VARIATIONAL ATLAS AUTOENCODERS AND SELF-SUPERVISED CHART SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Complex, high-dimensional data lie on manifolds that a single global latent coordinate system cannot represent without distortion. We introduce Variational Atlas Autoencoders, a mixture-of-charts framework that learns an atlas of local latent representations, each responsible for one region of the data manifold. The central difficulty in learning an atlas is deciding which chart should be responsible for which data point without labels. We show that this decision need not be a heuristic. By treating the chart index as a latent variable, the evidence lower bound yields the training objective and casts chart assignment as posterior inference, valid for any routing distribution. Within this framework, we propose *latent-evidence routing*, which assigns a point to a chart in proportion to the evidence of its encoding under that chart's latent prior. This has a simple interpretation: the closer a point lands to the center of a chart's latent space, and the more confidently it is placed there, the more likely it is to belong to that chart. Chart assignment thus emerges as posterior inference, depends on the encoder alone, and is fully self-supervised, so the charts discover structure in the data on their own, separating classes and subclasses such as distinct types of shoes or faces without access to labels. We further recover an atlas from the trained charts by identifying their overlaps and transition maps, enabling continuous interpolation within and across charts that stays close to the data manifold rather than traversing unsupported regions of latent space. Compared to capacity-matched baselines, Variational Atlas Autoencoders yields better-structured latent spaces and higher sample quality on image datasets including CelebA, while sampling requires evaluating only a single chart's decoder.

1 INTRODUCTION

Latent spaces learned by autoencoders underpin much of modern generative modeling, from latent diffusion (Rombach et al., 2022; Vahdat et al., 2021) to manifold-learning flows (Brehmer & Cranmer, 2020). These models are only as good as the learned coordinate system they are expressed in, and those coordinates almost always form a single global representation. Forcing all data into one representation is a design choice inherited from the autoencoder, not a property of the data.

Under the manifold hypothesis, data concentrate near a k-dimensional manifold $\mathcal{M} \subset \mathbb{R}^d$ with $k \ll d$ (Fefferman et al., 2016; Brown et al., 2023), and an autoencoder can be viewed as an approximate *chart* of $\mathcal{M}$: the encoder assigns coordinates to each data point, and the decoder maps them back. However, a single chart cannot cover every manifold without tearing or distortion. For example, a sphere cannot be flattened onto a plane, and data made of separate clusters, such as distinct classes, cannot be covered by one continuous latent space without filling the gaps between them. Autoencoders thus resort to non-smooth latent spaces whose interpolants decode to implausible samples (Arvanitidis et al., 2018; Oring et al., 2021). This is unavoidable for disconnected data: the image of a connected latent space is connected, so it must cross the gaps between components.

Differential geometry resolves this with an *atlas*: charts whose domains cover $\mathcal{M}$, overlap, and are related by transition maps. Chart autoencoders (Schonsheck et al., 2019) and mixtures of VAEs (Ye & Bors, 2022; Alberti et al., 2024) move in this direction by giving each component its own encoder–decoder pair, but leave two problems open.

Without labels, the model must decide the *responsibility* of each chart for each data point on the manifold. We make this decision part of the probabilistic model by treating the chart index as a

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

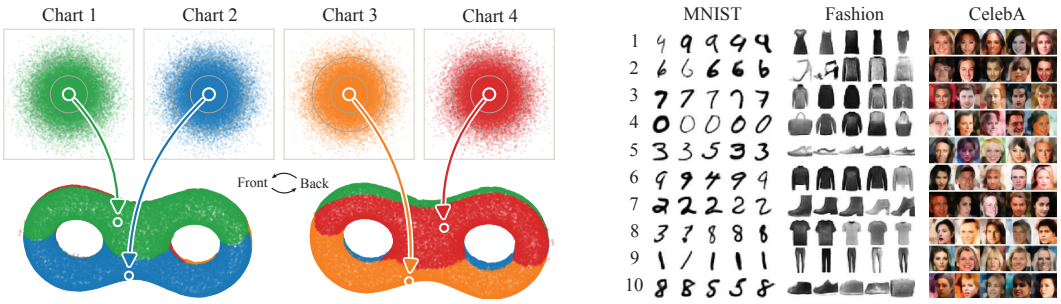


Figure 1: Variational Atlas Autoencoders specialize to meaningful subsets of the data manifold with no label supervision. **Left**: a genus-2 surface covered by four 2D charts, colored by the chart claiming each region, shows that multiple charts avoid the distortion of a single global 2D latent space. **Right**: generated samples per chart on three image datasets. On MNIST (LeCun et al., 1998) and Fashion-MNIST (Xiao et al., 2017) (16-D latents), a chart typically covers one class or a related pair; on CelebA (Liu et al., 2015) (128-D latents), charts split by facial features, illumination, and background.

latent variable. Responsibility then enters the ELBO as the variational posterior $q(c|x)$, which we call the routing function, and since the bound holds for any choice of $q(c|x)$, the question becomes which router trains best. Routing by each chart's ELBO (Alberti et al., 2024) ties assignment to how well a chart already reconstructs a point, which can favour charts that improve early. On the other hand, separate latent spaces use disjoint coordinate systems, so interpolation is undefined across charts, limiting the *connectivity* of the model. Without a way to transition between charts, the model cannot interpolate between arbitrary points on the data manifold, hindering downstream applications expressed in this coordinate system, such as latent-space diffusion.

Contributions. Variational Atlas Autoencoders address both problems. (i) We formulate a variational mixture of charts, each a complete VAE, in which the training objective follows from the ELBO. (ii) We propose *latent-evidence routing*, which assigns a point to a chart in proportion to the evidence of its encoding under that chart's prior: the closer the encoding lies to the chart's centre, and the more confident it is, the higher its responsibility. It depends on the encoders alone, meaning it does not require reconstructions for routing. It discovers classes and subclasses without labels, such as sandals and heeled shoes in Fashion-MNIST. (iii) We turn the trained charts into an atlas by fitting overlaps (*portals*) where the router ties and building a chart transition graph, enabling interpolation within and across charts that stays close to the data manifold. (iv) On MNIST, Fashion-MNIST, and CelebA, our method achieves the best sample FID among parameter-matched VAE and VaDE baselines, while sampling requires only a single (smaller) decoder pass.

2 Related Work

Manifold learning. The manifold hypothesis posits that high-dimensional data concentrate near a manifold of much lower intrinsic dimension. Fefferman et al. (2016) provide a rigorous framework for testing this hypothesis for sufficiently smooth manifolds. For image data, Brown et al. (2023) provide empirical evidence for a more general union-of-manifolds hypothesis: data concentrate near disjoint manifolds with potentially varying intrinsic dimensions. They further show that incorporating this structure into model design improves classification and generation.

These observations suggest that using a single continuous latent space is generally insufficient to represent the data manifold. Tanielian et al. (2020) quantify this phenomenon for GANs: covering separated data components involves generating low-quality samples in the gaps between them. Batson et al. (2021) demonstrate the same issue in the context of anomaly detection, giving low-dimensional examples where topological mismatch leads to reconstruction errors for otherwise ordinary data points. Our use of multiple charts allows separate representations of disconnected components and local coordinates within components whose topology precludes a single global chart, with transition maps relating coordinates on overlaps.

Latent geometry and interpolation. Geometry also matters for density estimation and interpolation. Manifold-learning flows (Brehmer & Cranmer, 2020) jointly learn an embedded manifold and a tractable density on it, while Arvanitidis et al. (2021) equip the ambient space with a Riemannian metric and pull it back through the generator to construct more meaningful interpolation paths. Together, these works motivate separating local parameterization from global structure. Although we do not aim to generate the shortest interpolation paths, our charts are well-suited to both intra-chart and cross-chart interpolation along straight lines thanks to their local coverage of the data manifold.

Multi-chart and mixture models. Mixtures of local linear latent-variable models (Ghahramani & Hinton, 1996; Tipping & Bishop, 1999) are the classical precursor to our model, with mixture responsibilities in the role of the routing posterior. Teh & Roweis (2002) and Brand (2002) align such local models into one global embedding, whereas we keep the local coordinates and learn the transition maps between them (Eq. 8).

In the nonlinear (deep) setting, variational deep embeddings (VaDE) (Jiang et al., 2017) place a Gaussian mixture prior on the latent space of a single VAE and read the cluster assignment from the mixture responsibilities. Since all components share the encoder and decoder, this is still a single chart, which cannot cover most non-trivial data manifolds (Sec. A.2). Mixtures of VAEs with a separate encoder–decoder pair per component (Ye & Bors, 2022; Alberti et al., 2024) remove this restriction and are the closest generative models to ours. To route data to a specific chart, Alberti et al. (2024) use the per-component ELBO. We show that this couples assignment to reconstruction quality and starves charts that start out worse, and we replace it with latent-evidence routing, which depends on the encoder alone. Neither work constructs overlaps or transition maps between components.

The chart auto-encoder (CAE) of Schonsheck et al. (2019) uses several deterministic encoder–decoder pairs and shows that a single simply connected latent space cannot faithfully represent a manifold with non-contractible topology. Charts are assigned by a separate network trained toward a softmax over per-chart reconstruction errors, and transition maps compose one chart's decoder with another's encoder, but the overlaps on which they should hold are never constructed. Stolberg-Larsen & Sommer (2022) unify such multi-chart models as *atlas generative models* and interpolate across charts on a graph over encoded data points. Two charts are linked only at individual data points whose routing exceeds a threshold in both, by connecting that point's two encodings. We differ in three respects: our chart assignment follows from the ELBO (Eq. 4) and is computed from the chart encoders rather than a separate network; overlaps are portals, distributions over a chart's latent space located where the router ties, rather than individual data points; and each portal comes with an explicit transition map, so charts can be crossed from anywhere within it (Sec. 3.3).

3 Methodology

The overall goal of a generative model is to represent a data distribution $p(\mathbf{x})$. In a latent-variable model, the distribution is often represented using a single continuous latent space with a simple prior, typically a multivariate standard normal distribution $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The encoder must therefore represent all samples $\mathbf{x}$ from the data distribution within a single latent space, even when the data lie on a highly nonlinear or disconnected manifold. Our proposed solution is to model the data distribution using multiple latent representations. Each latent component can specialize in a particular subset, mode, or locally coherent region of the data manifold.

