DRIFTING THROUGH ANNEALED CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Drifting models generate in a single forward pass, but learning coarse structure and fine detail together remains challenging. We turn soft clustering into a novel formulation of drifting that learns the data one scale at a time. Replacing the finite centroids of deterministic-annealing clustering with a neural generator yields a new likelihood objective: explain the data through a kernel-smoothed generated distribution. We characterize its optimum, showing that when exact matching is possible, the generator represents a sharpened version of the data while the kernel gives the remaining variation. Annealing the bandwidth progressively transfers this variation to the generator, turning broad clusters into increasingly detailed distributions. We derive a simple regression algorithm with expectation-maximization targets, in which generated samples compete to explain the data without explicit repulsion or manual cluster splitting, and use it to generate reliable samples on both synthetic multi-scale data (where drifting struggles) and realistic large datasets.

1 INTRODUCTION

Recent years have seen an explosion of progress in generative AI. Most models rely on the principle of mapping a simple prior distribution to a complex data distribution. These models typically generate samples by numerically solving an SDE or ODE, requiring multiple network evaluations as each sample is gradually constructed. This computational cost has motivated one-step and few-step generators, obtained either by distilling pretrained models or through direct training with flow-map, consistency-style, or velocity-matching objectives Luo et al. (2023); Yin et al. (2024); Salimans & Ho (2022). Drifting models (Deng et al., 2026) instead train a one-step generator by moving model samples along a kernel induced attraction-repulsion field. The kernel defines sample similarity and controls the strength of pairwise interactions. Recent work has connected drifting to score-based modeling for Gaussian kernels and developed kernel-gradient formulations that extend to more general kernels, Riemannian manifolds, and discrete data (Lai et al., 2026; Esteban-Casadevall et al., 2026).

Despite these developments, reliable training directly in data space remains challenging. A central difficulty is the choice of kernel temperature, which determines the spatial scale over which samples interact. At low temperatures, interactions become local, potentially providing weak signals between distant regions of the model and target distributions. At high temperatures, interactions span broader regions but may fail to resolve fine-scale structure. Existing high-performing image-generation implementations combine multiple temperatures with latent space generation to capture structure at different scales (Deng et al., 2026). Selecting these scales introduces additional design choices.

Annealing is an established statistical method Neal (2001) to start from a simple distribution, and to create a series of intermediate distributions leading to a distribution of interest. It has been applied in a variety of contexts as a method to organise learning from coarse to fine structure, including in variational inference Fogliani et al. (2026), and also in generative models Wu & Xie (2025). Previous work Brown et al. (2026) has attempted to resolve fine-scale structure in drifting with second-order methods, and Turan et al. (2026) has applied bandwidth annealing to conventional drifting.

Contributions. We approach this problem through the connection between drifting, classical clustering, and expectation-maximisation. By extending clustering algorithms to represent a continuous distribution of centroids, we derive a novel drifting formulation with a distinct loss function (without the "repulsive term" of the usual formulation) and a deterministic temperature-annealing schedule.

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.

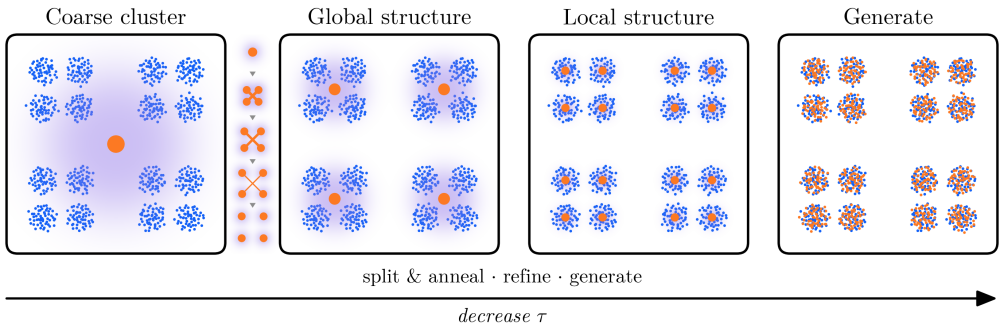

Figure 1: **Annealed cluster refinement.** As τ decreases, a single coarse cluster (orange center, shaded extent) repeatedly splits and anneals into increasingly local components, resolving global and then local structure of the data (blue), and the final components are used to generate samples (orange).

The schedule provides a principled progression from coarse to fine interactions, allowing the model to resolve structure across multiple scales during training. In experiments, this new method reproduces classical drifting's coarse-to-fine structure, captures multiple scales on synthetic data where drifting struggles, and generates healthy images from MNIST, FashionMNIST and CIFAR10.

2 Background

Drifting models

Drifting models (Deng et al., 2026) offer an exciting generative learning framework where samples generated by a neural network from a simple noise prior are evolved using optimisation on a stopgrad Mean-Squared Error (MSE) loss, under the action of a drifting field that attracts samples towards data, and repulses them from other samples.

More concretely, given a neural network f_θ and positive normalized kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{>0}$, drifting defines a discrepancy field

$$V_{p,q_\theta}(x) = V_p^+(x) - V_{q_\theta}^-(x),$$

where

$$V_p^+(x) = \underbrace{\frac{\mathbb{E}_{y\sim p}[k(x,y)(y-x)]}{\mathbb{E}_{y\sim p}[k(x,y)]}}_{\text{attraction to data}}, \qquad V_{q_\theta}^-(x) = \underbrace{\frac{\mathbb{E}_{x'\sim q_\theta}[k(x,x')(x'-x)]}{\mathbb{E}_{x'\sim q_\theta}[k(x,x')]}}_{\text{self-repulsion}}. \tag{1}$$

where q_θ is the model distribution, defined as the pushforward through the map f_θ of a simple structureless Gaussian

$$q_\theta = (f_\theta)_\# \mathcal{N}(0, I) \tag{2}$$

Thus, V_{p,q_θ} has an attractive-repulsive structure: it *attracts* model samples towards regions where the data distribution is underrepresented by the model, while *repelling* model samples from one another to promote coverage. During training, a noise sample $\epsilon \sim \mathcal{N}(0, I)$ is mapped to $x = f_\theta(\epsilon)$ and displaced according to

$$\widetilde{x} = x + \eta V_{p,q_\theta}(x),$$

where $\eta > 0$ is the step size. The generator is then trained to match these displaced samples using the stop-gradient loss

$$\mathcal{L}_{\text{drift}}(\theta) = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\|f_\theta(\epsilon) - \text{sg}(\widetilde{x})\|_2^2\right] = \eta^2\, \mathbb{E}_{x\sim q_\theta}\left[\|V_{p,q_\theta}(x)\|_2^2\right]. \tag{3}$$

Existing work Picard (2026) has shown that this standard formulation of drifting may be interpreted as an expectation-maximisation algorithm with the MMD^2 as the objective. We pursue a similar line of research to reformulate drifting itself, by optimising a new EM objective inspired by clustering.

3 CLUSTERING AND DRIFTING

Given a dataset, classical clustering approaches such as k-means (McQueen, 1967) provide an algorithm for assigning each point in the dataset to one of a discrete set of k clusters with centroids $\{x_j\}$. In such an approach, the aim is to give the data a simple structure by describing it with the cluster centroids, often as a preprocessing step for further algorithms.

3.1 REVIEW OF CLASSICAL CLUSTERING

Soft clustering methods, such as that of Rose, Gurewitz, and Foxin in Rose et al. (1990) (henceforth called ***RGF clustering***), do not assign each datapoint definitively to a cluster, but rather give a probability that a given datapoint y is assigned to a cluster with centre x_j, in the form $\rho(x_j|y)$. Just like the kernel $\kappa(x, y)$ in drifting, a natural choice of $\rho(x_j|y)$ is one which weights cluster points according to their distance from data points:

$$\rho(x_j|y) = \frac{\exp(-\beta d(x_j, y))}{Z(y)_\beta} \qquad Z(y)_\beta = \sum_{j=1}^{k} \exp(-\beta d(x_j, y)), \tag{4}$$

where $Z(y)_\beta$ is a partition function defined per data point y, $\beta := 1/T$ can be interpreted as an inverse temperature, and $d(x, y)$ is an appropriate distance metric. The temperature controls the scale at which the data clusters. Intuitively, $\rho(x_j|y)$ is often called the *responsibility*, as it is larger the greater the share of the data x_j explains.

