YOUR VAE KNOWS WHERE TO SAMPLE, NOT WHAT IT DOESN'T KNOW: MERGED AGGREGATED POSTERIORS AS DETERMINISTIC PRIORS AND CLUSTER MAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

The *aggregated posterior* of a trained VAE — the N-component Gaussian mixture of its per-sample posteriors — can be compressed to K components by moment-preserving hierarchical merging: one deterministic pass, with no seed and no refitting, yields the entire nested family, every K from N to 1. We evaluate this object in three roles. As a *latent density* for out-of-distribution detection it fails: a pre-registered study (MNIST, CIFAR-10; with CelebA-64 added later) refutes it in every form we could construct — post-hoc, co-trained, IWAE-trained, as a discrete codebook, substituted into the pixel likelihood, and at every K along the path — across three dataset regimes. On CelebA-64, latent density sits near or below chance against SVHN, but pixel based density (with a calibrated decoder) is even worse. The easy-regime (MNIST) verdict turns on the training recipe, not on the object: ordinary geometric augmentation with a convolutional architecture makes MNIST latent density beat that same model's pixel likelihood at β=0.5 ($10/10$ seeds), while the CIFAR-10/SVHN inversion barely moves. As a *structure recovery* device it succeeds: co-evolving the merge partition during training raises the clustering ceiling decisively over the best plain pipeline (NMI 0.74 vs. 0.66; plain ELBO plus a KL floor suffices), though not to the 0.86 of DEC, which gives up the generative model. A controlled comparison shows the hierarchy itself carries this structure ($+0.12$ NMI over post-hoc merging on the same encoder), while density depends only on the encoder ($\Delta \leq 0.008$ AUROC on MNIST). As a *generative prior* it succeeds too: drop-in sampling beats $\mathcal{N}(0, I)$ on $10/10$ seeds on every dataset, a frozen merged codebook beats learned codebooks in an SQ-VAE-style model, and, unlike an equal-K EM prior, the mixture cannot catastrophically corrupt the model's likelihood. Against EM, which samples as well, the merge's advantage is this safety, determinism and nesting, not accuracy. One principle organizes all three verdicts: moment-preserving compression preserves where the data lives — its support and cluster structure — not how likely it is.

1 INTRODUCTION

Variational autoencoders (Kingma & Welling, 2014) are both, theoretically interesting and used in practice. This theoretical and experimental study on out-of-distribution detection (OOD), structure recovery/clustering and the use of the latent space as generative prior tries to answer questions with direct practical use. Industrial anomaly detectors on sensor streams threshold, in units set per deployment, a reconstruction error (Fährmann et al., 2025), the term that dominates a VAE's pixel likelihood. A calibrated density on the latent code, needing one encoder pass and no decoder, is the natural alternative; whether it can be trusted is our first question. Neural audio codecs quantize into discrete codebooks (the VQ-VAE mechanism of van den Oord et al., 2017; Zeghidour et al., 2022; Défossez et al., 2022) and fight codebook collapse with balancing losses (Zheng et al., 2024) or distribution-based quantization (Niu et al., 2024). Molecular generators decode invalid molecules from parts of the latent space (Gómez-Bombarelli et al., 2018), which Wang et al. (2026) localise to low-density regions. Single-cell genomics clusters VAE latents into cell types (Lopez et al., 2018).

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.

Each of these relies on one of three roles of the aggregated posterior — density, cluster map, generative prior — and our verdicts differ by role: latent density can fail silently on inputs simpler than the training data, whereas a frozen codebook built from the aggregated posterior doubles utilization without heuristics, sampling from the merged mixture improves samples without retraining, and one merge pass yields clusters at every resolution. Every trained VAE thus already holds a cheap, deterministic record of where its data lives, but not a reliable measure of how likely an input is — which some applications assume.

The detection role is where that assumption is made most often. Discriminative networks are overconfident on inputs from other distributions, which makes out-of-distribution (OOD) detection "a fundamental requirement" for deployment (Lee et al., 2018). The generative remedy fits a model of the normal data and thresholds a probabilistic score — for a VAE, the reconstruction probability of An & Cho (2015) or the likelihood $p(x)$ itself — yet Nalisnick et al. (2019) showed that deep generative models can assign *higher* likelihood to OOD data than to in-distribution data, an *inversion*. The latent space offers a different object to threshold. A trained VAE maps each x_i to $q(z \mid x_i) = \mathcal{N}(\mu_i, \mathrm{diag}(\sigma_i^2))$, so the *aggregated posterior* $q(z) = \frac{1}{N}\sum_i \mathcal{N}(\mu_i, \mathrm{diag}(\sigma_i^2))$ is an N-component mixture that records exactly where encoded data lives. We compress it to K components by moment-preserving *hierarchical merging*: starting from the N per-point components, we merge mutually-nearest pairs in parallel rounds, preserving per-dimension first and second moments (Section 3.2 justifies this order). The merge is *post-hoc*: it uses only the frozen latents of a trained model. Whereas a Gaussian mixture fitted by expectation–maximisation (EM) is non-deterministic, re-fit per K and non-nested, one merge pass yields the entire nested family at every K, the *K-path*. We pre-registered the hypothesis that this compressed latent density would match or beat EM for OOD detection and sidestep the pixel-space pathology.

Finding: structure and support, not density. The test refutes the density hypothesis in every form we could construct. What the same experiments reveal instead is a principle: **moment-preserving compression preserves where the aggregated posterior puts its mass — the neighbourhoods of the encoded data and their cluster structure — not its likelihood ranking.** Support and structure are exactly what two standing problems need. For the prior–posterior mismatch (Hoffman & Johnson, 2016), whose uncovered regions of prior space degrade generation (Rosca et al., 2018; Dai & Wipf, 2019) and which learned mixture priors train their way around (Tomczak & Welling, 2018), the merged mixture is a training-free, deterministic drop-in. For codebook collapse, which Takida et al. (2022) hypothesize is driven by the heuristics of VQ training, a frozen merged codebook doubles utilization without them. The method's value is thus a combination of properties, not accuracy on one task: EM samples as well and fits held-out density better, k-means on the posterior means clusters MNIST better, and GPU EM is faster at small K (Section 5). The merge is deterministic, nested (every K from one pass), moment-preserving and *density-safe*: its components cannot become razor-thin, so it can replace the prior inside the model's own likelihood, where an EM mixture of the same size can crater it.

Contributions. We contribute a method, a training variant, three measured verdicts on the object it produces, and two baseline reproductions.

- **A deterministic, nested, density-safe compression of the aggregated posterior** (Section 3.2): moment-preserving merging under a mutual-nearest-neighbour rule. One GPU pass costing $O(N^2 d)$ yields the whole nested K-path with no seed and no re-fitting.
- **A co-trained variant** (Section 3.3) in which the merge partition evolves with the encoder, so the hierarchy becomes a property of the representation rather than of frozen latents.
- **Density: refuted** (Section 5). A pre-registered test with an exploratory CelebA-64 extension refutes it in every form we built (post-hoc, co-trained, IWAE-trained, codebook, inside the pixel likelihood) and at every K tested, in three dataset regimes, worsening with resolution.
- **Structure: confirmed against EM.** If you disentengle the latent space by increasing β, Kullback–Leibler (DL) divergence rate, the merged mixture does not degenerate, whereas equal-K EM degrades; co-training raises the clustering ceiling (conv-MNIST NMI 0.740 vs. 0.662), and an ablation isolates the co-evolving partition plus a KL floor as the

workhorse. Still, non-generative methods like k-means on the posterior means and DEC cluster MNIST better.

- **Support: confirmed.** As a drop-in prior the merged mixture beats $\mathcal{N}(0, I)$ on $10/10$ seeds on every dataset; as a frozen codebook it beats learned codebooks; and it is density-safe, whereas an equal-K EM prior can crater the model's own likelihood.

2 Related Work

VAEs for anomaly/OOD detection. Reconstruction-based scores form one family of three members, from point estimate to marginal. At the bottom is the reconstruction error at the posterior mean. Above it is that error's posterior expectation $\mathbb{E}_{q(z|x)}[\log p(x \mid z)]$ — the *reconstruction probability* of An & Cho (2015), the "posterior expected log-likelihood" of Morningstar et al. (2021), and, as a single-draw estimate, the score of Park et al. (2018) on sensor streams. At the top is the marginal likelihood, where the pathology of Nalisnick et al. (2019) lives. That pathology persists across model classes — flows, autoregressive models and a categorical VAE (Nalisnick et al., 2019), Glow and PixelCNN++ in the complexity-correction study (Serrà et al., 2020) — but no cited work varies a VAE's decoder likelihood, which is what we do here. Havtorn et al. (2021) argue that in a hierarchical VAE the ELBO's OOD signal rides on its KL term, which suggests the middle member carries little on its own. We report all three members under each model's own likelihood: Bernoulli, the calibrated Gaussian of Rybkin et al. (2021) (decoder scale σ set to its maximum-likelihood value; not evaluated for OOD there), and the discretized mixture of logistics of Salimans et al. (2017). A different line of work scores in feature space: the Mahalanobis distance of Lee et al. (2018) is a strong supervised baseline. We study the unsupervised OOD case, so one diagonal Gaussian on the latents similar to Lee et al. (2018), is the floor our latent densities must beat.

Corrected likelihood scores. Because of the findings of Nalisnick et al. (2019) research looked for *corrections* to the raw likelihood, the baselines a density claim is measured against: likelihood ratios against a background model (Ren et al., 2019); the input-complexity correction $S(x) = -\log_2 p(x) - L(x)$ of Serrà et al. (2020), with L a lossless code length (we report $-S$ so that higher means more in-distribution); likelihood regret (Xiao et al., 2020); ensemble WAIC (Choi et al., 2018); and, for hierarchical VAEs, the ratio of Havtorn et al. (2021) that isolates the top latents. Kirichenko et al. (2020) give the analogous diagnosis for normalizing flows, whose likelihoods follow local pixel correlations rather than semantics. Our question — is a compressed *aggregated posterior* a usable density at all? — is orthogonal, but we owe that comparison (Section 6).

Priors, mismatch, and latent mixtures. The gap between a VAE's prior and its aggregated posterior is a known failure source (Hoffman & Johnson, 2016; Rosca et al., 2018; Dai & Wipf, 2019), and learned mixture priors (Tomczak & Welling, 2018) and two-stage modeling (Dai & Wipf, 2019) train their way around it. Closest to our sampling result, Ghosh et al. (2020) fit a 10-component GMM to the codes *ex post* by EM and sample from it. Our mixture is deterministic and nested over every K, and we also test the mixture as a prior *density*, where an EM fit turns out not to be safe (Appendix D.7). On the discrete side, SQ-VAE (Takida et al., 2022) replaces the VQ heuristics by stochastic quantization with a shared trainable covariance, and NDVQ (Niu et al., 2024) gives each entry its own mean and variance. Our frozen merged codebook has NDVQ's form with SQ-VAE's stochastic assignment, and is fixed before training.

Clustering in a VAE's latent space. VaDE (Jiang et al., 2017) learns a global K-component GMM prior inside the ELBO; GMVAE (Dilokthanakul et al., 2016) conditions the mixture on an auxiliary latent; DEC (Xie et al., 2016) and DeepCluster (Caron et al., 2018) train the encoder against targets derived from a clustering of its own embedding. Our co-trained partition (Section 3.3) is closest to VaDE, but its mixture is never fitted: it is *merged* from the per-point posteriors each epoch, deterministically, and it is the partition, not a parametric prior, that enters the objective. A head-to-head against VaDE and GMVAE is future work; DEC we did run under our protocol (Appendix H). VaDE's published MNIST accuracy (94.46%) did not reproduce from our port or from the reference implementation; the reproduced mean over seeds is 0.85 (Appendix G).

3 METHOD

3.1 THE OBJECT, AND MOMENT-PRESERVING MERGING

We snapshot $(\mu_i, \log \sigma_i^2)$ once from the trained encoder in evaluation mode, giving the N-component diagonal mixture $q(z)$; these statistics are all the method consumes. Merging two components (μ_a, σ_a^2, w_a) and (μ_b, σ_b^2, w_b) preserves the mixture's per-dimension first and second moments:

$$w = w_a + w_b, \qquad \mu = \frac{w_a \mu_a + w_b \mu_b}{w}, \tag{1}$$

$$\sigma_:^2 = \frac{w_a(\sigma_{a,:}^2 + \mu_{a,:}^2) + w_b(\sigma_{b,:}^2 + \mu_{b,:}^2)}{w} - \mu_:^2. \tag{2}$$

Pairs are scored by the diagonal Bhattacharyya distance (Bhattacharyya, 1943), $D_B = \sum_d D_{B,d}$. While it is a distance (symmetric and non-negative), it does not fulfill the triangle inequality. It measures overlap, and works well with many dimensions. We constrain merging of components to keep a diagonal covariance matrices Σ_i, as the latent basis of full covariance matices is not identifiable (Section 6).

3.2 GPU PARALLEL MERGING: THE DISTANCE MATRIX SHRINKS WITH THE MIXTURE

The whole K-path costs one $O(N^2 d)$ distance computation, because the pairwise matrix D is built *once* and is thereafter the algorithm's state: every merge contracts it by one row and column. **One round** selects the pairs that can merge *simultaneously*. Selection is by *mutual nearest neighbour* — (i, j) is taken iff j is i's nearest eligible neighbour and i is j's — and such pairs are vertex-disjoint by construction, so the whole set falls out of two row-wise $\min$ operations on the GPU. Matched components are removed and rounds repeat on the shrinking submatrix until no pair is eligible (a *step*); a round with no mutual pair falls back to the global minimum. **One merge contracts the matrix.** Merging (i, j) into c replaces two rows and columns by one; every other distance carries over, and the new row costs $O(Nd)$, so the $N - K$ merges of the path cost $O(N^2 d)$ in total. The matrix at K is simply what the matrix at $K{+}1$ becomes after one merge, so the family is nested by construction; ordering each step's pairs by merge distance makes the merges a *total* order from N to 1, whose first $N - K$ merges give the mixture at K.

Why not the agglomerative order. Classical agglomerative clustering on this merge — one globally closest pair per step — collapses on VAE posteriors. VAEs train to seperate each individual points, leading to very peaked gaussian distributions. For two such points, the distance per dimension, with scales s_1, s_2, is:

$$D_{B,d} \;=\; \frac{1}{4}\,\frac{\Delta\mu^2}{s_1^2 + s_2^2} \;+\; \frac{1}{2}\ln\frac{s_1^2 + s_2^2}{2\,s_1 s_2}. \tag{3}$$

Merging two peaky points at gap $\Delta\mu$ yields a component of variance $\sigma^2 + \Delta\mu^2/4$: it inherits the gap as width. The mean term then falls like $1/s^2$ with that width while the variance term rises only like $\ln s$, so width reads as proximity, and the first closest-pair merges create a wide component that absorbs the rest one per step — single-linkage chaining. Merging *every* mutually-nearest pair of a round at once caps that growth, since a component can at most double per step; hence our mutual-nearest rule, which we do not claim is agglomerative clustering.

Every step is a perfect matching — a balance constraint we rely on. A step ends only when every active component is matched, so the count halves *exactly* per step. *This is a regulariser.* On a labelled synthetic mixture ($N{=}1500$, ten unequal clusters, ten seeds) agglomerative order puts 94% of the data into one component (NMI 0.14) and our rule 17% (NMI 0.71); with 3% outliers the agglomerative orders fall to NMI 0.01 while ours stays at 0.66. *It is also a bias.* Component sizes, and hence weights, are set by the rule (true sizes $490, \ldots, 6$ come back as $256, \ldots, 124$) — right for class-balanced benchmarks, not for imbalanced data.

Determinism and memory. Given the latents, the merge sequence is fixed with no seed: ties break by component index, a step cut short at the target K executes its closest pairs first (so runs to different

targets are nested), and a last-bit difference can flip exact ties, so determinism is a statement about a fixed numerical environment (Appendix K.1). The fast vectorised version uses this matrix and D is $O(N^2)$ memory. If not enough memory is available we keep an exact but slower sequence algoirthm: each component keeps a nearest-candidate list with a *certified radius* below which the list is complete, and the full matrix takes over once the survivors fit. Both phases select the same pairs, so the sequence equals a full-matrix run (verified exactly).

Model interface. Any model exposing one diagonal-Gaussian posterior over a flat latent is compatible (β-VAE, σ-VAE, IWAE-trained VAEs, the single-level VampPrior, the first stage of a two-stage VAE; Appendix K.1); hierarchical, flow, spatial and discrete latents are out of scope.

3.3 Co-training the partition during training

If post-hoc merging finds structure that EM cannot, can training against the merged mixture bake that structure into the latent itself? To test this we co-train a partition of the training points with the VAE (an exploratory extension, not pre-registered, run with the same multi-seed rigor; called *progressive* in the appendix tables). After W warmup epochs every point is a singleton cluster. The cluster count then shrinks on a *geometric* schedule (constant ratio per epoch) to a partition budget K_{target} at the final epoch: once per epoch each cluster's Gaussian is re-derived from its members' current posteriors by moment-preserving merging, and the nearest cluster pairs are merged under a size-weighted (Ward) squared 2-Wasserstein cost. The objective becomes cluster-aware. Within a batch, one cluster's members are moment-merged into one $\mathcal{N}(m, S)$, one $z^* \sim \mathcal{N}(m, S)$ is decoded to a prototype y^*, and each member's reconstruction of y^* enters the loss weighted by its membership probability, blended with the ordinary reconstruction at a weight ramping to $\alpha_{\max}$=0.5. The per-image KL is kept, with a per-dimension free-bits floor (Kingma et al., 2016) — a minimum KL per latent dimension below which no penalty applies — so singleton clusters recover the ELBO exactly. Each design element prevents an observed failure rooted in cluster-size asymmetry: without the per-image reconstruction term and the free bits, the posterior collapses, and under density-based assignment one cluster holds the dominace of points. All are needed (Appendix D.5). Strict nesting across epochs is given up; determinism is kept.

