# RESOLVING HOMOTOPY OBSTRUCTIONS IN NORMALIZING FLOWS VIA CONTRACTIBLE COVERS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Normalizing Flows are a powerful class of likelihood-based generative models that map complex target distributions into simple, tractable base distributions, usually Gaussians. As differentiably invertible functions trained via likelihood maximization, they enable exact density evaluation and efficient sampling. However, they also suffer from the fundamental topological rigidity of homeomorphic transformations: they cannot alter the topology, and consequently homotopy, of the space they transform without severely increasing their bi-Lipschitz constant. **Co**ntractible**Co**ver**Flow** (**CoCoFlow**) resolves these homotopy obstructions by partitioning the target support into a data-driven **contractible cover** and then training a posterior-regularized **mixture of Normalizing Flows** confined to the prescribed cover elements, which are guaranteed to be homotopically compatible with the Gaussian base by construction. The mixture model preserves exact likelihood evaluation, deterministic maximum likelihood training, and efficient sampling of standard Normalizing Flows. Experiments on synthetic and real-world data distributions demonstrate improved or competitive density estimation with increased continuity of the generative map compared to standard baselines. To our knowledge, CoCoFlow is the first mixture-of-normalizing-flows framework to provide per-component contractibility guarantees through a topology-aware, data-driven partition of the target support, offering a principled resolution to a long-standing structural limitation of the field.

## 1 INTRODUCTION

Normalizing Flows (NFs) are a well-established class of likelihood-based generative models that learn an invertible transformation mapping a complex target data distribution to a tractable base distribution, typically Gaussian (Tabak and Vanden-Eijnden, 2010; Rezende and Mohamed, 2016; Dinh et al., 2015). This formulation yields exact density evaluation, deterministic training objectives, and efficient sampling, making NFs a powerful tool with versatile applications, including density estimation (Dinh et al., 2017; Papamakarios et al., 2017; Durkan et al., 2019), Boltzmann distribution approximation (Noé et al., 2019), and variational inference (Rezende and Mohamed, 2016). These properties depend on the learned map being differentiably invertible and consequently a **homeomorphism**. However, the same requirement imposes a rigid topological constraint: the flow cannot alter the topology of the space it transforms. When the target distribution exhibits non-trivial **homotopy** features (loops, voids, or disconnected components) absent from the contractible Gaussian base, the learned map is forced to approximate a discontinuous function, resulting in large bi-Lipschitz constants (Cornish et al., 2020; Behrmann et al., 2021). This obstruction is an intrinsic topological constraint and cannot be overcome by adopting a more expressive parameterization or architecture, such as autoregressive (Papamakarios et al., 2017) or spline-based (Durkan et al., 2019) flows.

Although the base–target topological mismatch problem is widely acknowledged (Cornish et al., 2020; Stimper et al., 2022; Bevins and Handley, 2023), existing approaches fail to resolve it without incurring significant compromises. Riemannian NFs (Gemici et al., 2016; Mathieu and Nickel, 2020) operate in a setting where the manifold structure is assumed known, and so do not offer a data-driven solution to the general case. Continuously Indexed Flows (Cornish et al., 2020) relax the bijectivity constraint through latent-variable augmentation, sacrificing exact likelihood evaluation. Resampled base distributions (Stimper et al., 2022) retain a bijective flow but replace the simple prior with a learned rejection-sampling scheme, introducing substantial additional sampling machinery and



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.

abandoning the tractable base distribution. Closer to our work, mixture-based methods enrich the model with multiple components and impose an explicit partition in advance (Dinh et al., 2019; Bevins and Handley, 2023) or allow the partition of the support to emerge implicitly during training (Izmailov et al., 2019; Pires and Figueiredo, 2020; Yu et al., 2026). In all cases, the partitioning criterion is purely geometric, based on implicitly learned maps, classifiers, or spatial clustering such as $k$-means. Without explicit accounting for topology, such geometry based approaches neither guarantee that individual components are topologically simpler than the original space, nor that homotopy-theoretic obstructions are resolved at the component level.

To address this gap we introduce **Co**ntractible **Co**ver **Flow** (**CoCoFlow**), a three-step framework grounded in topological data analysis (TDA). Given a finite sample from the target distribution, we: (i) approximate the local geometry of its support via a **geometric graph**; (ii) construct a **contractible cover** of its support from this graph; and (iii) learn a posterior-regularized **mixture of Normalizing Flows**, where each component is regularized to be confined within the corresponding contractible cover element of the target distribution. Our main contributions are as follows:

**CoCoFlow: Summary of Contributions**

- **Contractible cover construction.** We introduce an algorithm that partitions the data support into contractible regions providing the first data-driven per-component contractibility guarantees without assuming known manifold structure.
- **Contractible posterior regularization.** We use posterior regularization to confine each component flow to a topologically simple subdomain, structurally eliminating homotopy obstructions for each component of the mixture model.
- **Topological mismatch resolution without compromise.** CoCoFlow resolves base-target homotopy mismatch at the component level without relaxing invertibility, augmenting the latent space, or replacing the tractable base distribution. The model preserves exact likelihood evaluation, deterministic MLE training, and efficient sampling of standard Normalizing Flows.

## 2 Preliminaries

**Normalizing Flows** approximate complex data distributions $P_X$ as pushforwards of a simple, typically Gaussian, base distribution $P_Z$ with density $p_Z$ through learned nonlinear, invertible, and differentiable transformations $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$. When the Jacobian determinant $\det J_{f_\theta}$ is computationally tractable, the optimal parameters $\theta^*$ can be learned via direct log-likelihood maximization, $\theta^* := \arg\max_\theta \mathbb{E}_{X\sim P_X}[\log p_\theta(X)]$, via gradient-based optimization together with the change-of-variables formula

$$p_\theta(x) = p_Z(f_\theta^{-1}(x)) \left|\det J_{f_\theta}(f_\theta^{-1}(x))\right|^{-1}.$$

However, Cornish et al. (2020) showed that when the support of $P_X$ is not homeomorphic to the support of $P_Z$, weak convergence of the pushforward $f_\theta \# P_Z \xrightarrow{D} P_X$ forces bi-Lipschitz constants of $f_\theta$ to diverge to infinity. More generally, parametrizing topologically non-trivial manifolds via homeomorphisms from $\mathbb{R}^d$ requires multiple charts (Yu et al., 2026).

**Mixture models** are convex combinations $p_\theta(x) = \sum_{i=1}^m p_\theta(C = i)\, p_\theta(x \mid C = i)$, where $C$ is a latent component variable with prior $p_\theta(C)$, and each conditional $p_\theta(x \mid C = i)$ is a component density. Given a finite sample $\mathcal{D} \sim P_X$, let $\hat{p}_X := \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}} \delta_x$ denote the empirical data distribution, and let $\pi_i(x) := \hat{p}_{C|X}(i \mid x)$ denote the empirical posterior over components, inducing the joint $\hat{p}_{X,C}(x,i) := \hat{p}_X(x)\pi_i(x)$ and empirical marginal $\hat{p}_C(i) := \sum_{x\in\mathcal{D}} \hat{p}_X(x)\pi_i(x)$. By the data-processing inequality for KL divergence under marginalization and the chain rule, assuming the model prior matches the empirical marginal $p_\theta(C) = \hat{p}_C(C)$,

$$\mathrm{KL}(\hat{p}_X \| p_\theta(X)) \le \mathrm{KL}(\hat{p}_{X,C} \| p_\theta(X, C)) = \sum_{i=1}^m \hat{p}_C(i)\, \mathrm{KL}\big(\hat{p}_{X|C=i} \,\|\, p_\theta(X \mid C = i)\big).$$

The global approximation error is thus controlled by the per-component approximation of $\hat{p}_{X|C=i}$. The results of Cornish et al. (2020) motivate seeking a component assignment $C|X$ such that each target distribution $\hat{p}_{X|C=i}$ has support homeomorphic to $\mathbb{R}^d$. Since likelihood maximization alone

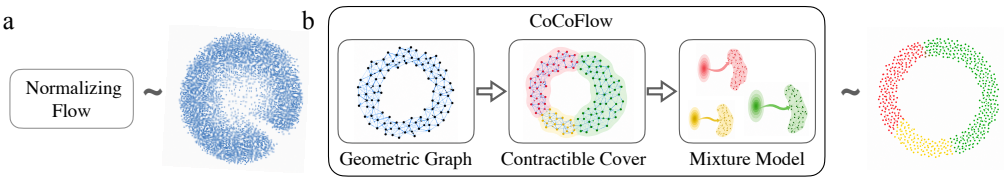


Figure 1: a. Standard Normalizing Flows map a contractible Gaussian base to the target distribution and distort non-contractible topology. b. CoCoFlow approximates the target support with a geometric graph, constructs a contractible cover, and trains a posterior-regularized mixture of flows that preserves the target loop structure during sampling.

does not reward such structure, we construct the assignment explicitly, prior to training, through a topology-aware partition of the support. However, since determining homeomorphic equivalence in general is undecidable (Friedl et al., 2026; Markov, 1958), we adopt a computationally practical relaxation and consider equivalence only up to homotopy.