The cluster positions themselves are evolved to minimize a free energy, i.e. an objective

$$F_{\text{cluster}} = -\frac{1}{\beta}\mathbb{E}_{y\sim p}\left[\log\left(\sum_{j=1}^{k}\exp[-\beta d(x_j, y)]\right)\right] . \tag{5}$$

Such a soft clustering approach therefore results in a description of the data with k centroids at a given temperature T. However, running a standard optimisation algorithm on F_{cluster} for general data $y \sim p$ at some fixed β can in general provide a poor description of the data. The problem is that F_{cluster} can be arbitrarily nonconvex as a function of the $\{x_i\}$, and so will have multiple local minima (Rose et al., 1990; Chen et al., 2024; Jin et al., 2016), distinct from the global minimum, which is the description of the data we seek.

One way to solve this problem is to anneal the temperature (Rose et al., 1990; Rose, 1998). In the high temperature limit $\beta \to 0$, it can be shown that there is a unique minimum of the free energy where all cluster centers accumulate at the global mean of the data. Gradually lowering the temperature so that β increases slowly from 0 to ∞, we remain on the same branch of minima[1] while the centroids spread apart, as in Figure 1. Note that in the limit of low temperature $\beta \to \infty$, the clustering becomes hard and the update reduces to the k-means update. There is no noise.

3.2 DRIFTING AND CLUSTERING

We now note that the collection of centroids can be thought of as defining a generative model (an idea suggested in other work such as Pan et al. (2026)). To make this precise, consider a limit where the number of cluster centroids k is taken to be very large; one should now imagine a continuous density of cluster centroids, where the probability of finding a cluster centroid at position x being given by a probability distribution $q(x)$. Below we will demonstrate how sampling from (a deformation of) $q(x)$ acts as a generative model.

The classical clustering algorithm thus corresponds to the case where $q(x)$ is

$$q_{\text{classical}}(x) = \sum_{j=1}^{k} \delta(x - x_j) \tag{6}$$

and is parametrized by a fixed number of centers $\{x_j\}$. In what follows, we will relax this assumption and allow an arbitrary $q(x)$. Generalizing equation 5 to the continuous case, we now consider

[1] In theory, assuming the anneal is slow enough that the system has time to equilibriate at every temperature.

the following free energy, which can be viewed as a function of the data distribution p and the model distribution q:

$$F_{\text{cluster}}(p, q) = -\frac{1}{\beta}\mathbb{E}_{y\sim p}\left[\log Z(y)_\beta\right] \qquad Z(y)_\beta = \mathbb{E}_{x\sim q} e^{-\beta d(x,y)} \tag{7}$$

This admits a natural expectation-maximization (Dempster et al., 1977) interpretation, similar to that of classical clustering Ueda & Nakano (1998): samples from q play the role of continuously distributed centroids, while the soft assignments introduced above act as responsibilities determining which parts of the data each centroid explains[2].

For the rest of this work we will specialize to the case where the distance measure $d(x, y) = \|x - y\|^2$. In this case we note that $Z(y)_\beta$ can itself be viewed as a probability distribution on y

$$q^\tau(y) = \left(\frac{\beta}{\pi}\right)^{\frac{d}{2}} Z(y)_\beta \tag{8}$$

corresponding to a smearing of the centroid distribution q by Gaussian noise of width $\tau := 1/\sqrt{2\beta}$. Throughout our work, we shall use both of the quantities τ, β according to convenience, alternating in this way between the drifting and clustering literature respectively. See Appendix B for further details on classical clustering.

Thus motivated, we introduce a new drifting loss, which is precisely the free energy above, and where the continuous centroid distribution $q = q_\theta$ is now defined in terms of a neural network f_θ mapping as in equation 2.

$$\mathcal{L}_{\text{cluster}}(\theta) = -\frac{1}{\beta}\mathbb{E}_{y\sim p}\left[\log\left(q_\theta^\tau(y)\right)\right] \qquad q_\theta^\tau(y) = (2\pi\tau)^{-\frac{d}{2}}\mathbb{E}_{x\sim q_\theta} e^{-\frac{\|x-y\|^2}{2\tau}} \tag{9}$$

For Gaussian kernels, this has a direct latent-variable interpretation: if $x \sim q_\theta$ and $y = x + \tau\epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$, then q_θ^τ is the marginal density of y, and $\mathcal{L}_{\text{cluster}}$ is, up to the factor $1/\beta$, its negative log-likelihood under the data distribution.

This objective is minimized as a function of θ when $p = q_\theta^\tau$, i.e. when the model distribution q smeared with Gaussian noise of strength τ is equal to the data distribution. The simplest way to see this is to note that the objective can equivalently be written as

$$\mathcal{L}_{\text{cluster}}(\theta) = \frac{1}{\beta} D_{KL}(p\|q_\theta^\tau) + \text{const.}, \tag{10}$$

where the additive constant measures the entropy of the data distribution and does not depend on θ. Thus only part of the variation of the data comes from the generator q_θ; part of it also comes from the noise τ. Equivalently, q_θ itself should be thought of as a *sharpened* version of the data distribution. Only at $\tau = 0$ is it exactly equal to the data distribution.

To use this loss in a drifting model, we anneal τ from τ_{max} to $\tau_{\text{min}} \approx 0$ with some schedule, precisely as in classical clustering. At each value of τ, we optimize $\mathcal{L}_{\text{cluster}}$ by adjusting the map f_θ. At inference time, we can generate samples in a single forward pass by sampling noise ϵ from a unit Gaussian, evaluating $f_\theta(\epsilon)$, and adding noise of magnitude τ_{min}. This provides a new formulation of drifting with a principled mechanism for choosing τ.

3.3 Implementation for drifting

Backpropagating through this clustering loss is computationally expensive in general. In implementation we use a slightly different formulation that uses stopgrad MSE loss as a proxy[3]

$$\mathcal{L}_{\text{eff}} = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{1})}\left[\left\|f_\theta(\epsilon) - \text{stopgrad}\left(f_\theta(\epsilon) + \mathbf{V}^+_{p,q_\theta}(f_\theta(\epsilon))\right)\right\|^2\right] \tag{11}$$

where we have a new definition of the attractive drifting field

$$\mathbf{V}^+_{p,q_\theta}(x) := \frac{\mathbb{E}_{y\sim p}\left[\rho(x|y)(y - x)\right]}{\mathbb{E}_{y\sim p}\left[\rho(x|y)\right]} \tag{12}$$

[2]Similar objectives have been proposed with hard nearest-sample dynamics Li & Malik (2019).

[3]While the definition of the clustering loss is specific to Gaussian kernels, i.e. $d(x, y) = \|x-y\|^2$, this MSE loss can be written with any choice of kernel, and may perform better with other choices in specific problems.

with $\rho(x \mid y)$ the continuous analogue of the assignment probability in classical clustering

$$\rho(x|y) := \frac{\kappa(x,y)}{\mathbb{E}_{x' \sim q_\theta}[\kappa(x',y)]} \qquad \kappa(x,y) \equiv e^{\frac{-\|x-y\|^2}{2\tau^2}} \tag{13}$$

Note the important absence of the repulsive term V^- found in standard drifting models as in equation 2. Instead, the information dynamics of clustering itself naturally induces an effective repulsion as centroids compete to describe the data [4].

The gradient of the MSE loss is equivalent to the gradient of the clustering loss in equation 9 up to a factor $\mathbb{E}_{x' \sim q_\theta}[\kappa(x',y)]$ dividing the former. Intuitively, the MSE loss weights each datapoint y's contribution to the expectation by how well the current estimate distribution covers this point and its neighbourhood[5].

3.4 Anisotropic smoothing at the end of clustering

From equation 9, at the conclusion of the clustering process - in either the original RGF form or our continuous generalization - we obtain a distribution q_θ of effective 'centroids' which we then in theory convolve with isotropic Gaussian noise of width τ to recover an estimate for the true data distribution p.