3.4 The merged mixture as prior and codebook

We study the use of our aggregeted posterior as sampling prior. Even though the density or OOD does not work, the data coverage (as seen in clustering) does stay intact, exactly wat a sampling prior needs. Whether this means it is a good sampling prios is what the hole distances and two-sample tests below measure. On plain ELBO-trained VAEs we therefore compare sampling from four priors on the same frozen decoder: $\mathcal{N}(0, I)$; the *tempered* $\mathcal{N}(0, t^2 I)$ with NVAE's temperatures (Vahdat & Kautz, 2020), transferred heuristically; the merged mixture at K=50; and EM at equal K. The priors are compared by a pixel-space two-sample test against held-out real images (leave-one-out 1NN accuracy; 0.5 means indistinguishable), a prior-hole score (distance of a sampled z to the nearest training latent), pixel-space diversity, the kNN precision and recall of Kynkäänniemi et al. (2019) (k=3, Inception-v3 features), and each prior's placement in latent space (Appendix K.1). Going one step further, we build an SQ-VAE-*inspired* model (Takida et al., 2022) whose codebook has K *Gaussian* entries (w_c, m_c, S_c) with their own covariances, as in NDVQ (Niu et al., 2024) — exactly the merged mixture's shape. Quantization is stochastic, $q(c \mid x) \propto w_c \mathcal{N}(z_e(x); m_c, S_c)$, with reparameterization within the chosen Gaussian, so the latent KL reduces to $\mathrm{KL}(q(c \mid x) \,\|\, w)$ and sampling draws within each component from the Gaussian the decoder was trained on: holes inside a component are excluded by construction; holes in the weights are measured, not excluded. We compare three codebooks of K=50 entries, with encoder and decoder started from the same first-stage model so that only the codebook differs: *learned from scratch* (entries start at 50 random encoded training images and are trained with the network), *initialized* (entries start at the merged mixture's components and are then trained), and *frozen* (entries fixed at the merged mixture; only encoder and decoder train).

Table 1: OOD AUROC (↑) at K=50 on full training latents. On MNIST, reconstruction based methods like IWAE or clustering methods like kNN perform better. In CIFAR-10 it depends of the smoothnes of the OOD samples (as shown in Nalisnick et al. (2019)), although all methods perform near chance in this case.

	MNIST in-dist.		CIFAR-10 in-dist.	
Score (latent unless noted)	FashionMNIST	KMNIST	SVHN	CIFAR-100
Pixel-space IWAE (Nalisnick et al., 2019)	*1.000*	*1.000*	0.578	0.443
kNN (k=10)	0.932	0.956	0.284	0.544
EM-GMM, K=50	0.922	0.939	0.280	0.534
Merged-GMM (ours), K=50	0.821	0.879	0.301	0.548
Mahalanobis (single Gaussian)	0.722	0.826	0.274	0.524

4 Experimental Protocol (Pre-registered)

The pre-registered protocol pairs MNIST with an MLP-VAE and CIFAR-10 with a convolutional VAE (we write MLP-MNIST, conv-MNIST and conv-CIFAR-10 for the encoder–dataset pairs), with β-VAE as an axis-aligned variant. The robustness studies add STL-10, and CelebA-64 (vs. SVHN / CIFAR-10) is an exploratory extension added after registration, with the same scores, baselines and tests (Appendix C). The mixture is built on *training-split* latents only and evaluated on held-out test and OOD sets (MNIST vs. FashionMNIST / KMNIST / Omniglot; CIFAR-10 vs. SVHN / CIFAR-100 / Textures). The score is the latent log-density $\log p_{\mathrm{GMM}}(z)$ at $z = \mu(x)$. **Baselines:** (1) a single diagonal-Gaussian Mahalanobis distance in latent space, the make-or-break baseline; (2) EM-GMM at equal K; (3) kNN in latent space; (4) the reconstruction error $-\|x - \mathrm{dec}(\mu(x))\|^2/D$ at the posterior mean (often used in industry); (4b) the reconstruction probability $\mathbb{E}_{q(z|x)}[\log p(x \mid z)]$ (An & Cho, 2015), (5) the VAE's *pixel-space* likelihood (Nalisnick et al., 2019), estimated with the k=16 importance-weighted bound (Burda et al., 2016) as an *evaluation-time estimator*, all of them computed from the sampe posterior samples under the model's own likelihood; and (6) merged-GMM (ours), fit post-hoc on the trained encoder — the *plain* pipeline, as opposed to the co-trained one of Section 3.3. Baselines (4), (4b) and (5) are one family: the reconstruction term at the mean, its posterior expectation, and the marginal. The estimator in (5) is a lower bound on $\log p(x)$ in expectation; a k-sample realisation is noisy, its gap is input-dependent and plausibly smaller in-distribution (which would flatter the pixel baseline), and cross-model pixel numbers compare bounds. Our conclusions do not rest on it, as latent density fails absolutely (inverted, AUROC < 0.5), not relative to pixel space. **Metrics:** AUROC and AUPR (in = positive), mean±std over seeds; the true-negative rate at 95% true-positive rate (TNR at 95% TPR), with the threshold set from in-distribution inputs alone (Lee et al., 2018); and, for clustering, normalised mutual information (NMI) and Hungarian-matched accuracy at K=10. **Statistics:** a *cell* is one architecture–dataset–hyper-parameter configuration, and a seed is one training run of it. We use 10 seeds per cell, paired Wilcoxon tests where the compared scores share one encoder per seed, and the Mann–Whitney U test (marked "U") between separately trained cells; claims spanning several cells form one family with Holm–Bonferroni adjusted p (Holm, 1979). With n=10 paired seeds the smallest Wilcoxon p is 0.002, i.e. "all ten seeds moved the same way". The headline results (Table 1) use one fixed seed at K=50.

5 Results & Analysis

Table 1 evaluates an MLP-VAE on MNIST (latent dim 16, $-\mathrm{ELBO} = 101.6$ nats) and a convolutional VAE on CIFAR-10, with mixtures on full training latents at K=50.

Wall-clock. At N=60k and K=50, one merge run (the whole path) took 18.6 s on an H100, while a GPU implementation of diagonal EM fits K=50 in 0.20 s and refits every K of $\{5, \ldots, 200\}$ in 1.5 s at CPU EM's held-out log-density: what the path provides is determinism and nesting, not throughput. At large K the balance tips to the merge, as EM refits grow costly and then infeasible — and large K is where density-OOD wants to operate (Figure 2). Test-time scoring runs the other way (one encoder pass and $O(Kd)$ per query vs. k=16 decoder passes): the affordable score fails.



Figure 1: MNIST latent mixtures (K=50; red crosses) with their exactly projected 2D densities (blue) under PCA (left) and LDA (right) of each pair; both maps are linear (Appendix D.1). **Left pair:** plain post-hoc merge (NMI ≈ 0.59). **Right pair:** the co-evolved partition of Section 3.3 (NMI $= 0.74$), visibly multimodal even in the unsupervised PCA view.

The OOD claim is refuted on MNIST, but clustering with the latent works, and the two tasks want opposite K. Merged-GMM density (Table 1) is worse than kNN, EM and the pixel likelihood, though better than Mahalanobis. This is consistent with theory: merging optimises a moment-preserving *compression* objective, not held-out likelihood, and greedy merges create "bridge" components that smear mass onto OOD regions. Nor is it an artifact of one K: over $K \in \{5, \ldots, 200\}$ merged trails EM at every K (Table 29). Density-OOD improves up to a large-K optimum (Figure 2), whereas clustering is best near the class count — and there merging wins: at full N the merged mixture has higher NMI than EM at K=10 on both datasets (Table 11; MNIST 0.589 vs. 0.534, CIFAR-10 0.089 vs. 0.038). The CIFAR-10 gap is a low-rate effect: these cells use a fixed σ=1, which keeps their per-image KL low (Appendix D.10), and the multi-seed study below shows that it survives only at low rate. The advantage is other methods to aggregate the posterior (wich means it can be used for sampling), not over clustering methods, k-means on $\mu(x)$ reaches 0.700/0.632 (conv/MLP-MNIST, K=10), or DEC wich reaches 0.740; (Appendix H).

Merged clustering is stable where EM is erratic; the advantage is regime-dependent. We move the VAE along its rate-distortion curve via β and a fixed decoder scale (σ^2 acts as β under a Gaussian likelihood, Appendix D.2) for three (architecture, dataset) pairs at 10 seeds (Table 12). Merged NMI stays in a tight band (across-seed std never above 0.04), whereas EM's reaches $\pm$0.12. Which method leads in NMI depends on the encoder's rate (its per-image KL to the prior). On conv-CIFAR-10, β moves the KL 9×, leaves merged NMI flat (0.084–0.103) while EM falls (0.109 $\rightarrow$ 0.043), so merging wins as the rate falls ($\beta \geq 2$: +0.023 to +0.057, 10/10, p=0.002; Appendix D.3). On high-rate MNIST merging loses on conv (-0.054, p=0.01), ties on MLP.

Latent density does not escape the pixel-space pathology. On MNIST the pixel score (a BCE pseudo-likelihood; Appendix F) already separates near-perfectly, so the decisive test is CIFAR-10 vs. SVHN, the canonical Nalisnick et al. (2019) failure. There every *latent* score falls below 0.5 AUROC — 0.27–0.30 in Table 1, 0.09–0.22 across the calibrated 10-seed β sweep — and the calibrated pixel score is inverted more strongly still (0.03–0.04). Merged-GMM density (0.30) is as inverted as the rest: the pathology survives compression, and our central hypothesis is refuted.

CelebA-64: both scores fail against SVHN, and the likelihood decides how it looks. On the official identity-disjoint partition of Liu et al. (2015) with the 140-pixel centre crop of the WAE protocol (Tolstikhin et al., 2018), a five-rung ladder (see Table 16 in Appendix) shows how much the likelihood matters. Trained and scored under a Bernoulli likelihood, the plain network gives pixel AUROC 0.769 against SVHN — which would suggest a third regime, pixel likelihood working where latent density fails. Trained and scored under a calibrated Gaussian σ instead, it gives 0.153, and every other CelebA cell agrees (0.08–0.40). Against SVHN, then, *both* scores fail — pixel strongly inverted, latent near chance (0.48–0.57 in the β, σ and resolution studies) or below it (0.38–0.43 on calibrated rungs B, D, E) — while against CIFAR-10 both work, with neither consistently ahead (Appendix D.4). Latent density *degrades* with resolution (Table 17) at every step against CIFAR-10 (0.845 $\rightarrow$ 0.706 from 16 to 64 px) and from 32 to 64 px against SVHN, where the pixel score is worse at every resolution. The paper's claim holds: latent density does not rescue OOD detection on natural images where the pixel likelihood fails.

The two scores fail differently; the latent one is not a complexity proxy. A measurement on *one* cell (CelebA β=1, ten seeds; not a sweep) separates the two failures by correlating each score with per-image PNG code length (Serrà et al., 2020). Within the CelebA test split the pixel likelihood

tracks complexity (Spearman $\rho = -0.60$ on every seed): simpler images are more likely, which is why a simpler dataset scores as in-distribution. The reconstruction error (4) and probability (4b) are that same score (identical ρ; SVHN AUROC 0.174 vs. pixel 0.172), so the industrial detector inherits the blind spot exactly. The merged-GMM density, by contrast, does *not* track complexity ($\rho = -0.08$); it fails against SVHN by being uninformative (0.51), not inverted. At 95% TPR the pixel and reconstruction scores reject 2% of SVHN, the latent score 8%.

The whole K-path: no escape at any K. One pass yields the OOD score at *every* K from N down to 5 (Figure 2). On MNIST moderate merging helps (its curve peaks at interior K, not at $K{=}N$), yet EM's best K matches or beats it; on CIFAR-10/SVHN every K from 5 to $15{,}000$ stays below 0.5 for both. Held-out likelihood picks a K within 0.033 AUROC of the optimum without OOD data (Appendix D.9).

The prior is not where pixel-space OOD lives. Substituting the merged mixture for $\mathcal{N}(0, I)$ *inside* the pixel score (only the prior term changes) moves its AUROC by a median of ≤ 0.001 on every dataset: the decoder term, hundreds of nats against the prior's tens, dominates. Nor does the training bound matter: IWAE-trained encoders ($k{=}8$, 10 seeds) leave CIFAR-10/SVHN latent AUROC unchanged (merged 0.116 vs. 0.115 for ELBO) and invert their own pixel score (0.026). The density hypothesis fails post-hoc, co-trained, IWAE-trained, as a codebook and inside the generative model.

The easy-regime (MNIST) verdict is about the recipe, not the object. Re-running the plain pipeline with ordinary geometric augmentation (training batches only; 10 seeds per cell) raises MNIST clustering (EM NMI by up to 0.21) and moves latent OOD sharply. Merged-GMM AUROC rises from 0.916 to 0.999 (conv-MNIST, $\beta{=}0.5$, FashionMNIST) and from 0.834 to 0.998 at $\beta{=}3$, every cell $10/10$ at $p{=}0.002$. Conv-MNIST latent density, which trailed the same model's pixel likelihood by 0.04–0.16 without augmentation, then beats it at $\beta{=}0.5$ (0.9992 vs. 0.9975 on FashionMNIST, 0.9979 vs. 0.9925 on KMNIST, $10/10$) and at $\beta{=}3$ only on FashionMNIST, pixel staying ahead on KMNIST (0.996 vs. 0.990). **The pathological regimes do not move at all** (conv-CIFAR-10 vs. SVHN, merged: $0.132 \rightarrow 0.132$; STL-10 within 0.008). Augmentation thus repairs the easy regime and leaves the inversion untouched, as expected of a mechanism acting on the posterior's *support*.

Co-training the partition raises the clustering ceiling. Three findings, at 10 seeds per cell (Table 23, Appendix D.5). *(1)* Where latent structure is strong, co-training raises the ceiling: on conv-MNIST the co-trained partition reaches $\mathrm{NMI} = 0.696 \pm 0.032$ at $\beta{=}3$, $K_{\mathrm{target}}{=}50$, beating post-hoc merging and EM on its own encoder ($10/10$, $p{=}0.002$) and the epoch-matched plain pipeline (0.639, $p{=}0.001$); training to a *large* partition budget and merging down reaches 0.740 ± 0.026, above training directly to 50 (U, $p{=}0.006$) and the best plain pipeline of any β (0.662; U, $p{=}0.0002$). MLP-MNIST behaves the same way; on conv-CIFAR-10 no budget effect survives Holm correction. *(2)* Co-trained NMI is nearly flat over a $20\times$ range of β (0.63–0.70 on conv-MNIST, where plain EM spans 0.385–0.662). *(3)* An ablation attributes the gain: without the cluster-reconstruction term ($\alpha_{\max}{=}0$: plain ELBO plus a KL floor, the partition still co-evolving) most of the MNIST gain remains (0.667 vs. 0.688, n.s.), the large-budget ceiling is reproduced (0.739 vs. 0.740), and the data-matched plain pipeline on conv-CIFAR-10 is beaten at both matched β (0.117 vs. 0.100, $10/10$). The recommended recipe is thus the simplest: plain ELBO, a KL floor, and a co-evolving partition at a large budget; the co-trained encoder, not the mixture fitter, carries the gain.

Construction matters for structure, not for density. Against a post-hoc hierarchy merged on the *same* encoder, a co-trained one ($K_{\mathrm{target}} \in \{100, \ldots, 5000\}$, then merged down) changes density-OOD by at most ± 0.008 AUROC at every common K, yet changes clustering by $+0.12$ NMI (0.740 vs. 0.614 at $K_{\mathrm{target}}{=}1000$, $10/10$, $p{=}0.002$). On conv-CIFAR-10 both move slightly (clustering $+0.010$ to $+0.018$, SVHN density up to 0.019 further into the inversion), so we state the separation as an MNIST result. Co-training leaves MNIST pixel OOD intact (0.997 vs. 0.999). On data-matched CelebA-64, co-training lifts latent AUROC against SVHN from 0.507 to 0.572 (U, $p{=}0.0002$; an encoder effect, still near chance) but not against CIFAR-10 (Appendix D.4). Co-training buys cluster structure, not usable density separation.