3.1 Variational Chart Selection

To model $p_\theta(\mathbf{x})$ using multiple charts, we introduce a discrete chart variable $c \in \{1, \dots, C\}$. As a modeling assumption, each chart is assigned equal prior probability, so that $p(c) = 1/C$. Each chart is further equipped with a continuous L-dimensional latent variable $\mathbf{z}_c \in \mathbb{R}^L$ with conditional prior $p(\mathbf{z}_c \mid c) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The resulting modeled data distribution is then given by:

$$p_\theta(\mathbf{x}) = \sum_{c=1}^{C} p(c) \int p_\theta(\mathbf{x} \mid \mathbf{z}_c, c)\, p(\mathbf{z}_c \mid c)\, \mathrm{d}\mathbf{z}_c. \tag{1}$$

Given an observation $\mathbf{x}$, inference requires the posterior distribution over the chart assignment $p_\theta(c \mid \mathbf{x})$ and the chart-specific latent posterior $p_\theta(\mathbf{z}_c \mid c, \mathbf{x})$. Since these posterior distributions

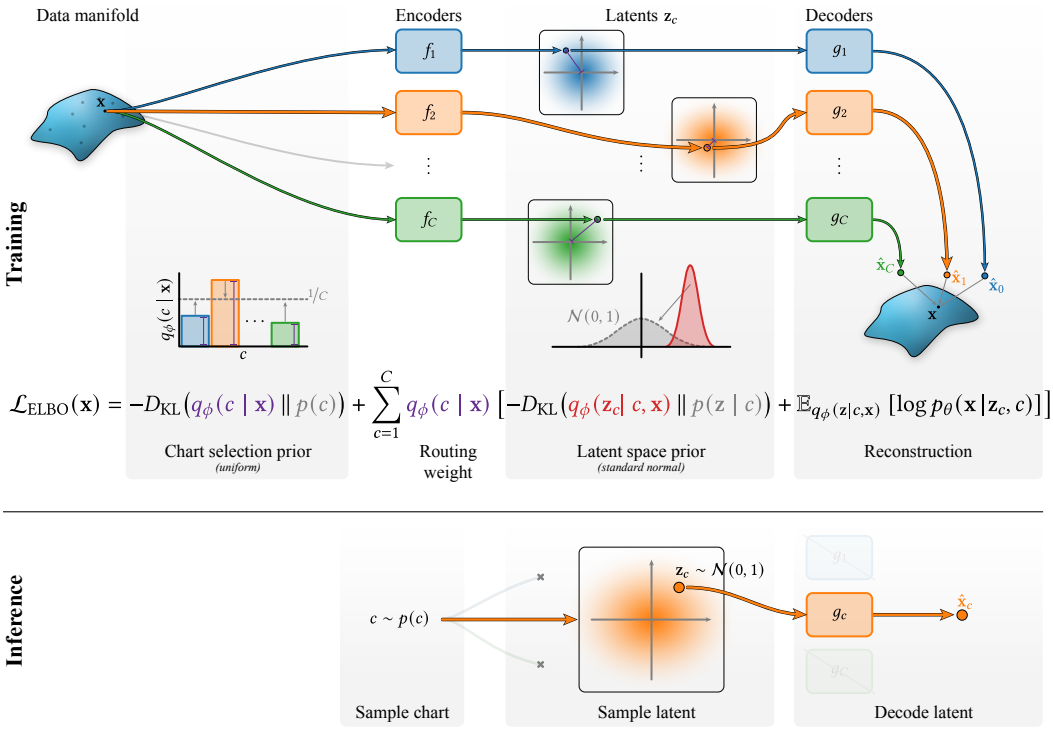


Figure 2: **Top**: we train C encoder–decoder pairs to model the data manifold using independent charts. Our training objective, the ELBO, decomposes into per-chart losses weighted by a routing probability. The KL divergence on the chart selection prior encourages all charts to be used with equal probability. **Bottom**: sampling from a chart is then as simple as selecting a chart c uniformly at random, sampling a latent $\mathbf{z}_c$ from the standard normal prior, and decoding it. Moment matching can optionally be added, see Sec. 3.2.

are generally intractable, we introduce the variational approximations $q_\phi(c \mid \mathbf{x})$ and $q_\phi(\mathbf{z}_c \mid c, \mathbf{x})$. Together, they define the approximate posterior

$$q_\phi(c, \mathbf{z}_c \mid \mathbf{x}) = q_\phi(c \mid \mathbf{x})\, q_\phi(\mathbf{z}_c \mid c, \mathbf{x}), \tag{2}$$

which approximates the true posterior $p_\theta(c, \mathbf{z}_c \mid \mathbf{x})$. Each chart c is realized by an encoder $f_c(\mathbf{x}) = (\boldsymbol{\mu}_c(\mathbf{x}), \boldsymbol{\Sigma}_c(\mathbf{x}))$, which parameterizes the diagonal Gaussian $q_\phi(\mathbf{z}_c \mid c, \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_c(\mathbf{x}), \boldsymbol{\Sigma}_c(\mathbf{x}))$, and a decoder g_c, whose output $\hat{\mathbf{x}} = g_c(\mathbf{z}_c)$ is the mean of $p_\theta(\mathbf{x} \mid \mathbf{z}_c, c)$. Throughout, ϕ and θ denote the parameters of all encoders and decoders, respectively. Direct evaluation and optimization of the marginal log-likelihood in Eq. 1 are generally intractable because the integral over the continuous latent variables cannot usually be computed in closed form. As in the single-chart VAE setting (Kingma & Welling, 2014), we therefore optimize a variational lower bound on $\log p_\theta(\mathbf{x})$,

$$\log p_\theta(\mathbf{x}) \geq \mathcal{L}_{\text{ELBO}}(\mathbf{x}) = \mathbb{E}_{q_\phi(c, \mathbf{z}_c \mid \mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z}_c, c)}{q_\phi(c, \mathbf{z}_c \mid \mathbf{x})}\right]. \tag{3}$$

Relating the lower bound to the exact posterior and expressing terms as KL divergences, we obtain

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\mathbf{x}) = &- D_{\text{KL}}(q_\phi(c \mid \mathbf{x}) \,\|\, p(c)) \\ &+ \sum_{c=1}^{C} q_\phi(c \mid \mathbf{x}) \left[\mathbb{E}_{q_\phi(\mathbf{z}_c \mid c, \mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z}_c, c)\right] - D_{\text{KL}}(q_\phi(\mathbf{z}_c \mid c, \mathbf{x}) \,\|\, p(\mathbf{z}_c \mid c))\right].\end{aligned} \tag{4}$$

Writing the chart router as an explicit finite sum makes it clear that the per-chart ELBO

$$\mathcal{L}_c(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}_c \mid c, \mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z}_c, c)] - D_{\text{KL}}(q_\phi(\mathbf{z}_c \mid c, \mathbf{x}) \,\|\, p(\mathbf{z}_c \mid c)) \tag{5}$$

is simply weighted by each chart's routing probability $q_\phi(c \mid \mathbf{x})$, and the prior routing term $D_{\text{KL}}(q_\phi(c \mid \mathbf{x}) \,\|\, p(c))$ regularizes the routing function. The role of each term is visualized in

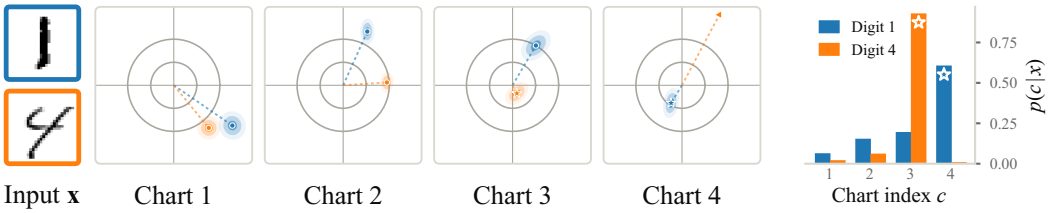


Figure 3: Illustration of latent-evidence routing. Two inputs (left) are encoded by every chart; the evidence of each encoded point under its chart's latent prior induces a distribution over charts, which is used for routing. For the histogram (right), the temperature parameter was artificially raised to $T = 10$ to make the distribution easier to visualize.

Figure 2. In practice, we minimize the weighted loss $\ell(\mathbf{x})$ of Eq. 17, in which $\mathcal{L}_{\text{recon}}(\mathbf{x}, g_c(\mathbf{z}_c))$ is either the reconstruction loss $-\log p_\theta(\mathbf{x} \mid \mathbf{z}_c, c)$ or a perceptual loss, and the two KL terms are weighted by hyperparameters β and γ (Appendix B, Appendix D).

The bound in Eq. 4 holds for *any* routing distribution $q_\phi(c \mid \mathbf{x})$, which leaves the choice of router as a design decision.

ELBO routing. A natural choice is to route by the evidence, $q_\phi(c \mid \mathbf{x}) \propto p(c) \exp(\mathcal{L}_c(\mathbf{x})/T)$, where T is a temperature parameter that anneals the chart assignment. Substituting this router into Eq. 4 collapses the bound into $\mathcal{L}_{\text{ELBO}}(\mathbf{x}) = T \log \sum_c p(c) \exp(\mathcal{L}_c(\mathbf{x})/T)$, which is tight at $T = 1$. We refer to this variant as *ELBO routing*, following related work (Alberti et al., 2024).

Latent-evidence routing. Instead, routing can be based on the latent prior alone, without involving the decoder. We propose *latent-evidence routing*, guided by the intuition that a chart's routing probability should be proportional to the likelihood of the encoded point under that chart's Gaussian prior. All variants below take the form $q_\phi(c \mid \mathbf{x}) \propto p(c) \exp(s_c(\mathbf{x})/T)$, a softmax over a per-chart score s_c with temperature T, and differ only in how s_c accounts for the posterior uncertainty $\mathbf{\Sigma}_c(\mathbf{x}) = \text{diag}(\sigma_{c,1}^2, \ldots, \sigma_{c,L}^2)$. Figure 3 provides a visual overview of latent-evidence routing.

The simplest score evaluates the prior at the posterior mean, $s_c(\mathbf{x}) = \log p(\mathbf{z}_c = \boldsymbol{\mu}_c(\mathbf{x}) \mid c)$, ignoring the posterior uncertainty entirely. We refer to this as *μ-based routing*. Alternatively, the *expected log-prior* score averages the log-prior over the posterior,

$$s_c(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}_c \mid c, \mathbf{x})}[\log p(\mathbf{z}_c \mid c)] = -\frac{1}{2}\left(\|\boldsymbol{\mu}_c\|^2 + \text{tr}\, \mathbf{\Sigma}_c\right) + \text{const.} \tag{6}$$

A single-sample Monte Carlo estimate of this expectation recovers the *resampling-based routing*, at the cost of additional variance in the routing weights.