In practice the optimization procedure is not perfect and a perfect equilibrium is generally not reached at each step along the anneal. We find that the quality of samples is greatly increased by smoothing instead with *anisotropic* Gaussian noise at the end. In practice, for each sample x obtained from q_θ we compute the following local covariance matrix

$$C(x) = \frac{\sum_{i \in \mathcal{D}_b} \rho(x|y_i)(x-y_i)(x-y_i)^T}{\sum_{k \in \mathcal{D}_b} \rho(x|y_k)} \tag{14}$$

where the sum is taken over a small batch of the data $\mathcal{D}_b$ (usually taken for convenience to be the same size of the batch of samples being generated). We then sample a noise vector $\eta(x)$ from $\mathcal{N}(0, C(x))$ and take the final sample to be:

$$x \to x + \eta \tag{15}$$

See Appendix C for a theoretical discussion of the principle behind this anisotropic smoothing.

This method generates good samples, but also adds some overhead to the forward pass, requiring an extra step. Methods such as Golub & Van Loan (2013); Higham (2008) could be used to more efficiently perform the sampling of the noise vector. We also note that a plausible alternative route to produce higher-diversity samples while retaining the single pass is to run clustering-style loss until a set point, and thereafter carry on evolving the same model with standard drifting, whose repulsive field may be more adept at producing diverse samples. This effectively amortizes the cost of this anisotropic smoothing. We discuss this direction in the conclusion.

4 Experiments

Our experiments are designed to answer a series of questions establishing progressively more sophisticated properties of our clustering-style learning formalism. We proceed from comparison to classical clustering and reproducing its coarse-to-fine learning progression, to establishing that our method can learn very different scales in the same target (difficult for standard drifting), to generating samples with clustering-style loss on realistic datasets with simple architecture and kernels.

To implement the clustering-style loss, the main design choice is the temperature anneal, for which we broadly use an exponential decay (with beginning and ending τ chosen with reference to critical

[4] See Appendix B.1 for a remark on this.

[5] Theoretically, this means that badly assigned points in the MSE loss retain a greater influence through the training process, though in practice, we find that the MSE loss both is much quicker and works well regardless on real, high-dimensional data.

temperatures from classical drifting - see Appendix B), and an extended dwell at the floor of τ to learn the finest detail. We give results mainly from locally-adaptive smoothing in this section.

Beginning with Deng et al. (2026), it is standard to attempt to circumvent the choice of temperature by combining multiple drifting fields and losses across a range of temperatures: we call this the *ensemble* approach, and compare our method against it. We note in general that ensembles of τ are chosen for a particular problem largely by trial-and-error. More exhaustive searching is possible but expensive and time-consuming in comparison to the robust anneal shape of the clustering-style loss. Further details, including architecture, anneals, and setup, may be found in Appendix D.

4.1 Continuous clustering captures coarse-to-fine behaviour

First, we establish that continuous annealed clustering replicates the coarse-to-fine behaviour of classical discrete clustering. We start by training using ADAM on the standard 2D checkerboard target, but instead of learning a generative f_θ, we optimise simply on the positions of generated points x_i. This amounts to continuous clustering-style learning on particles.

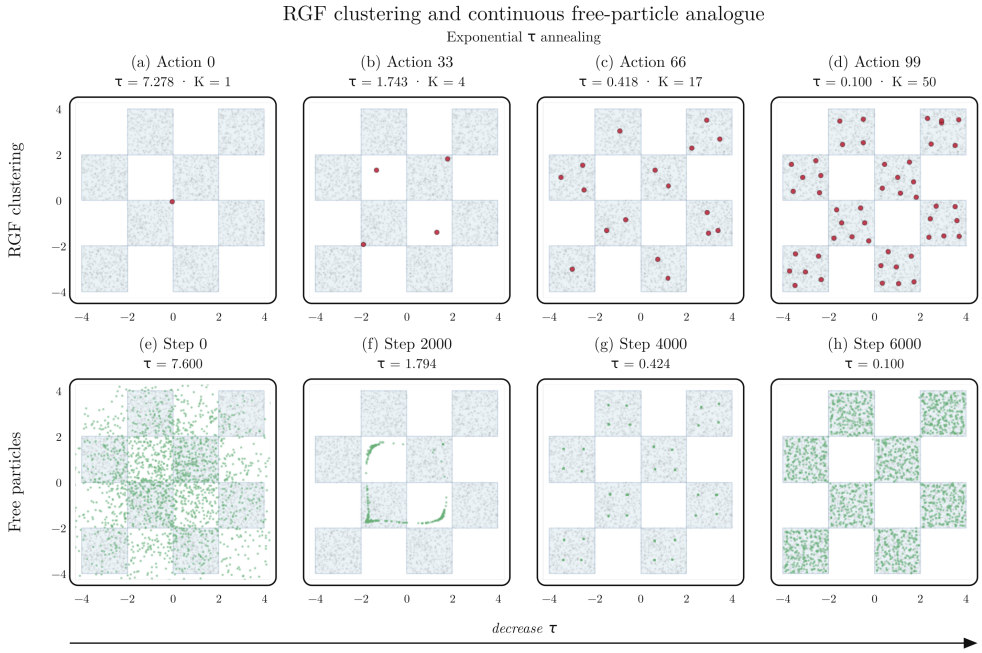

Figure 2: Evolution of clustering on a checkerboard target. 'Action' refers to a single temperature drop and equilibriation step of classical RGF clustering.

In Figure 2, both discrete RGF and our continuous annealed clustering resolve structure from coarse to progressively finer levels of detail. The splitting trajectory during classical clustering could be used to design the anneal for neural learning. We use this minimally in our implementations, but much more investigation is possible: see Appendix D.

4.2 A continuously clustering neural generator can resolve multiple scales

Next, we show that a learned network trained on the annealed clustering objective learns both global placement and local structure in a simple 2D dataset, where ordinary drifting struggles. We promote the usual checkerboard to two 2×2 checkerboards whose centres are at $(-6, 6)$ and $(6, -6)$, and whose cells are 2 units wide. There is thus a clear separation of intra- and inter-checkerboard scales.

This advantage of clustering-style learning is shown in the outputs in Figure 3. A summary of the metrics is presented in Table 1. The binary classifier C2ST metric is used for this simple dataset.

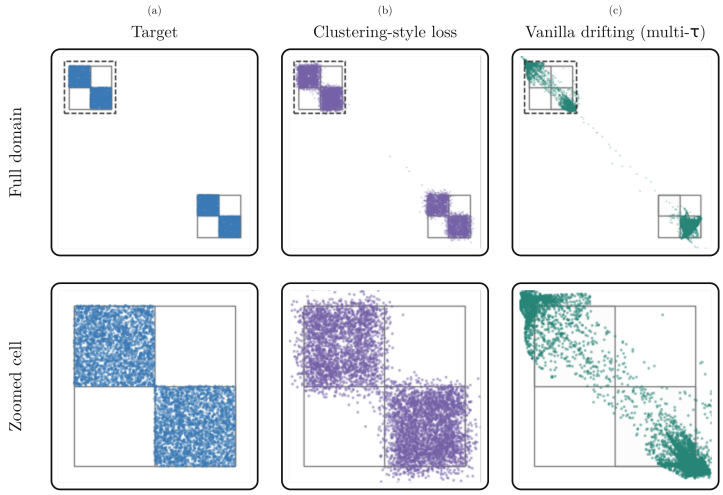

Figure 3: Results from a clustering-style loss with an exponential anneal ending at $\tau = 0.25$ and remaining there.

Table 1: Summary metrics for checkerboard of checkerboards

Method	Sliced W_2	MMD^2	Precision	Recall	C2ST (3 runs)
Clustering-style loss	**2.7494**	**0.0198**	**0.9180**	**0.9880**	**0.538 ± 0.004**
Ensemble drifting	2.9628	0.0455	0.7140	0.9100	0.793 ± 0.008

4.3 Annealed clustering generates images from realistic datasets

Finally, we show that neural networks trained on the annealed clustering objective can generate samples from realistic datasets with simple architecture and kernels. Our clustering-style loss scores comparably or better than drifting on FID in these experiments. These scores are inflated due to choices of encoder, which themselves have a floor reconstruction FID - see Appendix D. We believe that better FID scores for both methods are possible with more optimisation, but our controlled test with limited technology provides a proof of the viability of clustering-style loss.