Support: a drop-in generative prior, and density safety. On plain ELBO-trained VAEs (10 seeds per cell) the merged prior beats $\mathcal{N}(0, I)$ on the two-sample test on $10/10$ seeds at $p{=}0.002$ for *every* architecture, dataset and β (on CelebA at $K{=}50$; e.g. conv-MNIST $\beta{=}0.5$: 0.641 vs. 0.761), with *higher* diversity and, except on CelebA's residual models, smaller hole distances (Figure 3). On

CelebA it also lowers the Fréchet Inception distance (FID; Heusel et al., 2017) at 50k samples on every seed of every ladder rung (65.7 vs. 71.2 on the β=1 models; 68.8 vs. 137.9 on rung B, the plain network with a calibrated σ, where $\mathcal{N}(0, I)$ samples fail outright; Table 19). An EM prior samples as well or slightly better on good β (conv-MNIST 0.615; on CelebA up to 2.4 lower FID), so the merged mixture's advantage over EM is the density safety shown below. The tempered prior does not help a flat-latent (single-vector, non-hierarchical) VAE: on the calibrated CelebA β=1 cell, lowering t makes samples *more* separable (1NN $0.840 \rightarrow 0.879$) while the merged prior sits at 0.614, because the encoded training means lie on a shell (mean norm 11.2, 90% between 7.7 and 15.1): $\mathcal{N}(0, I)$ draws (radius 8.0) already fall inside it, tempering moves them further in, and merged and EM draw on the shell (Appendix D.9). As prior *densities* inside the pixel likelihood, the two mixtures diverge. The merged prior tracks $\mathcal{N}(0, I)$ within 0.001 AUROC, while the EM prior suffers seed-dependent collapses (pixel score $0.9997 \rightarrow 0.85/0.80$ at N=60k on MNIST; as low as 0.56 over 30 MLP-MNIST checkpoints, where the merged prior sits at 0.9987), because EM fits razor-thin components (smallest $\sigma = 0.011$), and a z missing one pays hundreds of nats, whereas merged components have a built-in width floor (smallest $\sigma = 0.276$).

Support: a frozen codebook beats a learned one. We compare codebooks learned from scratch, initialized from the merged mixture, or frozen from the merged mixture.. In the SQ-VAE-style model on conv-MNIST the ordering is frozen > initialized > scratch on every metric, 5/5 seeds (Table 27): code reconstruction 201.5 vs. 215.4 from scratch, codes in use 40.8 vs. 21.1 of 50 (collapse halved, no heuristics), and likewise on two-sample accuracy and NMI. The merged mixture is a *better* codebook than the model can learn, even when learning starts from it; on CIFAR-10 it leads on all four metrics, narrowly. Feeding the mixture back into *continuous* training as an auxiliary prior term, finally, is clustering-neutral and does not rescue latent-density OOD (SVHN 0.21–0.29; Appendix D.6).

In sum, the merge wins only against $\mathcal{N}(0, I)$ and learned codebooks; against EM its case is determinism, nesting and density safety, against k-means or DEC a density per cluster and a hierarchy.

6 Limitations

Four limitations bound the verdicts (in full: Appendix K.3). The method needs a continuous Gaussian posterior and merges diagonal-joint only, since the latent basis is not identifiable. Most of our cells are trained shorter than the work we compare against (plain CelebA cells: 4.1M samples against at most 11.4M for Ghosh et al. (2020), except ladder rungs A and B at 15.5–16.3M), which bounds the *positive* claims more than the negative result — the refutation strengthens as reconstruction improves. Our OOD baselines are uncorrected, so the claim is that the compressed aggregated posterior is not a usable density *on its own*, not that no corrected latent score (Ren et al., 2019; Serrà et al., 2020; Xiao et al., 2020; Choi et al., 2018; Havtorn et al., 2021) can work. And our clustering claims are against EM, not VaDE, GMVAE or DEC: DEC under our protocol clusters MNIST far better (0.861 vs. 0.740 NMI) at the price of the generative model (Appendices G, H).

7 Conclusion

We pre-registered moment-preserving merged mixtures as latent density models for OOD detection on MNIST and CIFAR-10 (CelebA-64 exploratory), and the hypothesis is *refuted in all three regimes*: they are beaten by kNN, EM and pixel space on MNIST (largely reversed by augmentation), inheriting the pixel-space pathology on CIFAR-10 vs. SVHN, near or below chance on CelebA-64, and degrading with resolution. What survives is a profile of properties rather than a score: one deterministic pass yields the whole nested K-path; every mixture on it is density-safe where EM's can crater the model's likelihood; its clustering is stable where EM's is erratic, better at low rate and raised by co-training, though k-means on the posterior means clusters MNIST better; and the mixture improves sampling as a drop-in prior and beats learned codebooks as a frozen one. The aggregated posterior knows where to sample, even though it does not know what it doesn't know.

USE OF GENERATIVE AI

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments; implement methods; propose or refine hypotheses (follow-up hypotheses and controls — the pre-registered hypothesis is ours); and interpret results. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, and generating synthetic data sets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, assisting with translation, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures; suggest experimental parameters; create or edit software code; create artifacts; draft parts of the paper; summarize or analyse existing literature; identify gaps; brainstorm; source and search for information; edit the paper to improve readability; identify relevant literature; format references; suggest a structure for the paper; and propose a title. We have reviewed all AI-assisted work: LLM-generated code runs through an automated test suite on every change, with each new test checked to fail on a deliberately introduced bug; every citation was checked against the full text of its source; and every summary, interpretation and number was checked against the raw result records (Appendix J). We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We release the code, the trained checkpoints and the per-cell result record behind every reported number (Appendix I). The repository's `README.md` documents installation, the automatic download of every dataset (including CelebA's official split and crop), running a single experiment cell, the command for each campaign — most drivers have a dry-run mode that prints their exact commands before anything runs — and regenerating every table of the paper. Dependencies are declared in `requirements.txt` (identical to `pyproject.toml`; a test keeps both complete), tested with Python 3.12 and PyTorch 2.11 on H100 GPUs and with Python 3.11 and PyTorch 2.6 on a consumer RTX 3080 Ti, which suffices for the MNIST cells and the baselines; an automated test suite, including a stubbed end-to-end run, checks an installation before any experiment is started. Every appendix table is generated from these records by the released code, and each record carries its seed, git revision, library versions, device, learning-rate provenance and content hashes of its data splits. One seed per run drives every random draw, with deterministic kernels, and the pipeline reproduces itself: a complete re-run of the CelebA campaign returned its validation-best checkpoints with best epoch, PSNR and per-dimension KL identical to the last bit, and evaluations re-run on them returned the recorded numbers exactly (Appendix I). Datasets, splits, preprocessing and likelihoods are specified in Appendix C, the merge's tie-breaking and its memory-bounded exact variant in Appendix K.1, the statistical protocol in Section 4, training budgets in Appendix K.2, and every trained grid in Table 4. Baseline ports are checked against the original implementations, archived verbatim, and against their published numbers (Appendices G and H).

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2015.

Anil Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations (ICLR)*, 2016.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

Hyunsun Choi, Eric Jang, and Alexander A. Alemi. WAIC, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations (ICLR)*, 2019.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

Daniel Fährmann, Malte Ihlefeld, Arjan Kuijper, and Naser Damer. Resource-efficient anomaly detection in industrial control systems with quantized recurrent variational autoencoder. *IET Collaborative Intelligent Manufacturing*, 7(1), 2025. doi: 10.1049/cim2.70032.

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael J. Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. In *International Conference on Learning Representations (ICLR)*, 2020.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

Jakob D. Havtorn, Jes Frellsen, Søren Hauberg, and Lars Maaløe. Hierarchical VAEs know what they don't know. In *International Conference on Machine Learning (ICML)*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL `https://arxiv.org/abs/1706.08500`.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2017.

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

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

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)*, pp. 1965–1972, 2017.

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

Diederik P. Kingma, Tim Salimans, Rafal Józefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Why normalizing flows fail to detect out-of-distribution data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.

Gabriel Loaiza-Ganem and John P. Cunningham. The continuous Bernoulli: Fixing a pervasive error in variational autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018.

Warren R. Morningstar, Cusuh Ham, Andrew G. Gallagher, Balaji Lakshminarayanan, Alexander A. Alemi, and Joshua V. Dillon. Density of states estimation for out of distribution detection. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations (ICLR)*, 2019.

Zhikang Niu, Sanyuan Chen, Long Zhou, Ziyang Ma, Xie Chen, and Shujie Liu. NDVQ: Robust neural audio codec with normal distribution-based vector quantization. *arXiv preprint arXiv:2409.12717*, 2024.

Daehyung Park, Yuuna Hoshi, and Charles C. Kemp. A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL `https://arxiv.org/abs/2104.11222`.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847*, 2018.

Oleh Rybkin, Kostas Daniilidis, and Sergey Levine. Simple and effective VAE training with calibrated decoders. *International Conference on Machine Learning (ICML)*, 2021.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations (ICLR)*, 2017.

Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F. Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations (ICLR)*, 2020.

Yuhta Takida, Takashi Shibuya, WeiHsiang Liao, Chieh-Hsin Lai, Junki Ohmura, Toshimitsu Uesaka, Naoki Murata, Shusuke Takahashi, Toshiyuki Kumakura, and Yuki Mitsufuji. SQ-VAE: Variational bayes on discrete representation with self-annealed stochastic quantization. In *International Conference on Machine Learning (ICML)*, 2022.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

Jakub M. Tomczak and Max Welling. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Xi Wang, Jiahan Li, Yuxuan Xia, Yingcheng Wu, Shaoyi Zheng, and Shengjie Wang. Smoothing dark areas in molecular latent diffusion. *arXiv preprint arXiv:2606.13955*, 2026.

Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational autoencoder. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning (ICML)*, pp. 478–487, 2016.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2022.

Rui-Chen Zheng, Hui-Peng Du, Xiao-Hang Jiang, Yang Ai, and Zhen-Hua Ling. ERVQ: Enhanced residual vector quantization with intra-and-inter-codebook optimization for neural audio codecs. *arXiv preprint arXiv:2410.12359*, 2024.

A Appendix

B Checkpoints, Result JSONS and Code

For complete reproduction, having all assets helps. This work uses extensive reporting in form of .json files and saves all the model checkpoints. This all currently resides in a private github project. Since supplementary material is only allowed to have 40MB a snapshot of the code and the most vital .jsons which is processed by an anonymity check is submitted as suplementary material. As soon as the paper is public, we will also make the repositoy with the full set of assets and code public.

C Datasets, splits and preprocessing

Table 2: Datasets, splits and preprocessing. Sizes are the standard splits; "model input" is the tensor the encoder sees after preprocessing. Pixel values are scaled to $[0, 1]$ and nothing else: no mean/variance normalisation, and no binarisation of MNIST (the Bernoulli likelihood is evaluated on continuous intensities). Each in-distribution dataset is paired with the two OOD sets listed, which are resized to the in-distribution model's input size.

Role	Dataset	Train	Test	Model input	Likelihood	OOD sets
in-dist	MNIST (MLP)	60 000	10 000	$1 \times 28 \times 28$	Bernoulli	FashionMNIST, KMNIST
in-dist	MNIST (conv)	60 000	10 000	$1 \times 32 \times 32$	Bernoulli	FashionMNIST, KMNIST
in-dist	CIFAR-10	50 000	10 000	$3 \times 32 \times 32$	Gaussian	SVHN, CIFAR-100
in-dist	STL-10	5 000	8 000	$3 \times 96 \times 96$	Bernoulli	SVHN, CIFAR-10
in-dist	CelebA-64	162 770	19 962	$3 \times 64 \times 64$	Gaussian	SVHN, CIFAR-10
OOD	FashionMNIST	—	10 000	as in-dist	—	—
OOD	KMNIST	—	10 000	as in-dist	—	—
OOD	SVHN	—	26 032	as in-dist	—	—
OOD	CIFAR-100	—	10 000	as in-dist	—	—
OOD	CIFAR-10	—	10 000	as in-dist	—	—

MNIST is resized $28 \to 32$ for the convolutional encoder only, because that encoder downsamples by 8. STL-10 uses the 5 000 *labelled* training images, because this allows MNI calculation after fitting unsupervised, and we investigated structure with it; its 100 000 unlabelled images are not used. CelebA uses the official identity-disjoint partition of Liu et al. (2015) (162 770 / 19 867 / 19 962 train / validation / test images of the aligned set), centre-cropped to 140×140 and resized to 64×64; its likelihood is Bernoulli only on ladder rung A. The resolution study runs CelebA at 16, 32 and 64 pixels with a calibrated σ-VAE. Latent dimension is 16 for MNIST, 256 for CIFAR-10 and STL-10, and 64 for CelebA (128 on CelebA ladder rungs D and E).

Table 2 lists every dataset, its split sizes and the tensor the encoder actually sees. Four datasets appear in-distribution, in five settings (MNIST with two architectures, CIFAR-10, STL-10, CelebA-64), and five as OOD sets. Three properties of the preprocessing matter for reading the results.

Pixel scaling only. Images are converted to $[0, 1]$ per channel by dividing trough 255 and nothing else: no mean/variance normalisation, no whitening, and no binarisation of MNIST. The Bernoulli likelihood is therefore evaluated on continuous intensities. Binary cross-entropy on continuous pixels is the log of an *unnormalised* density: the normaliser of the continuous Bernoulli, $C(\lambda)$, depends on the decoder's output at every pixel, so the omitted term is neither a constant nor a fixed proportion and the objective is a *pseudo*-likelihood (Loaiza-Ganem & Cunningham, 2019). Every MNIST pixel-space score in this paper is that pseudo-likelihood, labelled so; Appendix F reports what changes when the same decoders are read as continuous-Bernoulli models. For the natural-image datasets we use a calibrated Gaussian likelihood, which is both normalised and correctly scaled (Appendix D.10); CelebA ladder rung A uses the Bernoulli instead, the decoder-noise sweep replaces the calibrated scale with a fixed σ_{dec}, and the resolution study uses a calibrated σ-VAE with the analytic optimal global scale.

Table 3: Training augmentation. The pre-registered protocol and every result in the main text except the augmentation study use *no* augmentation. Where augmentation is used it is applied to training batches only: the per-epoch re-encoding that drives the partition, all mixture fits, and every OOD and clustering evaluation always see clean images. Each image in a batch receives an independent transform.

Dataset	Rotation	Scale	Translation	H-flip	Padding
MNIST	$\pm 10^\circ$	0.9–1.1	$\pm 0.1\,W$ (sub-pixel)	no	zeros
CIFAR-10	—	—	$\pm W/8$ (whole pixels)	yes	reflection
STL-10	—	—	$\pm W/8$ (whole pixels)	yes	reflection
CelebA-64	—	—	$\pm W/16$ (whole pixels)	yes	reflection

Digits are not mirror-symmetric, so MNIST gets no flip; its black background makes zero padding the natural choice, and small rotations and scalings are the standard nuisance transformations for handwriting. For natural images the policy is the usual pad-and-crop plus horizontal flip, with whole-pixel shifts so that the shifted image is an exact copy rather than a blurred resample. CelebA faces are aligned, so the shift is small ($\pm W/16$) and the flip horizontal. The reconstruction target is the augmented image itself, which means the model learns the augmented distribution; the alternative of reconstructing the clean image (denoising-style, which regularises the encoder without changing the modelled distribution) and stronger policies (RandAugment, Cutout, CutPaste) are implemented but not run here.

Resizing happens in two places, for two reasons. MNIST is resized from 28 to 32 pixels for the convolutional encoder only, because that encoder downsamples by a factor of eight; the MLP encoder sees the native 28×28. Separately, every OOD set is resized to the in-distribution model's *effective* input size, so that SVHN is upsampled from 32 to 64 pixels when scored against a CelebA model and to 96 against STL-10.

Splits. MNIST, CIFAR-10, STL-10 and the OOD sets use their standard splits. STL-10 contributes only its $5\,000$ labelled training images because we also wanted to compute the NMI. CelebA uses the official identity-disjoint partition of Liu et al. (2015), in the three-way split its release distributes ($162\,770$ / $19\,867$ / $19\,962$ images of the aligned set); each 178×218 aligned image is centre-cropped to 140×140 and resized to 64×64 with antialiasing. That crop is the CelebA preprocessing of the WAE protocol (Tolstikhin et al., 2018). Ghosh et al. (2020), our closest CelebA comparison, state only that they use the official splits, and their released code reads their own pre-cropped 64×64 copy of the dataset, so their crop is recoverable from neither; our CelebA numbers should not be read as directly comparable to theirs. The same indices are reused whenever two pipelines must be compared on identical data. Every CelebA cell fits its mixtures on the whole aggregated posterior ($162\,770$ latents) with the memory-bounded exact merge.

C.1 Augmentation

No augmentation is used anywhere in the pre-registered protocol or in any result outside the augmentation study. Where it is used, the exact policies are those of Table 3: an independent random transform per image, applied to training batches only. Evaluation never sees an augmented image, and neither does the per-epoch re-encoding that drives the co-trained partition, so augmentation changes what the model learns and not how it is measured. Translations for the natural-image datasets are whole-pixel, so a shifted image is an exact copy rather than a resampled one; MNIST uses sub-pixel affine transforms.

D Experiment details

Table 4 lists every grid that was run, with its cell count, datasets, seeds and epoch budget. A cell is one independent training run with its own seed; unless stated otherwise every grid uses ten seeds, mixtures at $K = 50$, clustering at $K = 10$, and paired Wilcoxon signed-rank tests across seeds when two methods are compared on the same encoders.

