with the closest predicted mode.