Equivariant Flow Matching for Symmetry-Breaking Bifurcation Problems

Fleur Hendriks
Department of Mechanical Engineering
Eindhoven University of Technology
Eindhoven, 5600 MB
f.hendriks@tue.nl

Ondřej Rokoš
Department of Mechanical Engineering
Eindhoven University of Technology
Eindhoven, 5600 MB
o.rokos@tue.nl

Martin Doškář
Faculty of Civil Engineering
Department of Mechanics
Czech Technical University in Prague
Prague 6, 166 29, Czech Republic
martin.doskar@cvut.cz

Marc G.D. Geers
Department of Mechanical Engineering
Eindhoven University of Technology
Eindhoven, 5600 MB
m.g.d.geers@tue.nl

Vlado Menkovski
Department of Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, 5600 MB
v.menkovski@tue.nl

Abstract

Bifurcation phenomena in nonlinear dynamical systems often lead to multiple coexisting stable solutions, particularly in the presence of symmetry breaking. Deterministic machine learning models are unable to capture this multiplicity, averaging over solutions and failing to represent lower-symmetry outcomes. In this work, we formalize the use of generative AI, specifically flow matching, as a principled way to model the full probability distribution over bifurcation outcomes. Our approach builds on existing techniques by combining flow matching with equivariant architectures and an optimal-transport-based coupling mechanism. We generalize equivariant flow matching to a symmetric coupling strategy that aligns predicted and target outputs under group actions, allowing accurate learning in equivariant settings. We validate our approach on a range of systems, from simple conceptual systems to physical problems such as buckling beams and the Allen-Cahn equation. The results demonstrate that the approach accurately captures multimodal distributions and symmetry-breaking bifurcations. Moreover, our results demonstrate that flow matching significantly outperforms non-probabilistic and variational methods. This offers a principled and scalable solution for modeling multistability in high-dimensional systems.

1 Introduction

Machine learning enables efficient data-driven modeling of complex systems, especially when traditional methods impose high complexity or are incomplete [1; 2; 3; 4; 5; 6]. Many such systems exhibit bifurcations that mark sudden changes in system behavior due to a small change in a control parameter. Bifurcations are crucial in fields such as fluid dynamics [7], climate science [8; 9; 10; 11],

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

biology [12], and crowd dynamics [13; 14; 15], where predicting transitions is essential. Many bifurcating systems exhibit multistability, where multiple distinct stable states coexist under the same input parameters, and the number of these states can change as the parameters vary. This phenomenon often arises due to symmetry breaking, where a system's state after the bifurcation possesses fewer symmetries than before. This results in a multiplicity of possible solutions, all of which are equally valid.

Simulating such systems with machine learning is nontrivial and remains an open problem [16, §2.9.1]. Deterministic machine learning models can only produce the average of multiple solutions, resulting in nonphysical predictions that do not represent any true solution. Geometric deep learning approaches that rely on equivariant models preserve the system's symmetry but cannot select asymmetric outcomes, limiting their ability to capture bifurcations. Approaches such as canonicalization [17], input perturbation [4], model ensembles [18], deterministic machine learning models combined with pseudo-arclength continuation [19; 20], and sampling group operations to artificially break symmetry [21; 17] offer partial solutions but often lack generalizability or precision.

We propose modeling the full output distribution using generative modeling. Generative models aim to learn mappings from simple base distributions (e.g., Gaussian noise) to complex, high-dimensional target distributions. However, the probability distributions we require are singular: the set of allowed values (the support) lies on a subspace that has lower dimensionality than the input space, e.g., a Dirac delta in 1D, or a circle in 2D space. This means the probability mass must be concentrated as sharply as possible around the allowed values. When the target distribution is singular, the necessary mapping becomes highly nonlinear: nearby points in the source distribution may map to distant points in the target space. Capturing such mappings is challenging for neural networks, as it requires representing high-frequency functions [22]. This is a fundamental limitation of direct generative approaches, which must learn a single, highly nonlinear transformation. For example, variational autoencoders (VAE) tend to generate blurry images due to their limitations in capturing highly concentrated, multimodal distributions [23; 24; 25].

In contrast, iterative methods such as diffusion models, denoising processes, and flow matching approximate this complex mapping as a sequence of small integration steps. This iterative structure distributes the required nonlinearity across many stages, making the learning problem more tractable, as each step only needs to model a small, smooth transformation. As a result, these methods are better suited for (i) modeling singular distributions, where the probability mass is concentrated on a low-dimensional manifold, and (ii) multimodal distributions, where the probability mass must split far apart [26].

Therefore, to address the challenge of modeling multistability, we utilize flow matching [27] to model the full output distribution. We show that flow matching excels at parameterizing distributions whose probability mass is highly concentrated in a low-dimensional subspace (e.g., very close to two Dirac deltas). We further systematically integrate flow matching with equivariant modeling to tackle symmetry-breaking. Additionally, we develop 'symmetric coupling', an approach to determine the optimal training target, similar to minimatch optimal transport [28] and equivariant flow matching [29; 30]. This approach substantially improves the training and the quality of generated flow paths. We demonstrate our method on abstract and physical systems, including coin flips, buckling beams, and phase separation via the Allen-Cahn equation. Our approach effectively captures bifurcations and multistability, with and without symmetry breaking.