Table 4: Every experiment grid behind the reported results. A cell is one independent training run with its own seed; rows counted in models are evaluations of already-trained models rather than new training runs. The CIFAR-10 and STL-10 cells of a grid that use a convolutional archiecture row are counted in the conv: Experiment rows only.

experiment	cells	datasets	seeds	epochs
Plain VAE: β and decoder-noise sweeps	180	MNIST	10	20
CelebA-64: architecture and likelihood ladder	50	CelebA-64	10	20, 25, 95, 100
CelebA-64: plain VAE, β sweep	20	CelebA-64	10	25
CelebA-64: plain VAE, decoder-noise sweep	20	CelebA-64	10	25
CelebA-64: resolution sweep	30	CelebA-64	10	35, 45
CelebA-64: co-trained partition	10	CelebA-64	10	25
Plain VAE, additional β values	40	MNIST	10	20
Plain VAE on MNIST, epoch-matched references	100	MNIST	10	30
Plain VAE trained with the importance-weighted bound	20	MNIST	10	30
Plain VAE, checkpoints for the sampling study	40	MNIST	10	30
Co-trained partition, $\beta = 0.5$	20	MNIST	10	30
Co-trained partition, $\beta \in \{1, 3, 4, 10\}$	80	MNIST	10	30
Co-trained partition, budget sweep and $\alpha_{\max} = 0$ control	70	MNIST	10	30
Prior tracked but excluded from the loss	100	MNIST	10	30
Prior term with averaging	100	MNIST	10	30
Prior term without averaging	100	MNIST	10	30
Prior term with averaging and shrinkage	40	MNIST	10	30
KL floor and partition only (no cluster-reconstruction term)	100	MNIST	10	30
Plain VAE retrained on the co-trained pipeline's images	40	MNIST	10	30
Plain VAE with augmentation, full data	40	MNIST	10	30
Plain VAE with augmentation, matched data	40	MNIST	10	30
Co-trained partition with augmentation	50	MNIST	10	30
Fresh VAE trained against a frozen merged prior	40	CIFAR-10, MNIST, STL-10	10	30, 60
Gaussian-codebook model, three codebook conditions	30	CIFAR-10, MNIST	5	60
OOD along the nested merge path	351 models	—	—	—
Sampling from alternative priors	120 models	—	—	—
Pixel likelihood under alternative priors	91 models	—	—	—
CelebA-64, β=1: tempered-prior sweep	10 models	—	—	—
CelebA-64, β=1: complexity and TNR analysis	10 models	—	—	—
Replication of one configuration	10	MNIST	10	30
conv: Plain VAE, β sweep	70	CIFAR-10	10	60
conv: Plain VAE, decoder-noise sweep	40	CIFAR-10	10	60
conv: Plain VAE, additional β values	80	CIFAR-10, STL-10	10	60
conv: Plain VAE trained with the importance-weighted bound	50	CIFAR-10	10	60
conv: Plain VAE retrained on the co-trained pipeline's images	20	CIFAR-10	10	60
conv: Plain VAE with augmentation, full data	40	CIFAR-10, STL-10	10	60
conv: Plain VAE with augmentation, matched data	20	CIFAR-10	10	60
conv: Co-trained partition, β sweep	100	CIFAR-10, STL-10	10	60
conv: Co-trained partition, budget sweep and $\alpha_{\max} = 0$ control	70	CIFAR-10	10	60
conv: KL floor and partition only	100	CIFAR-10, STL-10	10	60
conv: Prior tracked but excluded from the loss	100	CIFAR-10, STL-10	10	60
conv: Prior term with averaging	100	CIFAR-10, STL-10	10	60
conv: Prior term without averaging	100	CIFAR-10, STL-10	10	60
conv: Prior term with averaging and shrinkage	40	CIFAR-10, STL-10	10	60
conv: Co-trained partition with augmentation	50	CIFAR-10, STL-10	10	60

Table 5: Plain VAE, β sweep on MNIST (10 seeds per cell, 20 epochs). Reconstruction, KL and clustering NMI at $K = 10$.

model	beta	recon	KLD	NMI merged	NMI EM
MLP / MNIST	0.5	73.7 $\pm$.6	28.1 $\pm$.7	0.586 $\pm$.031	0.588 $\pm$.016
MLP / MNIST	1	81.8 $\pm$.6	19.6 $\pm$.3	0.564 $\pm$.030	0.501 $\pm$.119
MLP / MNIST	2	93.3 $\pm$.4	13.3 $\pm$.1	0.542 $\pm$.024	0.395 $\pm$.063
MLP / MNIST	4	110.4 $\pm$1.1	8.4 $\pm$.2	0.529 $\pm$.038	0.363 $\pm$.103
MLP / MNIST	8	135.4 $\pm$.3	4.6 $\pm$.0	0.513 $\pm$.023	0.475 $\pm$.043
conv / MNIST	0.5	116.1 $\pm$.1	30.4 $\pm$.2	0.612 $\pm$.034	0.638 $\pm$.022
conv / MNIST	1	121.1 $\pm$.1	23.2 $\pm$.1	0.594 $\pm$.030	0.641 $\pm$.022
conv / MNIST	2	130.6 $\pm$.3	16.6 $\pm$.0	0.599 $\pm$.021	0.650 $\pm$.021
conv / MNIST	4	147.5 $\pm$.3	10.6 $\pm$.0	0.563 $\pm$.025	0.631 $\pm$.014
conv / MNIST	8	175.1 $\pm$.4	5.9 $\pm$.0	0.540 $\pm$.038	0.480 $\pm$.079

D.1 PLAIN-VAE SWEEPS

Table 11 gives the comparison at $K = 10$; the sweeps vary the operating point around it two ways, the β of a β-VAE and a fixed Gaussian decoder noise scale σ_{dec}. On a Gaussian likelihood these

Table 6: Plain VAE, β sweep on MNIST (10 seeds per cell, 20 epochs). OOD AUROC at $K = 50$ for the merged mixture, EM at equal K and the model's pixel-space likelihood.

model	beta	FashionMNIST mrg	EM	pixel	KMNIST mrg	EM	pixel
MLP / MNIST	0.5	0.832 $\pm$.038	0.890 $\pm$.029	1.000 $\pm$.000	0.877 $\pm$.029	0.926 $\pm$.014	1.000 $\pm$.000
MLP / MNIST	1	0.793 $\pm$.055	0.860 $\pm$.041	0.999 $\pm$.000	0.814 $\pm$.040	0.892 $\pm$.034	0.999 $\pm$.000
MLP / MNIST	2	0.696 $\pm$.057	0.830 $\pm$.056	0.999 $\pm$.000	0.700 $\pm$.053	0.844 $\pm$.045	0.999 $\pm$.000
MLP / MNIST	4	0.627 $\pm$.055	0.808 $\pm$.057	0.999 $\pm$.000	0.576 $\pm$.061	0.796 $\pm$.037	0.998 $\pm$.000
MLP / MNIST	8	0.715 $\pm$.043	0.858 $\pm$.020	0.999 $\pm$.000	0.598 $\pm$.036	0.841 $\pm$.014	0.998 $\pm$.000
conv / MNIST	0.5	0.889 $\pm$.027	0.915 $\pm$.021	0.999 $\pm$.000	0.950 $\pm$.007	0.962 $\pm$.005	0.999 $\pm$.000
conv / MNIST	1	0.896 $\pm$.019	0.924 $\pm$.017	0.999 $\pm$.000	0.947 $\pm$.006	0.962 $\pm$.004	0.999 $\pm$.000
conv / MNIST	2	0.872 $\pm$.037	0.922 $\pm$.033	0.999 $\pm$.000	0.925 $\pm$.013	0.959 $\pm$.011	0.999 $\pm$.000
conv / MNIST	4	0.827 $\pm$.026	0.909 $\pm$.025	0.999 $\pm$.000	0.829 $\pm$.014	0.929 $\pm$.007	0.999 $\pm$.000
conv / MNIST	8	0.778 $\pm$.034	0.927 $\pm$.018	0.999 $\pm$.000	0.690 $\pm$.021	0.904 $\pm$.010	0.998 $\pm$.000

Table 7: Plain VAE, β sweep on CIFAR-10 (10 seeds per cell, 60 epochs). CIFAR-10 and STL-10 rows: calibrated decoder scale, convolutional architecture (configuration C of Table 30). Latent columns only (the pixel column is not reported for these cells). Reconstruction, KL and clustering NMI at $K = 10$.

model	beta	recon	KLD	NMI merged	NMI EM
conv / CIFAR-10	0.1	-4952.0 $\pm$9.7	926.5 $\pm$1.8	0.103 $\pm$.010	0.109 $\pm$.009
conv / CIFAR-10	0.3	-4917.4 $\pm$17.5	729.8 $\pm$2.4	0.094 $\pm$.012	0.096 $\pm$.010
conv / CIFAR-10	0.5	-4888.0 $\pm$11.0	635.0 $\pm$1.6	0.086 $\pm$.008	0.090 $\pm$.009
conv / CIFAR-10	1	-4807.6 $\pm$24.9	509.0 $\pm$3.9	0.084 $\pm$.008	0.082 $\pm$.009
conv / CIFAR-10	2	-4544.7 $\pm$76.9	367.9 $\pm$14.0	0.092 $\pm$.007	0.069 $\pm$.010
conv / CIFAR-10	4	-3981.0 $\pm$55.3	216.7 $\pm$6.9	0.092 $\pm$.008	0.054 $\pm$.005
conv / CIFAR-10	8	-3258.3 $\pm$27.0	108.2 $\pm$2.0	0.100 $\pm$.008	0.043 $\pm$.001

coincide (Section D.2), so the CIFAR-10 rows of the two sweeps trace one axis; on MNIST, whose likelihood is Bernoulli and has no decoder scale, β is a knob of its own. Each cell trains a VAE, then fits a merged mixture and an EM mixture at equal K on the same latents and scores both, alongside the model's pixel-space likelihood. Two patterns are visible directly in the tables: EM's clustering swings widely across the sweep and across seeds (down to 0.196 ± 0.054 on MLP-MNIST at $\beta = 3$) while the merged mixture stays in a narrow band, and every latent score on CIFAR-10 sits below 0.5 against SVHN at every setting.

D.2 The two rate–distortion knobs are one knob under a Gaussian likelihood

We naively started to explore our VAE with a σ decoder varying β and σ independently. We have cases where we do not use the σ decoder, so in that case a sweep of β still makes sens. We explain here that we found out that our coupling did not really make sense, or, stated differently, just leats to an experiment were we have a larger range of the same knob. Our β-VAE objective uses a per-image-summed squared error, $\mathcal{L} = \frac{1}{2}\|x-\mu\|^2 + \beta\,\mathrm{KL}$. Dividing by β leaves the optimum unchanged and gives $\|x-\mu\|^2/(2\beta) + \mathrm{KL}$, which is exactly the σ-VAE objective $\|x-\mu\|^2/(2\sigma^2) + \mathrm{KL}$ with

$$\sigma^2 \;=\; \beta. \tag{4}$$

(Rybkin et al. (2021) state the identity as $\sigma^2 = \beta/2$; their own Eq. 7, $\frac{D}{2}\,\mathrm{MSE} + \beta\,\mathrm{KL}$, gives $\sigma^2 = \beta$ as here. For a fixed σ the equivalence is one of objectives up to a positive scale, not of optimisation trajectories; with a learned σ the $D \log \sigma$ term stays.) A β sweep at fixed σ and a σ_{dec} sweep at β=1 therefore trace the same curve. The data confirm equation 4 directly: β=1 and σ_{dec}=1.0 are the same objective, and the two cells agree to within seed noise (per-image KL 13.1 vs. 13.0 nats), while the rest of the σ column interleaves with the β column as the identity requires (σ_{dec}=0.5 $\equiv$ β=0.25, KL 39.0, between the β=0.5 and β=0.1 cells). Two consequences: for the Gaussian-likelihood datasets the two sweeps are one rate–distortion axis sampled at different points, and the genuinely independent β sweep is the MNIST one; and the identity dissolves once the decoder scale is *calibrated* rather than fixed — with the analytic optimum substituted the reconstruction term

Table 8: Plain VAE, β sweep on CIFAR-10 (10 seeds per cell, 60 epochs). CIFAR-10 and STL-10 rows: calibrated decoder scale, convolutional architecture (configuration C of Table 30). Latent columns only. OOD AUROC at $K = 50$ for the merged mixture, EM at equal K and the model's pixel-space likelihood

model	beta	SVHN mrg	EM	CIFAR-100 mrg	EM
conv / CIFAR-10	0.1	0.178 ±.003	0.179 ±.003	0.543 ±.001	0.543 ±.002
conv / CIFAR-10	0.3	0.149 ±.005	0.148 ±.005	0.531 ±.002	0.531 ±.002
conv / CIFAR-10	0.5	0.132 ±.003	0.128 ±.004	0.527 ±.002	0.526 ±.003
conv / CIFAR-10	1	0.115 ±.002	0.102 ±.002	0.523 ±.002	0.522 ±.002
conv / CIFAR-10	2	0.118 ±.005	0.090 ±.003	0.521 ±.002	0.518 ±.002
conv / CIFAR-10	4	0.154 ±.006	0.093 ±.005	0.522 ±.001	0.515 ±.002
conv / CIFAR-10	8	0.222 ±.006	0.132 ±.005	0.524 ±.002	0.513 ±.002

Table 9: Plain VAE, decoder-noise sweep (σ_{dec}, Gaussian likelihood, 10 seeds per cell). CIFAR-10 rows: convolutional architecture, latent columns only. Reconstruction, KL and clustering NMI at $K = 10$.

model	decoder_sigma	recon	KLD	NMI merged	NMI EM
MLP / MNIST	0.1	-760.8 ±2.1	51.3 ±.1	0.575 ±.019	0.598 ±.020
MLP / MNIST	0.3	-175.2 ±1.0	26.9 ±.6	0.572 ±.031	0.526 ±.117
MLP / MNIST	0.5	202.8 ±.3	15.9 ±.2	0.541 ±.031	0.465 ±.121
conv / MNIST	0.1	-1192.7 ±3.5	52.0 ±.2	0.627 ±.026	0.661 ±.019
conv / MNIST	0.3	-258.4 ±.2	29.3 ±.1	0.610 ±.035	0.634 ±.033
conv / MNIST	0.5	249.5 ±.1	19.1 ±.1	0.596 ±.043	0.642 ±.025
conv / CIFAR-10	0.1	-3719.1 ±12.4	302.8 ±3.3	0.092 ±.008	0.065 ±.010
conv / CIFAR-10	0.3	-733.2 ±1.0	81.3 ±.4	0.106 ±.008	0.041 ±.001
conv / CIFAR-10	0.5	772.0 ±.3	38.6 ±.1	0.108 ±.008	0.040 ±.003
conv / CIFAR-10	1	2855.8 ±.1	12.8 ±.0	0.092 ±.007	0.046 ±.003

becomes $D(\frac{1}{2} + \log \sigma^* + \frac{1}{2} \log 2\pi)$, driven by $\log \mathrm{MSE}$, so β is then the only rate–distortion weight and the effective reconstruction weight, proportional to $1/\mathrm{MSE}$, rises as the model improves.

D.3 Clustering advantage versus the encoder's rate

The clustering comparison between merging and EM depends on the encoder's *rate* — the per-image $\mathrm{KL}(q(z \mid x) \,\|\, p(z))$ — and the calibrated-σ CIFAR-10 sweep separates the two cleanly (Table 13). We say rate rather than information deliberately: the KL is an *upper bound* on the mutual information $I(x; z)$, and the slack is exactly $\mathrm{KL}(q(z) \,\|\, p(z))$, the aggregated-posterior-to-prior gap this paper measures separately. Sweeping β from 0.1 to 8 moves the per-image KL over a $9\times$ range, $927 \to 108$ nats, at reconstruction quality between 26.3 and 21.5 dB. Across that range *merged NMI barely moves* (0.084–0.103, no monotone trend) while *EM falls monotonically* ($0.109 \to 0.043$). The two therefore tie, or EM edges ahead, when β is small ($\beta \leq 1$: differences of -0.006 to $+0.002$, none significant), and merging wins decisively once β is large ($\beta \geq 2$: $+0.023$, $+0.039$, $+0.057$, each $10/10$ seeds at p=0.002). This places the headline CIFAR-10 gap on the same axis: the headline cells, trained with a fixed σ=1, sit at KL 13 — beyond the right-hand end of Table 13 — precisely the regime where the gap is widest, and their EM value (0.045) is reproduced at β=8 here (0.043), where the latent is starved by β instead of by the decoder scale. Merging's robustness, not a uniform quality edge, produces the gap. The OOD columns move the opposite way (Section D.10): the inversion against SVHN is *deepest* where reconstruction is best and the latent richest (0.115 at β=1) and recovers toward 0.222 at β=8; against the semantically close CIFAR-100 every score stays near chance (0.52).