4.3.1 MNIST in pixel space

We first apply both ensemble ordinary drifting and clustering-style learning to learn MNIST in pixel space, to test the approach on large dimension (784) and raw Euclidean geometry. The values of τ for the ensemble approach were chosen in consultation with the details of the original implementation in Deng et al. (2026).

Figure 4 shows outputs generated by the multi-τ ensemble across three values $\tau \in \{0.01, 0.09, 0.72\}$ against clustering-style learning (batch 2048 for both). Metrics are summarised in Table 2.

Table 2: Summary metrics for MNIST in pixel space (batch size = 2048)

Method	Sliced W_2	Precision	Recall	FID
Clustering-style loss, local smoothing	0.0040	**0.572**	**0.693**	**172.4**
Ensemble drifting	**0.0022**	0.086	0.597	254.4

4.3.2 FashionMNIST in latent space

Next, we run both formulations of drifting on a 20-D latent-space representation of FashionMNIST. Details of the encoding may be found in Appendix D.

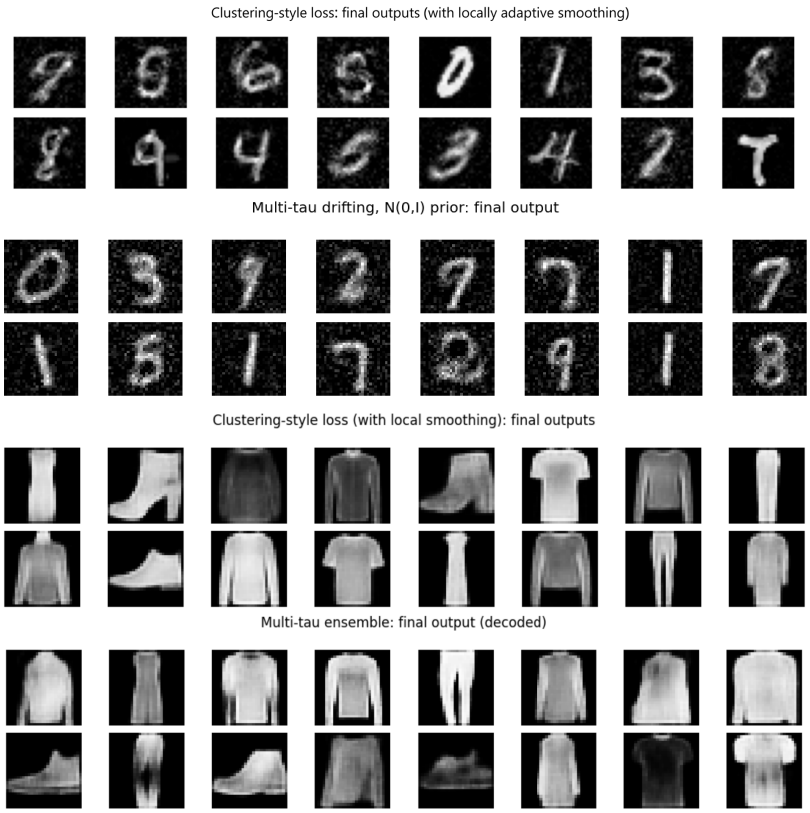

Figure 4: Results from MNIST pixel space, and FashionMNIST in 20-D latent space.

In Figure 4, we see that the clustering-style loss produces samples that are cleaner and free of the aberrations and discolorations present in part of the output of the ensemble of multiple τs. Statistics are summarised in Table 3, computed in pixel space rather than latent space for fairness.

Table 3: Summary metrics for FashionMNIST in latent space

Method	**Sliced W_2**	**Precision**	**Recall**	**FID**
Clustering-style loss, local smoothing	0.0220	**0.838**	0.764	53.99
Ensemble drifting	0.0208	0.559	**0.845**	55.56

4.3.3 CIFAR10 IN LATENT SPACE

Finally, we run the clustering-style loss with a more sophisticated DiT architecture on the CIFAR10 dataset in latent space, against an ensemble of multiple τs. The outputs are shown in Figure 5. Summary metrics are presented in Table 4, as before, measured against raw data pixels.

Table 4: Summary metrics for CIFAR10 in latent space

Method	**Sliced W_2**	**Precision**	**Recall**	**FID**
Clustering-style loss, local smoothing	**0.0300**	**0.8520**	**0.7627**	**99.25**
Ensemble drifting	0.0416	0.1020	0.4327	137.918

4.3.4 ABLATIONS: ANNEAL, CHOICE OF SMOOTHING AND BATCH SIZE

We use MNIST in pixel space as a case study to determine the effects of various design choices. Firstly, exponential annealing on standard drifting alone does not improve performance in our ex-

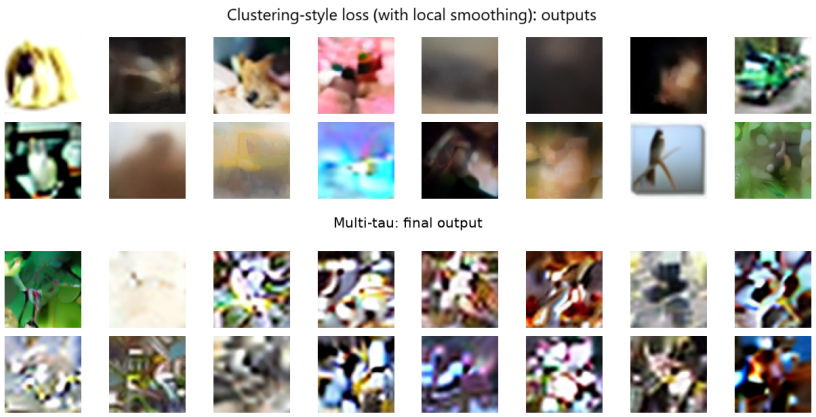

Figure 5: Results from CIFAR10 in latent space.

periments; instead, it makes it actively harder for drifting to learn a moving target. An example of collapse resulting from this can be seen at batch size 256 in Figure 9 in Appendix D.2.

Further, Table 5 shows that clustering-style learning is fairly robust with respect to batch size: the key metrics barely change. Meanwhile, traditional drifting deteriorates more. We also see at this batch size how important locally adaptive smoothing is for sample diversity and quality.

Table 5: Isotropic vs local smoothing for MNIST in pixel space

Method	**Batch**	**Sliced W_2**	**Precision**	**Recall**	**FID**
Clustering-style loss, local smoothing	2048	0.0040	0.572	0.693	172.4
Ensemble drifting	2048	0.0022	0.086	0.597	254.4
Clustering-style loss, local smoothing	256	0.0039	0.513	0.623	181.0
Clustering-style loss, isotropic smoothing	256	0.0061	0.657	0.200	274.7
Ensemble drifting	256	0.0026	0.001	0.443	283.2

A possible explanation for this phenomenon is that a sufficient batch size is required for repulsion between samples to have influence. Our clustering-style method, free of traditional repulsion, does not seem to suffer as measurably from this problem, based on this limited set of experiments.

5 Conclusions and Future Directions of Research

We have shown that a continuous version of annealed clustering with an EM-inspired objective leads to a novel reformulation of drifting models. With only a choice of temperature annealing to be made, this method reliably learns multi-scale data and realistic datasets. Our work has laid out the applicability and soundness of drifting through annealed clustering; we expect that by combining this new loss with other architectural optimizations we can push this method towards existing benchmarks and analyse its performance with more capable architectures.

In high dimensions, sufficient diversity needs to be carefully ensured across the entirety of the data distribution. In our experiments, we used local smoothing at the end to counteract this limitation, which overhead to the single forward pass for generation. However, the annealed clustering method could also be used as a guide to first resolve the data up to a suitably fine scale, before using standard drifting with its repulsion mechanism to finish learning and generating diverse samples (or another neural-network-based method, rather than the KDE method we now have). We believe such a two-phase drifting paradigm may allow drifting to reach better performance on many targets, and that it would be interesting to investigate its properties.

More fundamentally, clustering was shown to reveal scale structure in the data in simple examples. It would be interesting to understand these phenomena more thoroughly, both for theoretical interest and to guide learning by improving the anneal.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets, formulate mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat datasets, and to support qualitative and thematic data analysis.