The code is available at `https://github.com/FHendriks11/bifurcationML/`.

2 Methods

Equivariance: A map $f : \mathcal{X} \to \mathcal{Y}$ is equivariant with respect to a group G if:

$$g \cdot y = f(g \cdot x), \quad \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \forall g \in G. \tag{1}$$

If an input x is invariant (i.e., self-similar) under a subset of the group actions (i.e., $g_x \cdot x = x$ for all $g_x \in G_x$, with $G_x \subseteq G$), then equivariance implies that output y is also invariant (i.e., $g_x \cdot y = y$). However, this preservation of symmetry inherently prevents a model from expressing symmetry-breaking behavior. To capture bifurcations and transitions into asymmetric states, equivariant models must be adapted to allow for symmetry-breaking solutions.

If there is symmetry-breaking in an equivariant system, there are multiple coexisting solutions for the same input. Meaning, if y is a solution, then so are all $g_x \cdot y, \forall g_x \in G_x$ (i.e., the orbit of y under G_x).

The equivariance in Equation (1) then still holds for the set of solutions $\{g_x \cdot y | g_x \in G_x\}$.

This is similar to the definition of 'relaxed equivariance' by Kaba et al. [31; 32] in that both account for input symmetries via stabilizer subgroups (i.e., groups G_x whose actions leave the input x invariant), but it differs by preserving equivariance over sets of outputs (i.e., orbits of solutions), whereas relaxed equivariance modifies the equivariance condition itself, by allowing a correction with an element from the input's stabilizer subgroup.

As a probability distribution: When using a generative model, we consider the set of solutions $\{y\}$ as a singular probability distribution $p(y|x)$, where the support of the distribution is the set of allowed solutions. The equivariance condition in Equation (1) then translates to:

$$p(y|x) = p(g \cdot y | g \cdot x) \quad \forall x \in \mathcal{X}, y \in \mathcal{Y}, g \in G. \tag{2}$$

To ensure that the predictions by the model satisfy Equation (2), we use (i) a model that is equivariant under G to parametrize the probability distribution and (ii) a prior $p(y_0)$ that is invariant under G. This ensures that the generated probability distribution respects the symmetry of the problem [33].

Flow matching: To model this probability distribution, we use flow matching [27], which learns a vector field $u(y_t, t, x)$ that transforms samples from an uninformed prior $p(y_0)$ into samples y_1 from the target distribution $p(y|x)$ over a continuous pseudo-time variable $t \in [0, 1]$. The vector field is learned by minimizing the expected squared error between the model prediction and the target vector field along linear interpolation paths between prior and target samples.

Symmetric coupling: Additionally, we leverage the fact that there are multiple equivalent outputs to improve our predictions, using *symmetric coupling*. During training, for each latent sample y_0 and target y_1, we find the closest equivalent under G_x to the current prediction, i.e., find

$$y_1' = \tilde{g}_x \cdot y_1 \tag{3}$$
$$\tilde{g}_x = \text{argmin}_{g_x \in G_x} c(y_0, g_x \cdot y_1), \tag{4}$$

where c is the cost function, for which the Euclidean distance squared $c = ||y_0 - y_1||^2$ is a possible choice. [1]

This is a way of straightening the flow paths similar to minibatch optimal transport [28], but instead of finding the optimal coupling between a minibatch of the prior samples and target samples, we find the optimal coupling between one prior sample y_0 and all symmetric equivalents $g_x \cdot y_1, g_x \in G_x$ of its corresponding target sample y_1. Our approach is similar to equivariant flow matching as described by Klein [29] and Song [30], except we only consider the group G_x that leaves the input x invariant. If this group contains permutations or rotations, we can therefore follow their lead and use the Hungarian algorithm for permutations or the Kabsch algorithm for rotations. Other options are checking all possible reflections of the target or using FFT cross-correlation to determine the optimal periodic translation of the target.

See Figure 1(a) for a visualization of the concepts of equivariance and self-similarity in the context of a buckling beam problem.

3 Experiments

Here, we showcase the performance of the proposed approach with four toy problems and two physically-grounded examples. See appendix A for more details on the datasets used and appendix B for implementation details.

[1] Note our notation is slightly different from the standard flow matching notation, since we already condition on an input x; therefore, we denote the prior sample as y_0 and the target sample as y_1, to stay consistent with the standard notation of equivariance between input x and output y.