Table 10: Plain VAE, decoder-noise sweep (σ_{dec}, Gaussian likelihood, 10 seeds per cell). CIFAR-10 rows: convolutional architecture, latent columns only. OOD AUROC at $K = 50$ for the merged mixture, EM at equal K and the model's pixel-space likelihood; — where a score was not computed for that cell.

model	decoder_sigma	FashionMNIST mrg	EM	pixel	KMNIST mrg	EM	pixel
MLP / MNIST	0.1	0.882 ±.011	0.907 ±.013	0.999 ±.000	0.922 ±.012	0.943 ±.011	0.999 ±.000
MLP / MNIST	0.3	0.815 ±.034	0.880 ±.039	0.999 ±.000	0.887 ±.013	0.929 ±.019	0.999 ±.000
MLP / MNIST	0.5	0.700 ±.040	0.833 ±.038	0.999 ±.000	0.738 ±.028	0.864 ±.025	0.999 ±.000
conv / MNIST	0.1	0.933 ±.021	0.948 ±.016	0.999 ±.000	0.957 ±.007	0.966 ±.005	0.999 ±.000
conv / MNIST	0.3	0.877 ±.020	0.906 ±.013	0.999 ±.000	0.946 ±.006	0.959 ±.005	0.999 ±.000
conv / MNIST	0.5	0.843 ±.022	0.890 ±.023	0.999 ±.000	0.925 ±.008	0.951 ±.008	0.999 ±.000
conv / CIFAR-10	0.1	0.132 ±.005	0.092 ±.003	—	0.523 ±.001	0.518 ±.002	—
conv / CIFAR-10	0.3	0.242 ±.003	0.149 ±.002	—	0.527 ±.001	0.509 ±.002	—
conv / CIFAR-10	0.5	0.288 ±.005	0.241 ±.005	—	0.535 ±.001	0.508 ±.003	—
conv / CIFAR-10	1	0.317 ±.008	0.273 ±.018	—	0.551 ±.002	0.532 ±.002	—

Table 11: Clustering at K=10 (vs. class labels), full N. Merged-GMM has higher NMI on both datasets; Hungarian-matched accuracy is mixed. The CIFAR-10 columns are the fixed-σ=1 cells; the calibrated-σ sweep is in Section D.3.

	MNIST		CIFAR-10	
Method	NMI	acc.	NMI	acc.
EM-GMM, K=10	0.534	**0.618**	0.038	0.153
Merged-GMM (ours)	**0.589**	0.601	**0.089**	**0.215**

D.4 CelebA-64 and the resolution study

CelebA carries no class labels, so these cells have no clustering scores. All share one protocol: using the same official split, learning rates and training lengths are approved on the validation split, and mixtures wehre fittet on the whole aggregated posterior. The ladder's first step (Table 16) isolates the likelihood: from rung A, trained and scored with a Bernoulli likelihood on continuous RGB, to rung B, the same network trained and scored under the calibrated σ, the pixel score against SVHN falls from 0.769 to 0.153. Under a Gaussian likelihood both kinds of score fail against SVHN: the pixel likelihood is inverted (0.16–0.40 in the β, σ and resolution sweeps, 0.08–0.17 on the calibrated rungs), and latent density sits near chance (0.48–0.57) or below it (rungs B, D and E: 0.38–0.43). Against CIFAR-10 both work (pixel 0.72–0.93, latent 0.71–0.85 in the three sweeps), but by no constant margin. Pixel space leads by 0.04–0.12 across resolutions; in the β and σ sweeps EM is level with it or ahead (0.744–0.797 vs. 0.724–0.761), and so is the merged mixture except at σ_{dec}=1 (0.742 vs. 0.761); on the largest rungs, D and E, the pixel score drops to 0.66–0.67 and the latent scores lead by 0.07–0.09. Under the fixed σ_{dec}=1 the three reconstruction-based scores come apart (SVHN: pixel 0.286, reconstruction probability 0.241, error at the mean 0.237; Table 15); under the calibrated σ they are one score. The resolution study runs CelebA at 16, 32 and 64 pixels with a calibrated σ-VAE (Table 17): latent density degrades with resolution — at every step against CIFAR-10, from 32 to 64 px against SVHN — but the pixel-vs-latent gap does *not* widen in pixel space's favour against SVHN, where pixel space is the worse of the two at every resolution. Plausibly, at fixed latent dimension a growing input makes the encoder discard proportionally more input-specific information, and the aggregated posterior drifts toward a generic natural-image prior that also covers the OOD sets. An example of samples are shown in Figure 5.

Co-training and sampling. Co-training (Table 18) moves the latent score against SVHN from 0.507 to 0.572 (U, p=0.0002), and a post-hoc merge on the co-trained encoder scores the same (0.564): the gain lies in the encoder's posterior, and at test time the trained-in mixture is interchangeable with a fresh merge of it — as it must be, since both are merges of the same posterior. The encoder changed because its objective did, in two ways. Half of each image's reconstruction is decoded from a draw of its cluster's merged Gaussian, which contracts every cluster toward its component and leaves low-density space between clusters. And the co-trained cells carry the free-bits floor that the plain cells lack. All these scores stay near chance. The gain is not a checkpoint effect: the co-trained cells report their last weights and the plain cells their validation-best, and scored at

Table 12: Clustering NMI at $K{=}10$ over the rate-distortion sweeps, 10 seedsvper cell: range of per-value means (min–max), with the largest across-seed stdvin the sweep in parentheses. Merged-GMM is stable in both senses; EM swingsvacross the sweep and across seeds. Which method is ahead depends on the regimev(see text and the Wilcoxon tests). CIFAR-10 rows: calibrated decoder scaling, convolutional architecture, $\beta \in \{0.1, \ldots, 8\}$ and $\sigma_{\text{dec}} \in \{0.1, \ldots, 1\}$.

Model / data	Sweep	Merged NMI (max std)	EM NMI (max std)
MLP / MNIST	β	0.51–0.59 ($\pm$0.04)	0.36–0.59 ($\pm$0.12)
MLP / MNIST	σ_{dec}	0.53–0.58 ($\pm$0.03)	0.43–0.60 ($\pm$0.12)
conv / MNIST	β	0.54–0.61 ($\pm$0.04)	0.48–0.65 ($\pm$0.08)
conv / MNIST	σ_{dec}	0.54–0.63 ($\pm$0.04)	0.54–0.66 ($\pm$0.04)
conv / CIFAR-10	β	0.08–0.10 ($\pm$0.01)	0.04–0.11 ($\pm$0.01)
conv / CIFAR-10	σ_{dec}	0.09–0.11 ($\pm$0.01)	0.04–0.07 ($\pm$0.01)

Table 13: CIFAR-10 β sweep under the calibrated decoder (analytic optimal σ, $z{=}256$), 60 epochs, 10 seeds per cell, mean$\pm$std. σ and PSNR are recovered from the reconstruction term. p is a paired Wilcoxon test of merged vs. EM clustering NMI across seeds. Merged NMI is almost flat over a $9\times$ range of latent rate while EM falls monotonically with it; the fixed-$\sigma{=}1$ cells sat at KL 13, i.e. beyond the right-hand end of this table.

β	σ	PSNR (dB)	KL (nats)	merged NMI	EM NMI	diff	wins	p	SVHN merged	SVHN pixel
0.1	0.048	26.3	927	0.103	0.109	-0.006	2/10	0.084	0.178	0.575
0.3	0.049	26.2	730	0.094	0.096	-0.003	2/10	0.131	0.149	0.554
0.5	0.049	26.2	635	0.086	0.090	-0.005	2/10	0.193	0.132	0.539
1.0	0.051	25.9	509	0.084	0.082	$+0.002$	7/10	0.322	0.115	0.520
2.0	0.055	25.2	368	0.092	0.069	$+0.023$	10/10	0.002	0.118	0.523
4.0	0.066	23.6	217	0.092	0.054	$+0.039$	10/10	0.002	0.154	0.558
8.0	0.084	21.5	108	0.100	0.043	$+0.057$	10/10	0.002	0.222	0.584

For reference, the fixed-$\sigma{=}1$ cells ($z{=}64$) gave merged 0.089, EM 0.045, SVHN merged 0.311 at KL 13.0 and 16.7 dB. EM's 0.043 at $\beta{=}8$ reproduces that EM value once the latent is starved by a legitimate mechanism.

their last weights the plain cells reach 0.498 against SVHN ($+0.075$ for the partition, U, $p{=}0.0002$). On sampling (Table 19), both $K{=}50$ mixtures beat $\mathcal{N}(0, I)$ on FID and on the two-sample test at every rung, on every seed. On the residual rungs C–E, however, their draws sit further from the nearest training latent than $\mathcal{N}(0, I)$ draws do; on rung C (the $\beta{=}1$ models, whose latent radii we measured), the $\mathcal{N}(0, I)$ draws lie inside the shell that the training latents occupy (Appendix D.9).

D.5 Progressive merge-training

Progressive cells co-train a partition of the training points with the VAE: after a warmup the partition shrinks on a geometric schedule from singletons to K_{target}. Pairs are merged under a W_2^2 cost weighted by cluster sizes ($\frac{n_i n_j}{n_i + n_j} W_2^2$, closed-form for diagonal Gaussians), so that merging two large clusters is expensive; a capped fraction of points may move to a better cluster each epoch (damped k-means-style reassignment); and any cluster growing past a fixed share of the data is split. Each design element closes an empirically observed failure channel rooted in cluster-size asymmetry: without the anchor and free bits the objective feeds a posterior-collapse loop (KL 21 $\rightarrow$ 3.6 nats), and with density-based assignment or plain nearest-pair merging a single wide cluster ended holding 14,949 of 15,000 points; the damped reassignment, Ward weighting, split guard and geometric schedule each close one channel, and all are needed. Each cell evaluates three mixtures on its own encoder — the co-trained partition, a post-hoc merge, and a GPU EM at equal K — so the effect of the objective and the effect of the mixture are separable. Table 22 sweeps the partition budget and includes the $\alpha_{\max} = 0$ control, which removes the cluster-reconstruction term and leaves plain ELBO, the KL floor (*free bits*, Kingma et al., 2016) and the co-evolving partition.

Table 14: CelebA-64, β sweep (OOD only; CelebA carries no class labels). Typical used split, (Tolstikhin et al., 2018) based crop, calibrated σ with the analytic optimal scale, 25 epochs at learning rate 3×10^{-4}, validation-best checkpoint, mixtures on the whole aggregated posterior, 10 seeds per cell; recon and KLD are the last training epoch's. OOD AUROC at $K = 50$ for the merged mixture, EM at equal K, the model's pixel-space likelihood, the reconstruction probability $\mathbb{E}_{q(z|x)}[\log p(x \mid z)]$ (rprob) and the reconstruction error of the decoded posterior mean (recon).

model	beta	SVHN mrg	EM	pixel	rprob	recon	CIFAR-10 mrg	EM	pixel	rprob	recon
residual / CelebA-64	1	0.504 $\pm$.014	0.502 $\pm$.008	0.172 $\pm$.005	0.174 $\pm$.006	0.174 $\pm$.006	0.739 $\pm$.006	0.744 $\pm$.008	0.742 $\pm$.006	0.743 $\pm$.006	0.744 $\pm$.006
residual / CelebA-64	4	0.494 $\pm$.025	0.489 $\pm$.025	0.165 $\pm$.007	0.168 $\pm$.007	0.168 $\pm$.007	0.753 $\pm$.012	0.763 $\pm$.011	0.737 $\pm$.008	0.739 $\pm$.008	0.739 $\pm$.008

Table 15: CelebA-64, decoder-noise sweep (fixed σ_{dec}, Gaussian likelihood), otherwise as Table 14. OOD AUROC at $K = 50$ for the merged mixture, EM at equal K, the model's pixel-space likelihood, the reconstruction probability $\mathbb{E}_{q(z|x)}[\log p(x \mid z)]$ (rprob) and the reconstruction error of the decoded posterior mean (recon).

model	decoder_sigma	SVHN mrg	EM	pixel	rprob	recon	CIFAR-10 mrg	EM	pixel	rprob	recon
residual / CelebA-64	0.3	0.538 $\pm$.064	0.524 $\pm$.068	0.159 $\pm$.014	0.149 $\pm$.016	0.149 $\pm$.015	0.762 $\pm$.023	0.769 $\pm$.025	0.724 $\pm$.015	0.721 $\pm$.018	0.720 $\pm$.018
residual / CelebA-64	1	0.493 $\pm$.010	0.495 $\pm$.016	0.286 $\pm$.005	0.241 $\pm$.007	0.237 $\pm$.006	0.742 $\pm$.004	0.797 $\pm$.008	0.761 $\pm$.004	0.743 $\pm$.006	0.728 $\pm$.007

CIFAR-10 and STL-10 grids. We use a onvolutional architecture, a calibrated decoder, 15k training images, report the last weights. The co-trained partition beats a post-hoc merge on its own encoder in 9 of 10 cells ($+0.009$ to $+0.023$ NMI; not significant only on CIFAR-10 at β=1). Against EM on its own encoder it wins on STL-10 for every $\beta \geq 1$; on CIFAR-10 it wins narrowly at β=0.5, ties for $\beta \in \{1, 3, 4\}$ and loses at β=10 (0.087 vs. 0.104, 0/10 seeds, p=0.002). The co-trained objective costs reconstruction — 17–20 dB PSNR against 23–24 dB for the data-matched plain model — and removing the cluster-reconstruction term recovers 2–4 dB on CIFAR-10. On CIFAR-10 the partition budget helps little: NMI plateaus at 0.109–0.113 for K_{target} from 100 to 4000 (best 2000 vs. 50: $+0.011$, 8/10, p=0.04) and drops to 0.105 at 5000; none of these survives Holm correction over the six-budget family (adjusted $p \geq 0.22$). The same-encoder advantage over post-hoc merging shrinks with the budget ($+0.018$ at 2000, $+0.007$ at 4000, none at 5000), while the advantage over EM grows ($+0.016$ to $+0.020$ for $K_{\mathrm{target}} \geq 500$) as EM's NMI declines from 0.110 to 0.090. At larger budgets the co-trained partition scores OOD slightly more inverted than a post-hoc merge (-0.011 to -0.019 AUROC against SVHN for $K_{\mathrm{target}} \geq 2000$, 0/10 seeds), and the model's own pixel score inverts further as reconstruction improves (0.166 $\rightarrow$ 0.057 at 18.3 $\rightarrow$ 22.0 dB PSNR).

MNIST budgets and the ablation, in full. On conv-MNIST the co-trained partition reaches 0.696 ± 0.032 at β=3, K_{target}=50; the epoch-matched plain pipeline at the same β gives 0.639 (p=0.001) and the best plain pipeline of any β 0.662 ± 0.035 (p=0.06). At K_{target}=1000 the trained hierarchy merged to 10 reaches 0.740 ± 0.026, a plateau of 0.733–0.740 for all $K_{\mathrm{target}} \geq 1000$ (U, p=0.006 against direct-to-50 training; p=0.0002 against the best plain pipeline). Co-trained NMI is nearly flat over β: 0.63–0.70 on conv-MNIST (plain EM 0.385–0.662), 0.087–0.115 on conv-CIFAR-10, 0.153–0.162 on conv-STL-10; on MLP-MNIST it ties the plain pipeline (0.510–0.588 vs. 0.523–0.587) and on CIFAR-10 EM fitted to the co-trained encoder is steadier still (0.101–0.110). Removing the cluster-reconstruction term ($\alpha_{\max}$=0) retains most of the MNIST gain (0.667 vs. 0.688, n.s.); on conv-CIFAR-10 it beats the full objective at β=1 (0.126 vs. 0.108, 9/10, p=0.004) and β=3 (0.117 vs. 0.102, 8/10, p=0.04), ties at β=0.5 and loses at β=10 (0.065 vs. 0.087, p=0.02); it beats the data-matched plain pipeline at both matched β (0.117 vs. 0.100, 10/10, p=0.002; 0.117 vs. 0.103, 9/10, p=0.03); on STL-10 it changes nothing. At K_{target}=1000 the control reproduces the conv-MNIST ceiling (0.739 ± 0.020 vs. 0.740 ± 0.026, p=0.85) and ties on conv-CIFAR-10 (0.111 vs. 0.110) while reconstructing 1.6 dB better. On conv-CIFAR-10 the co-trained encoder lifts EM from 0.042 to 0.101 at β=3 and holds it at 0.10–0.11 across all five β; that lift needs the cluster term (without it EM on the same encoder falls back to 0.069 at β=3 and to 0.051 at K_{target}=1000), and a frozen-prior distillation control (a fixed-σ=1 cell) did not reproduce it. Training the hierarchy with $K_{\mathrm{target}} \in \{100, \ldots, 5000\}$ instead of building it post hoc on the same encoder changes density-OOD by at most ±0.008 AUROC at every common K on MNIST; on conv-CIFAR-10 the clustering gain is $+0.010$ to $+0.018$ up to K_{target}=2000 and vanishes at 5000, while the SVHN density drifts up to 0.019 further into the inversion (Holm-supported at five of six budgets, against three of six for the structure gain). At matched large K the trained construc-

Table 16: OOD AUROC (↑), CelebA-64 in-distribution, split from (Liu et al., 2015), centre-crop 140 then resize 64 (Appendix C), mean±std over seeds, mixtures (K=50) on the whole aggregated posterior. **Rows A–E are a ladder: each adds one change to the row above**, and each rung is trained at its own validation-approved learning rate and length (samples seen = epochs × 162,770 training images, in millions; late best = seeds whose validation-best epoch lies in the last quarter of training, i.e. still improving at the approved length — the lengths of A and B extrapolate a 75-epoch probe whose best epoch was at its end). A trains and scores with a Bernoulli likelihood on continuous RGB; B trains and scores the same network on the same data under the calibrated σ. Row C is the same ten models as the β=1 row of Table 14.