We have not used generative AI tools to help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, or to interpret results. Assisting with translation is not applicable to this work.

Additionally, we used generative AI tools to create or modify scientific figures or images, suggest experimental parameters, create or edit software code, summarize or analyse existing literature, brainstorm, sourcing/searching for information, identify relevant literature. We have reviewed all AI-assisted work. Suggestions regarding locally adaptive smoothing were derived independently and theoretically motivated by authors. LLM-generated code was verified for correctness by authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper's aim is to advance understanding of generative models in the field of Machine Learning. There are potentially societal consequences of this work, none of which we feel are within the scope here.

REPRODUCIBILITY STATEMENT

Code that produced the majority of the experimental results in this notebook is attached as a zip file with the submission. To reproduce further experiments, such as ablations, Appendix D provides details and motivations for all the experiments in this paper.

REFERENCES

Drake Brown, Yuhao Huang, Shih-Hsin Wang, and Bao Wang. Second order drifting models, 2026. URL https://arxiv.org/abs/2608.07924.

Yudong Chen, Dogyoon Song, Xumei Xi, and Yuqian Zhang. Local minima structures in gaussian mixture models. *IEEE Transactions on Information Theory*, 70(6):4218–4257, 2024. doi: 10.1109/TIT.2024.3374716.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22, 1977.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Maria Esteban-Casadevall, Jorge Carrasco-Pollo, Max Welling, Jan-Willem van de Meent, Erik J Bekkers, and Floor Eijkelboom. Kernel-gradient drifting models. *arXiv preprint arXiv:2605.10727*, 2026.

Luigi Fogliani, Bruno Loureiro, and Marylou Gabrié. Annealing in variational inference mitigates mode collapse: A theoretical study on gaussian mixtures, 2026. URL https://arxiv.org/abs/2602.12923.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations - 4th Edition*. Johns Hopkins University Press, Philadelphia, PA, 2013. doi: 10.56021/9781421407944. URL https://epubs.siam.org/doi/abs/10.1137/1.9781421407944.

Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008. ISBN 978-0-898716-46-7. doi: 10.1137/1.9780898717778.

Chi Jin, Yuchen Zhang, Sivaraman Balakrishnan, Martin J. Wainwright, and Michael I. Jordan. Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. In *Neural Information Processing Systems*, 2016. URL https://api.semanticscholar.org/CorpusID:3200184.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/0234c510bc6d908b28c70ff313743079-Paper.pdf.

Chieh-Hsin Lai, Bac Nguyen, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, Stefano Ermon, and Molei Tao. A unified view of score-based and drifting models. *arXiv preprint arXiv:2603.07514*, 2026.

Ke Li and Jitendra Malik. Implicit maximum likelihood estimation, 2019. URL https://openreview.net/forum?id=rygunsAqYQ.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 76525–76546. Curran Associates, Inc., 2023. doi: 10.52202/075280-3344. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f115f619b62833aadc5acb058975b0e6-Paper-Conference.pdf.

James B McQueen. Some methods of classification and analysis of multivariate observations. In *Proc. of 5th berkeley symposium on math. stat. and prob.*, pp. 281–297, 1967.

Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

Yuangang Pan, Yinghua Yao, Atsushi Nitanda, Joey Tianyi Zhou, and Ivor W. Tsang. Auto-clustering with continuous distribution estimation on centroids. *Mach. Learn.*, 115(4):86, 2026. doi: 10.1007/S10994-025-06973-0. URL https://doi.org/10.1007/s10994-025-06973-0.

David Picard. An expectation-maximization interpretation of generative modeling via drifting. 2026. URL https://davidpicard.github.io/pdf/An_Expectation_Maximization_interpretation_of_Generative_model_drifting.pdf.

K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *IEEE Proc.*, 86(11):2210–2239, 1998. doi: 10.1109/5.726788.

Kenneth Rose, Eitan Gurewitz, and Geoffrey Fox. A deterministic annealing approach to clustering. *Pattern Recognition Letters*, 11(9):589–594, 1990.

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/f7696a9b362ac5a51c3dc8f098b73923-Paper.pdf.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Erkan Turan, Nicolas Dufour, and Maks Ovsjanikov. Generative drifting is secretly score matching: a spectral and variational perspective, 2026. URL https://arxiv.org/abs/2603.09936.

Naonori Ueda and Ryohei Nakano. Deterministic annealing em algorithm. *Neural Networks*, 11(2):271–282, 1998. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(97)00133-0. URL https://www.sciencedirect.com/science/article/pii/S0893608097001330.

Dongze Wu and Yao Xie. Annealing flow generative models towards sampling high-dimensional and multi-modal distributions. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=iT8qIzNSAl.

Tianwei Yin, Michael Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

A Optimum of clustering loss

Here we demonstrate directly that the clustering loss has an extremum when the noised model distribution is equal to the data distribution. We note that this can also be directly read off from the positive-definiteness of the KL divergence in the formulation of the loss in equation 10, but we feel a direct computation is still instructive.

Our starting point is equation 7, which we write explicitly in terms of integrals over the data and model distributions $p(y)$ and $q(x)$:

$$F_{\text{cluster}}(p,q) = -\frac{1}{\beta}\int dy p(y)\left[\log Z(y)_\beta\right] \qquad Z(y)_\beta = \int dx q(x) e^{-\beta\|x-y\|^2} \tag{16}$$

Let us now vary the free energy with respect to $q(y)$ in an attempt to find the extrema of the free energy. We supplement the system with a Lagrange multiplier to enforce normalization of the probability distribution $q(y)$

$$\delta_q\left[F(p,q) + \lambda\left(\int dx q(x) - 1\right)\right] = 0 \tag{17}$$

This gives the functional equation

$$-\frac{1}{\beta}\int dy \frac{p(y)}{Z(y)_\beta}\exp(-\beta\|x-y\|^2) + \lambda = 0 \tag{18}$$

which must hold for every yx, and where the q dependence is implicitly inside $Z(y)$. To find a solution we note that the term in λ is x-independent, and thus the first term must also be. This will happen ifthe ratio $p(y)/Z(y)$ is independent of y. Thus we have

$$Z(y)_\beta = p(y)c \tag{19}$$

which becomes

$$\int dy q(y)\exp\left(-\beta|x-y|^2\right) = p(y)c \tag{20}$$

where the value of c will be fixed by normalization and ultimately determines the saddle-point value of λ. Thus, as claimed, at the extremum of the free energy, the model distribution $q(y)$ convolved with a Gaussian of width controlled by β is equal to the data distribution $p(y)$.

B Details on classical clustering

Classical RGF clustering with deterministic annealing in Rose et al. (1990), Rose (1998) operates in the following manner. The anneal begins at a high temperature (hence large τ, low β) and updates the centroid positions x_i using fixed-point iteration obtained from the gradient of the free energy objective F in equation 5:

$$\frac{\partial F}{\partial x_i} = \mathbb{E}_{y\sim p}\left[\frac{(x_i - y)e^{-\beta d(x_i,y)}}{\sum_{j=1}^k e^{-\beta d(x_j,y)}}\right] \tag{21}$$

Rearranging which the natural fixed-point update is derived:

$$x_i^{(n+1)} = \mathbb{E}_{y\sim p}\left[\frac{y e^{-\beta d(x_i^{(n)},y)}}{\sum_{j=1}^{k} e^{-\beta d(x_j^{(n)},y)}}\right] \tag{22}$$

Once these fixed-point updates have converged to a stable state at a given β, Rose (1998) derives a criterion for computing the next critical (inverse) temperature β_c at which it becomes energetically favourable for a centroid to split into two. The second derivative of the free energy under such a perturbation can be calculated, and the moment it is zero, a split will become favourable. A calculation shows that this occurs when

$$\beta_c^{-1} = 2\lambda_{\max} \tag{23}$$

where $\lambda_{\max}$ is the largest eigenvalue of the weighted covariance matrix

$$C_{y|x} = \sum_y p(y|x)(x-y)(x-y)^T \tag{24}$$

These eigenvalues can be computed for each generated point x, from which the lowest β_c is then chosen as the value to next anneal to. The centroid in question splits in two, and the cycle then repeats, with equilibriation and further β_c calculations, until, in most implementations, we reach either a desired temperature or a desired number of clusters.