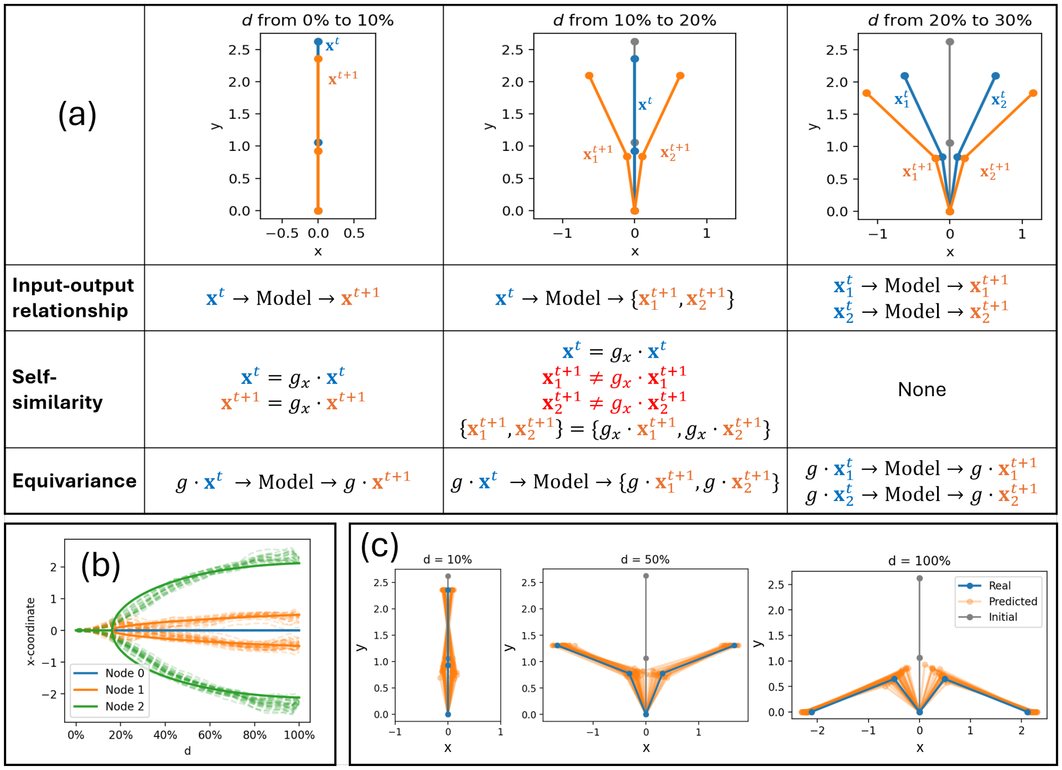

Figure 1: (a) Self-similarity and equivariance illustrated with the buckling beam problem, in which the input is the state at the current time step t and the target output is the next time step $t+1$. d is the relative vertical downward displacement of the beam tip, which is increased over time. The relevant symmetry group G here consists of only two elements: identity and reflection in the y-axis. In this case, the self-similarity of the beam before buckling happens to be described by the same group, i.e., here $G_x = G$. (b) The x-coordinates of each node over the course of a trajectory, showing 50 predictions of a trained flow matching model (dashed lines) compared to the two possible ground truths (solid lines). (c) The same 50 predictions compared to the two possible ground truths, showing the beam deformation.

Toy Models: We use 4 simple toy datasets to compare flow matching to non-probabilistic models and VAEs. Results in the form of the mean Wasserstein distance between the predicted and actual distribution of allowed outcomes are provided in Table 1, comparing a non-probabilistic model, a VAE, and flow matching.

For the 4-node graph, we additionally show how symmetric coupling further improves the results. Symmetric coupling in this case amounts to picking the permutation of the target that has the smallest Euclidean distance squared to the noise sample. Only two permutations are relevant, because there are only two possible solutions; the identity permutation and the one that swaps node 0 with 2 and node 1 with 3.

Buckling: To demonstrate the capability of flow matching to capture bifurcations with multiple solutions in a more realistic scenario, we test it on a buckling beam problem in 2D. We use an approach similar to [6], which uses a random walk prior, an EGNN [34] for interactions between nodes and a UNet for the time evolution. The symmetric coupling in this case consists of picking the closest reflection of the target trajectory. The results in Table 1 show that, as expected, symmetric coupling improves the results. See Figure 1(b,c) for an example of the predictions of a trained flow matching model compared to the two possible ground truths.

Allen-Cahn: Finally, we test flow matching on the 1D Allen-Cahn equation with periodic boundary conditions, which shows complex bifurcation behavior, and in which the number of solutions can quickly become infinite, including (but not limited to) all possible circular shifts of a solution. The

Table 1: Performance of various approaches for the different test systems. All metrics except for the Allen-Cahn are the Wasserstein distance of a set of 100 predictions to the actual distribution of allowed outcomes for each test data point, averaged over all test data points. For the Allen–Cahn system, we use the residual of the governing equation as a performance metric, acknowledging its limitations due to sensitivity to small fluctuations in the prediction, particularly in the Laplacian term.