rung (CelebA)	n	epochs	lr	samples (M)	late best	PSNR	SVHN pixel	merged	EM	CIFAR-10 pixel	merged	EM
A: plain conv, Bernoulli likelihood	10	100	10^{-3}	16.3	10/10	22.38 ±.02	0.769 ±.000	0.398 ±.009	0.375 ±.013	0.762 ±.000	0.724 ±.006	0.734 ±.006
B: + calibrated σ	10	95	3×10^{-3}	15.5	6/10	22.48 ±.02	0.153 ±.003	0.419 ±.048	0.426 ±.039	0.716 ±.003	0.731 ±.020	0.745 ±.019
C: + residual encoder/decoder	10	25	3×10^{-4}	4.1	3/10	23.09 ±.01	0.172 ±.005	0.504 ±.014	0.502 ±.008	0.742 ±.006	0.739 ±.006	0.744 ±.008
D: + z=128	10	25	3×10^{-4}	4.1	7/10	25.07 ±.15	0.083 ±.009	0.400 ±.041	0.396 ±.038	0.659 ±.016	0.744 ±.027	0.747 ±.028
E: + wider encoder/decoder	10	20	10^{-4}	3.3	3/10	25.03 ±.02	0.089 ±.007	0.395 ±.014	0.384 ±.013	0.666 ±.011	0.732 ±.008	0.732 ±.008

Table 17: Input resolution, CelebA in-distribution (calibrated σ-VAE with the analytic optimal scale, official split and crop, mixtures on the whole aggregated posterior, 10 seeds per cell; 45/35/35 epochs at learning rates 10^{-3}/3×10^{-4}/3×10^{-4} for 16/32/64 px, each approved on validation). The OOD sets are resized to the in-distribution resolution. OOD AUROC at $K = 50$ for the merged mixture, EM at equal K, the model's pixel-space likelihood, the reconstruction probability $\mathbb{E}_{q(z|x)}[\log p(x \mid z)]$ (rprob) and the reconstruction error of the decoded posterior mean (recon).

model	resolution	SVHN mrg	EM	pixel	rprob	recon	CIFAR-10 mrg	EM	pixel	rprob	recon
residual / CelebA-64	16	0.553 ±.020	0.549 ±.023	0.398 ±.017	0.389 ±.021	0.390 ±.020	0.845 ±.006	0.849 ±.007	0.893 ±.007	0.892 ±.008	0.892 ±.008
residual / CelebA-64	32	0.568 ±.016	0.561 ±.016	0.319 ±.005	0.320 ±.005	0.321 ±.005	0.811 ±.007	0.817 ±.006	0.927 ±.001	0.928 ±.001	0.928 ±.001
residual / CelebA-64	64	0.478 ±.028	0.475 ±.023	0.175 ±.010	0.177 ±.010	0.177 ±.010	0.706 ±.011	0.713 ±.010	0.748 ±.010	0.750 ±.010	0.750 ±.010

tion crosses EM on MNIST (K=4000: 0.927 vs. 0.912, p=0.03; K=5000: 0.928 vs. 0.900). Pixel OOD is unimpaired on MNIST (0.997 vs. 0.999); on conv-CIFAR-10 the co-trained model's pixel score is less inverted than the data-matched plain model's (SVHN 0.06–0.17 vs. 0.03–0.04), still far below chance. On CelebA-64, co-training moves the *latent* AUROC against SVHN from 0.507 to 0.572 (U, p=0.0002), still near chance, and does not improve it against CIFAR-10 (0.733 vs. 0.740; Table 18).

D.6 Prior-term ablation

The prior arms feed the compressed mixture back into training as an auxiliary cross-entropy term, leaving the analytic KL to $\mathcal{N}(0, I)$ untouched. The arms add one ingredient at a time: the prior term alone; then an exponential moving average of the prior's parameters across epochs (matched component-to-component between updates), which keeps the prior from chasing the posterior it is shaping; then shrinkage of the prior toward $\mathcal{N}(0, I)$. A further arm tracks the prior without letting it enter the loss, which isolates the averaged mixture as a passive density estimator. The last column scores latent OOD under the trained-in prior itself. On the convolutional CIFAR-10 and STL-10 grids no prior arm changes clustering significantly in any of 24 cells ($|\Delta\mathrm{NMI}| \leq 0.008$). The trained-in mixture scores OOD slightly less inverted than a post-hoc merge on the same encoder ($+0.01$ to $+0.03$ AUROC against SVHN), but the arm that tracks the prior without training against it shows the same gain, so the gain comes from the averaged mixture as an estimator, not from the loss term. Every score stays inverted (0.21–0.29).

D.7 Relation to ex-post density estimation

Ghosh et al. (2020) introduce *ex-post density estimation*: after training, they fit a density to the aggregated latent codes and draw samples from it instead of from $\mathcal{N}(0, I)$, comparing the prior (or, for their deterministic autoencoders, a single full-covariance Gaussian fitted to the codes) against a 10-component GMM, and report improved FID on CelebA-64 ($48.12 \rightarrow 45.52$ for a vanilla VAE; $44.74 \rightarrow 40.95$ for RAE-SN, from a single full-covariance Gaussian fitted to its codes to the mixture). *What is shared* is the premise: the aggregated posterior does not match the prior, and sampling from an estimate of it improves generation without retraining the decoder; our drop-in prior result reproduces that conclusion on different models and metrics and should be read as confirmation. *What*

Table 18: Co-training on CelebA-64 against the plain β=1 cells it is matched to: the same 162,770 training images, 25 epochs, learning rate, batch and seeds ($\alpha_{\max}$=0.5; batches drawn uniformly, not grouped by cluster). OOD AUROC (↑), mean±std, and the difference of the means to the plain cell's counterpart (the trained-in partition against plain post-hoc merged) with the two-sided Mann–Whitney U test (separately trained runs, marked U; Section 4). Both sides are scored by the same code, the co-trained pipeline's evaluator; the plain cells' own evaluator (Table 14) gives values within 0.025 per seed and 0.004 in the mean. The co-trained cells report their last weights, the plain cells their validation-best.

encoder	score	n	SVHN	Δ vs. plain	CIFAR-10	Δ vs. plain
co-trained	trained-in partition	10	0.572 ±.013	+0.065 (U, p=0.0002)	0.733 ±.010	-0.007 (U, p=0.0890)
co-trained	post-hoc merged	10	0.564 ±.018	+0.057 (U, p=0.0002)	0.734 ±.010	-0.006 (U, p=0.0890)
co-trained	EM	10	0.564 ±.011	+0.064 (U, p=0.0002)	0.728 ±.005	-0.015 (U, p=0.0017)
co-trained	pixel IWAE	10	0.178 ±.011	+0.006 (U, p=0.2413)	0.746 ±.010	+0.004 (U, p=0.3075)
plain	post-hoc merged	10	0.507 ±.013	—	0.740 ±.007	—
plain	EM	10	0.500 ±.013	—	0.743 ±.008	—
plain	pixel IWAE	10	0.172 ±.005	—	0.742 ±.006	—

PSNR co-trained 22.93 ±.03 dB vs. plain 23.09 ±.01 dB; largest cluster 0.041 ±.005 of the training images and $\exp H$ = 46.1 ±.8 effective clusters of K_{target}=50.

differs: (i) construction — their mixture is fitted by EM ("at most 100 iterations") with full covariances, seed-dependent and re-fit per component count, ours by one deterministic moment-preserving pass whose every K is a coarsening of the previous one; (ii) selection of K — they fix K=10, noting that "fitting GMMs with up to 100 components only improved results marginally"; we measure the entire path, observe the same saturation, and select K by held-out in-distribution likelihood without OOD access; (iii) which role is tested — they evaluate the mixture only as a *sampler*, whereas used as a prior *density* inside the model's own likelihood an equal-K EM mixture is unsafe and craters the pixel-space score on some seeds where the merged mixture does not (Section 3.4). Sampling cannot expose this, because samples are drawn *from* components, whereas density evaluation must score arbitrary z. What is ours is the construction — deterministic, nested, EM-free — and the finding that the choice of estimator becomes safety-relevant as soon as the mixture is used as a density rather than only as a sampler.

D.8 Matched-data control and augmentation

The matched control retrains the plain pipeline on exactly the 15 000 images the progressive pipeline uses for the same seed, and additionally scores it with the progressive pipeline's own evaluator, so that mixtures, EM implementation and the number of importance samples are identical on both sides and only training differs. On CIFAR-10 (convolutional architecture, calibrated recipe) the co-trained partition beats the matched plain pipeline at β=0.5 (0.115 vs. 0.101, 9/10, p=0.004) and ties it at β=3 (0.102 vs. 0.104, p=0.49); co-training lowers the rate by 35–44% and costs 4.0–4.7 dB PSNR.

The augmentation study repeats the plain pipeline (on the full and on the matched training set) and the co-trained pipeline with the geometric policies of Table 3. On CIFAR-10 and STL-10 augmentation moves no clustering NMI by more than 0.009 and moves latent SVHN AUROC by at most 0.008 in either direction (conv-CIFAR-10 vs. SVHN, merged: 0.132 → 0.132 on full data and 0.170 → 0.173 on matched data). On conv-MNIST it raises clustering (EM NMI by up to 0.21, 10/10) and moves latent OOD sharply — merged-GMM AUROC 0.916 → 0.999 at β=0.5 (FashionMNIST) and 0.834 → 0.998 at β=3 — and, compared within the same trained model, the latent density then beats the model's own pixel likelihood (0.9992 vs. 0.9975 on FashionMNIST and 0.9979 vs. 0.9925 on KMNIST at β=0.5, 10/10; 0.9983 vs. 0.9956 on FashionMNIST at β=3, 9/10, p=0.004), except on KMNIST at β=3, where pixel stays ahead (0.9959 vs. 0.9899, 0/10). In each reversal of the pixel-over-latent ordering, 86–98% of the change is the latent score improving, the rest a small pixel-score penalty (−0.002 to −0.007), the expected cost of training on the augmented distribution.

Table 19: Sampling on CelebA-64 from alternative priors on each ladder rung's frozen decoder (Table 16), mean±std over 10 seeds. Reconstructions are not sampled images, but reconstructions from test images. FID (Heusel et al., 2017) is computed with clean-fid (Parmar et al., 2022) against the official test split (19,962 images) at 10k and 50k generated images (reconstructions: 10k test images) — a protocol not comparable with Ghosh et al. (2020)'s; 1NN is the leave-one-out two-sample accuracy against the first 2,000 test images (0.5 = indistinguishable; the tempered-prior sweep of Section 5 draws a random 2,000 and reads lower on the same models — by 0.02–0.03 for $\mathcal{N}(0, I)$ and the tempered prior, by 0.05–0.06 for the K=50 mixtures — with the same ordering); hole is the mean distance from a sampled z to the nearest of the 162,770 training latents, not reported for the tempered prior; diversity is the mean pairwise pixel distance of 512 samples. The tempered prior exists on the residual rungs only. Lower is better for FID, 1NN and hole.

rung	prior	FID@10k	FID@50k	1NN	hole	diversity
A: plain conv, Bernoulli likelihood	reconstructions	63.7 ±.9	—	—	—	—
	$\mathcal{N}(0, I)$	72.1 ±1.0	71.4 ±.9	0.684 ±.008	7.14 ±.01	34.4 ±.3
	EM, K=10	71.4 ±.9	70.7 ±.9	0.638 ±.005	6.65 ±.05	34.5 ±.5
	merged, K=10	70.9 ±1.0	70.2 ±1.0	0.657 ±.008	7.07 ±.03	35.1 ±.5
	EM, K=50	69.5 ±.9	68.7 ±.9	0.612 ±.006	6.61 ±.04	34.9 ±.4
	merged, K=50	70.1 ±.9	69.4 ±.9	0.630 ±.008	7.03 ±.02	34.9 ±.2
B: + calibrated σ	reconstructions	56.5 ±1.5	—	—	—	—
	$\mathcal{N}(0, I)$	139.2 ±10.4	137.9 ±10.4	0.724 ±.025	7.19 ±.03	33.0 ±.5
	EM, K=10	70.6 ±3.2	69.9 ±3.4	0.688 ±.008	6.02 ±.16	34.8 ±.4
	merged, K=10	72.2 ±3.6	71.5 ±3.5	0.708 ±.013	6.16 ±.16	34.8 ±.5
	EM, K=50	67.2 ±2.9	66.4 ±2.9	0.643 ±.008	5.87 ±.15	35.3 ±.4
	merged, K=50	69.5 ±2.7	68.8 ±2.8	0.668 ±.008	6.03 ±.15	34.9 ±.4
C: + residual encoder/decoder	reconstructions	67.8 ±.9	—	—	—	—
	$\mathcal{N}(0, I)$	71.9 ±1.4	71.2 ±1.3	0.868 ±.002	7.81 ±.03	20.7 ±.2
	tempered, t=0.6	97.2 ±1.4	96.7 ±1.3	0.882 ±.004	—	12.3 ±.2
	EM, K=10	67.1 ±1.5	66.3 ±1.5	0.707 ±.006	9.53 ±.09	33.3 ±.3
	merged, K=10	66.7 ±1.5	66.0 ±1.5	0.691 ±.014	9.85 ±.09	31.8 ±.4
	EM, K=50	66.4 ±1.5	65.7 ±1.5	0.659 ±.007	9.24 ±.09	34.3 ±.4
	merged, K=50	66.6 ±1.6	65.7 ±1.5	0.660 ±.011	9.59 ±.10	33.0 ±.3
D: + z=128	reconstructions	56.3 ±1.5	—	—	—	—
	$\mathcal{N}(0, I)$	70.9 ±4.6	70.2 ±4.7	0.875 ±.011	11.75 ±.17	20.8 ±1.1
	tempered, t=0.6	83.3 ±5.5	82.8 ±5.4	0.905 ±.002	—	12.2 ±.6
	EM, K=10	65.8 ±4.7	65.1 ±4.8	0.734 ±.012	13.32 ±.57	32.1 ±.3
	merged, K=10	67.4 ±4.5	66.7 ±4.5	0.687 ±.025	13.69 ±.57	30.4 ±.3
	EM, K=50	63.6 ±4.6	62.9 ±4.7	0.692 ±.012	13.00 ±.54	33.4 ±.4
	merged, K=50	65.4 ±4.5	64.7 ±4.7	0.676 ±.015	13.40 ±.53	31.7 ±.3
E: + wider encoder/decoder	reconstructions	55.1 ±1.5	—	—	—	—
	$\mathcal{N}(0, I)$	70.1 ±1.9	69.4 ±1.8	0.877 ±.006	11.72 ±.03	21.0 ±.1
	tempered, t=0.6	85.2 ±1.4	84.7 ±1.5	0.901 ±.002	—	12.2 ±.1
	EM, K=10	64.6 ±2.6	63.9 ±2.6	0.745 ±.012	13.26 ±.14	32.0 ±.4
	merged, K=10	66.1 ±3.0	65.4 ±2.9	0.690 ±.009	13.60 ±.13	30.2 ±.4
	EM, K=50	62.6 ±2.7	61.8 ±2.6	0.697 ±.007	12.93 ±.11	33.3 ±.2
	merged, K=50	64.2 ±2.7	63.4 ±2.7	0.680 ±.005	13.34 ±.11	31.7 ±.5

D.9 THE MERGE PATH, SAMPLING, CODEBOOKS AND THROUGHPUT

Table 25 traces OOD along the nested merge path: one pass yields every K from N down to 5, while each EM entry is an independent refit, and at full N an EM fit at $K = 20\,000$ was memory-infeasible (77 GB responsibility matrix). The path is a single merge run, recorded at each K; it uses the Ward-weighted merger of the co-trained pipeline, which merges in distance order by construction. The post-hoc cells elsewhere instead call the full-matrix merger once per K; those runs are nested by the total order of Section 3.2. At N=60k on MNIST the merged curve peaks at interior $K \approx$ 4–8k (0.955/0.963 against FashionMNIST/KMNIST) and dips at K=N (0.936/0.934): the raw posterior is over-sharp, and moderate merging smooths it. EM keeps rising there ($0.964 \rightarrow 0.970$ from K=1000 to 5000), but on every 15k-latent MNIST cell it peaks at K=1000 and declines by 5000 (e.g. $0.952 \rightarrow 0.928$ on conv-MNIST), its component scales pinned at its covariance floor. Selecting K by held-out in-distribution likelihood lands within 0.001–0.033 AUROC of the path optimum. On CIFAR-10/SVHN the merged curve rises monotonically from K=10 upward ($0.354 \rightarrow 0.417$; K=5 sits slightly higher at 0.376) and is significantly less inverted than EM anywhere ($10/10$ seeds, p=0.002); these CIFAR-10 numbers come from the fixed-σ=1 encoders.