A large gap between β_{crit}^k and $\beta_{\text{crit}}^{k+1}$ in the course of the anneal indicates the presence of a plateau where a large injection of noise is required to cluster the data more finely. This corresponds to particularly stable, accurate descriptions of the data.

This sort of structure ostensibly offers an important guide for how to schedule clustering in the continuous generative learning case. Near such transition points, where the classical deterministic annealing algorithm has long plateaus (we see in Figure 2, for example, that there is one around 8 clusters, corresponding to the 8 checkerboards), the temperature anneal in our training process should also be appropriately slow, to ensure equilibriation near the critical temperature. Elsewhere, it can be faster.

In practice, however, with the sort of high-dimensional targets generative models are often run on, we find that there seldom exist such clear gaps between different scales in the data, so a slow exponential anneal is usually sufficient, ending in a small value of τ at which we judge the learning of the target to be fine enough. We choose this value per specific dataset, and it is important in higher dimensions after the exponential anneal to allow training fixed at this floor value of τ for some time at the end to ensure learning of the finest scale of detail.

Note that the covariance matrix is the same object we use to locally smooth our data at the end. It provides the key to understanding why we need to do so at all. If we finish at τ_{floor}, after an extended period of equilibriation, the maximal eigenvalues of the covariance matrix would all necessarily be lower than τ^2, since otherwise a further splitting event could occur. This means that the required smoothing in any direction would never exceed τ.

However, due to the limitations of a real implementation - namely, that equilibrium is not always reached before the anneal continues - we end up needing to smooth by more than τ in some directions. At the small scales of τ_{floor}, it is also conceivable that equilibriation begins to take far longer. We leave this line of inquiry open for future investigation.

B.1 Competition in clustering

The responsibility term in classical clustering reweights attraction to the data while preserving a centroid-directed update. To see this in our generative modelling context, define the probability measure

$$\pi_{\mathbf{x}}(d\mathbf{y}) := \frac{\rho(\mathbf{x} \mid \mathbf{y})}{\mathbb{E}_{\mathbf{z}\sim p}\left[\rho(\mathbf{x} \mid \mathbf{z})\right]}\, p(d\mathbf{y}). \tag{25}$$

Then the attractive field can be written as

$$\mathbf{V}^+_{p,q_\theta}(\mathbf{x}) = \mathbb{E}_{\mathbf{y}\sim\pi_{\mathbf{x}}}[\mathbf{y}] - \mathbf{x}. \tag{26}$$

Thus, the update target is a weighted centroid of the data distribution. In particular, for bounded target support,

$$\left\|\mathbf{V}^{+}_{p,q_\theta}(\mathbf{x})\right\| \leq \sup_{\mathbf{y}\in\mathrm{supp}(p)} \|\mathbf{y}-\mathbf{x}\|\,. \tag{27}$$

For fixed $\mathbf{x}$, this bound is independent of the kernel bandwidth: annealing changes the relative responsibilities but cannot make the normalized field arbitrarily large.

B.2 Splitting structure in classical clustering

We here reproduce the loss and splitting trajectory plots for classical and free-particle neural clustering on a checkerboard.

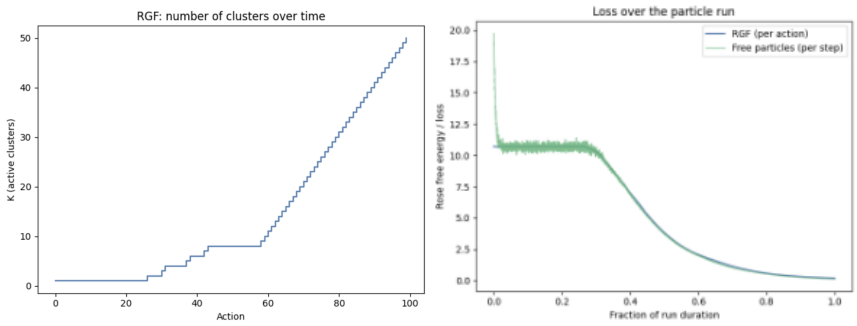

Figure 6: Evolution of classical clustering free energy/loss, and number of clusters, on a checkerboard.

Note from Figure 6 that the loss function (the clustering free energy) evolves more or less identically in both discrete and continuous versions. The long plateau in the loss and the number of clusters is suggestive and suggests a sharp transition between macro-structure (the 8 cells) and finer detail. We aim to investigate this behaviour in future work.

C Estimator of the data distribution

In the RGF clustering formalism of Rose et al. (1990), the responsibility of centroid x_i for datapoint y_j is

$$\rho(x_i|y_j) = \frac{e^{-d(x_i,y_j)}}{\sum_l e^{-d(x_l,y_j)}} \tag{28}$$

Let us now consider that we are given a fixed set of N datapoints y_j, and and the assignment of the x_i is probabilistic. The data distribution $p(y)$ is then the empirical distribution

$$p(y) = \frac{1}{N}\sum_{k=1}^{N}\delta(y-y_k) \tag{29}$$

Thus we have

$$\rho(y|x_i) = \frac{\rho(x_i|y)p(y)}{p(x_i)} \tag{30}$$

Furthermore, given the responsibilities (which can be viewed as conditional probabilities), we have:

$$p(x_i) = \int dy \rho(x_i|y)p(y) = \frac{1}{N}\sum_{k=1}^{N}\rho(x_i|y_k) \tag{31}$$

Inserting this and the empirical distribution equation 29 into equation 30 we have

$$\boxed{\rho(y|x_i) = \frac{\sum_j \rho(x_i|y_j)\delta(y-y_j)}{\sum_k \rho(x_i|y_k)}} \tag{32}$$

This is the true data estimator given centroids $\{x_i\}$. Taking the covariance matrix of this conditional distribution therefore gives exactly C_j, the estimated covariance matrix in equation 14. The idea is that we estimate what the covariance of the conditional distribution of a generated point *would* be if it were among the N datapoints.

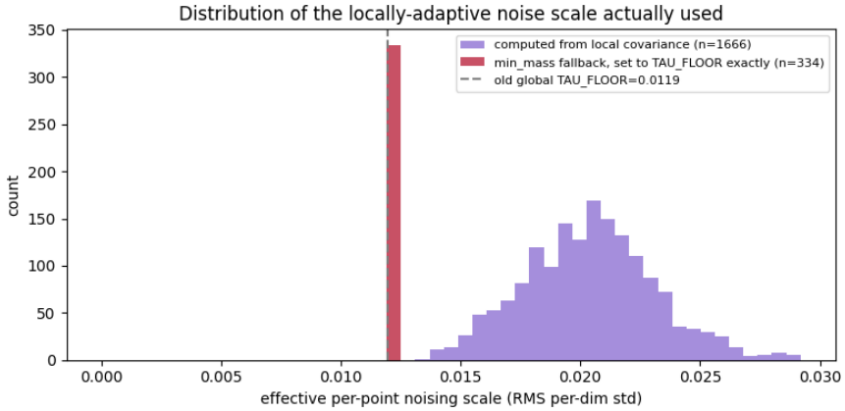

Figure 7: Distribution of smoothing noise added under locally adaptive scheme, on MNIST pixels.

In practice, smoothing using the covariance matrix of this conditional distribution results in noising in certain directions often much greater than τ, and in other directions less than τ (for which we cap off at a lower limit that is the finishing value of $\tau = \tau_{\text{floor}}$, since these components are noised by τ_{frloor} anyway). The resulting adaptive smoothing is shown for the end of the MNIST pixel space run in Figure 7.

D FURTHER DETAILS ON EXPERIMENTS

Experiments for each dataset are conducted with both clustering-style (MSE proxy) loss and drifting with an ensemble of fixed τ values. Both of these are calculated using a Gaussian kernel, i.e. $d(x, y) = \|x - y\|^2$. For each dataset, the network architecture for the model (deliberately simple, but MLPs suffice to learn most of the datasets reasonably well and are easy to train), the number of steps, the optimiser and the learning rate scheduling are kept the same to ensure a fair comparison between the methods.