**Homotopy** induces a broad class of topological invariants that capture properties preserved under continuous deformations. Formally, homotopy is an equivalence relation $\sim$ between continuous maps $f_0, f_1 : X \to Y$ such that $f_0 \sim f_1$ if and only if there exists a continuous deformation $H : X \times [0,1] \to Y$ satisfying $H(x,0) = f_0(x)$ and $H(x,1) = f_1(x)$ for all $x \in X$. This induces a notion of equivalence between spaces: two topological spaces $X$ and $Y$ are homotopy equivalent, denoted $X \simeq Y$, if there exist continuous maps $f : X \to Y$ and $g : Y \to X$ such that $g \circ f \sim \mathrm{id}_X$ and $f \circ g \sim \mathrm{id}_Y$. Homotopy equivalent spaces share a wide range of invariants, including path connectedness, fundamental groups, and all homology and cohomology groups. We highlight **contractible** spaces which are homotopy equivalent to a point ($X \simeq \{*\}$) and thus homotopy trivial. Examples include Euclidean spaces $\mathbb{R}^d$ and closed balls $\mathbb{D}^d$. In contrast, tori $\mathbb{T}^d$ and spheres $\mathbb{S}^d$ are not contractible.

## 3 Methods

Given a finite sample from the target distribution $\mathcal{D} \sim P_X$, we approximate the topology of its support via a set of landmark points $V \subseteq \mathcal{D}$, on which we build a symmetrized $k$-nearest neighbor graph $G = (V, E)$ and interpret its clique complex $\mathcal{K}(G)$ as a combinatorial proxy for $\mathrm{supp}(P_X)$. Using a novel algorithm, we construct a contractible cover $\mathcal{C}_G \coloneqq \{R_i \subseteq V\}_{i=1}^m$ of the landmark set, which is then lifted to the entire dataset $\mathcal{C}_\mathcal{D} \coloneqq \{U_i \subseteq \mathcal{D}\}_{i=1}^m$. We learn $P_X$ with a mixture of Normalizing Flows $p_\theta(x) = \sum_{i=1}^m p_\theta(C = i)\, p_\theta(x \mid C = i)$, where each component $p_\theta(x \mid C = i)$ is regularized to operate within $U_i$.

### 3.1 Clique Complex Approximation of the Support

To focus purely on the topology of the support $\mathrm{supp}(P_X)$ and leave density estimation to the flow model, we subsample a uniformly spaced set of landmark points with Farthest Point Sampling (FPS) (Gonzalez, 1985). We start from a point $x_1$ furthest along the first principal component of $\mathcal{D}$ and iteratively construct a sequence of points $\{x_i\}_{i \geq 1}$ that maximize the distance to the set of all previously selected points $x_i \coloneqq \arg\max_{x \in \mathcal{D}} d(x, \{x_j\}_{j<i})$. The set of landmarks $V$ is defined as the sequence prefix $\{x_i\}_{i \leq n}$. The size of the landmark set is treated as a hyperparameter to be selected according to the desired level of geometric fidelity.

Following the manifold learning literature (Belkin and Niyogi, 2003; McInnes et al., 2020), we approximate the local geometry of the set $V$ via a symmetric $k$NN graph $G = (V, E)$, where two points $u, v \in V$ are connected by an edge $(u, v) \in E$ if $u$ is among the $k$ nearest neighbors of $v$, or vice versa. To approximate the higher-dimensional geometry induced by $G$, we consider its clique complex (Edelsbrunner and Harer, 2010b) $\mathcal{K}(G)$, the set of all cliques (complete subgraphs) of $G$, where each clique of size $d'$ defines a $(d'-1)$-simplex given by the affine span of its vertices.

### 3.2 Constructing Contractible Covers

Given the $k$NN graph $G = (V, E)$, with $|V| = n$, we introduce an algorithm to construct a **contractible cover** $\mathcal{C}_G \coloneqq \{R_i \subseteq V\}_{i=1}^m$ of the vertex set $V$ according to the topology of the clique

complex $\mathcal{K}(G)$. Formally, the vertex subset $R_i$ is considered to be contractible if its induced subgraph clique complex $\mathcal{K}(G[R_i])$ is contractible. We call contractible vertex subsets $R_i$ *regions*. The central part of the algorithm is the **link-dismantlable expansion** step, which starting from single element root region greedily expands it while maintaining contractibility invariant. The cover is constructed via concurrent link-dismantlable expansion from multiple roots. To minimize the overlap between the elements of the cover we perform **star-subsumed retraction**, a topology-aware vertex pruning which preserves the contractibility property of each region. Finally, we lift the graph cover $\mathcal{C}_G$ to the full dataset cover $\mathcal{C}_\mathcal{D} := \{U_i \subseteq \mathcal{D}\}_{i=1}^m$ via nearest-sets projection which preserves the contractibility of regions under suitable topology modeling assumption.

**Link-dismantlable expansion.** We denote the neighborhood of $v$ in $G$ as $N[v] := \{v\} \cup \{u \in V \mid (u, v) \in E\}$. We say that graph $H$ is *dismantlable* if its vertices can be eliminated one at a time so that each removed vertex $u$ is *dominated* by some neighbor $w$ in the current subgraph, i.e. $N[u] \subseteq N[w]$ (Nowakowski and Winkler, 1983). We define *star* of $v$ restricted to $R$ as $\mathrm{St}_R(v) := \mathcal{K}(G[N[v] \cap R \cup \{v\}])$, and the *link* of $v$ restricted to $R$ as $\mathrm{Lk}_R(v) := \mathcal{K}(G[N[v] \cap R \setminus \{v\}])$.

Given any root point $r \in V$ we expand a contractible region $R \subset V$ by greedily adding its neighboring vertex $v \in V \setminus R$ (and thus adding star $\mathrm{St}_R(v)$ to clique complex $\mathcal{K}(G[R])$) only when graph of $\mathrm{Lk}_R(v)$ is *dismantlable*, a procedure visualized in Figure 2. In practice, we traverse neighbors of added vertices in priority based on minimal distance from root $r$ and visit each point at most once. We denote the maximum vertex degree in graph $G$ as $\Delta$. Note that for symmetric $k$NN graphs and uniformly sampled points $\Delta \approx k$. The dismantlability of each link can be checked in $O(\Delta^3)$ time using an efficient implementation operating on bit-mask representations of the adjacency matrix. The total computational complexity of expansion from a single root $r$ is $O(n\Delta^3)$, which is dependent on dimension $d$ only indirectly through $k$ used in $k$NN graph construction, selected to model the appropriate topology. Our algorithm can be used to construct contractible sets that respect high dimensional topology. This is in stark contrast to other TDA algorithms, which computational complexity typically scales exponentially with respect to dimension $d$. For example, computing Betti numbers of a clique complex is #P hard while multiplicative approximation is NP-hard (Schmidhuber and Lloyd, 2023). Detailed description of our algorithm is provided in the Algorithm 1. We prove by induction that $\mathcal{K}(G[R])$ is contractible throughout the execution of the algorithm with help of the following well-known theorems.

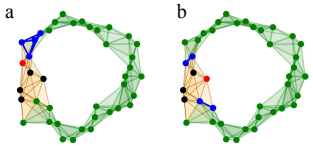

Figure 2: a. Vertex (red) can be added to contractible region (green) because its link (blue) is dismantlable. b. The vertex cannot be added.