Test System	Non-prob.	VAE	FM	FM w/ symm. coupling
Two Delta Peaks	1.0*	0.25	**0.091**	-
Heads or Tails	56.2	33.0	**8.30**	-
Three Roads	21.4	17.3	**3.88**	-
Four Node Graph	10.0	9.89	2.02	**1.19**
Buckling Beam	-	-	43.2	**14.8**
Allen-Cahn	-	-	255	**244**

*Theoretical result, a non-probabilistic model would always predict the mean 0.

results in Table 1 show that using symmetric coupling to find the closest circular shift (determined using FFT cross-correlation), as well as the best reflection and sign flip, improves the results. The trained model additionally manages to recreate the pitchfork bifurcation diagram as the parameter μ is varied, see the appendix C for more detailed results of this system.

4 Conclusions

We introduced a probabilistic, equivariant flow matching framework to address the challenge of modeling multiple coexisting solutions in symmetry-breaking bifurcation problems. By leveraging generative modeling and symmetric coupling, our approach captures multimodal output distributions while respecting system symmetries. Through experiments on both abstract and physically grounded systems, we demonstrated that our method outperforms traditional and variational models in accurately representing bifurcation behavior. This work opens new avenues for integrating symmetry-aware generative models into the analysis of complex dynamical systems.

Acknowledgments and Disclosure of Funding

This project has received funding from the Eindhoven Artificial Intelligence Institute (EAISI). MD's work was co-funded by the European Union under the project Robotics and Advanced Industrial Production (reg. no. CZ.02.01.01/00/22_008/0004590).

References

[1] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[2] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual review of fluid mechanics*, 52(1):477–508, 2020.

[3] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International conference on learning representations*, 2020.

[4] Fleur Hendriks, Vlado Menkovski, Martin Doškář, Marc GD Geers, and Ondřej Rokoš. Similarity equivariant graph neural networks for homogenization of metamaterials. *Computer Methods in Applied Mechanics and Engineering*, 439:117867, 2025.

[5] Koen Minartz, Fleur Hendriks, Simon Martinus Koop, Alessandro Corbetta, and Vlado Menkovski. Discovering interaction mechanisms in crowds via deep generative surrogate experiments. *Scientific Reports*, 15(1):10385, 2025.

[6] Kiet Bennema ten Brinke, Koen Minartz, and Vlado Menkovski. Flow matching for geometric trajectory simulation. *arXiv preprint arXiv:2505.18647*, 2025.

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

[8] David I Armstrong McKay, Arie Staal, Jesse F Abrams, Ricarda Winkelmann, Boris Sakschewski, Sina Loriani, Ingo Fetzer, Sarah E Cornell, Johan Rockström, and Timothy M Lenton. Exceeding 1.5 c global warming could trigger multiple climate tipping points. *Science*, 377(6611):eabn7950, 2022.

[9] Simon Willcock, Gregory S Cooper, John Addy, and John A Dearing. Earlier collapse of anthropocene ecosystems driven by multiple faster and noisier drivers. *Nature Sustainability*, 6(11):1331–1342, 2023.

[10] 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.

[11] 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.

[12] Mehrshad Sadria and Thomas M Bury. Fatenet: an integration of dynamical systems and deep learning for cell fate prediction. *Bioinformatics*, 40(9):btae525, 2024.

[13] Masakuni Muramatsu, Tunemasa Irie, and Takashi Nagatani. Jamming transition in pedestrian counter flow. *Physica A: Statistical Mechanics and its Applications*, 267(3-4):487–498, 1999.

[14] François Gu, Benjamin Guiselin, Nicolas Bain, Iker Zuriguel, and Denis Bartolo. Emergence of collective oscillations in massive human crowds. *Nature*, 638(8049):112–119, 2025.

[15] William Warren and Trenton Wirth. A bifurcation in visually-guided behavior when following a crowd. *Journal of Vision*, 22(14):4317–4317, 2022.

[16] Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. *Foundations and Trends® in Machine Learning*, 18(4):385–912, 2025.

[17] Hannah Lawrence, Vasco Portilheiro, Yan Zhang, and Sékou-Oumar Kaba. Improving equivariant networks with probabilistic symmetry breaking. *arXiv preprint arXiv:2503.21985*, 2025.

[18] Zongren Zou, Zhicheng Wang, and George Em Karniadakis. Learning and discovering multiple solutions using physics-informed neural networks with random initialization and deep ensemble. *arXiv preprint arXiv:2503.06320*, 2025.

[19] Gianluca Fabiani, Hannes Vandecasteele, Somdatta Goswami, Constantinos Siettos, and Ioannis G Kevrekidis. Enabling local neural operators to perform equation-free system-level analysis. *arXiv preprint arXiv:2505.02308*, 2025.

[20] Alessandro Della Pia, Dimitrios G Patsatzis, Gianluigi Rozza, Lucia Russo, and Constantinos Siettos. Surrogate normal-forms for the numerical bifurcation and stability analysis of navier-stokes flows via machine learning. *arXiv preprint arXiv:2506.21275*, 2025.