In each experiment, the choice of generative prior is found to play an important role. For the clustering-style loss, it is essential that at the beginning of the anneal, the distribution is initialised to a prior that is centred around the mean of the dataset, a computation that is cheap and places the process at the start of classical RGF clustering. without it, the generated probability mass swiftly begins to develop imbalances.

However, starting at this point turns out to lead to collapse for ensemble drifting in almost every experiment. Rather, for those experiments, the best results were found when the prior had a large spread, and usually not centred around the data mean. We later explain initialisation for each of the experiments in turn.

Table 6: Details of experimental setups.

Dataset	Dim.	Optimiser	Number of steps	Architecture
Checkerboard of checkerboards	2	Adam	12000	MLP, 6 hidden layers, width=512
MNIST (pixel)	784	Adam	5000	MLP, 3 hidden layers, width=768
Fashion MNIST (latent)	20	Adam	9000	MLP, 3 hidden layers, width=384
CIFAR10 (latent)	256	AdamW	15000	DiT, patch=1, hidden=384, depth=12, heads=6, MLP ratio=4.0

An Adam optimiser is used consistently, only being replaced with an AdamW for CIFAR10, where it is the standard choice for the DiT architecture used on that dataset.

In these experiments, we use several variants on the same idea of a simple exponential τ schedule. The two really important variables are therefore the initial τ_{init} and final τ_{floor} values. These were chosen in ways detailed for each of the specific experiments later.

The important common ingredient in higher dimensions is to spend a period of time at τ_{floor} after the end of the anneal, to allow equilibriation and thus learning of detail at the smallest scale.

In some implementations, we also used the theoretical insight of critical splitting temperatures from Appendix B to calculate the highest (so first encountered in the anneal) value of τ at which a split in classical RGF clustering occurs. This τ_{crit} was then used as a marker, since before it learning can be fast while the generated distribution remains roughly at the centre of mass of the data. Afterwards, learning needs to be slower as structure is beginning to be learned.

This two-rate anneal - an initial faster burst followed by a slower long process - is, as far as our experiments show, generally of entirely comparable performance to a single exponential decay between τ_{init} and τ_{floor} within the same budget. We used it nevertheless in some experiments as an equal alternative, and it may be interesting to explore optimising the schedule further in future.

An example of a scheduling combining all these features is that for the FashionMNIST run, whose anneal is shown in Figure 8. In general, the shape of the anneal could plausibly be much better optimised and we leave this open for future work.

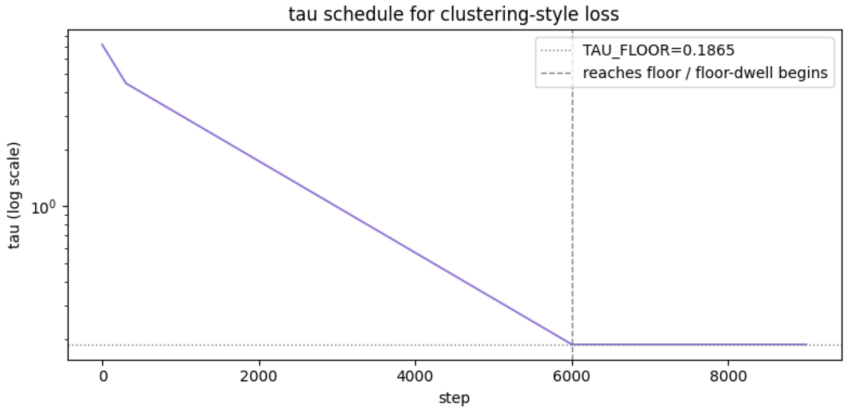

Figure 8: An example of the exponential temperature anneal with an initially faster period, for FashionMNIST.

To calculate Precision and Recall metrics for all experiments, the standard approach for generative models from Kynkäänniemi et al. (2019) is used, itself following the paradigm proposed by Sajjadi et al. (2018).

D.1 CHECKERBOARD OF CHECKERBOARDS

The schedule here is 12000 steps of exponential decay, with a short period of faster decay at the beginning, before the first τ_{crit}. No dwell at $\tau_{\text{floor}} = 0.25$ was needed here, as opposed to higher dimensions, where it is actively necessary. The prior is also simply $\mathcal{N}(0, I)$ for both methods - it performs adequately for annealed clustering here on account of the dataset's exact symmetry. For the ensemble drifting, $\tau \in \{0.25, 8.63\}$ was used.

Cosine annealing for the learning rate is used for both runs, with a batch size of 512. Data samples are generated through sampling from a sum of uniform distributions in 2D since the data distribution is here conveniently exact. Another consequence of such exact structure is that uniform (isotropic) smoothing by τ at the end suffices to learn the data distribution to a good degree of accuracy, and in any case locally adaptive smoothing is not particularly informative at such low dimension.

D.2 MNIST IN PIXEL SPACE

For MNIST in pixel space, we first perform a light preprocessing of the data in line with standard implementations, such as those in Deng et al. (2026). Raw pixels are rescaled to lie in $[0, 1]$, flattened from 28×28 to dimension 784, and then L^2 normalised: each 784-dimensional vector $\mathbf{x}$ is divided by its own L^2 norm $|x|^2$. No class-conditioning is used in the model.

Here, the clustering-style run is initialised around the dataset mean as $\mathcal{N}(\mu, 0.0001)$, since clustering collapses very quickly to the data mean at high temperature anyway. The schedule consists of 3000 steps of exponential annealing, with an initial faster burst of 150 steps to reach $\tau_{\text{crit}} = 0.2387$ (having started from $3\tau_{\text{crit}}$), followed by 2000 steps of dwelling at $\tau_{\text{floor}} = 0.05\tau_{\text{crit}}$.

Meanwhile, drifting is carried out with an ensemble spanning several scales $\tau \in \{0.0012, 0.24, 0.72\}$, which were found after trial-and-error through a coarse sweep of values (most of which led to unmitigated collapse). The prior is the standard $\mathcal{N}(0, I)$ - initialising at a dataset mean-centred normal also led to collapse.

The model in both runs was identity-initialised, to avoid collapse in the case of random initialisation (since both methods are sensitive to the choice of prior).

Figure 9 shows the collapse resulting from a temperature anneal applied to ordinary drifting.

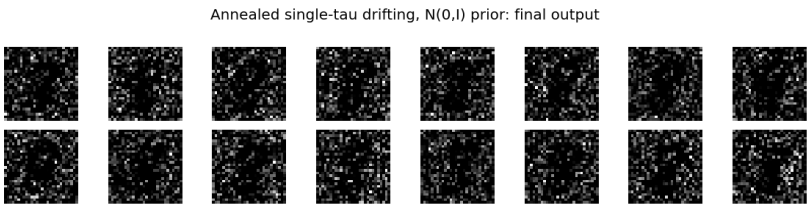

Figure 9: Run of MNIST in pixel space with standard drifting, exponential τ anneal.

D.3 FASHIONMNIST IN LATENT SPACE

To encode FashionMNIST, a VAE with a slightly weaker KL term is used to accurately capture the geometry of the underlying data. Sample reconstructions may be seen in Figure 10. The latent space is 20-dimensional, and reconstruction from this latent space to raw pixels has a base FID of 49.57, before learning is then conducted on it. No class-conditioning is used in the model.

The schedule for clustering-style learning consisted of two-rate exponential annealing for 6000 steps with an initial 300-step burst to $\tau_{\text{crit}} = 3.73$, followed by 3000 steps of dwelling at $\tau_{\text{floor}} = 0.05\tau_{\text{crit}}$, in the same prescription as we consistently use.



Figure 10: Sample reconstructions of FashionMNIST data samples using VAE.

Both clustering and drifting runs here were found to work well with the same shared prior of $\mathcal{N}(\mu, 2.54)$, making the comparison especially controlled. (Starting drifting from $\mathcal{N}(0, I)$ produced identical results, in all likelihood due to the low latent dimension). Drifting was carried out with an ensemble, in analogy with the MNIST run, weighted relative to τ_{crit} (and found to be the best of a coarse sweep of candidates): $\tau \in \{0.05\tau_{\text{crit}}, \tau_{\text{crit}}, 3\tau_{\text{crit}}\} = \{0.186, 3.73, 11.2\}$.