**Proposition 1.** *Dismantlable graph $H \implies$ Contractible clique complex $\mathcal{K}(H)$ (Barmak, 2011).*

**Proposition 2** (Corollary of Nerve Lemma (Hatcher, 2002))**.** *If $A$ and $B$ are subcomplexes of a simplicial complex such that $A$, $B$, and $A \cap B$ are all contractible, then $A \cup B$ is contractible.*

Initially $R = \{r\}$ which is trivially contractible. Suppose that $v$ is a vertex accepted by the algorithm. Then the graph of $\mathrm{Lk}_R(v)$ is dismantlable therefore by Proposition 1 the link $\mathrm{Lk}_R(v)$ is contractible. By induction assumption $\mathcal{K}(G[R])$ is contractible. The star $\mathrm{St}_R(v)$ is a cone with apex $v$ and thus contractible by construction. Since $\mathcal{K}(G[R]) \cap \mathrm{St}_R(v) = \mathrm{Lk}_R(v)$, by Proposition 2 the union $\mathcal{K}(G[R]) \cup \mathrm{St}_R(v) = \mathcal{K}(G[R \cup \{v\}])$ is contractible.

**Contractible cover.** To construct a contractible cover $\mathcal{C}_G$ we decompose $G$ into connected components and find a cover for each component independently. To cover a non-contractible component, we must find a set of regions expanded from more than one root. Although independent link-dismantlable expansion from all roots would yield a correct contractible cover, the resulting regions would have complex geometry and intersections, which is unsuitable for our downstream applications that use the contractible cover for posterior regularization. We therefore introduce a **concurrent version** of link-dismantlable expansion, where we consider a set of roots $\{r_1, r_2, \ldots, r_m\}$ whose subsequent neighbors are processed based on a joint priority queue ordered by distance from the parent root. We add a stopping criterion for root expansion: when $v$ is added to region $R_i$, it can add its neighbors to the queue only if it is not an element of some other region $R_j$ for which $2d(v, r_j) \leq d(v, r_i)$. The stopping criterion allows for moderate intersection between regions but limits extreme cases. Each vertex can be visited at most $m$ times, resulting in $O(mn\Delta^3)$ computational complexity of the concurrent version of link-dismantlable expansion. Detailed description of the procedure is provided in Algorithm 2. To find contractible covers for the connected components of $G$ using as few starting

roots as possible, we incrementally increase the number of roots $m$ for each component, selecting candidate roots at each step via FPS sampling, as in the landmark selection step. We terminate at the smallest $m$ for which concurrent link-dismantlable expansion covers the entire component. The total computational complexity of finding a contractible cover with $m$ regions is therefore $O(m^2 n \Delta^3)$.

**Star subsumed retraction.** Given the cover $\mathcal{C}_G = \{R_i\}_{i=1}^m$, we reduce region intersections by backtracking the inductive chain of Algorithm 1 for each region. Recall that Algorithm 1 produces, for each accepted vertex $v \in R_i$, an explicit record of the link $\mathrm{Lk}_{R_{i,v}}(v)$ used to certify its addition, where $R_{i,v}$ is the state of the region at the time $v$ was added. This yields a natural dependency directed acyclic graph $\mathcal{T}_i$ on the vertices of $R_i$: vertex $v$ is a child of each $u \in \mathrm{Lk}_{R_{i,v}}(v)$, since those vertices were required to already reside in $R_{i,v}$ at the moment $v$ was accepted. The root of $\mathcal{T}_i$ is the seed vertex of $R_i$, interior nodes are vertices whose presence enabled the admission of at least one later vertex, and leaves are vertices that were never used to certify any subsequent admission. A leaf $v$ may be safely removed without breaking contractibility because its removal leaves the inductive chain intact. Moreover, if the insertion-time star $\mathrm{St}_{R_{i,v}}(v)$ was fully subsumed by some other region $R_j$, the cover property is preserved. The retraction iterates over regions in decreasing order of size, greedily removing leaf vertices whose insertion-time stars are subsumed by another region. Each removal prunes the corresponding leaf from $\mathcal{T}_i$, potentially exposing new leaves among its parents. This process repeats until no further removals are possible across all regions.

**Lifting the contractible cover.** Given $G = (V, E)$ and its cover $\mathcal{C}_G = \{R_i\}_{i=1}^m$, we lift the construction to a cover $\mathcal{C}_\mathcal{D} = \{U_i\}_{i=1}^m$ of the full dataset $\mathcal{D}$ by assigning each $x \in \mathcal{D}$ to all nearest regions $U_i := \{x \in \mathcal{D} : i \in \arg\min_j d(x, R_j)\}$. This reflects a modeling assumption that $G$ represents the topology of the distribution support given uniformly spaced subsample and $G_\mathcal{D} = (\mathcal{D}, E \cup \{(u, v^*) : u \in \mathcal{D}, v^* = \arg\min_{v \in V} d(u, v)\})$, represents additional local noise around each landmark point. The nearest set projection is naturally compatible with the topology of $\mathcal{K}(G_\mathcal{D})$, for which it deformation retracts each $\mathcal{K}(G_\mathcal{D}[U_i])$ onto $\mathcal{K}(G[R_i])$ and thus each set $U_i$ in contractible with respect to $\mathcal{K}(G_\mathcal{D})$.

### 3.3 Posterior Regularized Mixture of Normalizing Flows

Given a contractible cover $\mathcal{C}_\mathcal{D} = \{U_i\}_{i=1}^m$ of dataset $\mathcal{D}$, we model the target distribution $P_X$ as a mixture $p_\theta(x) = \sum_{i=1}^m p_\theta(C = i)\, p_\theta(x \mid C = i)$, where each conditional $p_\theta(x \mid C = i)$ is a normalizing flow and $p_\theta(C)$ is a non-parametric estimate from the data.

The model is trained via an EM procedure in which the cover $\mathcal{C}_\mathcal{D}$ constrains the component-membership posteriors throughout training. Concretely, we represent each point $x$ by a membership probability vector $\pi(x) \in \Delta^{m-1}$, whose $i$-th entry $\pi_i(x)$ approximates the posterior $p_\theta(C = i \mid x)$. This vector is initialized by distributing probability mass uniformly over the cover elements $\pi_i^{(0)}(x) \propto \mathbf{1}[x \in U_i]$. In the M-step, given current assignments $\{\pi(x)\}_{x \in \mathcal{D}}$, we first update the component prior as the empirical mean $p_\theta(C = i) = \frac{1}{|\mathcal{D}|} \sum_x \pi_i(x)$, then maximize the **posterior-regularized objective** (Ganchev et al., 2010)

$$\mathcal{L}_\alpha(\theta) := \mathbb{E}_{X \sim P_X} \left[ (1 - \alpha) \log p_\theta(X) + \alpha \sum_{i=1}^m \pi_i(X) \log p_\theta(X \mid C = i) \right],$$

where $\alpha \in [0, 1]$ interpolates between standard mixture MLE ($\alpha = 0$) and a fully assignment-supervised objective ($\alpha = 1$). Throughout the training we anneal $\alpha$ from $0.5$ to $0$. Providing posterior guidance in early stages of the training locks in topology-aware component assignment of the model, which is then relaxed to adjust the behavior at the cover overlaps, where several regions compete for the same points. In the E-step, the membership vectors are assigned probability vectors proportional to the region-confined posterior $\pi_i^{(\text{new})}(x) \propto p_\theta(C = i \mid x) \cdot \mathbf{1}[x \in U_i]$. This ensures that, whenever $\alpha > 0$, in the subsequent M-step the mixture component $p_\theta(x|C = i)$ is penalized for accumulating mass outside its assigned region $U_i$.

During inference, the model is treated as an ordinary mixture model, for which sampling is defined as $i \sim p_\theta(C)$ followed by $x \sim p_\theta(x \mid C = i)$. All additional computational complexity of posterior regularization is attributed to cover construction $\mathcal{C}_\mathcal{D}$ which is done once, prior to the training, and then discarded.

## 4 Related Work

**Normalizing Flows** research traces back to NICE (Dinh et al., 2015) and RealNVP (Dinh et al., 2017). Subsequent work focused on increasing the expressivity of architectures, either by using autoregressive transforms (Papamakarios et al., 2017) or more expressive monotonic spline couplings (Durkan et al., 2019). Recent works showcase that Normalizing Flows are competitive with other generative methods (Zhai et al., 2024; Gu et al., 2025; Ghugare and Eysenbach, 2026). Nevertheless, by design, all normalizing flow models share the same structural premise of a homeomorphism from a contractible Gaussian base, and therefore inherit the topological obstruction that motivates our work.