Finally, we propose *prior-marginal routing*, which scores each chart by the prior mass its posterior covers, i.e. the evidence of the posterior under the chart's latent prior,

$$s_c(\mathbf{x}) = \log \int q_\phi(\mathbf{z}_c \mid c, \mathbf{x})\, p(\mathbf{z}_c \mid c)\, \mathrm{d}\mathbf{z}_c = -\frac{1}{2}\left(\|\boldsymbol{\mu}_c\|^2_{(\mathbf{I}+\mathbf{\Sigma}_c)^{-1}} + \log\det(\mathbf{I} + \mathbf{\Sigma}_c)\right) + \text{const}, \tag{7}$$

where $\|\boldsymbol{\mu}_c\|^2_{(\mathbf{I}+\mathbf{\Sigma}_c)^{-1}} = \boldsymbol{\mu}_c^\top (\mathbf{I}+\mathbf{\Sigma}_c)^{-1} \boldsymbol{\mu}_c$. This is the marginal likelihood $\mathcal{N}(\boldsymbol{\mu}_c(\mathbf{x}); \mathbf{0}, \mathbf{I}+\mathbf{\Sigma}_c(\mathbf{x}))$ of the encoder output regarded as a noisy observation of a latent $\mathbf{z}_c \sim p(\mathbf{z}_c \mid c)$ with observation noise $\mathbf{\Sigma}_c(\mathbf{x})$. All three scores are available in closed form for diagonal Gaussian posteriors. Whereas ELBO routing measures the evidence of $\mathbf{x}$ in data space and therefore depends on the decoder, s_c measures the evidence of the encoded point in latent space and depends on the encoder only. It formalizes the latent-centrality intuition of Sec. 1: a point belongs to the chart whose posterior places it closest to the center of the chart's latent domain.

The three scores differ only in how they treat uncertainty, and in the limit $\mathbf{\Sigma}_c \to \mathbf{0}$, all of them coincide with μ-based routing. The expected log-prior (Eq. 6) is a lower bound on the prior marginal by Jensen's inequality, but it penalizes the full Euclidean distance $\|\boldsymbol{\mu}_c\|^2$ regardless of the posterior spread and merely adds a variance penalty $\text{tr}\, \mathbf{\Sigma}_c$. The prior marginal of Eq. 7 instead measures this distance in the Mahalanobis norm of $\mathbf{I} + \mathbf{\Sigma}_c$, so that uncertain encodings are penalized less.

Impact on training. The choice of router also affects training: since Eq. 4 is a valid lower bound for any $q_\phi(c \mid \mathbf{x})$, we substitute the router in and maximize $\mathcal{L}_{\text{ELBO}}$ directly, so the encoders receive gradients through both the per-chart bounds $\mathcal{L}_c$ and the routing weights. Under ELBO routing, a chart's weight is determined by how well it already reconstructs $\mathbf{x}$. Charts that start out worse receive smaller weights, hence smaller gradients, and fall further behind. Latent-evidence routing removes this feedback loop by assigning points according to where they land in latent space rather than how well they are reconstructed, and the KL term in Eq. 4 keeps the routing weights from collapsing onto a single chart. We adopt prior-marginal routing (Eq. 7) as our default due to its robustness and report ablation results for all variants in Table 5 in the appendix.

3.2 Sampling

Moment matching (optional). After training, the aggregate posterior $q_\phi(\mathbf{z}_c \mid c) = \mathbb{E}_{p(\mathbf{x})}\, q_\phi(\mathbf{z}_c \mid c, \mathbf{x})$ of a chart need not coincide exactly with its prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and sampling from regions of the prior that carry no posterior mass produces poor samples (Hoffman & Johnson, 2016). We therefore optionally replace the prior by the moment-matched aggregate posterior $\tilde{p}(\mathbf{z}_c \mid c) := \mathcal{N}(\mathbf{m}_c, \mathbf{S}_c)$, whose mean and covariance are those of $q_\phi(\mathbf{z}_c \mid c)$ estimated once over the training set. Without moment matching, we use $\tilde{p} = p$. The chart prior is treated likewise: the uniform $p(c) = 1/C$ is replaced by the empirical routing mass $\hat{p}(c) = \mathbb{E}_{p(\mathbf{x})}\, q_\phi(c \mid \mathbf{x})$, so that charts claiming more of the training data are sampled more often. Both corrections are cheap, require no retraining, and are applied in all sampling experiments unless otherwise stated.

Unconditional sampling. Generation follows Algorithm 2: draw a chart $c \sim \hat{p}(c)$ and a latent $\mathbf{z}_c \sim \tilde{p}(\mathbf{z}_c \mid c)$, and decode $\hat{\mathbf{x}} = g_c(\mathbf{z}_c)$. Only the selected decoder is evaluated, so the sampling cost is that of a single VAE with the same backbone. Sampling within one chart yields samples from the corresponding region of the data manifold (Figure 9), which makes the chart index an interpretable, label-free handle on the generated content.

Encoding. In contrast, encoding a data point requires all C encoders to evaluate the routing weights $q_\phi(c \mid \mathbf{x})$. For applications where encoding cost matters, one may train a lightweight classifier to predict the routing distribution so that only the relevant encoder is evaluated.

3.3 Overlaps, Transition Maps, and Portals

An atlas on a manifold is a family of charts whose domains cover the manifold and whose *transition maps* are homeomorphisms on the overlaps (Sec. A.1). The model described in Sec. 3.1 supplies coordinate maps, but the routing posterior $q_\phi(c \mid \mathbf{x})$ induces a *partition* of the data, and a partition has no overlaps. Here, we describe how overlaps and transition maps between charts can be constructed to recover an atlas and thereby enable movement from any point on the manifold to any other.

Transition map. The decoder g_c and the encoder mean $\boldsymbol{\mu}_c$ play the roles of the parametrization and the coordinate map of chart c (Sec. A.3), so the transition map from chart a to chart b and its round trip are

$$\psi_{ab}(\mathbf{z}) = \boldsymbol{\mu}_b\big(g_a(\mathbf{z})\big), \qquad R_{ab} = \psi_{ba} \circ \psi_{ab}. \tag{8}$$

The atlas condition on the overlap is $R_{ab} = \mathrm{id}$. We measure latent distances in the Mahalanobis metric of the sampling distribution $\tilde{p}(\mathbf{z}_c \mid c) = \mathcal{N}(\mathbf{m}_c, \mathbf{S}_c)$ of Sec. 3.2, $\|\mathbf{u}\|_c^2 = \frac{1}{L}\,\mathbf{u}^\top \mathbf{S}_c^{-1}\mathbf{u}$, normalised so that a typical latent of chart c has $\|\boldsymbol{\mu}_c(\mathbf{x}) - \mathbf{m}_c\|_c^2 \approx 1$.

Portals as variational overlaps. An overlap is a region of the data manifold that *both* charts claim, and the router $q_\phi(c \mid \mathbf{x})$ is the model's own statement of which chart claims a point. We therefore represent the overlap of the ordered pair (a, b) by a Gaussian $q_{ab}(\mathbf{z}) = \mathcal{N}(\mathbf{m}_{ab}, \mathbf{S}_{ab})$ on the latent space of chart a, with $\mathbf{S}_{ab}$ parameterized by its Cholesky factor, fitted to lie where the router assigns equal probability to a and b (a tie). We refer to these overlaps as "portals", as they will be used to transition across charts. The encoders' and decoders' weights are kept frozen while optimizing portal locations. Writing $\mathbf{x} = g_a(\mathbf{z})$, $(R_{ab})_{\#} q_{ab}$ for the pushforward of q_{ab} under the round trip R_{ab}, and $\mathrm{mm}(\cdot)$ for the Gaussian with the same mean and covariance as its argument, q_{ab} minimizes

$$\mathcal{L}_{\text{portal}}(q_{ab}) = -\mathbb{E}_{q_{ab}}\big[\tfrac{1}{2}\big(\log q_\phi(a \mid \mathbf{x}) + \log q_\phi(b \mid \mathbf{x})\big)\big] + \lambda\, D_{\mathrm{KL}}\Big(\mathrm{mm}\big((R_{ab})_{\#} q_{ab}\big) \,\Big\|\, q_{ab}\Big). \tag{9}$$

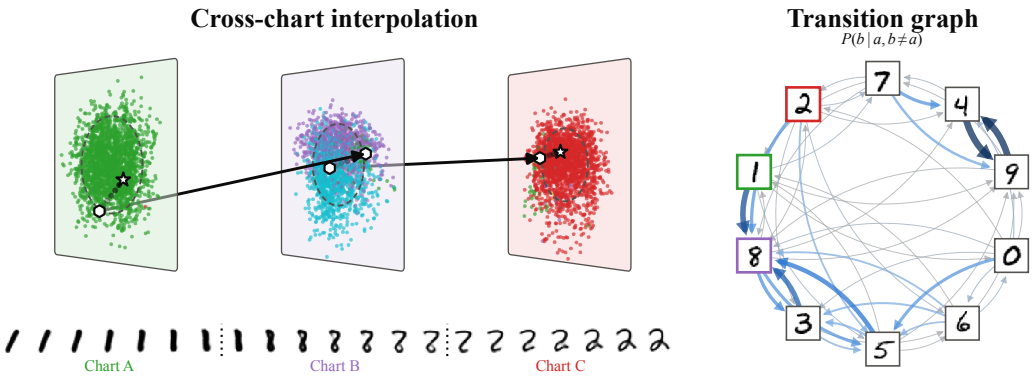


Figure 4: An atlas of chart autoencoders can create smooth latent interpolation within each chart as well as across multiple charts. After training, we precompute a transition graph (right). Interpolation between any two examples in any two charts can then be achieved by following the shortest path in that graph. Within each chart's latent space, we interpolate linearly (left).

The first term is minimized at an exact tie $q_\phi(a \mid \mathbf{x}) = q_\phi(b \mid \mathbf{x}) = \frac{1}{2}$, since the two probabilities sum to at most one, and on its own already locates the overlap. The second term favours the part of the tie set on which the round trip preserves q_{ab} as a distribution. This on its own does not enforce pointwise invertibility of ψ_{ab}, but the defect can be repaired by an optional post-processing step proposed in Appendix C. Finally, expectations are estimated by reparameterized Monte Carlo, and each fit is initialized at the data point where the router most nearly ties between the two charts. This differs from Stolberg-Larsen & Sommer (2022), who cross charts directly at individual data points whose routing exceeds a threshold in both charts, so that the correspondence between charts exists only at those points. A portal instead is a region of the overlap, and the transition map of Eq. 8 lets charts be crossed from anywhere within it.