The model in both runs was again identity-initialised, to avoid collapse. Experimental results everywhere are produced with a batch size of 256.

In latent space, the diversity shortfall of isotropic smoothing at the end of training is more evident than in pixel space, perhaps due to the concentration of information in each of the dimensions. For FashionMNIST, we can see this shortfall particularly clearly by plotting the recall per class with isotropic τ-smoothing versus local smoothing. These may be seen in Figure 11, and confirm the necessity of locally adaptive smoothing to capture the diversity of the data distribution in latent space, even though both methods have high precision and generate high-quality samples.

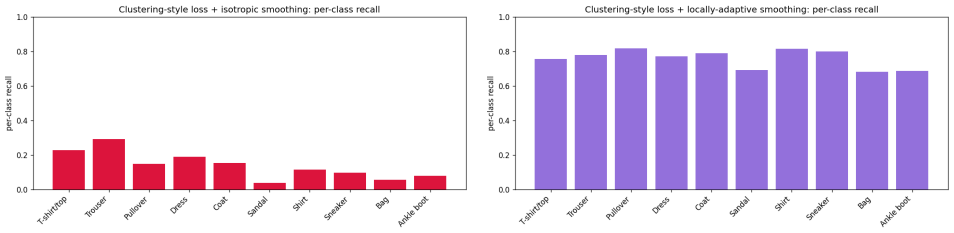

Figure 11: Clustering-style learning on FashionMNIST: per-class recall with isotropic vs adaptive smoothing.

D.4 CIFAR10 IN LATENT SPACE

In this experiment, a much larger batch size of 8192 was used to generate results due to the much more complex dataset, and the need for detail. Ablations were performed at batch size 2048 and 1024, both of which led to no improvement in performance.

The encoding used in this experiment was from the standard Stable Diffusion VAE, imported from HuggingFace at `https://huggingface.co/stabilityai/sd-vae-ft-mse`, into a $4 \times 8 \times 8$ latent space, flattened into dimension 256. This encoding has a base FID$= 51.12$, before any learning is conducted on it. Sample reconstructions of real data samples using this encoder are shown for reference in Figure 12.

Here, $\tau_{\text{crit}} = 4.707$, and a single exponential anneal starting as in all our experiments at $3\tau_{\text{crit}}$ and ending at $\tau_{\text{floor}} = 0.10$, found here after a coarse sweep to produce the best performance. The anneal lasts 9000 steps, followed by 6000 steps of dwelling at τ_{floor}. After a coarse sweep, best performance at $\tau_{\text{floor}} = 0.10$. No class-conditioning is used in the model. The prior used was $\mathcal{N}(\mu, 0.0865)$.

For usual drifting, on the other hand, the usual prior of $\mathcal{N}(0, I)$ was found to still not be wide enough to prevent total collapse, so a much-widened prior of $\mathcal{N}(0, \tau_{\text{crit}}^2)$ was used, together with the same ensemble of $\tau \in \{0.05\tau_{\text{crit}}, \tau_{\text{crit}}, 3\tau_{\text{crit}}\}$ (again, tested over a coarse sweep).



Figure 12: Sample reconstructions of CIFAR10 data samples using SD-VAE encoder.

Note that the results presented in the body of the text are from training a generative model without any class-conditioning (a feature of many implementations). To show the clustering-style loss producing more recognisable high-quality images, in an attempt to capture some of the presumable advantage of class-conditioning, we also train the same model with the same training parameters on just one class in CIFAR10: birds.

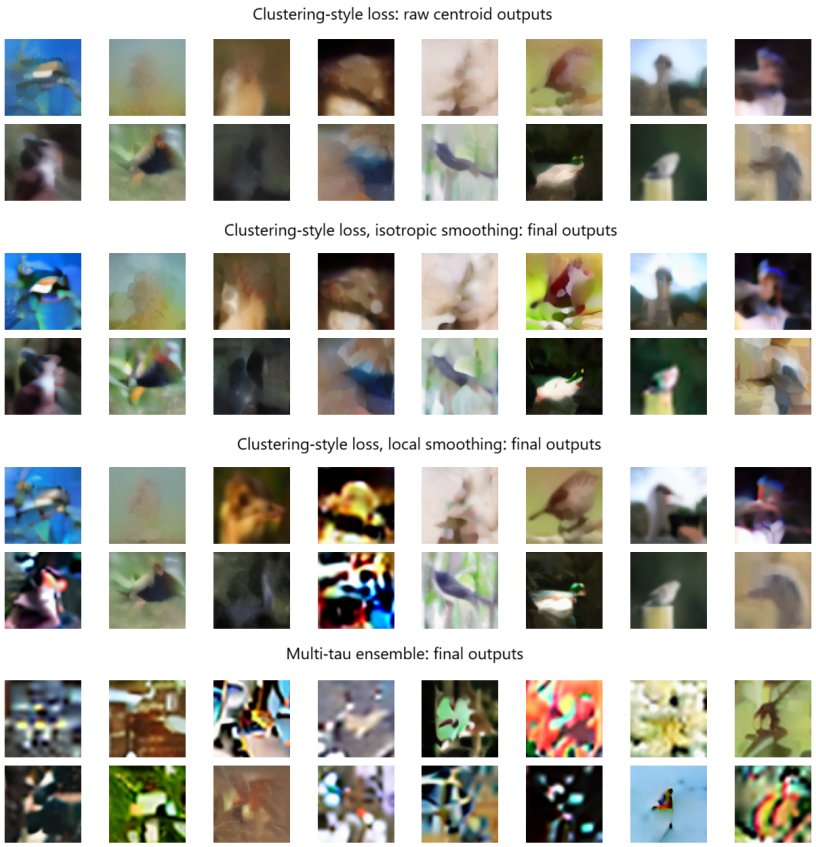

Figure 13: Results from training on only birds in CIFAR10 in latent space.

Results are shown in Figure 13. For a visual demonstration of smoothing at the end of our annealed clustering process, we provide the raw centroid outputs and two different smoothing processes apply to them: the isotropic smoothing of width τ, and the locally adaptive smoothing. We see that the latter clearly turns vague outlines into crisp images, and indeed nearly all outputs here are clearly recognisable as birds. The ensemble of multiple τs is also provided for comparison - once again, given the same conditions, it fails to produce coherent output.

Of course, even in this simple architecture and setup, ordinary drifting needs further tuning, but the ease of producing results with very little tuning using clustering-style loss is yet another datapoint in favour of its robustness.

D.5 DIVERSITY/MEMORISATION

Since producing samples of sufficient diversity is a point that needs careful addressing within the annealed clustering generative paradigm, we here investigate the diversity of generated samples and concerns of memorising training data.

We concentrate on MNIST pixel space (a raw, high-dimensional space) and Fashion MNIST latent space (a lower-dimensional representation) to diagnose the effects of clustering-style generation on these properties of the generated samples.

We diagnose both features through nearest-neighbour distance, a reliable indicator of whether datapoints are collapsing to certain over-represented members of the training data. Diversity is defined as the average nearest-neighbour distance within the generated population, divided by the nearest-neighbour distance within the real data population (estimated from a sample the same size as the generated batch size). Memorisation is defined as the average distance between a generated sample and its nearest real datapoint, divided by the nearest distance from the real datapoint to its nearest data neighbour. Both metrics, therefore, indicate healthy collapse-free learning when they are around 1.

Table 7: Diversity and memorisation metrics for MNIST (pixel) and FashionMNIST (latent)

Run	**Diversity**	**Memorisation**
MNIST: multi-tau ensemble	0.717	1.760
MNIST: clustering-style loss, local smoothing	1.062	1.285
FashionMNIST: multi-tau ensemble	1.493	1.782
FashionMNIST: clustering-style loss, local smoothing	0.680	1.055

The results for these datasets may be seen in Table 7. We see that the clustering-style learning produces slightly under-diverse samples in FashionMNIST, but that it never comes near memorisation. On the other hand, normal drifting produces under-diverse samples on MNIST pixels, but seems to be slightly too diverse, or spread-out, in the case of FashionMNIST. However, neither method comes near full collapse.