[21] YuQing Xie. *Equivariant symmetry breaking sets*. Massachusetts Institute of Technology, 2024.

[22] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.

[23] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.

[24] Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, Carl-Johann Simon-Gabriel, and Bernhard Schoelkopf. From optimal transport to generative modeling: the vegan cookbook. *arXiv preprint arXiv:1705.07642*, 2017.

[25] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016.

[26] Antoine Salmona, Valentin De Bortoli, Julie Delon, and Agnes Desolneux. Can push-forward generative models fit multimodal distributions? *Advances in Neural Information Processing Systems*, 35:10766–10779, 2022.

[27] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[28] 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. *arXiv preprint arXiv:2302.00482*, 2023.

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

[30] Yuxuan Song, Jingjing Gong, Minkai Xu, Ziyao Cao, Yanyan Lan, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Equivariant flow matching with hybrid probability transport for 3d molecule generation. *Advances in Neural Information Processing Systems*, 36:549–568, 2023.

[31] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pages 15546–15566. PMLR, 2023.

[32] Sékou-Oumar Kaba and Siamak Ravanbakhsh. Symmetry breaking and equivariant neural networks. *arXiv preprint arXiv:2312.09016*, 2023.

[33] Koen Minartz, Yoeri Poels, Simon Koop, and Vlado Menkovski. Equivariant neural simulators for stochastic spatiotemporal dynamics. *Advances in Neural Information Processing Systems*, 36:38930–38957, 2023.

[34] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

A Datasets

Gaussian to 2 Dirac Deltas: This toy problem is intended to demonstrate the ability of flow matching to capture multi-modal distributions with very sharp peaks. There is no data set size, because during training, we simply keep sampling new data points. The input x_i is sampled from a Gaussian distribution $\mathcal{N}(0,1)$; the output y_i is sampled from $\{-1,1\}$ with equal probability, or equivalently from $p(y_i) = 0.5\delta(y_i - 1) + 0.5\delta(y_i + 1)$.

Coin Flip: To model a coin flip, we take as input the amount of money one bets (positive for heads, negative for tails), and then predict the winnings as an output. This means the outcome is either the same as the input (winning the bet amount) or the negative of the input (losing the bet amount). Our dataset consists of 1000 inputs x_i sampled uniformly from the range $[-100, 100]$, with corresponding outputs is $y_i \in \{x_i, -x_i\}$ with equal probability. 800 of these data points are used for training, 200 for testing.

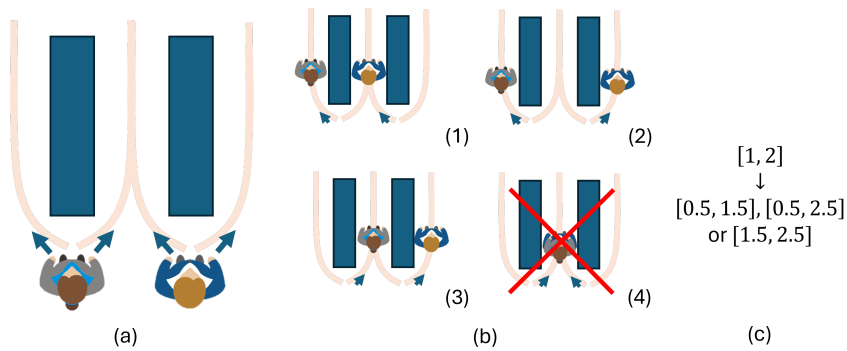

Figure 2: Illustration of the 3 roads problem. (a) Input, (b) possible outputs, (c) what that looks like in the actual data.

3 Roads: This test system models bifurcation in a situation where there are two entities (can be nodes in a graph or something else) that need to coordinate. One can imagine that these are two people that are in a store with very narrow aisles. Both of them are standing right in front of an obstacle separating the aisles, and want to avoid this obstacle by either swerving left or right. However, they don't want to bump into each other, so if one of them chooses the middle aisle (e.g., going left for the right person, going right for the left person), the other will not pick the same aisle. See Figure 2 for an illustration. This is important in prediction of, e.g., pedestrian dynamics, where 1) people coordinate to avoid bumping into each other, and 2) the distributions of their possible paths are highly multimodal when obstacles are involved; there are multiple ways to go around an obstacle, but none to go through it [5; 6]. We choose to model this as a very simple system with two input features and two output features. The input and output features $[x_i^{(1)}, x_i^{(2)}]$ and $[y_i^{(1)}, y_i^{(2)}]$ are two numbers that indicate the horizontal position of both entities before and after deciding their path, respectively. The possible values of $[y_i^{(1)}, y_i^{(2)}]$ are

$$[y_i^{(1)}, y_i^{(2)}] \in \begin{cases} [x_i^{(1)} - d/2, x_i^{(2)} + d/2] \\ [x_i^{(1)} - d/2, x_i^{(2)} - d/2] \\ [x_i^{(1)} + d/2, x_i^{(2)} + d/2] \end{cases} \quad (5)$$

with $d = x_i^{(2)} - x_i^{(1)}$.