Transition measure and chart graph. To interpolate between points in different charts, we need to know which charts are adjacent. We use the model's own generative process to determine this. A sample drawn from chart a is routed by q_ϕ to some chart b with some probability which defines the Markov kernel

$$P(b \mid a) = \int q_\phi\big(b \mid g_a(\mathbf{z})\big)\, \tilde{p}(\mathbf{z} \mid a)\, d\mathbf{z}. \tag{10}$$

We estimate this probability with Monte Carlo. The kernel is row-stochastic, and its diagonal is the probability of staying in chart a. Moreover, it is generally asymmetric, since $P(b \mid a)$ and $P(a \mid b)$ involve different decoders. The charts and the off-diagonal entries of P form a directed graph, the chart graph, with an edge $a \to b$ of weight $P(b \mid a)$ (Figure 4, right). Its integrand is the same routing probability whose tie defines the portal q_{ab} in Eq. 9, so every edge of the graph comes with a portal through which it can be traversed.

Interpolation. With the transition kernel (Eq. 10) precomputed, interpolating between points in charts a and b reduces to a shortest-path problem on the chart graph with edge costs $-\log P(b \mid a)$. Since costs are additive, a chart sequence $a = c_0 \to c_1 \to \cdots \to c_K = b$ has cost $-\sum_k \log P(c_{k+1} \mid c_k)$, which corresponds to the negative log-probability of that sequence under the Markov chain defined by P. The shortest path, found with Dijkstra's algorithm, is therefore the most probable route from a to b. It passes through zero or more intermediate charts, entering and leaving each through a portal. Within a chart we interpolate linearly to the portal of the next edge, represented by its mean $\mathbf{m}_{c_k c_{k+1}}$ or a sample drawn from $q_{c_k c_{k+1}}$, and transfer through the transition map of Eq. 8.

4 Results

We train our model on three image datasets and compare it with a standard Variational Auto-Encoder (VAE) and VaDE (Jiang et al., 2017). We also attempted to compare it with CAE (Schonsheck et al.,

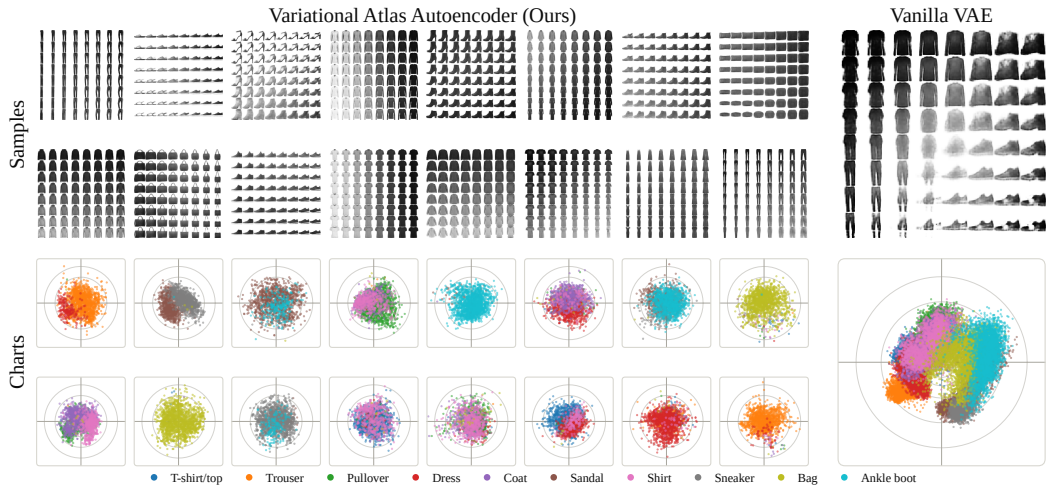


Figure 5: Sampling a model trained on Fashion-MNIST (Xiao et al., 2017) along the two dominant latent PCA axes generates fewer degenerate samples than a parameter-matched vanilla VAE (Kingma & Welling, 2014). The charts also specialize without supervision, e.g. by placing bags without handles in a separate chart. For readability, we show 16 out of the 20 charts.

Table 1: Sampling and reconstruction quality for a plain VAE (Kingma & Welling, 2014), VaDE (Jiang et al., 2017), and our atlas model (10 charts), all using the same number of parameters and training iterations. Quality is measured by the Fréchet Inception Distance (Inception-v3, Szegedy et al. (2016)), averaged over three runs (mean ± std). Lower is better.

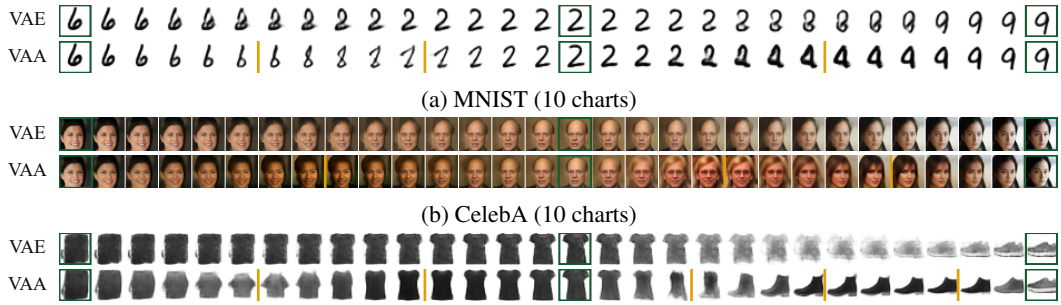
	Sample FID			Reconstruction FID		
Dataset	Plain VAE	VaDE	VAA (ours)	Plain VAE	VaDE	VAA (ours)
MNIST	19.84 ± 0.34	14.95 ± 0.54	**13.74 ± 0.55**	**16.81 ± 0.23**	20.54 ± 0.12	18.25 ± 0.13
Fashion	83.27 ± 1.62	45.07 ± 1.62	**37.73 ± 1.12**	**31.27 ± 0.32**	43.49 ± 0.15	32.83 ± 0.38
CelebA	30.43 ± 0.59	28.01 ± 0.26	**27.84 ± 0.42**	**16.66 ± 0.18**	22.09 ± 0.07	17.24 ± 0.25

2019) and MLMM-VAE (Alberti et al., 2024), but did not succeed in fitting these models to the datasets: one or two charts covered the entire dataset, while the remaining charts were inactive.

All methods are scaled to use a similar number of parameters (model size), and hyperparameters are selected for each method and each dataset via grid search. See Appendix D for a detailed description of the experimental setup. Additional results are available in Appendix F, including ablations, reconstructions and samples. Throughout this paper, 2D visualizations of higher-dimensional latent spaces use the first two PCA directions.

Geometric examples. In a geometric setting, such as the double torus shown in Figure 1, our charts cover the surface with low distortion and clear separation. Interpolation trajectories follow the data manifold closely, as shown in Figure 8.

Reconstruction & generation. Our method faithfully reconstructs input examples, as shown in Figure 7. Reconstruction performance for our method and the capacity-matched baselines is also reported in Table 1. For reconstruction alone, the vanilla VAE tends to perform best, which can be attributed to its larger decoder capacity in comparison to a single chart. This tendency is reversed in image generation, where our method outperforms the two baselines. This can be explained by the better latent structure and the lower incidence of samples drawn from low-density regions in the latent space (see Figure 5). We can see that each chart's latent space is closer to a Gaussian than the combined latent space of the VAE. As a result, the generated samples have fewer artifacts. More examples generated with our model can be found in Figure 1 and Figure 9 in the appendix.



(a) MNIST (10 charts)

(b) CelebA (10 charts)

(c) Fashion-MNIST (20 charts)

Figure 6: Interpolations between sequences of randomly selected training samples (green). Our method may walk through one or more charts via portals (Sec. 3.3), denoted by orange vertical bars. The vanilla VAE traverses its single global latent space in a straight line, which leads to several nonsensical generations along the way. In contrast, our method consistently stays on the data manifold, even if it entails covering more distance in latent space.

Self-supervised structure discovery. Our variational chart selection objective uncovers dataset structure by regularizing reconstruction smoothness while allowing disconnected charts when needed. Without supervision, our 20-chart VAA uncovers subclasses in Fashion-MNIST and other datasets (e.g., sandals and shoes with heels, see Figure 5). Moreover, we observe that the data closely follow the Gaussian prior in the latent domain, whereas this assumption breaks down in the capacity-matched VAE.

Interpolation. Clustering is not the only structure our method discovers. The transition kernel (Eq. 10) connects the charts into a graph that reflects semantic similarity without any supervision. On MNIST (Figure 4), charts of similar digits such as 4 and 9 are strongly connected, while dissimilar ones such as 2 and 0 are nearly disconnected. Interpolating along this graph (see Figure 4, Figure 6) yields realistic samples along the entire path, including the passage through intermediate charts. In contrast, the capacity-matched VAE produces degenerate samples whenever its straight-line path crosses a region of low data density. The Variational Atlas Autoencoder avoids such regions because every segment of its path stays within a chart's latent support and hence close to the data manifold.

5 Conclusion

We have introduced an end-to-end probabilistic method to train variational atlas autoencoders, which are inspired by the manifold hypothesis. We proposed latent-evidence-based routing, which connects the probability of each data point under each chart's prior to the confidence that the chart should encode it. Finally, we showed how the mixture of charts can be turned into an atlas with a post-training step, allowing cross-chart interpolation trajectories that remain close to the data manifold. Our method self-supervises the input data, resulting in smooth charts while staying faithful to the data manifold, and enables high-quality reconstruction and generation. Due to our chart selection not all model parameters are active during sampling, resulting in a lower sampling cost while outperforming relevant baselines on sampling quality.

In future work, determining an adequate chart count automatically, for example by adding or removing charts adaptively during training based on the data manifold's complexity, would improve the method's ease of use and utility. Training can be further accelerated by training a lightweight routing proxy model, so that only the encoders in the most likely subset need to be evaluated, rather than all of them. Next, because a single transition point is selected for each pair of charts, our cross-chart interpolations sometimes cover more distance than needed. Introducing more portals should help minimize total interpolation distance. Finally, scaling the method to larger datasets will likely require combining it with more advanced architectures such as latent-space diffusion or vector-quantized VAEs.