Table 20: Progressive merge-training across β ($K_{\text{target}} = 50$, 10 seeds). The co-trained partition, a post-hoc merge and a GPU EM are all evaluated on the same encoder. CIFAR-10 and STL-10 rows: calibrated decoder scale, convolutional architecture (configuration C of Table 30). Clustering NMI at $K = 10$.

model	β	NMI co-trained	NMI post-hoc	NMI EM
MLP / MNIST	0.5	0.560 $\pm$.031	0.519 $\pm$.043	0.493 $\pm$.050
MLP / MNIST	1	0.587 $\pm$.055	0.533 $\pm$.038	0.441 $\pm$.080
MLP / MNIST	3	0.588 $\pm$.026	0.531 $\pm$.044	0.365 $\pm$.049
MLP / MNIST	4	0.584 $\pm$.031	0.535 $\pm$.024	0.424 $\pm$.036
MLP / MNIST	10	0.510 $\pm$.025	0.478 $\pm$.017	0.495 $\pm$.031
conv / MNIST	0.5	0.658 $\pm$.026	0.547 $\pm$.031	0.579 $\pm$.024
conv / MNIST	1	0.655 $\pm$.035	0.579 $\pm$.019	0.580 $\pm$.052
conv / MNIST	3	0.696 $\pm$.032	0.608 $\pm$.032	0.628 $\pm$.029
conv / MNIST	4	0.679 $\pm$.023	0.592 $\pm$.054	0.609 $\pm$.029
conv / MNIST	10	0.634 $\pm$.024	0.576 $\pm$.033	0.596 $\pm$.054
conv / CIFAR-10	0.5	0.115 $\pm$.009	0.107 $\pm$.010	0.110 $\pm$.009
conv / CIFAR-10	1	0.108 $\pm$.009	0.106 $\pm$.014	0.104 $\pm$.009
conv / CIFAR-10	3	0.102 $\pm$.011	0.087 $\pm$.009	0.101 $\pm$.010
conv / CIFAR-10	4	0.102 $\pm$.011	0.090 $\pm$.009	0.105 $\pm$.004
conv / CIFAR-10	10	0.087 $\pm$.012	0.077 $\pm$.013	0.104 $\pm$.011
conv / STL-10	0.5	0.153 $\pm$.009	0.143 $\pm$.009	0.146 $\pm$.014
conv / STL-10	1	0.154 $\pm$.013	0.135 $\pm$.009	0.145 $\pm$.015
conv / STL-10	3	0.159 $\pm$.012	0.145 $\pm$.007	0.151 $\pm$.007
conv / STL-10	4	0.162 $\pm$.008	0.138 $\pm$.014	0.142 $\pm$.014
conv / STL-10	10	0.157 $\pm$.008	0.135 $\pm$.009	0.139 $\pm$.005

Table 21: Progressive merge-training across β ($K_{\text{target}} = 50$, 10 seeds). The co-trained partition, a post-hoc merge and a GPU EM are all evaluated on the same encoder. CIFAR-10 and STL-10 rows: calibrated decoder scale, convolutional architecture (configuration C of Table 30). OOD AUROC for the co-trained partition and for the model's pixel-space likelihood; OOD-1/OOD-2 are the two OOD sets of each in-distribution dataset in the order of Table 2.

model	β	OOD-1 co-trained	OOD-1 pixel	OOD-2 co-trained	OOD-2 pixel
MLP / MNIST	0.5	0.796 $\pm$.051	0.996 $\pm$.002	0.853 $\pm$.025	0.989 $\pm$.003
MLP / MNIST	1	0.790 $\pm$.075	0.997 $\pm$.001	0.828 $\pm$.046	0.991 $\pm$.002
MLP / MNIST	3	0.776 $\pm$.105	0.997 $\pm$.001	0.752 $\pm$.060	0.992 $\pm$.002
MLP / MNIST	4	0.769 $\pm$.057	0.998 $\pm$.000	0.765 $\pm$.035	0.993 $\pm$.001
MLP / MNIST	10	0.604 $\pm$.096	0.998 $\pm$.000	0.502 $\pm$.117	0.992 $\pm$.001
conv / MNIST	0.5	0.806 $\pm$.039	0.994 $\pm$.001	0.918 $\pm$.010	0.992 $\pm$.001
conv / MNIST	1	0.811 $\pm$.030	0.997 $\pm$.002	0.916 $\pm$.010	0.994 $\pm$.001
conv / MNIST	3	0.786 $\pm$.046	0.997 $\pm$.001	0.895 $\pm$.019	0.993 $\pm$.001
conv / MNIST	4	0.760 $\pm$.052	0.997 $\pm$.001	0.874 $\pm$.016	0.993 $\pm$.001
conv / MNIST	10	0.669 $\pm$.050	0.996 $\pm$.001	0.767 $\pm$.039	0.990 $\pm$.001
conv / CIFAR-10	0.5	0.222 $\pm$.012	0.115 $\pm$.028	0.540 $\pm$.004	0.542 $\pm$.011
conv / CIFAR-10	1	0.231 $\pm$.017	0.142 $\pm$.039	0.537 $\pm$.005	0.550 $\pm$.007
conv / CIFAR-10	3	0.251 $\pm$.020	0.166 $\pm$.066	0.536 $\pm$.004	0.544 $\pm$.004
conv / CIFAR-10	4	0.251 $\pm$.013	0.160 $\pm$.033	0.532 $\pm$.009	0.549 $\pm$.008
conv / CIFAR-10	10	0.272 $\pm$.023	0.157 $\pm$.026	0.532 $\pm$.008	0.546 $\pm$.008
conv / STL-10	0.5	0.207 $\pm$.011	0.040 $\pm$.019	0.445 $\pm$.011	0.149 $\pm$.024
conv / STL-10	1	0.205 $\pm$.016	0.043 $\pm$.018	0.443 $\pm$.008	0.150 $\pm$.027
conv / STL-10	3	0.218 $\pm$.023	0.042 $\pm$.013	0.444 $\pm$.012	0.154 $\pm$.027
conv / STL-10	4	0.210 $\pm$.020	0.045 $\pm$.025	0.441 $\pm$.005	0.162 $\pm$.027
conv / STL-10	10	0.226 $\pm$.033	0.051 $\pm$.024	0.441 $\pm$.017	0.176 $\pm$.035

Table 26 is the training-free sampling study on frozen decoders. On the calibrated CelebA β=1 cell, with mixtures on the whole posterior, the tempered sweep gives 1NN 0.840 at t=1, 0.841 at 0.85, 0.852 at 0.7, 0.862 at 0.6 and 0.879 at 0.5, against merged 0.614 (a 15k-subsample merge: 0.615) and EM 0.598; precision runs $0.150 \rightarrow 0.085$ across the tempered priors against 0.182 (merged)

Table 22: Partition-budget sweep (conv encoders, $\beta = 3$, 10 seeds). The partition is trained at K_{target} and merged down to $K = 10$ for NMI. The last row of each block removes the cluster-reconstruction term ($\alpha_{\max} = 0$), leaving plain ELBO, the KL floor and the co-evolving partition. CIFAR-10 and STL-10 rows: calibrated decoder scale, convolutional architecture (configuration C of Table 30).

in-dist	K_{target}	NMI co-trained	NMI post-hoc	NMI EM	OOD-1 co-trained	OOD-2 co-trained
MNIST	100	0.693 ±.031	0.604 ±.032	0.620 ±.019	0.832 ±.040	0.916 ±.006
MNIST	500	0.723 ±.025	0.613 ±.025	0.617 ±.031	0.892 ±.019	0.948 ±.006
MNIST	1000	0.740 ±.026	0.614 ±.029	0.617 ±.032	0.888 ±.038	0.956 ±.007
MNIST	2000	0.733 ±.016	0.604 ±.024	0.620 ±.031	0.915 ±.027	0.959 ±.007
MNIST	4000	0.735 ±.022	0.607 ±.025	0.615 ±.028	0.927 ±.022	0.960 ±.006
MNIST	5000	0.738 ±.019	0.598 ±.030	0.615 ±.047	0.928 ±.016	0.960 ±.003
MNIST	1000 ($\alpha_{\max}$=0)	0.739 ±.020	0.584 ±.026	0.612 ±.031	0.912 ±.019	0.960 ±.005
CIFAR-10	100	0.109 ±.010	0.098 ±.008	0.110 ±.008	0.228 ±.009	0.535 ±.006
CIFAR-10	500	0.111 ±.006	0.093 ±.005	0.095 ±.008	0.218 ±.007	0.534 ±.002
CIFAR-10	1000	0.110 ±.009	0.096 ±.012	0.096 ±.009	0.213 ±.006	0.532 ±.002
CIFAR-10	2000	0.113 ±.009	0.095 ±.012	0.092 ±.008	0.214 ±.007	0.533 ±.002
CIFAR-10	4000	0.109 ±.006	0.102 ±.010	0.091 ±.007	0.213 ±.008	0.533 ±.002
CIFAR-10	5000	0.105 ±.009	0.106 ±.008	0.090 ±.006	0.214 ±.007	0.533 ±.001
CIFAR-10	1000 ($\alpha_{\max}$=0)	0.111 ±.006	0.094 ±.015	0.051 ±.005	0.195 ±.009	0.528 ±.002

Table 23: Co-trained (progressive) partition vs. post-hoc merge and EM on the *same* encoder, and vs. the plain-VAE pipelines, as clustering NMI at K=10: min–max of per-β means over $\beta \in \{0.5, 1, 3, 4, 10\}$, 10 seeds per cell. Progressive has the highest minimum on conv-MNIST and STL-10 (rate-distortion robustness at the pipeline level) and the highest maximum on conv-MNIST (0.696 vs. 0.662); on MLP-MNIST it ties the plain pipeline at both ends (0.510–0.588 vs. 0.523–0.587). CIFAR-10 and STL-10 rows: calibrated recipe, convolutional architecture; plain references at $\beta \in \{0.5, 1, 3, 10\}$ (and 4 on CIFAR-10) on full training data. On CIFAR-10 the highest minimum belongs to EM fitted to the co-trained encoder.

	co-trained encoder			plain VAE	
Pair	Progressive	Post-hoc	EM	Merged	EM
MLP / MNIST	0.510–0.588	0.478–0.535	0.365–0.495	0.523–0.587	0.385–0.584
conv / MNIST	**0.634–0.696**	0.547–0.608	0.579–0.628	0.537–0.623	0.491–0.662
conv / CIFAR-10	0.087–0.115	0.077–0.107	**0.101–0.110**	0.087–0.105	0.041–0.088
conv / STL-10	**0.153–0.162**	0.135–0.145	0.139–0.151	0.136–0.146	0.136–0.152

and 0.205 (EM), and recall is ≤ 0.005 for every prior. The training latents lie on a shell of radius 11.2 (5th percentile 7.7; the aggregated posterior is wider than unit variance at a KL of 223 nats on the validation split). Draws from $\mathcal{N}(0, I)$ lie at radius 8.0 and the tempered draws at 6.8, 5.6, 4.8 and 4.0, whereas merged and EM draws lie at 11.2–11.3. Table 27 compares codebook conditions in the SQ-VAE-style model (on CIFAR-10: code reconstruction 56.9 vs. 57.4/57.5, perplexity 37.0 vs. 31.4/28.6, two-sample 0.596 vs. 0.605/0.607, NMI 0.079 vs. 0.061/0.064 for frozen vs. initialised/scratch). Substituting the mixtures for $\mathcal{N}(0, I)$ inside the pixel-space score moves the AUROC by a median $|\Delta|$ of 0.0001–0.001 across every dataset and 10–30 checkpoints per cell; on the IWAE-trained encoders (k=8, β=1, 10 seeds) the CIFAR-10/SVHN latent picture is unchanged (merged 0.116 vs. 0.115, EM 0.104 vs. 0.102 for ELBO training, both n.s., at a 14-nat lower KL). Table 28 gives the wall-clock comparison behind the throughput claim, with held-out log-density alongside so that speed is not compared across different-quality fits.

D.10 Decoder likelihood scaling, capacity, and architecture

A Gaussian-likelihood VAE trained with neither a fixed decoder scale nor a calibrated one has a reconstruction term of $\frac{1}{2}\|x - \mu\|^2$ — an implicit σ=1 against pixels in $[0, 1]$. Reconstruction is then worth about 0.01 nats per pixel while the KL runs at full strength, and the cheapest available solution is to emit each image's mean colour; the headline CIFAR-10 cells are of this kind. Because a β-VAE with a constant decoder variance is a σ-VAE with $\sigma^2 = \beta$ (Eq. 4), σ=1 is not a mis-set β but a mis-calibrated σ: the calibrated values of these models are $\sigma^* \approx 0.043$ (residual) to 0.053

Table 24: Prior-term ablation at $\beta = 3$, $K_{\text{target}} = 50$, 10 seeds. Each arm adds one ingredient: the auxiliary cross-entropy to the cluster prior, then averaging of the prior parameters across epochs, then shrinkage toward $\mathcal{N}(0, I)$; the last arm instead removes the cluster-reconstruction term. OOD columns are the first OOD set of each in-distribution dataset. CIFAR-10 and STL-10 rows: calibrated decoder scale, convolutional architecture (configuration C of Table 30).

model	arm	NMI co-trained	OOD co-trained	OOD trained-in prior	OOD pixel
conv / MNIST	no prior term	0.696 ±.032	0.786 ±.046	—	0.997 ±.001
conv / MNIST	prior tracked, not in the loss	0.688 ±.039	0.805 ±.029	0.686 ±.052	0.996 ±.001
conv / MNIST	prior term, no averaging	0.694 ±.039	0.781 ±.061	0.691 ±.076	0.997 ±.001
conv / MNIST	prior term + averaging	0.696 ±.028	0.777 ±.036	0.649 ±.065	0.998 ±.001
conv / MNIST	+ shrinkage to $\mathcal{N}(0, I)$	0.689 ±.028	0.772 ±.042	0.469 ±.076	0.997 ±.001
conv / MNIST	$\alpha_{\max} = 0$	0.667 ±.030	0.824 ±.027	—	0.998 ±.001
conv / CIFAR-10	no prior term	0.102 ±.011	0.251 ±.020	—	0.166 ±.066
conv / CIFAR-10	prior tracked, not in the loss	0.102 ±.011	0.251 ±.020	0.264 ±.041	0.166 ±.066
conv / CIFAR-10	prior term, no averaging	0.101 ±.014	0.244 ±.021	0.257 ±.026	0.163 ±.040
conv / CIFAR-10	prior term + averaging	0.105 ±.008	0.234 ±.013	0.254 ±.009	0.148 ±.024
conv / CIFAR-10	+ shrinkage to $\mathcal{N}(0, I)$	0.099 ±.013	0.239 ±.010	0.260 ±.033	0.170 ±.043
conv / CIFAR-10	$\alpha_{\max} = 0$	0.117 ±.010	0.194 ±.009	—	0.068 ±.018

Table 25: OOD AUROC along the nested merge path: one pass yields every K, while each EM entry is an independent refit. Columns are $K = N$ (the raw aggregated posterior) followed by decreasing K.

study (OOD set)	K=N	5000	1000	200	50	10
MNIST, full N (vs. FashionMNIST)	0.936	—	0.945	0.910	0.878	0.792
EM refits	—	0.970	0.964	0.940	0.920	0.815
conv MNIST, plain (vs. FashionMNIST)	0.936 ±.031	—	0.929 ±.046	0.918 ±.041	0.899 ±.038	0.860 ±.036
EM refits	—	0.928 ±.025	0.952 ±.019	0.947 ±.020	0.926 ±.025	0.883 ±.034
conv CIFAR-10, co-trained (vs. SVHN)	0.417 ±.031	—	0.397 ±.034	0.383 ±.032	0.373 ±.027	0.354 ±.021
EM refits	—	0.337 ±.015	0.313 ±.014	0.304 ±.013	0.296 ±.015	0.292 ±.018
conv STL-10, co-trained (vs. SVHN)	0.308 ±.022	0.308 ±.022	0.284 ±.025	0.282 ±.024	0.290 ±.025	0.295 ±.018
EM refits	0.274 ±.022	0.274 ±.022	0.277 ±.026	0.283 ±.028	0.283 ±.030	0.287 ±.029

(convolutional), i.e. $\sigma^{*2} \approx 0.002$–$0.003$ (Table 30), so the reconstruction term is weighted 360–$540\times$ too weakly — the equivalent of $\beta \approx 400$ at the calibrated scale. MNIST is unaffected: its binary cross-entropy weights the reconstruction at ~ 100 nats per image against a KL of 7–27, so the latent is never starved. The diagnostic is the per-image KL, which separates the two populations without overlap — CIFAR-10 cells at 2–20 nats against MNIST's 7–27 and STL-10's 56–111.

Table 30 separates three axes at matched epochs and seeds (10 seeds, 60 epochs, CIFAR-10, β=1). Three findings follow.

Scaling dominates; capacity is second; architecture is marginal. Replacing the implicit σ=1 with the analytic optimal $\sigma^{*2} = \text{MSE}(x, \mu)$ gains 4.4 dB, and raising the latent from 64 to 256 a further 4.6 dB. Doubling the channel width adds nothing (+0.0 dB) and the full residual encoder/decoder — GroupNorm/SiLU, two residual blocks per stage, $5.7\times$ the parameters — adds 0.8 dB. The fixed σ=1 therefore produces formless reconstructions that no architectural change repairs: the decoder scale, not capacity, is the first-order choice.

Architecture is marginal for reconstruction, not for the verdicts. The 0.8 dB that the residual encoder/decoder adds (C → F: 25.8 → 26.6 dB) is small, but it moves the downstream measurements the paper is about. Merged-vs-EM clustering goes from a tie under C (0.085 ± 0.008 vs. 0.083 ± 0.010, p=0.77) to EM ahead under F (0.095 ± 0.004 vs. 0.105 ± 0.011, merged ahead on $1/10$ seeds, p=0.037), and the latent SVHN AUROC falls from 0.116 to 0.086 (merged) and from 0.102 to 0.076 (EM). A result obtained with the convolutional network therefore cannot be assumed to hold for the residual one, and conclusions are stated per configuration.