For example, for the input $[x_i^{(1)}, x_i^{(2)}] = [1.0, 2.0]$, the output could be $[0.5, 2.5]$, $[0.5, 1.5]$, or $[1.5, 2.5]$. In this test system, we have chosen to consider all these possibilities equally likely. Our dataset consists of 2000 data points, with $[x_i^{(1)}, x_i^{(2)}]$ sampled uniformly from the range $[-50, 50]$. Of the 2000 data points, 1600 are used for training and 400 for testing.

4 Node Graph: For this problem, the input is a graph consisting of 4 nodes, connected in a square. Each node has the same value $x_i \in [-100, 100]$ as a node embedding. The output is an embedding for each node which is equal to $x_i \pm 5$, with connected nodes choosing the opposite sign. See Figure 3 for a visualization. This system is equivariant with respect to permutation. Additionally, the input is self-similar under some permutations, whereas each individual output is self-similar under fewer permutations, which means there is symmetry-breaking. Our dataset contains 2000 data points, of which 70% is used for training, 30% for testing. We sample x_i uniformly from the range $[-50, 50]$.

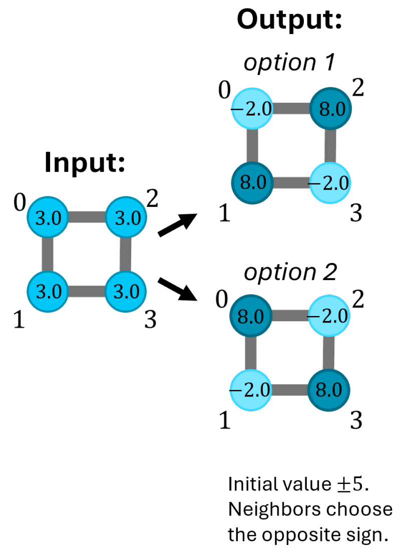

Figure 3: Illustration of the 4 node graph problem.

Buckling Beam: For this problem, we consider a beam that is attached to a base on the bottom, and then prescribe a vertical displacement to the top of the beam, while leaving the horizontal displacement free. For a small displacement, the beam will compress but stay straight, but when the displacement reaches a critical value, the system becomes unstable and the beam will buckle, which will make it bend to either the left or right side, with either option equally likely. We model the beam as a series of n connected segments, which connect $n+1$ nodes, where node 0 is attached to the base. We consider n ranging from 2 to 10. Each segment has a length L_i and a stiffness C_i. At each node except for the top one the beam can bend, and therefore has a rotational stiffness K_i. The input d is the vertical displacement applied to the top node. We solve the deformation of the beam by finding the Hencky strains ϵ_i and angles q_i that minimize the total energy:

$$U(\mathbf{q}, \epsilon) = \frac{1}{2}\left[K_1 q_1^2 + \sum_{i=2}^{n} K_i (q_i - q_{i-1})^2 + \sum_{i=1}^{n} L_i C_i \epsilon_i^2\right] \tag{6}$$

subject to the constraint that the top of the beam is at the right displacement d:

$$\sum_{i=1}^{n} L_i \left[1 - \exp(\epsilon_i)\cos q_i\right] = d. \tag{7}$$

We solve this using Lagrange multipliers, combined with stability analysis. The solution is perturbed using the lowest eigenmode of the stiffness matrix to break symmetry when the system becomes unstable. In total, we generate 1000 beams. We sample n from the integers 2 to 11, and sample L_i, C_i and K_i from the log-uniform distribution between 0.5 and 2. For each beam we solve for its trajectory as d is increased from zero. We choose the range of d such that at the end of a trajectory the top node is at the ground, and we divide this range into 200 time steps. 70% of the resulting trajectories are used for training, 30% for testing.

Allen-Cahn: The Allen-Cahn equation is a reaction-diffusion equation and describes the time evolution of a binary mixture of phases, e.g., ice and water at zero degrees Celsius. It is given by

$$\frac{\partial u}{\partial t} = \epsilon^2 \frac{\partial^2 u}{\partial x^2} - (u^3 - \mu u), \tag{8}$$

where u is the phase field variable, representing the concentration of one of the phases. ϵ is parameter that controls size the of the interfaces between phases, and $u^3 - \mu u$ is the derivative of a double-well potential with minima at $\pm\sqrt{\mu}$ if $\mu > 0$, for $\mu \leq 0$ there is only one minimum at 0. We solve this equation on a 1D domain of length 1.0 with periodic boundary conditions, from $t = 0$ to $t = 100$, using a finite difference scheme, with 200 spatial points and a time step of 0.1. Each trajectory starts from $u = 0$ homogeneously, however, because this is an unstable solution when $\mu > 0$, we add a small random perturbation sampled from $\mathcal{N}(0, 0.001)$ to each spatial point. We generate 400 trajectories in total, of which 300 are used for training and 100 for testing. For each trajectory, ϵ is randomly sampled from a loguniform distribution between 0.001 and 0.1., and μ from $\mathcal{U}(-0.1, 1.0)$.