AI USE STATEMENT

In this work, we used large language models (LLMs) to prototype and develop code and to verify the concepts presented in this paper. We have not used these tools to generate the core idea or the mathematics associated with it. Additionally, we used LLM tools to help find and summarize related work, revise portions of the manuscript, and write code to build figures. We have reviewed all AI-assisted work for correctness. In particular, LLM-generated code was verified and tested by one author. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

While we do not foresee risks introduced by this method beyond those shared by all generative methods, foundational research can have a wide range of uses in subsequent work that translates it into practical applications. We employed standard academic datasets and performed all experiments *in silico*.

REFERENCES

Giovanni S. Alberti, Johannes Hertrich, Matteo Santacesaria, and Silvia Sciutto. Manifold learning by mixture models of VAEs for inverse problems. *Journal of Machine Learning Research*, 25 (202):1–35, 2024.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Georgios Arvanitidis, Søren Hauberg, and Bernhard Schölkopf. Geometrically enriched latent spaces. In *The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 631–639. PMLR, 2021.

Joshua Batson, C. Grace Haaf, Yonatan Kahn, and Daniel A. Roberts. Topological obstructions to autoencoding. *Journal of High Energy Physics*, 2021(4):280, 2021.

Matthew Brand. Charting a manifold. In *Advances in Neural Information Processing Systems*, volume 15, pp. 961–968, 2002.

Johann Brehmer and Kyle Cranmer. Flows for simultaneous manifold learning and density estimation. *Advances in neural information processing systems*, 33:442–453, 2020.

Bradley C. A. Brown, Anthony L. Caterini, Brendan Leigh Ross, Jesse C. Cresswell, and Gabriel Loaiza-Ganem. Verifying the union of manifolds hypothesis for image data. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Zoubin Ghahramani and Geoffrey E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto, 1996.

Matthew D. Hoffman and Matthew J. Johnson. ELBO surgery: yet another way to carve up the variational evidence lower bound. In *Workshop on Advances in Approximate Bayesian Inference, NeurIPS*, 2016.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pp. 1965–1972. AAAI Press, 2017. ISBN 9780999241103.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

John M. Lee. *Introduction to Smooth Manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, 2 edition, 2012.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Alon Oring, Zohar Yakhini, and Yacov Hel-Or. Autoencoder image interpolation by shaping the latent space. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8281–8290. PMLR, 18–24 Jul 2021.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR*, 2016.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685, 2022.

Stefan Schonsheck, Jie Chen, and Rongjie Lai. Chart auto-encoders for manifold structured data. *arXiv preprint arXiv:1912.10094*, 2019.

Jakob Stolberg-Larsen and Stefan Sommer. Atlas generative models and geodesic interpolation. *Image and Vision Computing*, 122:104433, 2022.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jérémie Mary. Learning disconnected manifolds: a no GAN's land. In *International Conference on Machine Learning*, pp. 9418–9427. PMLR, 2020.

Yee Whye Teh and Sam T. Roweis. Automatic alignment of local representations. In *Advances in Neural Information Processing Systems*, pp. 841–848. MIT Press, 2002.

Michael E. Tipping and Christopher M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Fei Ye and Adrian G. Bors. Deep mixture generative autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5789–5803, 2022.

Dejiao Zhang, Yifan Sun, Brian Eriksson, and Laura Balzano. Deep unsupervised clustering using mixture of autoencoders. *arXiv preprint arXiv:1712.07788*, 2017.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Appendices

A Preliminaries

We collect the geometric notions used in the main text and fix notation. Everything in Sec. A.1 is standard; we follow the conventions of Lee (2012).

A.1 Manifolds, charts, and atlases

Manifolds. A *k-dimensional manifold* is a space $\mathcal{M}$ that is locally Euclidean: every point has an open neighborhood homeomorphic to an open subset of $\mathbb{R}^k$. Throughout, we assume that the *manifold hypothesis* (Fefferman et al., 2016) holds: the data lie on (or close to) a manifold $\mathcal{M} \subset \mathbb{R}^d$ with $k \ll d$.

Charts. A *chart* is a pair (U, φ) where $U \subset \mathcal{M}$ is open and $\varphi \colon U \to \mathbb{R}^k$ is a homeomorphism onto an open subset of $\mathbb{R}^k$. The map φ assigns local coordinates to points of U, and its inverse φ^{-1} parametrizes U. In autoencoder terms, the encoder plays the role of φ and the decoder that of φ^{-1}.

Atlases and transition maps. An *atlas* is a collection of charts $\{(U_c, \varphi_c)\}_{c=1}^{C}$ whose domains cover $\mathcal{M}$. Where two charts overlap, their coordinates are related by the *transition map*

$$\psi_{ab} = \varphi_b \circ \varphi_a^{-1}, \tag{11}$$

defined on $\varphi_a(U_a \cap U_b)$. Transition maps are themselves homeomorphisms, satisfy $\psi_{ba} = \psi_{ab}^{-1}$, and compose consistently on triple overlaps: $\psi_{bc} \circ \psi_{ab} = \psi_{ac}$. Sec. 3.3 measures how well the learned charts satisfy these conditions.

Ball charts. The charts learned in this paper are slightly more restrictive than the general definition. Each decoder is defined on all of $\mathbb{R}^L$ and the latent prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ has full support, so each chart domain is (approximately) a deformed copy of $\mathbb{R}^L$ rather than an arbitrary open set. We refer to such charts as *ball charts*.

A.2 Why a single chart is not enough

A standard VAE is a single ball chart: one decoder maps a connected latent space $\mathbb{R}^L$ onto the data. This can only faithfully represent $\mathcal{M}$ if $\mathcal{M}$ itself looks like $\mathbb{R}^L$, which excludes most manifolds of interest.

Proposition 1. *Let $g \colon \mathbb{R}^L \to \mathbb{R}^d$ be continuous and let $\mathcal{M} \subset \mathbb{R}^d$ be compact or disconnected. Then g is not a homeomorphism onto $\mathcal{M}$. In particular, $\mathcal{M}$ cannot be covered by a single ball chart.*

Proof. Homeomorphisms preserve compactness and connectedness (Lee, 2012). Since $\mathbb{R}^L$ is connected and not compact, so is any homeomorphic image of it. □

Both hypotheses are typical of real data. Bounded data such as images lie on compact sets, and multiclass data such as MNIST lie on several disconnected pieces. In either case, at least two charts are needed. The sphere, for example, requires exactly two. The proposition also explains the spurious interpolants of single-latent models discussed in Sec. 1. The image $g(\mathbb{R}^L)$ of a single decoder is connected, so if it covers a disconnected $\mathcal{M}$ it must also pass through points outside $\mathcal{M}$. Off-manifold decoding along latent paths between components is therefore unavoidable, not a failure of optimization.

A.3 Autoencoders as approximate charts

Autoencoders. An autoencoder consists of an encoder $f \colon \mathbb{R}^d \to \mathbb{R}^L$ and a decoder $g \colon \mathbb{R}^L \to \mathbb{R}^d$ trained to minimize the reconstruction error $\mathbb{E}_{p(\mathbf{x})} \|\mathbf{x} - g(f(\mathbf{x}))\|^2$. With perfect reconstruction, (f, g) would be a chart with $U = \mathcal{M}$. Reconstruction is never perfect, so we call (f, g) an *approximate chart*: a chart up to the reconstruction error. The atlas properties studied in Sec. 3.3 hold to the same accuracy.

Variational autoencoders. A VAE (Kingma & Welling, 2014) places a prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ on the latent, models $p_\theta(\mathbf{x} \mid \mathbf{z})$ with the decoder, and approximates the posterior by $q_\phi(\mathbf{z} \mid \mathbf{x})$ using the encoder. It is trained by maximizing the evidence lower bound

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - D_{\mathrm{KL}}\big(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})\big), \tag{12}$$

which is the single-chart case $C = 1$ of Eq. 4.

Chart autoencoders. Chart autoencoders (Schonsheck et al., 2019) use C encoder–decoder pairs, one per chart. Since a plain autoencoder has no likelihood, they require a separate network and loss to decide which chart a point belongs to. In our model this decision follows from the ELBO itself (Sec. 3).

A.4 Notation

For a map f and a distribution μ, $f_\# \mu$ denotes the pushforward of μ under f, i.e. the distribution of $f(\mathbf{u})$ for $\mathbf{u} \sim \mu$. For a distribution ν on $\mathbb{R}^L$, $\mathrm{mm}(\nu)$ denotes the Gaussian with the same mean and covariance. A set of non-negative functions $\{w_c\}$ with $\sum_c w_c(\mathbf{x}) = 1$ for every $\mathbf{x}$ is a *partition of unity*; the routing posterior $q_\phi(c \mid \mathbf{x})$ is one by construction.

B Detailed derivation of the objective function

We provide the detailed derivation of the objective function. To model $p_\theta(\mathbf{x})$ using multiple charts, we introduce a discrete chart variable $c \in \{1, \ldots, C\}$. As a modeling assumption, each chart is assigned the same prior probability, so that $p(c) = 1/C$. Each chart is further equipped with a continuous L-dimensional latent variable $\mathbf{z}_c \in \mathbb{R}^L$ with conditional prior $p(\mathbf{z}_c \mid c) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Repeating Eq. 1 for convenience, the resulting modeled data distribution is then given by:

$$p_\theta(\mathbf{x}) = \sum_{c=1}^{C} p(c) \int p_\theta(\mathbf{x} \mid \mathbf{z}_c, c)\, p(\mathbf{z}_c \mid c)\, \mathrm{d}\mathbf{z}_c. \tag{13}$$

Given an observation $\mathbf{x}$, inference requires the posterior distribution over the chart assignment $p_\theta(c \mid \mathbf{x})$ and the chart-specific latent posterior $p_\theta(\mathbf{z}_c \mid c, \mathbf{x})$. Since these posterior distributions are generally intractable, we introduce the variational approximations $q_\phi(c \mid \mathbf{x})$ and $q_\phi(\mathbf{z}_c \mid c, \mathbf{x})$. Together, they define the approximate posterior (repeating Eq. 2 for convenience)

$$q_\phi(c, \mathbf{z}_c \mid \mathbf{x}) = q_\phi(c \mid \mathbf{x})\, q_\phi(\mathbf{z}_c \mid c, \mathbf{x}), \tag{14}$$

which approximates the true posterior $p_\theta(c, \mathbf{z}_c \mid \mathbf{x})$.