**Methods resolving topological mismatch** in Normalizing Flows fall into 4 groups: (i) **Known manifold assumption**, in which the topological obstruction problem is side-stepped by applying projection through a chart of the manifold (Gemici et al., 2016), coupling layers tailored to tori and spheres (Rezende et al., 2020), or geometry-aware ODE flows on the manifold (Mathieu and Nickel, 2020) all of which are unavailable in purely data-driven regime. (ii) **Relaxing bijectivity** through introduction of extra continuous latent variables (Cornish et al., 2020) or surjective and stochastic layers (Nielsen et al., 2020) which remove the homotopy obstructions at the cost of closed-form density evaluation. (iii) **Modifying the base distribution** via rejection resampling (Stimper et al., 2022), replacing simple analytic prior by a learned sampling procedure, abandoning the tractability of base distribution. (iv) **Mixture and piecewise flows** that decompose the target into pieces, each modeled by a separate flow. Approaches include training a variational mixture of independent flows with partition defined by posterior network (Pires and Figueiredo, 2020), introducing a discrete latent for indexing a learned piecewise-invertible map (Dinh et al., 2019), learning an atlas via a VQ-autoencoder (Sidheekh et al., 2022), clustering the data and learning independent flows (Bevins and Handley, 2023), and learning a mixture model without any constraint on the posterior (Yu et al., 2026). In every case the target decomposition is geometric and offers no guarantee that the resulting pieces are individually homotopy trivial, whereas components derived from our contractible cover algorithm are provably contractible wrt. the clique complex approximation of the support.

**Topological Data Analysis** methods compute topological invariants from finite samples, typically approximated by a simplicial complex (Carlsson, 2009). Quantities of interest include: (i) Betti numbers, which count the $k$-dimensional holes (Edelsbrunner and Harer, 2010a), (ii) persistent homology, which tracks how homology groups evolve across scale (Edelsbrunner et al., 2003; Zomorodian and Carlsson, 2005), (iii) good covers, which yield a homotopy-equivalent simplicial sparsification via the nerve theorem (Hatcher, 2002; Singh et al., 2007). A central obstacle limiting the use of these invariants in machine learning applications is the computational cost which usually scales exponentially (or worse) with dimension. Our contractible cover algorithm is distinguished by a computational complexity that depends on dimension only through polynomial function of maximum vertex degree.

## 5 Experiments

We first evaluate CoCoFlow on synthetic (Section 5.1) and real-world Earth&Climate Science (Section 5.2) data. We compare CoCoFlow that is regularized with contractible cover and scheduled-$\alpha$ to: (i) $k$-means PNF (Bevins and Handley, 2023) ($k$-means cluster cover, fixed $\alpha = 1$), (ii) $k$-means Sched. (an ablation with scheduled-$\alpha$), (iii) Mixture (Yu et al., 2026) (cover elements are the entire dataset, $\alpha = 0$), (iv) Single component model. We use 2 normalizing flow architectures as backbones for each method: Small Affine (2-layer RealNVP (Dinh et al., 2017)), Big Spline (8-layer, 8-bin Neural Spline Flow (Durkan et al., 2019)) implemented using Zuko (Rozet et al., 2022) library. We train the models with noising (Zhai et al., 2024). See Appendix B for the detailed training recipe.

We measure generative performance using log-likelihood (LL) computed on the test set. Motivated by the theoretical results of Cornish et al. (2020), we additionally report the Lipschitz constant of each generative model, approximated as the 99th percentile of Jacobian norms:

$$\mathrm{Lip}_{.99} = \|\nabla_z f_\theta(z \mid c)\|_{0.99}, \quad c \sim p_\theta(C),\ z \sim \mathcal{N}(0, I).$$

Note that **comparing Lipschitz constants alone can be misleading**, since trivial maps may exhibit low Lipschitz constants while being poor generative models (e.g., the identity map has a Lipschitz constant of 1, and a constant map has a Lipschitz constant of 0). We therefore first compare models

Table 1: Test Log-Likelihood (LL) and 99-th quantile of Jacobian norm in generative direction (Lip.$_{99}$) on synthetic datasets. Mean±std estimated over 3 runs. In each column, we **bold** the best and underline 2nd-best LL score, highlight both rows, and **bold** the better Lip.$_{99}$ score from the two.

| | | Mickey | | Concentric | | Rings | | Cyclooctane | |
|---|---|---|---|---|---|---|---|---|---|
| | | Lip.$_{99}$ ↓ | LL ↑ | Lip.$_{99}$ ↓ | LL ↑ | Lip.$_{99}$ ↓ | LL ↑ | Lip.$_{99}$ ↓ | LL ↑ |
| Small Affine | Single | 22.2 ± 3.5 | -1.45 ± 0.04 | 5.4 ± 0.2 | -2.14 ± 0.02 | 11.9 ± 0.2 | -2.43 ± 0.01 | 3.4 ± 0.2 | -8.75 ± 0.24 |
| | Mixture | 12.3 ± 7.6 | -0.04 ± 0.05 | 4.1 ± 2.3 | -1.02 ± 0.13 | **8.9 ± 1.4** | -2.13 ± 0.04 | **3.2 ± 0.4** | -7.10 ± 0.29 |
| | $k$-means PNF | 3.5 ± 0.9 | -0.36 ± 0.03 | 5.0 ± 0.3 | -1.04 ± 0.04 | 7.9 ± 0.1 | -2.23 ± 0.06 | 2.2 ± 0.2 | -7.58 ± 0.87 |
| | $k$-means Sched. | 3.6 ± 0.5 | -0.32 ± 0.03 | 5.4 ± 0.1 | -1.01 ± 0.02 | 7.8 ± 0.7 | -2.17 ± 0.03 | 2.7 ± 0.7 | -7.23 ± 0.44 |
| | CoCoFlow | **2.1 ± 0.2** | **0.03 ± 0.01** | **1.5 ± 0.4** | **-0.98 ± 0.14** | 9.0 ± 0.6 | **-2.08 ± 0.05** | 3.4 ± 0.1 | **-6.66 ± 0.08** |
| Big Spline | Single | 22.6 ± 4.8 | 0.02 ± 0.02 | 4.4 ± 0.1 | -0.91 ± 0.01 | 8.5 ± 0.5 | -1.76 ± 0.01 | 17.3 ± 1.8 | -2.07 ± 0.18 |
| | Mixture | 22.8 ± 5.8 | 0.06 ± 0.01 | 4.4 ± 0.2 | -0.88 ± 0.01 | 11.1 ± 2.3 | **-1.70 ± 0.02** | **17.2 ± 1.6** | -0.40 ± 0.24 |
| | $k$-means PNF | 4.4 ± 1.3 | 0.06 ± 0.00 | 3.1 ± 1.3 | -0.89 ± 0.01 | 7.7 ± 1.1 | -1.73 ± 0.00 | 16.3 ± 0.8 | -1.68 ± 0.02 |
| | $k$-means Sched. | 5.4 ± 0.9 | 0.06 ± 0.01 | 3.2 ± 0.4 | -0.88 ± 0.02 | 8.0 ± 0.9 | -1.72 ± 0.01 | 17.5 ± 0.9 | -0.51 ± 0.45 |
| | CoCoFlow | **4.4 ± 3.1** | **0.07 ± 0.01** | **3.4 ± 0.4** | **-0.88 ± 0.01** | **10.0 ± 0.3** | -1.71 ± 0.02 | 19.2 ± 0.6 | **-0.27 ± 0.27** |

based on generative performance, and only compare Lipschitz constants between the two best-performing generative models. We do not measure Lipschitz constants in the data-to-noise direction, as these are expected to be large given the lower intrinsic dimensionality of the datasets relative to the ambient space. The modern Normalizing Flow training recipes we employ successfully control data-to-noise continuity via noise dequantization (Gu et al., 2025; Zhai et al., 2024).