B Architectures & Training

All models were trained with the Adam optimizer.

Gaussian to 2 Dirac Deltas: For the VAE, we used an MLP with 2 hidden layers of 64 neurons per layer as an encoder and another one as a decoder, with a latent dimension of 16. For the flow model, we used an MLP with 3 hidden layers, 64 neurons per layer. Both Flow and VAE were trained for 10000 epochs with a batch size of 256. For the flow model we used a learning rate of 10^{-3} and for the VAE a learning rate of 10^{-2}.

Coin Flip: As a non-probabilistic model we used an MLP with 2 hidden layers, with each 32 neurons per layer. It was trained for 52 epochs (determined by early stopping), with a batch size of 64 and a learning rate of 10^{-4}. For the VAE, we used an MLP with 2 hidden layers of 32 neurons as an encoder and a decoder, and a latent dimension of 16. We use another similar MLP to condition on the betted amount. The VAE was trained for 1000 epochs with a learning rate of 10^{-3}. For the flow model, we used a similar MLP to predict the flow field, and as a prior we used Gaussian noise with standard deviation 1.0.

3 Roads: We used a set model on the 3 roads dataset, which applies MLPs on each element separately and each ordered pair of elements, aggregated in such a way to respect the permutation equivariance. As a prior we used Gaussian noise with standard deviation 1.0.

4 Node Graph: As a non-probabilistic model, we used a message-passing graph neural network with 3 message-passing layers. For the VAE, we used such a message-passing GNN as encoder and another as decoder, and another to condition on the input graph. For the flow matching, we used a similar message-passing GNN. As a prior, we used the input value plus Gaussian noise with standard deviation 1.0.

Buckling Beam: For the buckling beam, we used a random walk prior similar to the approach from Brinke et al. [6]. We used an EGNN [34] to predict interactions between nodes and a UNet for the time evolution.

Allen-Cahn: We used a random walk prior, which consists of cumulative, scaled random Gaussian noise. We then use a 2D UNet to predict the flow field, which maps an entire random walk trajectory to the entire final trajectory. It is periodic in only the spatial dimension. The UNet has 4 downsampling steps and 32 channels.

C Additional Results

Gaussian to 2 Dirac Deltas: See Figure 4 for a comparison of the predicted distributions from the VAE and the flow model, as well as a visualization of the flow field.

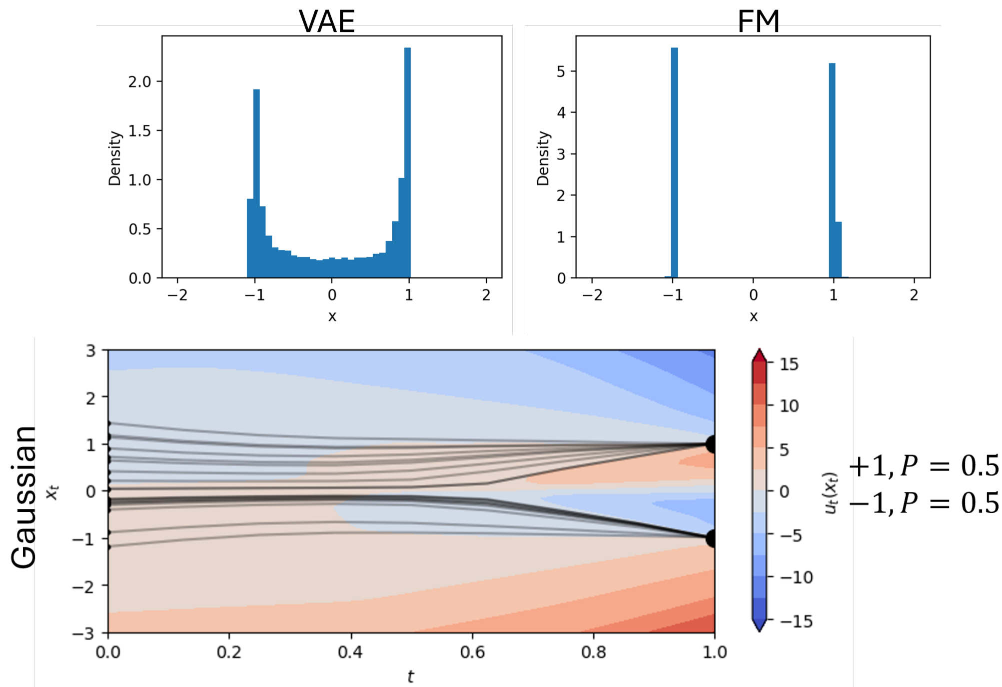

Figure 4: Top: the results of predicting a probability distribution consisting of two Dirac deltas, using a VAE and flow matching. Bottom: the learned flow field by the trained flow matching model (contour plot), showing how samples from the Gaussian prior are pushed towards the two delta peaks (black lines).