Direct evaluation and optimization of the marginal log-likelihood in Eq. 13 are generally intractable because the integral over the continuous latent variables cannot usually be computed in closed form. As in the single-chart VAE setting (Kingma & Welling, 2014), we therefore optimize a variational lower bound on $\log p_\theta(\mathbf{x})$.

Using $q_\phi(c, \mathbf{z}_c \mid \mathbf{x}) = q_\phi(c \mid \mathbf{x}) q_\phi(\mathbf{z}_c \mid c, \mathbf{x})$, the marginal likelihood can be rewritten as

$$\log p_\theta(\mathbf{x}) = \log \sum_{c=1}^{C} \int q_\phi(c, \mathbf{z}_c \mid \mathbf{x}) \frac{p_\theta(\mathbf{x}, \mathbf{z}_c, c)}{q_\phi(c, \mathbf{z}_c \mid \mathbf{x})}\, \mathrm{d}\mathbf{z}_c. \tag{15}$$

Applying Jensen's inequality yields

$$\begin{aligned} \log p_\theta(\mathbf{x}) &\geq \sum_{c=1}^{C} q_\phi(c \mid \mathbf{x}) \int q_\phi(\mathbf{z}_c \mid c, \mathbf{x}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z}_c, c)}{q_\phi(c, \mathbf{z}_c \mid \mathbf{x})}\, \mathrm{d}\mathbf{z}_c \\ &= \mathbb{E}_{c \sim q_\phi(c|\mathbf{x})} \left[\mathbb{E}_{\mathbf{z}_c \sim q_\phi(\mathbf{z}_c|c,\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z}_c, c)}{q_\phi(c, \mathbf{z}_c \mid \mathbf{x})} \right] \right]. \end{aligned}$$

This defines the evidence lower bound for the multi-chart model,

$$\log p_\theta(\mathbf{x}) \geq \mathcal{L}_{\mathrm{ELBO}}(\mathbf{x}) = \mathbb{E}_{q_\phi(c,\mathbf{z}_c|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z}_c,c)}{q_\phi(c,\mathbf{z}_c \mid \mathbf{x})}\right].$$

The lower bound can equivalently be related to the exact posterior through

$$\log p_\theta(\mathbf{x}) = \mathcal{L}_{\mathrm{ELBO}}(\mathbf{x}) + D_{\mathrm{KL}}(q_\phi(c,\mathbf{z}_c \mid \mathbf{x}) \,\|\, p_\theta(c,\mathbf{z}_c \mid \mathbf{x})).$$

Since the KL divergence is non-negative, it follows that $\log p_\theta(\mathbf{x}) \geq \mathcal{L}_{\mathrm{ELBO}}(\mathbf{x})$. Using the factorization $p_\theta(\mathbf{x},\mathbf{z}_c,c) = p_\theta(\mathbf{x} \mid \mathbf{z}_c,c)\,p(\mathbf{z}_c \mid c)\,p(c)$ and $q_\phi(c,\mathbf{z}_c \mid \mathbf{x}) = q_\phi(\mathbf{z}_c \mid c,\mathbf{x})\,q_\phi(c \mid \mathbf{x})$, the ELBO can be written as

$$\begin{aligned}
\mathcal{L}_{\mathrm{ELBO}}(\mathbf{x}) &= \mathbb{E}_{c\sim q_\phi(c|\mathbf{x})}\mathbb{E}_{\mathbf{z}_c\sim q_\phi(\mathbf{z}_c|c,\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x} \mid \mathbf{z}_c,c)\,p(\mathbf{z}_c \mid c)\,p(c)}{q_\phi(\mathbf{z}_c \mid c,\mathbf{x})\,q_\phi(c \mid \mathbf{x})}\right] \\
&= \mathbb{E}_{c\sim q_\phi(c|\mathbf{x})}\mathbb{E}_{\mathbf{z}_c\sim q_\phi(\mathbf{z}_c|c,\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z}_c,c)\right] \\
&\quad + \mathbb{E}_{c\sim q_\phi(c|\mathbf{x})}\mathbb{E}_{\mathbf{z}_c\sim q_\phi(\mathbf{z}_c|c,\mathbf{x})}\left[\log \frac{p(\mathbf{z}_c \mid c)}{q_\phi(\mathbf{z}_c \mid c,\mathbf{x})}\right] \\
&\quad + \mathbb{E}_{c\sim q_\phi(c|\mathbf{x})}\left[\log \frac{p(c)}{q_\phi(c \mid \mathbf{x})}\right].
\end{aligned}$$

Expressing the last two terms as KL divergences gives

$$\begin{aligned}
\mathcal{L}_{\mathrm{ELBO}}(\mathbf{x}) &= \mathbb{E}_{c\sim q_\phi(c|\mathbf{x})}\mathbb{E}_{\mathbf{z}_c\sim q_\phi(\mathbf{z}_c|c,\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z}_c,c)\right] \\
&\quad - \mathbb{E}_{c\sim q_\phi(c|\mathbf{x})}\left[D_{\mathrm{KL}}(q_\phi(\mathbf{z}_c \mid c,\mathbf{x}) \,\|\, p(\mathbf{z}_c \mid c))\right] \\
&\quad - D_{\mathrm{KL}}(q_\phi(c \mid \mathbf{x}) \,\|\, p(c)).
\end{aligned}$$

Writing the chart router as an explicit finite sum makes it clear that the per-chart ELBO

$$\mathcal{L}_c(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}_c|c,\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z}_c,c)] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}_c \mid c,\mathbf{x}) \,\|\, p(\mathbf{z}_c \mid c)) \tag{5}$$

is simply weighted by each chart's routing probability $q_\phi(c \mid \mathbf{x})$, while the prior routing term $D_{\mathrm{KL}}(q_\phi(c \mid \mathbf{x}) \,\|\, p(c))$ regularizes the routing function, yielding Eq. 4. In practice, we replace the log-likelihood by a reconstruction loss $\mathcal{L}_{\mathrm{recon}}(\mathbf{x}, g_c(\mathbf{z}_c))$, which defaults to $-\log p_\theta(\mathbf{x} \mid \mathbf{z}_c,c)$ but may also be a perceptual loss (Appendix D), and weight the regularizers by the hyperparameters β (KL weight). This yields the per-chart loss

$$\ell_c(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}_c|c,\mathbf{x})}\left[\mathcal{L}_{\mathrm{recon}}(\mathbf{x}, g_c(\mathbf{z}_c))\right] + \beta\, D_{\mathrm{KL}}(q_\phi(\mathbf{z}_c \mid c,\mathbf{x}) \,\|\, p(\mathbf{z}_c \mid c)) \tag{16}$$

and the training loss

$$\ell(\mathbf{x}) = D_{\mathrm{KL}}(q_\phi(c \mid \mathbf{x}) \,\|\, p(c)) + \sum_{c=1}^{C} q_\phi(c \mid \mathbf{x})\,\ell_c(\mathbf{x}), \tag{17}$$

where $q_\phi(c \mid \mathbf{x})$ is the router of Eq. 7 at temperature T. The loss coincides with $-\mathcal{L}_{\mathrm{ELBO}}$ from Eq. 4 when $\beta = 1$ and $\mathcal{L}_{\mathrm{recon}} = -\log p_\theta$.

C Reducing the roundtrip error

In the portal objective Eq. 9, the second term preserves q_{ab} as a distribution. This locates the overlap but does not make ψ_{ab} invertible on it. To fix this, we propose an optional lightweight fine-tuning step in which the expected pointwise defect,

$$\mathcal{L}_{\mathrm{atlas}} = \mathbb{E}_{(a,b)\sim\pi}\Big[\mathbb{E}_{\mathbf{z}\sim q_{ab}}\|R_{ab}(\mathbf{z}) - \mathbf{z}\|_a^2 + \mathbb{E}_{\mathbf{w}\sim(\psi_{ab})_{\#}q_{ab}}\|R_{ba}(\mathbf{w}) - \mathbf{w}\|_b^2\Big], \tag{18}$$

is minimized in both directions, jointly with the training loss Eq. 17 as a retention term. Distances are measured in the Mahalanobis metric $\|\cdot\|_c$ defined above, so that a defect of one is the size of a typical latent and defects are comparable across charts. The outer expectation is estimated by drawing pairs from π with replacement and the inner expectations by reparameterized Monte Carlo with gradients through ψ_{ab}. The retention term ensures that the model retains its sampling and reconstruction performance.

D EXPERIMENTAL SETUP

This appendix provides additional details on the setup summarized in Sec. 4, specifically on the architecture (Sec. D.1), the chart-balance term and its weight γ (Sec. D.2), per-dataset hyperparameters (Secs. D.3-D.4), and how the vanilla VAE and VaDE baselines are made comparable to ours (Sec. D.5).

D.1 NETWORK ARCHITECTURE

We use two different network architectures: an MLP-only architecture for the double torus dataset, and a convolutional architecture for the image-based datasets.

MLP (double torus). The encoders and decoders are `depth`-layer, `hidden`-wide dense networks with ReLU nonlinearities, where each encoder/decoder pair is private to its respective chart. This can be seen as C separate pairs of dense MLPs, one per chart; although in practice we implement this as a single batched matrix multiply over all C charts at once, rather than looping over C separate layers, which is significantly faster on a GPU. The encoder maps the 3D ambient coordinates that lie on the surface of the torus to a $2\times$`latent_dim` latent vector containing the mean and log-variance; the decoder mirrors this back to 3D, and its raw output is used directly as the reconstruction, with no output nonlinearity (Sec. D.4).

DCGAN (MNIST, Fashion-MNIST, CelebA). For image datasets, we use an architecture that can exploit their spatial structure more effectively. The encoder and decoder follow the convolutional architecture of Radford et al. (2016), with the channel width doubling at every downsampling stage and, conversely, halving at every upsampling stage. In every tuned configuration reported in this paper, this entire convolutional network (encoder and decoder) is shared identically across all charts (Table 2). Charts only differ at the very end: the shared encoder's final feature map is flattened and passed through one linear layer per chart, mapping it to that chart's own $2\times$`latent_dim` latent vector containing the mean and log-variance; the decoder mirrors this with one linear layer per chart that expands z back to that same feature-map size, before the (again shared) network decodes it into an image. These per-chart linear layers are the only chart-specific parameters in this backbone. The VAE and VaDE baselines (Sec. D.5) reuse this same convolutional network at $C = 1$.

The architecture also supports making some of the deepest convolutional stages chart-specific as well, rather than only the two linear layers. Reducing the size of the shared encoder/decoder model improves the generation speed, but comes with the trade-off that each chart needs to learn the mapping from ambient to latent space from scratch, effectively re-learning the feature extractor.