On each train split of the data we search the appropriate $k$NN graph construction. We FPS subsample $n = \min(1000, \text{len(data)}/2)$ landmark points and, for each $k$NN graph with $k$ in the range $[\min(2\log n, 2d, 2^d),\ 2\max(2\log n, 2d, 2^d)]$, compute the contractible cover size $m(k)$, where $d$ is the dimension of the data. As $k$ varies, $m(k)$ exhibits plateaus of constant value. We identify the longest such plateau and set $k$ to its central value, since a wide plateau indicates a cover size that is robust to the choice of $k$, and the plateau center is the value least sensitive to the plateau boundaries. For a fair comparison, all compared mixture models use $m(k)$ components, matching CoCoFlow. We perform all experiments on 3 random 80-20 train-test splits.

In Section 5.3, we decompose the Lipschitz constant per component to see what the topology-aware assignment does and does not control. Finally, we study the contractible cover on its own, comparing its construction cost to Betti-number computation and testing what its size reveals about the topology of the support (Section 5.4).

### 5.1 Synthetic Data

We consider 4 topologically rich datasets, described in detail in Appendix C: Mickey, Concentric, Rings, and a 24-dimensional dataset of Cyclooctane molecular conformations (Martin et al., 2010).

**Topologically aware posterior regularization improves generative performance and continuity.** The results in Table 1 show that CoCoFlow achieves the highest generative performance (LL) in 7/8 evaluation settings. The biggest improvements are visible on high-dimensional Cyclooctane dataset, both for the Small Affine model ($-6.66 \pm 0.08$ vs. 2nd-best $-7.10 \pm 0.29$) and Big Spline model ($-0.27 \pm 0.27$ vs. 2nd-best $-0.4 \pm 0.24$). Moreover, comparing models with best and 2nd-best LL, CoCoFlow has lower Lipschitz constant than the other model in 5/8 settings. In the remaining 3/8 scenarios, either Lipschitz constants are comparable or CoCoFlow achieves considerably better LL (as in the Cyclooctane dataset). Additionally, observe that $k$-means model trained using our $\alpha$-schedule recipe achieves higher LL compared to $k$-means PNF in all 8/8 scenarios.

**Topology-aware covers are necessary for faithful reconstruction.** Figure 3 shows that CoCoFlow is the only method whose components reasonably align with the topological structure of the data. On Mickey and Concentric, it decomposes each annulus into contractible arcs and reconstructs every region faithfully. The baselines fail in ways that expose why topology-unaware assignment is insufficient: the unconstrained Mixture routinely assigns disconnected parts of the data to a single component, while $k$-means either assigns a topologically non-trivial region to a single component (the upper-left an-

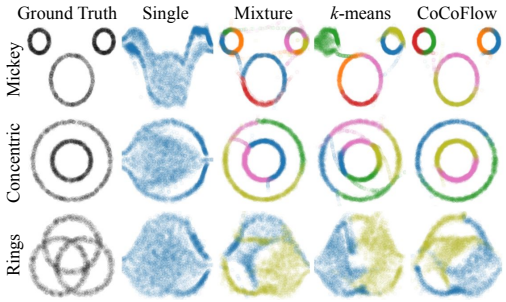


Figure 3: Samples generated with Small Affine models trained on synthetic datasets. Colors represent samples from each component of the mixture model.

Table 2: Test Log-Likelihood (LL) and 99-th quantile of Jacobian norm in generative direction (Lip.$_{99}$) on Earth&Climate Science datasets. Mean±std estimated over 3 runs. In each column, we **bold** the best and underline 2nd-best LL score, highlight both rows, and **bold** the better Lip.$_{99}$ score from the two.

| | | Volcano | | Earthquake | | Flood | | Fire | |
|---|---|---|---|---|---|---|---|---|---|
| | | Lip.$_{99}$ ↓ | LL ↑ | Lip.$_{99}$ ↓ | LL ↑ | Lip.$_{99}$ ↓ | LL ↑ | Lip.$_{99}$ ↓ | LL ↑ |
| Small Affine | Single | 7.8 ± 2.1 | -1.16 ± 0.59 | 25.4 ± 21.3 | -0.76 ± 0.16 | 4.5 ± 0.5 | -1.61 ± 0.55 | 8.8 ± 3.9 | -0.08 ± 0.12 |
| | Mixture | 4.3 ± 1.5 | **0.69 ± 0.69** | 2.3 ± 1.2 | 0.72 ± 0.42 | 2.4 ± 0.7 | -0.04 ± 0.07 | **1.4 ± 0.7** | 2.07 ± 0.01 |
| | $k$-means PNF | 4.4 ± 1.8 | -0.51 ± 2.26 | 2.0 ± 1.2 | 0.55 ± 0.40 | 2.0 ± 0.9 | -0.03 ± 0.09 | 0.8 ± 0.2 | 2.02 ± 0.01 |
| | $k$-means Sched. | 6.4 ± 3.2 | -0.14 ± 2.01 | **1.7 ± 0.7** | 0.73 ± 0.47 | **1.9 ± 0.7** | **0.05 ± 0.17** | 0.8 ± 0.2 | 2.05 ± 0.07 |
| | CoCoFlow | **2.8 ± 0.7** | 0.27 ± 0.26 | 2.5 ± 1.5 | **0.75 ± 0.35** | 2.1 ± 1.6 | -0.02 ± 0.12 | 1.4 ± 0.5 | **2.07 ± 0.08** |
| Big Spline | Single | 9.7 ± 0.5 | 2.81 ± 0.05 | 11.2 ± 3.1 | 1.53 ± 0.01 | 9.9 ± 0.7 | 0.62 ± 0.06 | 12.1 ± 1.4 | 2.37 ± 0.06 |
| | Mixture | 10.0 ± 0.3 | **3.34 ± 0.15** | 9.6 ± 0.3 | **1.90 ± 0.12** | 9.9 ± 1.5 | **0.99 ± 0.07** | 5.5 ± 3.4 | 2.83 ± 0.03 |
| | $k$-means PNF | 7.7 ± 0.3 | 3.20 ± 0.20 | 3.6 ± 2.1 | 1.79 ± 0.11 | 2.7 ± 0.8 | 0.82 ± 0.07 | 1.6 ± 0.7 | 2.76 ± 0.07 |
| | $k$-means Sched. | 7.6 ± 2.2 | 3.20 ± 0.20 | 2.6 ± 1.0 | 1.83 ± 0.13 | 2.3 ± 0.6 | 0.86 ± 0.06 | 1.5 ± 0.3 | 2.75 ± 0.05 |
| | CoCoFlow | **8.9 ± 1.0** | 3.29 ± 0.17 | **4.2 ± 1.8** | 1.85 ± 0.12 | **6.7 ± 2.6** | 0.93 ± 0.02 | **2.9 ± 0.1** | **2.84 ± 0.05** |

nulus in Mickey) or produces a partition that introduces extra discontinuities (Concentric). As a result, these models infill the interiors of the circles or create artificial bridges between disconnected components. The Rings dataset marks the limit of our approach: here the cover does partition the data into two contractible regions, but the high curvature of those regions prevents accurate reconstruction.

### 5.2 Real world Earth&Climate Science data

We consider Earth&Climate Science datasets including Volcano (NOAA, 2020b), Earthquake (NOAA, 2020a), Flood (Brakenridge, 2017), Fire (EOSDIS, 2020) data, compiled by Mathieu and Nickel (2020). These datasets are defined on 2D sphere, fragment into many small components due to islands and contain holes because of bays, sees, and lakes.

As shown in Table 2, CoCoFlow attains the best or 2nd-best generative performance (LL) in all 8/8 evaluation settings. CoCoFlow yields lower Lipschitz constant than the other top-2 generative model in 5/8 settings and is comparable in the rest. The gain is most consistent for the more expressive Big Spline backbone, where CoCoFlow achieves a lower Lipschitz constant than the unconstrained Mixture for all four datasets while matching its log-likelihood to within 0.06 and surpassing it on Fire. On Earthquake, CoCoFlow more than halves the Lipschitz constant (4.2 vs. 9.6) at a log-likelihood of 1.85 against 1.90. Finally, our scheduled-$\alpha$ variant of the $k$-means-based model improves log-likelihood over $k$-means PNF in 6/8 settings and is comparable in the remaining 2/8. We provide generated sample visualizations in Appendix D.