Coin flip: See Figure 5 a comparison between predictions from a deterministic model, a VAE and a flow matching model.

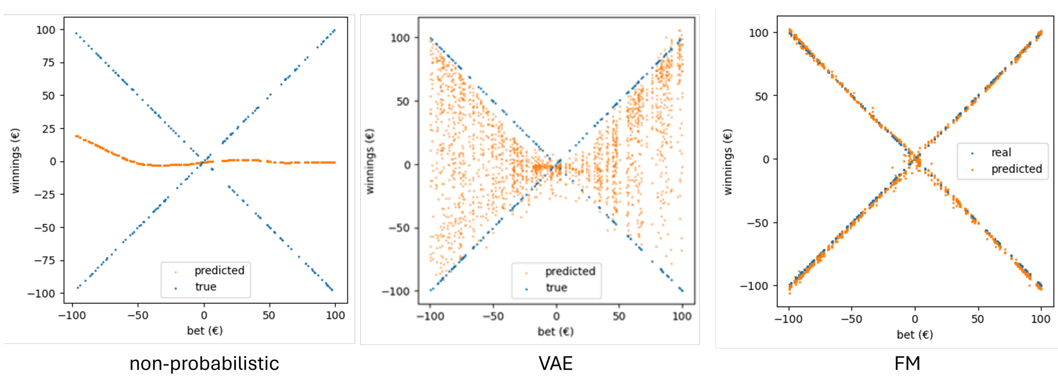

Figure 5: The results of predicting outcomes of a coin flip using three different methods: (a) a regular neural network, (b) a conditional VAE, and (c) flow matching. Each method makes 10 predictions per test data point.

Allen-Cahn: See Figure 6 for several predictions by the trained flow model, and Figure 7 for a bifurcation diagram recreated by the trained model.

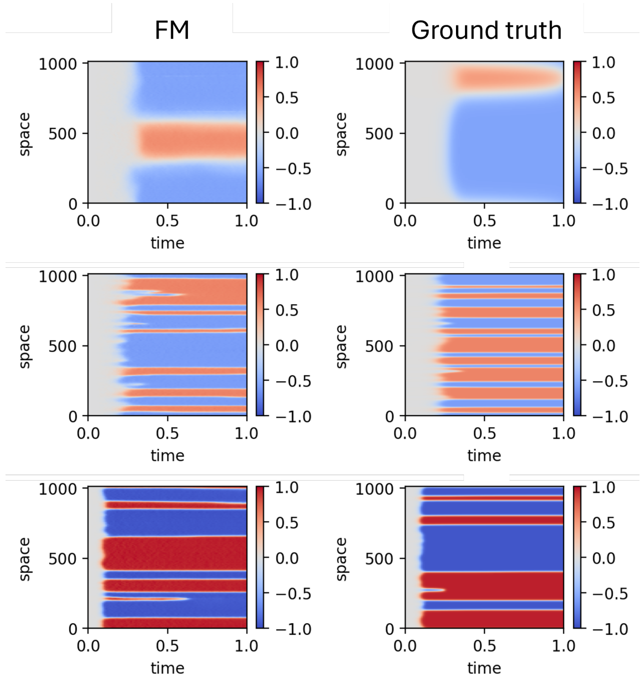

Figure 6: Comparison of predicted Allen-Cahn trajectories by the flow model with symmetric coupling with examples of possible ground truth trajectories.

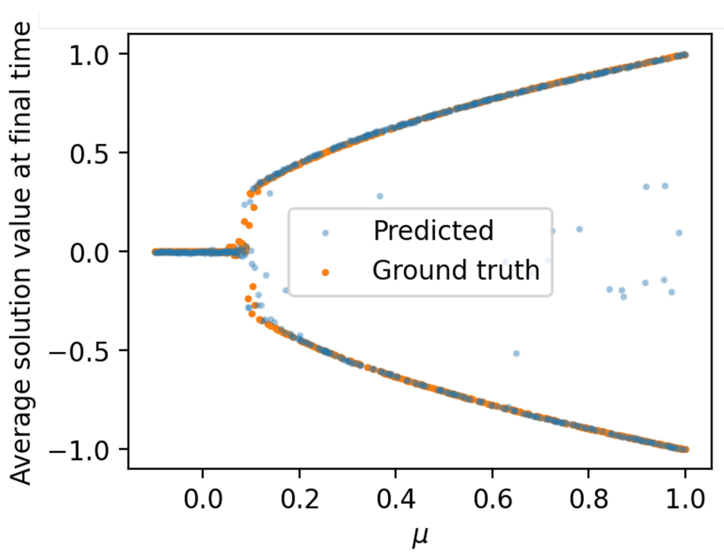

Figure 7: Bifurcation diagram of the Allen-Cahn equation as the parameter μ is varied, keeping ϵ constant at 0.1. The trained model is able to model the pitchfork bifurcation.