Table 2: Resolved DCGAN dimensions per dataset: channel widths of the shared convolutional network and the size of the feature map passed to the per-chart linear layers.

	MNIST	Fashion-MNIST	CelebA
Convolutional stages	2	2	4
Channel widths	64, 128	64, 128	64, 128, 256, 512
Bottleneck channels	13	6	51
Bottleneck resolution	7×7	7×7	4×4

D.2 THE BALANCE WEIGHT γ

Although Eq. 4 already contains a routing KL toward the uniform chart prior, we keep that term at weight 1 and do not tune it. The hyperparameter γ instead weights a separate *balance term*, which compares the uniform distribution U over the C charts with the routing averaged over a minibatch, $\bar{r}_c = \frac{1}{B}\sum_{i=1}^{B} q_\phi(c \mid x_i)$. The loss we minimize is therefore

$$\mathcal{L} = -\mathcal{L}_{\mathrm{ELBO}} + \gamma\, D_{\mathrm{KL}}(U \,\|\, \bar{r}), \tag{19}$$

with $\gamma = 50$ for the entire run in every configuration.

The two terms push in different ways. The routing KL of Eq. 4 acts on each sample and pulls every $q_\phi(c \mid x)$ toward uniform, which also discourages confident chart assignments. The balance term acts only on the batch average: individual points may still be routed sharply to a single chart, as long as all charts are used about equally overall. To see why, note that the batch-averaged routing KL splits as $\frac{1}{B}\sum_i D_{\mathrm{KL}}(r_i \,\|\, U) = \hat{I}(X;C) + D_{\mathrm{KL}}(\bar{r} \,\|\, U)$, with $r_i = q_\phi(c \mid x_i)$ and $\hat{I}(X;C) = \frac{1}{B}\sum_i D_{\mathrm{KL}}(r_i \,\|\, \bar{r})$ the empirical mutual information between inputs and charts. The first part penalizes confident routing, the second unequal chart usage. The γ term amplifies only the latter (in reversed form), so charts are kept alive without blurring the assignments. This is the same role played by the auxiliary load-balancing loss in related work (Zhang et al., 2017), which prevents a router from sending all inputs to a few expert charts.

D.3 Image Datasets

Hyperparameters (Table 3) are selected independently per dataset via grid search. The VAE and VaDE baselines reuse the architecture of Sec. D.1 and the reconstruction loss below, with only width rescaled to match parameter count (Sec. D.5).

Table 3: Tuned hyperparameters for MNIST, Fashion-MNIST, and CelebA.

Hyperparameter	MNIST	Fashion-MNIST	CelebA
Charts	10	20	10
Latent dimension	16	16	128
Reconstruction loss	Bernoulli (BCE)	Bernoulli (BCE)	L1 + LPIPS, scale $\times 3$
T (routing temperature)	Fixed at 1.0	Fixed at 1.0	Annealed $6.0 \to 2.0$
β (KL weight)	0.5	0.1	0.1
γ (load-balancing weight)	50, entire run	50, entire run	50, entire run
Learning rate	5×10^{-4}	2×10^{-3}	5×10^{-4}
LR schedule	Cos-annealed to 1%	Cos-annealed to 1%	Cos-annealed to 1%
Batch size	256	256	128
Epochs	200	200	200

A per-pixel likelihood scores each pixel independently, which rewards blurry reconstructions. For CelebA specifically, we replace it with a mix of L1 error in pixel space and LPIPS (Zhang et al., 2018), a common perceptual loss used by latent-diffusion autoencoders. Rather than comparing raw pixels, LPIPS compares distances in a pretrained latent space that tracks human judgments more closely. We weight the two terms so that each contributes about as many nats as the part of the BCE loss it replaces, split evenly, then scale the sum by 3 (see Table 3).

Two optimizations keep this loss practical. LPIPS is by far the most expensive term to compute, so we evaluate it once per image rather than once per chart: every chart still gets its own L1 term, but all charts share the LPIPS score of whichever chart reconstructs best. This leaves the ranking between charts unchanged while avoiding the slowdown from scoring every chart separately. We also run the LPIPS network at reduced precision, which roughly halves its cost with no measurable effect on training.

D.4 Double Torus

The double torus is a synthetic point cloud sampled uniformly by area from a genus-2 surface embedded in $\mathbb{R}^3$ (the surface of Figure 1). Targets are 3D coordinates rather than pixel intensities, so we use a squared-error reconstruction term directly on the decoder's raw output, rather than passing it through a sigmoid first as we do for pixel intensities. This term is small relative to the KL term and induces posterior collapse if left uncorrected. We counteract this by scaling up the reconstruction term ($200\times$), with no other change to the loss, routing, or training procedure. Table 4 lists the full configuration.

Table 4: Tuned hyperparameters for the double torus.

Hyperparameter	Value
Backbone	MLP, hidden width 512, depth 5
Charts	4
Latent dimension	2
Reconstruction loss	Squared error, no output squashing, scale $\times 200$
T (routing temperature)	annealed $1.0 \to 0.2$
β (KL weight)	1.5
γ (load-balancing weight)	50, for the entire run
Learning rate	1×10^{-3}, cosine-annealed to 1%
Batch size	2048
Epochs	100
Surface points	1,000,000

D.5 Baselines

Both baselines are given our model's total trainable parameter count on each dataset: we perform a binary search over the baseline's width parameter (channel count for DCGAN, hidden width for the MLP) to match that target, rather than altering depth or other architectural choices, and train for the same number of epochs with the same reconstruction loss. The vanilla VAE control is simply $C = 1$ (Sec. A.3), so matching parameters amounts to retuning the backbone width at $C = 1$.

For VaDE, we port the original authors' code to modern PyTorch, verify it reproduces their reported numbers, and then swap its Gaussian-mixture prior onto our own single-chart DCGAN backbone (Sec. D.1) in place of their original autoencoder. So the architecture, parameter count, and training length are identical across all three methods, and only the modeling choice (charts vs. GMM prior vs. single Gaussian prior) differs.

As noted in Sec. 4, we also attempted CAE (Schonsheck et al., 2019) and MLMM-VAE (Alberti et al., 2024) but did not succeed in fitting either to the image-based datasets: one or two charts covered the whole dataset, while the remaining charts were effectively inactive. As a result these methods were only using a small subset of the available model-parameter capacity resulting in significantly higher FID when sampling new data from the model. For that reason, we exclude them from the main comparison.

E Algorithm

Algorithm 1 Training procedure for Variational Atlas Autoencoders. All C encoders are evaluated for each minibatch to obtain the routing weights and per-chart losses; encoders and decoders are updated jointly.

Require: data $\{\mathbf{x}\}$, encoder–decoder pairs $\{(f_c, g_c)\}_{c=1}^{C}$, temperature T, weights β, γ

while not converged **do**
  sample a minibatch $\mathbf{x}$
  for $c = 1, \dots, C$ **do**
    $(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \leftarrow f_c(\mathbf{x})$ ▷ encode with every chart
    $s_c \leftarrow \log \mathcal{N}(\boldsymbol{\mu}_c;\, \mathbf{0},\, \mathbf{I} + \boldsymbol{\Sigma}_c)$ ▷ latent evidence, Eq. 7
    $\mathbf{z}_c \leftarrow \boldsymbol{\mu}_c + \boldsymbol{\Sigma}_c^{1/2} \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ reparameterized sample
    $\ell_c \leftarrow \mathcal{L}_{\text{recon}}(\mathbf{x}, g_c(\mathbf{z}_c)) + \beta\, D_{\text{KL}}(q_\phi(\mathbf{z}_c \mid c, \mathbf{x}) \,\|\, p(\mathbf{z}_c \mid c))$ ▷ per-chart loss, Eq. 16
  end for
  $q_\phi(c \mid \mathbf{x}) \leftarrow \text{softmax}_c(s_c / T + \log p(c))$ ▷ routing
  $\ell \leftarrow \sum_c q_\phi(c \mid \mathbf{x})\, \ell_c + \gamma\, D_{\text{KL}}(q_\phi(c \mid \mathbf{x}) \,\|\, p(c))$ ▷ weighted loss, Eq. 17
  update ϕ, θ by gradient descent on ℓ
end while

Algorithm 2 Sampling from Variational Atlas Autoencoders. Only the selected decoder is evaluated, so the cost is independent of C. With uniform $p(c)$ and standard normal priors, $\hat{p}(c) = 1/C$ and $(\mathbf{m}_c, \mathbf{S}_c) = (\mathbf{0}, \mathbf{I})$; the moment-matched values provide an optional correction described in Sec. 3.2.

Require: trained decoders $\{g_c\}_{c=1}^{C}$, chart weights $\hat{p}(c)$, sampling distributions $\{\tilde{p}(\mathbf{z}_c \mid c) = \mathcal{N}(\mathbf{m}_c, \mathbf{S}_c)\}_{c=1}^{C}$

1: $c \sim \hat{p}(c)$ ▷ pick a chart
2: $\mathbf{z}_c \sim \tilde{p}(\mathbf{z}_c \mid c)$ ▷ sample its latent
3: $\hat{\mathbf{x}} \leftarrow g_c(\mathbf{z}_c)$ ▷ decode
4: **return** $\hat{\mathbf{x}}$

Table 5: Sample FID (mean ± std over seeds) by routing function, DCGAN backbone, $C = 10$ charts, and iteration counts matched within each dataset. Lower is better; best in bold, second best underlined.

		Latent-evidence routing			
Dataset	ELBO	μ-based	Resampling	Expected log-prior	Prior-marginal
MNIST	**11.80 ± 0.10**	12.99 ± 0.22	13.79 ± 1.00	12.77 ± 0.33	<u>12.62 ± 0.53</u>
Fashion-MNIST	43.64 ± 0.34	<u>38.42 ± 0.49</u>	39.77 ± 1.46	**38.03 ± 0.64**	38.53 ± 0.45
CelebA	47.14 ± 0.19	**43.46 ± 0.53**	44.05 ± 0.24	43.89 ± 0.55	<u>43.69 ± 0.24</u>

Algorithm 1 and 2 summarize the training and sampling procedures. During training, every encoder is evaluated on each minibatch. The encoder outputs determine both the routing weights via the latent-evidence score of Eq. 7 and, through a reparameterized sample, the per-chart losses of Eq. 16. These are combined into the weighted objective of Eq. 17, and all encoders and decoders are updated jointly. Sampling requires only a chart index, a latent sample from that chart's prior, and a single decoder pass.