### 5.3 The impact of component assignment on Lipschitz constants

The aggregate Lipschitz constants in Tables 1 and 2 summarize continuity at the level of the whole model, but they conceal how it varies across components. The theory of Cornish et al. (2020) implies that a base–target topological mismatch forces the bi-Lipschitz constant to diverge, so resolving the mismatch removes a *necessary* source of Lipschitz inflation. It does not remove geometric sources such as high curvature, which can inflate Jacobian norms even where the assigned region is contractible. To separate these two contributions, we decompose them per component and visualize the Jacobian norms of the generative maps alongside the generated samples for the Mickey dataset (Figure 4).

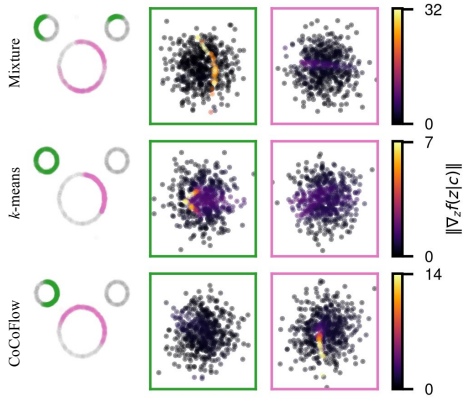


Figure 4: Jacobian norms of the selected Big Spline mixture components trained on Mickey dataset. In column 1, ground truth data (grey) and generated component samples (green, pink). In columns 2-3, Gaussian noise space of each mixture component colored proportionally to the Jacobian norm of the respective generative maps.

For the baselines, the largest norms are attributable to topology. The Mixture component that reconstructs the disconnected parts of the dataset (row 1, green) has the highest Lipschitz constant ($\approx 32$) of all components considered. For the $k$-means model, the component approximating the full circle (row 2, green) exhibits a larger Lipschitz constant ($\approx 6$) than the component approximating the topologically simple arc (row 2, pink), which is expected given the topological mismatch between base and target dis-

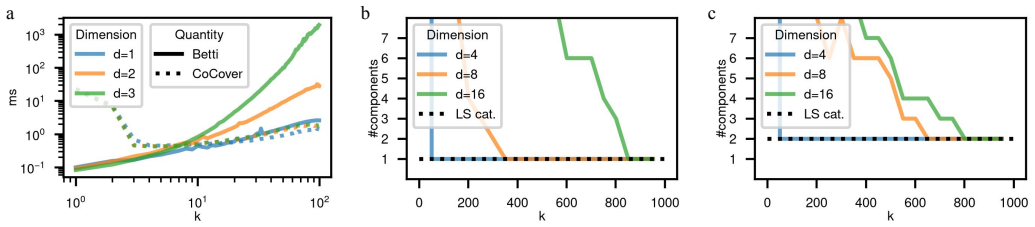


Figure 5: a. Wall-clock time (ms) required to compute Betti numbers and contractible cover (CoCover) of $d$-dimensional data for a range of $k$NN graphs. b-c. Number of components in contractible cover constructed on sample from $d$-dimensional Gaussian (b.) and $(d-1)$-dimensional uniform sphere (c.) across a range of $k$NN graphs, compared with reference Lusternik—Schnirelmann category (LS cat.) of the space.

tributions. Both CoCoFlow mixture components approximate contractible arcs (row 3), so neither carries a topological source of inflation. One arm nonetheless develops a high Jacobian norm ($\approx 13$). Since both arms are topologically identical, this cannot stem from a topological obstruction, and we attribute it to the geometry of the region and the training dynamics of the flow, neither of which can be resolved by a topology-aware partition alone. Aggregate Lipschitz constants must therefore be read with care. A single Lipschitz number cannot separate the topological inflation from the geometric inflation, so a low constant alone is not evidence of a well-resolved topology.

### 5.4 Contractible Cover Analysis

We measure the wall-clock time in ms required to perform full contractible cover search and compare it with the time required to compute Betti numbers for a given graph using GUDHI (Project, 2026). We perform measurements on $d \in \{1, 2, 3\}$ dimensional clique complexes of graphs with $n = 100$ vertices and $k \in \{1, 2, \ldots, n\}$ neighbors. The moderate scale of the experiment is due to computational infeasibility of computing Betti numbers for larger graphs and higher dimensions. Figure 5a shows the payoff of using our algorithm: the required time to perform contractible cover search remains stable across tested dimensions, whereas Betti numbers computation time grows exponentially with the dimension.

Beyond computational cost, we test whether contractible cover size recovers the Lusternik–Schnirelmann (LS) category, the minimum number of contractible sets needed to cover a space. For each $d \in \{4, 8, 16\}$ we draw $10^4$ samples from a $d$-dimensional normal distribution, whose support is contractible (LS cat. $= 1$), and from the uniform distribution on the $(d-1)$-dimensional sphere $S^{d-1}$ (LS cat. $= 2$), and select $n = 1000$ landmarks. As shown in Figure 5b-c, for sufficiently high values of $k \in \{0, 50, \ldots, 1000\}$, the cover size converges to the LS category in every tested dimension, correctly returning 1 for the Gaussian and 2 for the sphere. The cover size thus gives an upper bound on the LS category of high-dimensional spaces. This bound need not be tight in general, but the exact recovery observed here across $d \in \{4, 8, 16\}$ indicates it is informative in practice.

## 6 Discussion

Our **contribution** is CoCoFlow, a method that resolves the homotopy obstructions in Normalizing Flows through a topology-aware decomposition of the target distribution support into homotopy trivial subsets via novel contractible cover algorithm and posterior regularization during training. Crucially, the framework retains exact likelihood evaluation, deterministic maximum-likelihood training, and efficient sampling. Empirically, this translates into improved or competitive density estimation and lower Lipschitz constants of the generative map. Two **limitations** of our method are worth naming: (i) Homotopy equivalence between target and base distributions is necessary but not sufficient for a well-conditioned homeomorphism. High curvature can still inflate Jacobian norms of the specific flow component even when its assigned region is contractible. (ii) The construction of contractible cover relies heavily on accurate graph approximation of the target support. To **summarize**, Prior methods resolving base–target topological mismatch have generally given up at least one canonical property of Normalizing Flows: bijectivity, exact likelihoods, or tractable base distribution. Other approaches rely on naive topology-unaware partition of data. CoCoFlow is the first principled and practical TDA-based solution to this problem.

## AI USE STATEMENT

In this work, we used generative AI tools for retrieval and discovery of related work and to polish writing. We have not used generative AI tools for research ideation or execution, to draft sections of the paper, for generating synthetic datasets, or for proving mathematical claims.

We checked if LLM-retrieved related work contains appropriate content. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES


Jonathan A. Barmak. *Algebraic Topology of Finite Topological Spaces and Applications*. Springer Berlin Heidelberg, 2011. ISBN 9783642220036. doi: 10.1007/978-3-642-22003-6. URL http://dx.doi.org/10.1007/978-3-642-22003-6.

Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger B. Grosse, and Jörn-Henrik Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *AISTATS*, pages 1792–1800, 2021. URL http://proceedings.mlr.press/v130/behrmann21a.html.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. doi: 10.1162/089976603321780317.

Harry Bevins and Will Handley. Piecewise Normalizing Flows. *arXiv e-prints*, art. arXiv:2305.02930, May 2023.

G. R. Brakenridge. Global active archive of large flood events. http://floodobservatory.colorado.edu/Archives/index.html, 2017.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46:255–308, 01 2009. doi: 10.1090/S0273-0979-09-01249-X.

Rob Cornish, Anthony Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2133–2143. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/cornish20a.html.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015. URL https://arxiv.org/abs/1410.8516.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Laurent Dinh, Jascha Sohl-Dickstein, Razvan Pascanu, and Hugo Larochelle. A RAD approach to deep mixture models, 2019. URL https://openreview.net/forum?id=HJeZNLIt_4.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/7ac71d433f282034e088473244df8c02-Paper.pdf.

Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. 01 2010a. ISBN 978-0-8218-4925-5. doi: 10.1007/978-3-540-33259-6_7.

Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction.* American Mathematical Society, 2010b. ISBN 978-0-8218-4925-5.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. 01 2003.

EOSDIS. Active fire data. https://earthdata.nasa.gov/earth-observation-data/near-real-time/firms/active-fire-data, 2020. Land, Atmosphere Near real-time Capability for EOS (LANCE) system operated by NASA's Earth Science Data and Information System (ESDIS).

Stefan Friedl, Tobias Hirsch, and Marc Kegel. The unsolvability of the homeomorphism problem, 2026. URL https://arxiv.org/abs/2603.23630.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(67):2001–2049, 2010. URL http://jmlr.org/papers/v11/ganchev10a.html.

Mevlana C. Gemici, Danilo Rezende, and Shakir Mohamed. Normalizing flows on riemannian manifolds, 2016. URL https://arxiv.org/abs/1611.02304.

Raj Ghugare and Benjamin Eysenbach. Normalizing flows are capable models for continuous control, 2026. URL https://arxiv.org/abs/2505.23527.

Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. ISSN 0304-3975. doi: https://doi.org/10.1016/0304-3975(85)90224-5. URL https://www.sciencedirect.com/science/article/pii/0304397585902245.

Jiatao Gu, Tianrong Chen, David Berthelot, Huangjie Zheng, Yuyang Wang, Ruixiang Zhang, Laurent Dinh, Miguel Angel Bautista, Josh Susskind, and Shuangfei Zhai. Starflow: Scaling latent normalizing flows for high-resolution image synthesis. *NeurIPS*, 2025.

Allen Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge, 2002. ISBN 0-521-79160-X; 0-521-79540-0.

Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows. *arXiv preprint arXiv:1912.13025*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

A. Markov. Insolubility of the problem of homeomorphy. *Doklady Akademii Nauk SSSR*, 121, 01 1958.

Shawn Martin, Aidan Thompson, Evangelos A. Coutsias, and Jean-Paul Watson. Topology of cyclooctane energy landscape. *The Journal of Chemical Physics*, 132(23), 2010. ISSN 1089-7690. doi: 10.1063/1.3445267. URL http://dx.doi.org/10.1063/1.3445267.

Emile Mathieu and Maximilian Nickel. Riemannian continuous normalizing flows, 2020. URL https://arxiv.org/abs/2006.10605.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020. URL https://arxiv.org/abs/1802.03426.

Didrik Nielsen, Priyank Jaini, Emiel Hoogeboom, Ole Winther, and Max Welling. Survae flows: Surjections to bridge the gap between vaes and flows, 2020. URL https://arxiv.org/abs/2007.02731.

NOAA. National geophysical data center / world data service (ngdc/wds): Ncei/wds global significant earthquake database. https://data.nodc.noaa.gov/cgi-bin/iso?id=gov.noaa.ngdc.mgg.hazards:G012153, 2020a.

NOAA. National geophysical data center / world data service (ngdc/wds): Ncei/wds global significant volcanic eruptions database. https://data.nodc.noaa.gov/cgi-bin/iso?id=gov.noaa.ngdc.mgg.hazards:G10147, 2020b.

Richard Nowakowski and Peter Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43(2):235–239, 1983. ISSN 0012-365X. doi: https://doi.org/10.1016/0012-365X(83)90160-7. URL https://www.sciencedirect.com/science/article/pii/0012365X83901607.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators – sampling equilibrium states of many-body systems with deep learning, 2019. URL https://arxiv.org/abs/1812.01729.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6c1da886822c67822bcf3679d04369fa-Paper.pdf.

Guilherme G. P. Freitas Pires and Mário A. T. Figueiredo. Variational mixture of normalizing flows, 2020. URL https://arxiv.org/abs/2009.00585.

The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.13.0 edition, 2026. URL https://gudhi.inria.fr/doc/3.13.0/.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016. URL https://arxiv.org/abs/1505.05770.

Danilo Jimenez Rezende, George Papamakarios, Sebastien Racaniere, Michael Albergo, Gurtej Kanwar, Phiala Shanahan, and Kyle Cranmer. Normalizing flows on tori and spheres. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8083–8092. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/rezende20a.html.

François Rozet et al. Zuko: Normalizing flows in pytorch, 2022. URL https://pypi.org/project/zuko.

Alexander Schmidhuber and Seth Lloyd. Complexity-theoretic limitations on quantum algorithms for topological data analysis. *PRX Quantum*, 4(4), December 2023. ISSN 2691-3399. doi: 10.1103/prxquantum.4.040349. URL http://dx.doi.org/10.1103/PRXQuantum.4.040349.

Sahil Sidheekh, Chris Barton Dock, Tushar Jain, Radu Balan, and Maneesh Kumar Singh. VQ-flows: Vector quantized local normalizing flows. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022. URL https://openreview.net/forum?id=rn4xnDUjcl9.

Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. pages 91–100, 01 2007. doi: 10.2312/SPBG/SPBG07/091-100.

Vincent Stimper, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Resampling Base Distributions of Normalizing Flows. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151, pages 4915–4936, 2022.

Esteban G. Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217 – 233, 2010.

Hanlin Yu, Søren Hauberg, Marcelo Hartmann, Arto Klami, and Georgios Arvanitidis. Learning geometry and topology via multi-chart flows, 2026. URL https://arxiv.org/abs/2505.24665.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. 2024. URL https://arxiv.org/abs/2412.06329.

Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33:249–274, 02 2005. doi: 10.1007/s00454-004-1146-y.

# A ALGORITHMS

**Algorithm 1** Link-Dismantlable Expansion

**Require:** Graph $G = (V, E)$; root $r \in V$
**Ensure:** Contractible region $R$; Corresponding vertex links $L$

$R \leftarrow \{\}$
$L \leftarrow \{\}$
$Q \leftarrow \mathrm{PriorityQueue}\big(\{(0, r)\}\big)$
**while** $Q$ is not empty **do**
    $(_, v) \leftarrow \mathrm{PopMin}(Q)$
    **if** $v$ is *visited* **then continue**
    **end if**
    Mark $v$ as *visited*
    $l_v \leftarrow N[v] \cap R \setminus \{v\}$
    **if** $\mathrm{IsDismantlable}\big(G[l_v]\big)$ **then**
        $R \leftarrow R \cup \{v\}$
        $L \leftarrow L \cup \{l_v\}$
        **for** each $w \in N[v]$ not yet *visited* **do**
            $\mathrm{Push}(Q, (\mathrm{dist}(w, r), w))$
        **end for**
    **end if**
**end while**
**return** $R, L$

**function** ISDISMANTLABLE($H = (V', E')$)
    **while** $|V'| > 1$ **do**
        Find $u, w \in V'$, $u \neq w$, such that $N[u] \subseteq N[w]$
        **if** no such pair exists **then return false**
        **end if**
        $V' \leftarrow V' \setminus \{u\}$
    **end while**
    **return true**
**end function**

**Algorithm 2** Concurrent Link-Dismantlable Expansion

**Require:** Graph $G = (V, E)$; roots $\{r_1, \ldots, r_m\} \subset V$; distance ratio $\tau$
**Ensure:** Contractible regions $R_1, \ldots, R_m$; corresponding vertex links $L_1, \ldots, L_m$

**for** $i = 1, \ldots, m$ **do**
  $R_i \leftarrow \{r_i\}$; $L_i \leftarrow \emptyset$
**end for**
$Q \leftarrow \mathrm{PriorityQueue}()$
**for** $i = 1, \ldots, m$ **do**
  **for** each $v \in N[r_i] \setminus \{r_i\}$ **do**
    $\mathrm{Push}(Q, (d(v, r_i),\ v,\ i))$
  **end for**
**end for**
**while** $Q$ is not empty **do**
  $(d_v, v, i) \leftarrow \mathrm{PopMin}(Q)$
  **if** $v$ is *visited* for root $i$ **then continue**
  **end if**
  Mark $v$ as *visited* for root $i$
  $l_v \leftarrow N[v] \cap R_i \setminus \{v\}$
  **if** $\mathrm{IsDismantlable}\big(G[l_v]\big)$ **then**
    $R_i \leftarrow R_i \cup \{v\}$
    $L_i \leftarrow L_i \cup \{l_v\}$
    $\mathit{propagate} \leftarrow \mathrm{true}$
    **for** $j = 1, \ldots, m$ **do**
      **if** $v \in R_j$ **and** $d_v \geq \tau \cdot d(v, r_j)$ **then**
        $\mathit{propagate} \leftarrow \mathrm{false}$; **break**
      **end if**
    **end for**
    **if** $\mathit{propagate}$ **then**
      **for** each $w \in N[v]$ not yet *visited* or *queued* for root $i$ **do**
        $\mathrm{Push}(Q, (d(w, r_i),\ w,\ i))$
      **end for**
    **end if**
  **end if**
**end while**
**return** $\{R_i\}_{i=1}^{m},\ \{L_i\}_{i=1}^{m}$