F Additional results

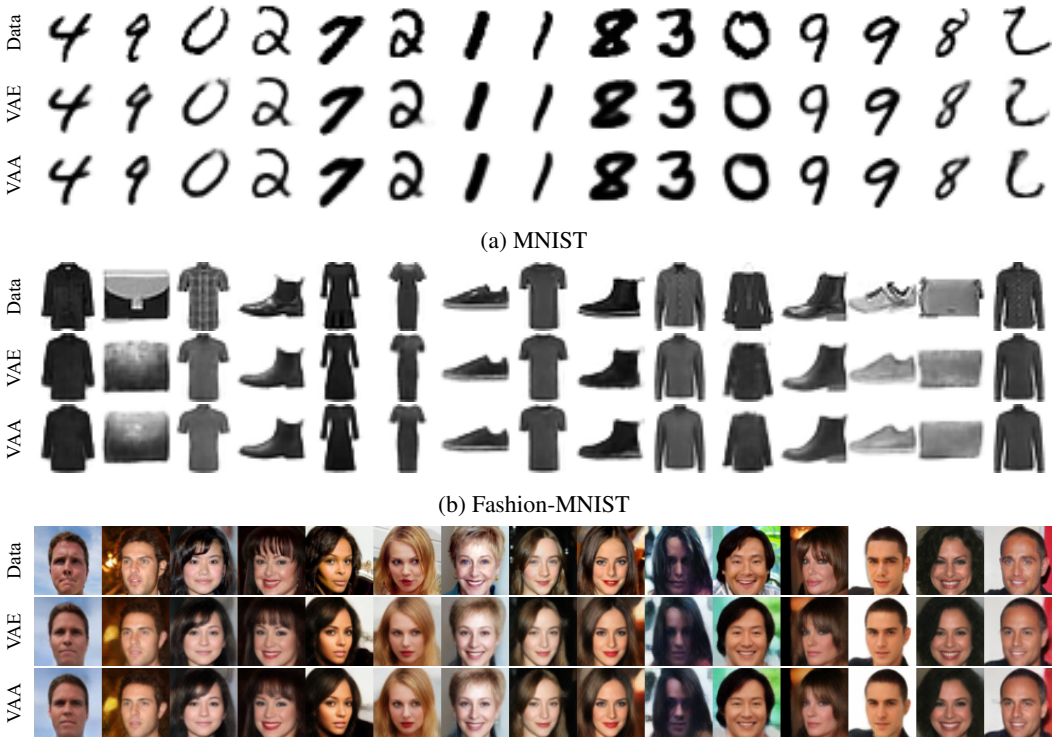
We present additional results for 10-chart variants of the variational atlas autoencoder trained on MNIST, Fashion-MNIST, and CelebA. Figure 7 depicts training samples and their reconstructions from the fully converged model. Figure 9 depicts samples drawn from each of the 10 charts within each VAA.

Choice of routing probability. Table 5 shows the effect of the routing function on the sample FID averaged over three seeds. The different variants of latent-evidence routing produce the best outcome in most cases. For our experiments, we choose prior-marginal routing, but we find that both μ-based and expected log-prior routing produce comparable results.

Reconstruction error. Figure 7 depicts additional reconstruction samples computed with our 10-chart VAA and compares them with those from the capacity-matched VAE.

Generated samples. Figure 9 includes additional samples generated from each of our charts and from a capacity-matched vanilla VAE baseline. Samples are drawn from each chart's prior at random. Sampling from our VAA model, only requires the sampled chart's weights to be active, resulting in faster sampling compared to our baselines which need all the model parameters.

Geometric interpolation results. Figure 8 shows interpolation trajectories over the double torus, including paths across chart boundaries through portals.



(a) MNIST

(b) Fashion-MNIST

(c) CelebA

Figure 7: 10-chart Variational Atlas Autoencoders (ours) and 1-chart capacity-matched VAE reconstructions of randomly selected held-out inputs from MNIST, Fashion-MNIST and CelebA.

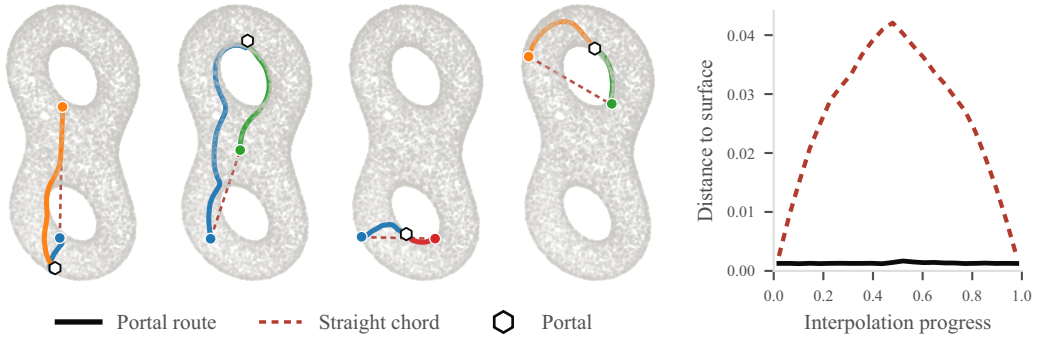
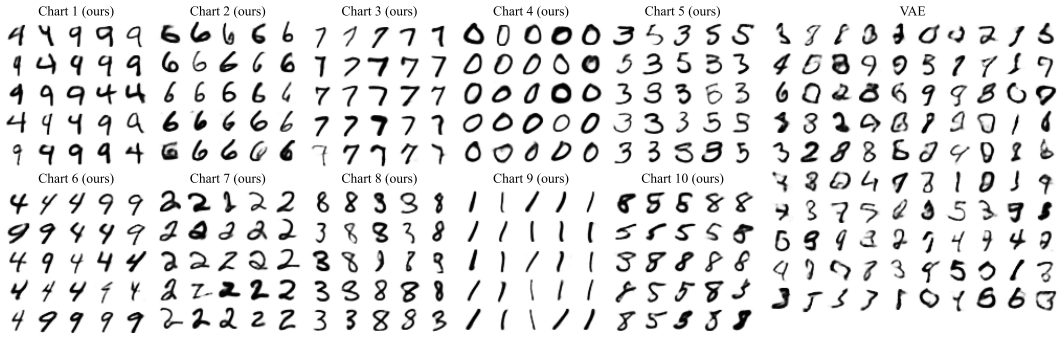
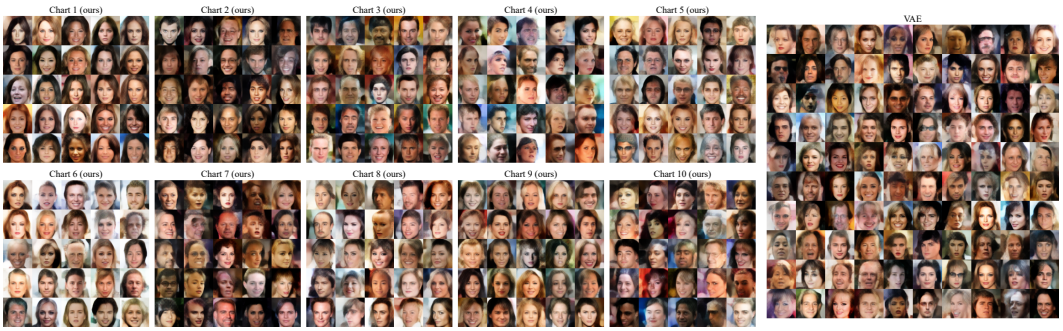


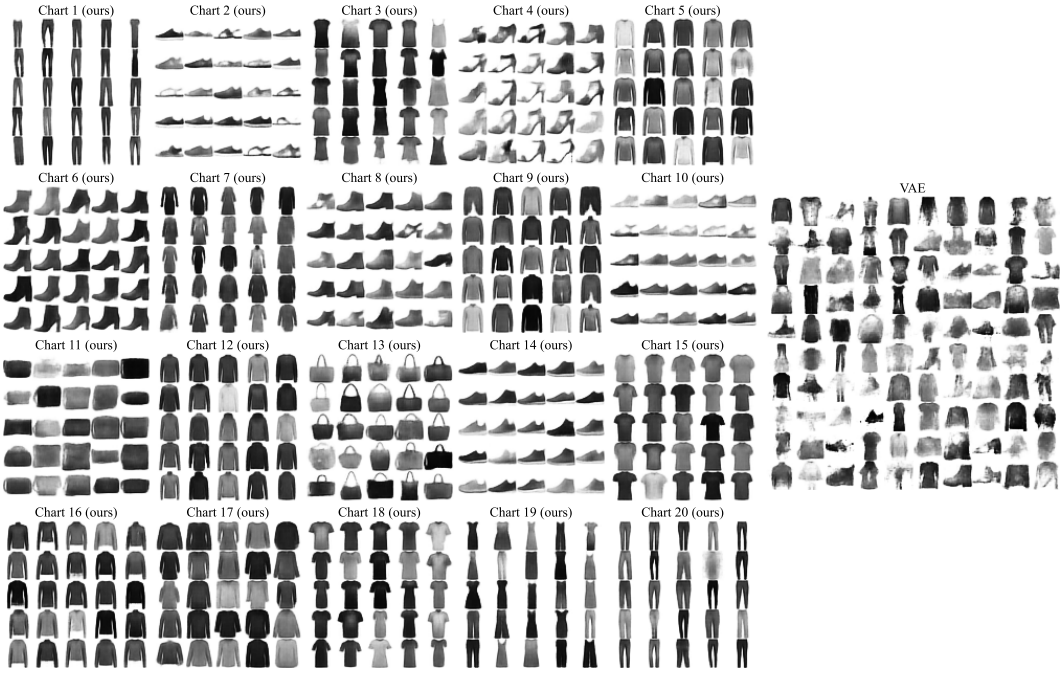
Figure 8: Interpolation trajectories between two charts through a portal on the double-torus geometry. For any two points on the torus, we can create interpolation trajectories in our charts' latent spaces that remain close to the torus surface (the data manifold). The red line illustrates straight-line interpolation through a global latent space, which leaves the surface and thus passes through implausible points. Optimizing portal placement to select shorter paths is left as future work.



(a) MNIST: 10 charts



(b) CelebA: 10 charts



(c) Fashion-MNIST: 20 charts

Figure 9: Sample images drawn from 10-, 10-, and 20-chart VAAs trained on MNIST, CelebA, and Fashion-MNIST, respectively. Each grid represents samples generated from a distinct chart. Samples from a capacity-matched VAE are shown on the right.