**function** IsDismantlable($H = (V', E')$)
  **while** $|V'| > 1$ **do**
    Find $u, w \in V'$, $u \neq w$, such that $N[u] \subseteq N[w]$
    **if** no such pair exists **then return false**
    **end if**
    $V' \leftarrow V' \setminus \{u\}$
  **end while**
  **return true**
**end function**

# B Normalizing Flows Training Recipe

We train all models with the same recipe: 512 batch size, 1024 iterations per epoch, 100 epochs; AdamW (Loshchilov and Hutter, 2019) optimizer with weight decay $10^{-2}$ and learning rate $10^{-4}$ cosine annealed to $10^{-8}$; noise dequantization (Zhai et al., 2024) with noise standard deviation $\sigma$ calibrated so that $\sigma = \varepsilon/\sqrt{\chi^2_{0.95}(d)}$, where $\varepsilon$ is the largest distance from a training points to the landmark set; cosine annealing schedule of $\alpha$ from 0.5 to 0 in the first half of the training. We divide the training set into train and validation splits (80–20)—selecting the checkpoint achieving the highest log-likelihood on the validation split.

# C Detailed Information on Synthetic Datasets

## C.1 Synthetic Datasets

All synthetic datasets contain $n = 10^4$ points in $\mathbb{R}^2$, generated with a fixed random seed and split into train/validation/test partitions in proportions $0.64/0.16/0.20$ (an initial $80/20$ train/test split, followed by an $80/20$ split of the training portion into train and validation).

**Mickey.** This dataset is a mixture of three noisy annuli arranged as a "Mickey Mouse" silhouette: one large ring (the head) and two smaller rings (the ears). The head is centered at the origin with radius $r_h = 1.5$, and the ears have radius $r_e = 0.7$. Ear centers are placed at distance

$$d = (r_h + r_e) \cdot s \tag{1}$$

from the origin, with offset scale $s = 1.5$, at angle $\phi = \pi/4$ from vertical:

$$c_{\text{head}} = (0, 0), \qquad c_{\text{ear}_{\ell,r}} = (\mp d \sin\phi,\ d\cos\phi). \tag{2}$$

Each of the $n$ points is assigned to one of the three components with equal probability $1/3$. For a point assigned to component $k$ (center $c_k$, radius $r_k$), an angle $\theta \sim \text{Unif}(0, 2\pi)$ is drawn, and the point is placed as

$$x = c_k + \big(r_k + \epsilon_{\text{rad}}\big)\,(\cos\theta, \sin\theta) + \epsilon_{\text{tan}}\,(-\sin\theta, \cos\theta), \tag{3}$$

where the radial perturbation combines a uniform "ring thickness" term and Gaussian jitter,

$$\epsilon_{\text{rad}} \sim \text{Unif}(-t/2,\, t/2) + \mathcal{N}(0, \sigma^2), \qquad t = 0.1,\ \sigma = 0.02, \tag{4}$$

and the tangential perturbation is $\epsilon_{\text{tan}} \sim \mathcal{N}(0, (0.65\sigma)^2)$. This produces three ring-shaped clusters of finite width with additional smoothing along the ring.

**Rings.** This dataset consists of three annuli of equal inner and outer radius, $r_{\text{in}} = 1.5$ and $r_{\text{out}} = 1.8$, centered at

$$c_1 = (-0.75, -0.15), \qquad c_2 = (0.75, -0.15), \qquad c_3 = (0, 1.15), \tag{5}$$

chosen with equal probability $1/3$. To ensure points are distributed *uniformly by area* within each annulus (rather than uniformly in radius, which would bias density toward the center), the radius is sampled as

$$r = \sqrt{u\,(r_{\text{out}}^2 - r_{\text{in}}^2) + r_{\text{in}}^2}, \qquad u \sim \text{Unif}(0, 1), \tag{6}$$

together with an angle $\theta \sim \text{Unif}(0, 2\pi)$, giving

$$x = c_k + r\,(\cos\theta, \sin\theta). \tag{7}$$

**Concentric.** This dataset consists of two concentric noisy circles of radius 1 and 0.5, each populated with $n/2$ points. An angle $t \sim \text{Unif}(0, 2\pi)$ is drawn independently for each point, giving base points $(\cos t, \sin t)$ on the unit circle; the inner circle is obtained by scaling these points by 0.5. Isotropic Gaussian noise is then added to all points:

$$x = \big[(\cos t, \sin t) \text{ or } 0.5\,(\cos t, \sin t)\big] + \mathcal{N}(0, \sigma^2 I_2), \qquad \sigma = 0.02. \tag{8}$$

# D EARTH&CLIMATE SCIENCE DATA GENERATED SAMPLES

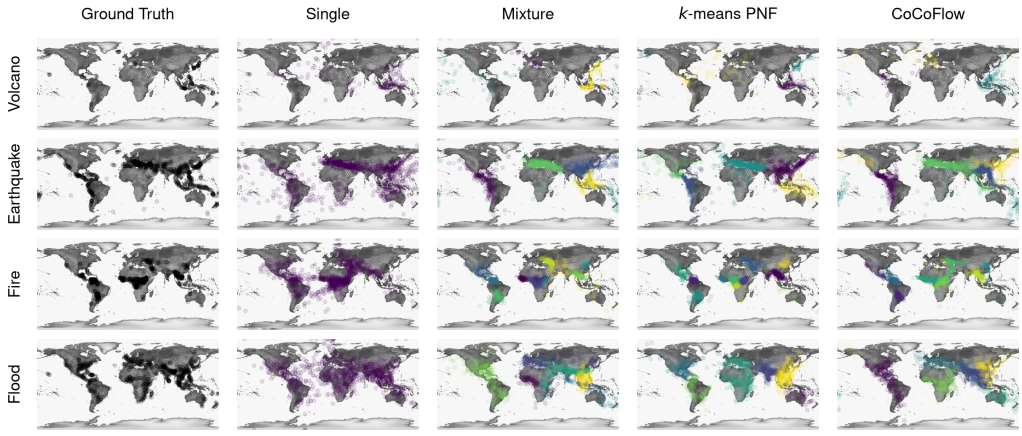


Figure 6: Samples generated with Small Affine models trained on Earth&Climate Science datasets. Colors represent samples from each component of the mixture model.

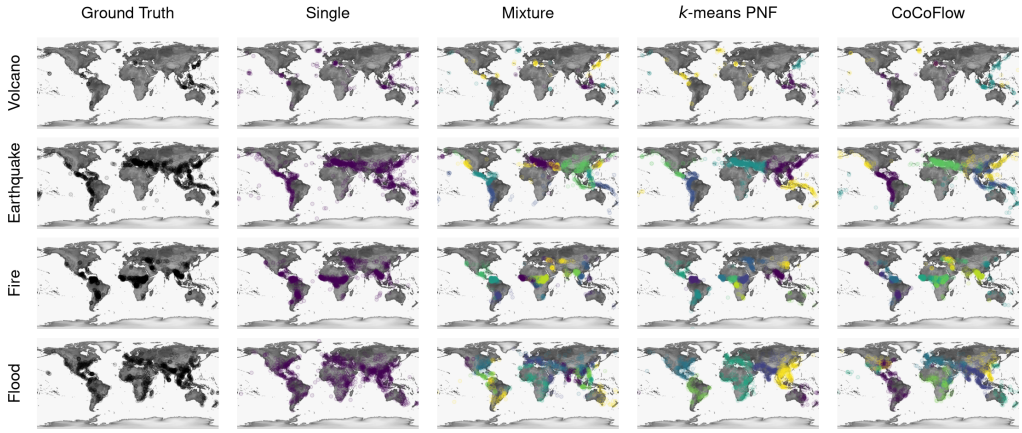


Figure 7: Samples generated with Big Spline models trained on Earth&Climate Science datasets. Colors represent samples from each component of the mixture model.