The merged-vs-EM gap on CIFAR-10 is a starved-latent effect. With a fixed σ=1, merged clustering beats EM decisively (0.093 vs. 0.040, $10/10$ seeds, p=0.002). With the calibrated σ the gap disappears, and under the residual architecture EM is ahead (0.105 vs. 0.095, p=0.037). The mechanism is visible in the columns: merged NMI is nearly flat across every configuration (0.085–

Table 26: Generative sampling from alternative priors on frozen decoders (training-free). Each cell is 1NN two-sample accuracy / mean distance from a sampled z to the nearest training latent (the prior-hole score); lower is better for both. The tempered prior was not run on these models (—). CelebA-64, with the tempered prior at t=0.6, is in Table 19.

trained model	n	$\mathcal{N}(0, I)$	tempered	merged	EM	trained-in prior
conv CIFAR-10, co-trained (β=0.5)	10	0.593 $\pm$.014 / 7.37 $\pm$.04	—	0.551 $\pm$.011 / 6.33 $\pm$.17	0.568 $\pm$.015 / 3.61 $\pm$.26	—
conv CIFAR-10, co-trained (β=3)	10	0.572 $\pm$.024 / 7.44 $\pm$.02	—	0.553 $\pm$.020 / 5.19 $\pm$.10	0.613 $\pm$.035 / 1.85 $\pm$.13	0.554 $\pm$.018 / 6.01 $\pm$.03
conv MNIST, co-trained (β=0.5)	10	0.808 $\pm$.007 / 2.84 $\pm$.03	—	0.725 $\pm$.008 / 1.54 $\pm$.05	0.732 $\pm$.008 / 1.39 $\pm$.08	—
conv MNIST, co-trained (β=3)	10	0.775 $\pm$.010 / 2.80 $\pm$.03	—	0.747 $\pm$.017 / 2.03 $\pm$.05	0.767 $\pm$.020 / 1.18 $\pm$.11	0.757 $\pm$.010 / 2.42 $\pm$.03
conv STL-10, co-trained (β=0.5)	10	0.689 $\pm$.014 / 7.52 $\pm$.04	—	0.549 $\pm$.019 / 4.32 $\pm$.34	0.552 $\pm$.021 / 4.14 $\pm$.35	—
conv STL-10, co-trained (β=3)	10	0.625 $\pm$.016 / 7.55 $\pm$.03	—	0.537 $\pm$.006 / 5.47 $\pm$.20	0.541 $\pm$.006 / 4.54 $\pm$.26	0.542 $\pm$.009 / 6.35 $\pm$.07
MLP MNIST, co-trained (β=0.5)	10	0.741 $\pm$.006 / 2.87 $\pm$.02	—	0.725 $\pm$.013 / 1.83 $\pm$.05	0.736 $\pm$.013 / 1.34 $\pm$.08	—
MLP MNIST, co-trained (β=3)	10	0.740 $\pm$.011 / 3.09 $\pm$.04	—	0.744 $\pm$.012 / 2.20 $\pm$.07	0.770 $\pm$.013 / 0.85 $\pm$.05	0.735 $\pm$.008 / 2.61 $\pm$.03
conv MNIST, plain (β=0.5)	10	0.761 $\pm$.009 / 2.63 $\pm$.01	—	0.641 $\pm$.009 / 2.30 $\pm$.02	0.615 $\pm$.008 / 2.08 $\pm$.02	—
conv MNIST, plain (β=3)	10	0.664 $\pm$.007 / 2.64 $\pm$.02	—	0.622 $\pm$.005 / 2.56 $\pm$.03	0.612 $\pm$.006 / 1.51 $\pm$.04	—
MLP MNIST, plain (β=0.5)	10	0.735 $\pm$.008 / 2.65 $\pm$.02	—	0.637 $\pm$.008 / 2.48 $\pm$.03	0.616 $\pm$.006 / 2.04 $\pm$.10	—
MLP MNIST, plain (β=3)	10	0.598 $\pm$.007 / 3.02 $\pm$.05	—	0.580 $\pm$.005 / 3.00 $\pm$.05	0.582 $\pm$.006 / 0.79 $\pm$.07	—

Table 27: SQ-VAE-style model with a Gaussian codebook, 5 seeds per condition. The codebook is either learned from a random initialization, initialized from the merged mixture and then learned, or frozen at the merged mixture. Perplexity is the effective number of codes used out of $K = 50$; C2ST is the 1NN two-sample accuracy of samples against real images (0.5 is indistinguishable).

dataset	codebook	code recon	perplexity	sample C2ST	NMI	OOD-1
MNIST	scratch	215.4 $\pm$3.9	21.1 $\pm$4.6	0.876 $\pm$.010	0.372 $\pm$.051	0.391 $\pm$.185
MNIST	merged-init	207.7 $\pm$2.3	30.7 $\pm$4.3	0.859 $\pm$.006	0.438 $\pm$.031	0.574 $\pm$.107
MNIST	merged-frozen	201.5 $\pm$1.9	40.8 $\pm$2.5	0.828 $\pm$.013	0.475 $\pm$.063	0.382 $\pm$.087
CIFAR-10	scratch	57.4 $\pm$1.2	31.4 $\pm$7.6	0.605 $\pm$.015	0.061 $\pm$.003	0.561 $\pm$.058
CIFAR-10	merged-init	57.5 $\pm$.7	28.6 $\pm$3.0	0.607 $\pm$.008	0.064 $\pm$.007	0.500 $\pm$.030
CIFAR-10	merged-frozen	56.9 $\pm$.7	37.0 $\pm$3.0	0.596 $\pm$.012	0.079 $\pm$.003	0.506 $\pm$.015

0.115) while EM's rises from 0.040 to 0.083–0.112. EM is not weak on this dataset; at σ=1 it is fit to a latent that carries almost no information.

What the decoder likelihood optimises. For a Gaussian decoder with mean $\mu(z)$ and scale σ, the per-image reconstruction term over D pixels is

$$-\log p(x \mid z) \;=\; \frac{\|x-\mu\|^2}{2\sigma^2} \;+\; D\log\sigma \;+\; \tfrac{D}{2}\log 2\pi. \tag{5}$$

With no scale set the term collapses to $\frac{1}{2}\|x-\mu\|^2$; with σ fixed it is rescaled by $1/\sigma^2$, the β knob (Eq. 4); with σ calibrated the minimiser is available in closed form, $\sigma^{*2} = \mathrm{MSE}(x, \mu)$, and substituting it gives $-\log p(x \mid z) = D\,(\frac{1}{2} + \log\sigma^* + \frac{1}{2}\log 2\pi)$, an objective driven by $\log \mathrm{MSE}$ that is scale-free. We use the analytic form. The reported reconstruction term is a log-*density* on $[0, 1]^D$, comparable across models on the same data as long as σ stays well above the 8-bit quantisation step of $1/255$; our calibrated values, $\sigma^* \approx 0.04$–0.07, sit an order of magnitude above the step, and we report PSNR and per-pixel MSE alongside. Evaluation reuses the *training* σ, since recomputing it per batch would let the scale adapt to an OOD set. The standard alternative for images, a discretized mixture of logistics (Salimans et al., 2017) as used by NVAE and VDVAE, predicts a scale per pixel, is multimodal, and is a genuinely discrete likelihood; converting our single-sample ELBO to the 8-bit grid, bits/dim $= (-\mathrm{ELBO}/D + \ln 255)/\ln 2$, puts this model near 6.1 bits/dim against 2.87 for VDVAE, 2.91 for NVAE and 4.70 for a multivariate Gaussian on the same data (van den Oord et al., as tabulated in Table 2 of Kingma et al., 2016). We attribute most of that gap to the isotropic single-σ likelihood, but architecture, training and the bound estimator also differ across these numbers, so this is a hypothesis; reconstruction quality and density quality are different axes, and we keep the calibrated Gaussian for the results in this paper.

The density refutation strengthens as the model improves. Latent OOD against SVHN falls monotonically with reconstruction quality, from 0.311 at 16.7 dB to 0.086 at 26.6 dB: better models are *more* inverted, not less. The pixel-space score follows, reaching 0.436 for the residual model: the strong pixel-space inversion Nalisnick et al. (2019) report appears in our own setup once the decoder is adequate. No scaling, capacity or architecture change rescues latent density.

Table 28: Wall-clock on one H100 at full $N = 60000$, $d = 16$ (MNIST latents). Held-out is the mean test log-density of the fitted mixture, so speed is not compared across different-quality fits.

procedure	seconds	held-out $\log p$
CPU EM (sklearn), K=50	2.81	-4.03
GPU EM (torch), K=50	0.20	-4.01
merge pass to K=50	18.59	-19.31
GPU EM over the whole K-grid	1.52	—
merge pass, whole K-path	18.69	—

Table 29: OOD AUROC vs. mixture size K (MNIST in-dist, 15k subsample, $K \leq 200$). In this range AUROC rises monotonically with K for both methods and merged-GMM trails EM-GMM at every K — the refutation is not an artifact of one K; the continuation along the full nested path to $K = N$ is analyzed in Figure 2. Note K=10 (the class count) is among the *worst* settings for density-OOD, the opposite of its role for clustering (Table 11).

K	5	10	20	50	100	200
Merged (FashionMNIST)	0.612	0.654	0.707	0.754	0.784	0.823
EM (FashionMNIST)	0.835	0.839	0.848	0.870	0.908	0.941
Merged (KMNIST)	0.647	0.690	0.749	0.796	0.826	0.861
EM (KMNIST)	0.801	0.819	0.845	0.880	0.910	0.932

E GENERATED SAMPLES AND THE QUALITATIVE CHECK

Looking is part of the protocol. A decoder with a fixed σ=1 on $[0, 1]$ pixels reconstructs each image as its mean colour, and no metric flags it: likelihoods, clustering scores and OOD AUROCs are all computable. A grid of samples flags it at once, so every study renders, from the checkpoint of seed 0 of every cell it trains, one grid with five rows: real test images; their reconstructions $\mathrm{decode}(\mu(x))$, the decoder's ceiling; samples with $z \sim \mathcal{N}(0, I)$; samples from the merged aggregated posterior at the cell's K; and samples with $z \sim \mathcal{N}(0, t^2 I)$ at the temperature the literature shows the dataset at (t=0.6 for CelebA-64 and 0.7 for CIFAR-10 in NVAE, Vahdat & Kautz, 2020). Each row is compared with what it can be compared with, and the numbers are recorded next to the grid: the reconstruction with its own originals; every sample row with the reconstruction row, since a sample cannot be sharper than what the decoder renders; the merged row with the $\mathcal{N}(0, I)$ row, the same decoder under two priors; and the diversity of every sample row (mean pairwise pixel distance) with that of the real row, which is what a collapsed model fails. None of these is a threshold: the grid is the test and the number its caption. The figures below show what each prior produces when its samples are decoded by a *frozen* decoder: nothing is retrained between panels, only the distribution the latent code is drawn from changes.

F THE MNIST PIXEL SCORE UNDER A NORMALISED LIKELIHOOD

Binary cross-entropy on continuous pixels omits the continuous-Bernoulli normaliser $\sum_{\text{pixels}} \log C(\lambda)$ (Loaiza-Ganem & Cunningham, 2019), which depends on the decoder's output and therefore differs per image. Whether that moves the OOD rankings is an empirical question: the same Bernoulli-trained decoders are scored under both readings from identical posterior draws — the IWAE bound and the reconstruction probability under the pseudo-likelihood the paper reports, and under the normalised density — and the per-image rank correlation between the two, the AUROCs and the TNR at 95% TPR against FashionMNIST and KMNIST are recorded for every seed (Table 31).

The omitted term is not small: 1,500–2,100 nats per test image, of the order of the score itself, and it is not a constant — the per-image rank agreement between the two readings is 0.92–0.95. The pseudo-likelihood *flatters* the pixel score: read as a normalised density, the same decoders lose up to 0.03 AUROC and 0.10 TNR at 95% TPR against KMNIST (conv β=3: $0.999 \to 0.970$ and $0.997 \to 0.901$), less on the MLP cells and against FashionMNIST. The verdict of Table 1 does not

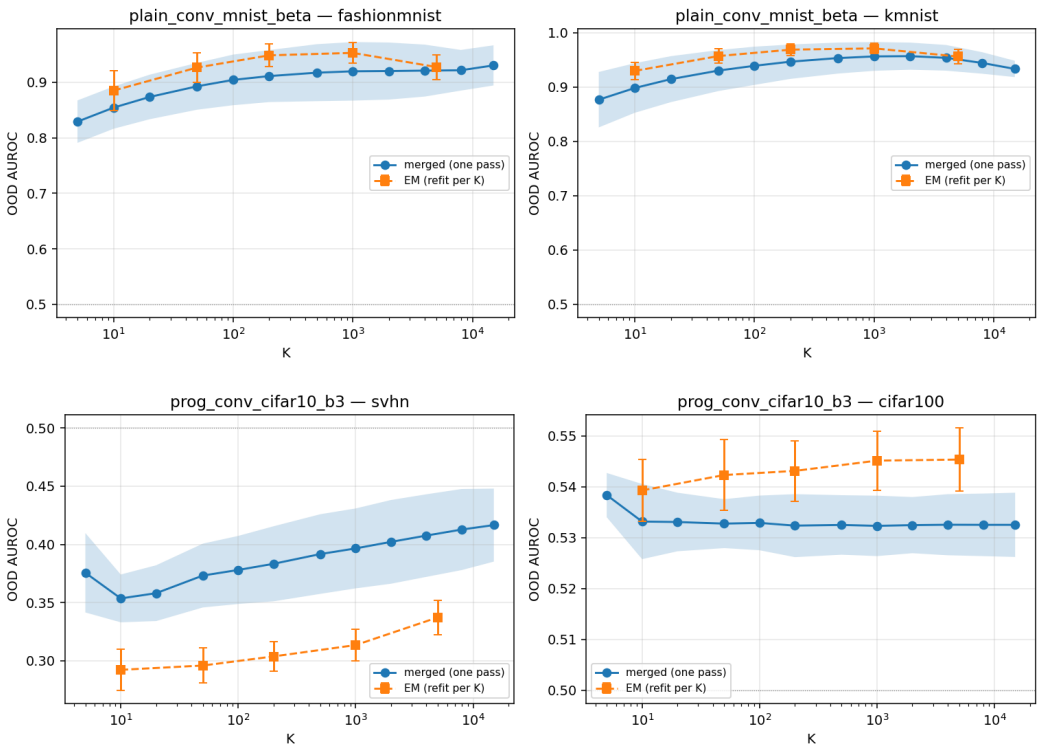


Figure 2: OOD AUROC along the nested merge path (bands: mean±std) vs. EM refits at selected K (points). Top: conv-MNIST, 20 ELBO-trained encoders ($\beta \in \{0.5, 3\}$) — both methods want large K; EM peaks at K=1000 and declines on razor components; the merged path saturates or peaks at interior K. Bottom: CIFAR-10 (10 encoders) — no K crosses 0.5 against SVHN for either method (the pathology spans the entire compression axis), with the merged path significantly less inverted throughout.

Table 30: Decoder scaling, latent capacity and architecture on CIFAR-10 (β=1, 60 epochs, 10 seeds, mean±std). "opt-σ" is the analytic optimal decoder scale $\sigma^{*2} = \mathrm{MSE}(x, \mu)$; row A is the fixed-$\sigma$=1 setting. p is a paired Wilcoxon test of merged vs. EM clustering NMI across seeds.

	decoder / latent	params	PSNR (dB)	KL (nats)	merged NMI	EM NMI	p	SVHN merged	SVHN pixel
A	σ=1, z=64	0.7M	16.7	13.1	0.093	0.040	0.002	0.311	0.579
B	opt-σ, z=64	0.7M	21.2	160	0.115	0.112	0.43	0.280	0.586
C	opt-σ, z=256	1.9M	25.8	509	0.085	0.083	0.77	0.116	0.521
D	opt-σ, z=256, wide	4.5M	25.8	527	0.090	0.091	0.77	0.110	0.505
E	σ=0.1, z=256	1.9M	24.5	303	0.093	0.067	0.002	0.131	0.536
F	residual, opt-σ, z=256	10.8M	**26.6**	568	0.095	**0.105**	0.037	**0.086**	0.436
G	residual, σ=0.1, z=256	10.8M	25.2	328	0.098	0.086	0.020	0.102	0.508

turn on it — the normalised pixel score, 0.97–0.99, still beats every latent score on MNIST — but "near-perfect" is a statement about the pseudo-likelihood. The main text's MNIST pixel columns are the BCE values the models were trained under, labelled as such; a model *trained* under the continuous Bernoulli is the proper object (Loaiza-Ganem & Cunningham, 2019) and is not run here, since it would change every MNIST encoder and with it every MNIST latent number.



(a) $z \sim \mathcal{N}(0, I)$ (b) $z \sim$ merged mixture

Figure 3: Samples from the *same* plain-trained conv-MNIST VAE decoder (β=0.5, seed 0) under two priors. The standard prior (a) places mass in aggregated-posterior holes and produces malformed hybrid glyphs; the merged aggregated posterior (b) — built post-hoc, training-free, deterministic — yields predominantly clean digits. Two-sample 1NN accuracy $0.761 \rightarrow 0.641$ (10/10 seeds, p=0.002).



(a) $\mathcal{N}(0, I)$ (b) merged mixture (c) trained-in prior

Figure 4: MNIST samples from one frozen MLP decoder under three priors. The standard prior places mass in regions the aggregated posterior never occupies and decodes them into malformed hybrid glyphs; the merged mixture and the trained-in prior stay on the data's support.

G Reproducing VaDE

The clustering half of our results compares the co-trained partition against EM and post-hoc merging on the same encoder, not against learned mixture priors (Section 2). This appendix documents the attempt to add VaDE (Jiang et al., 2017) as a reproduced baseline, because the outcome is itself a finding about the baseline: neither our term-by-term port nor the reference PyTorch implementation reaches the published 94.46% MNIST clustering accuracy, and the training budget is not the reason.

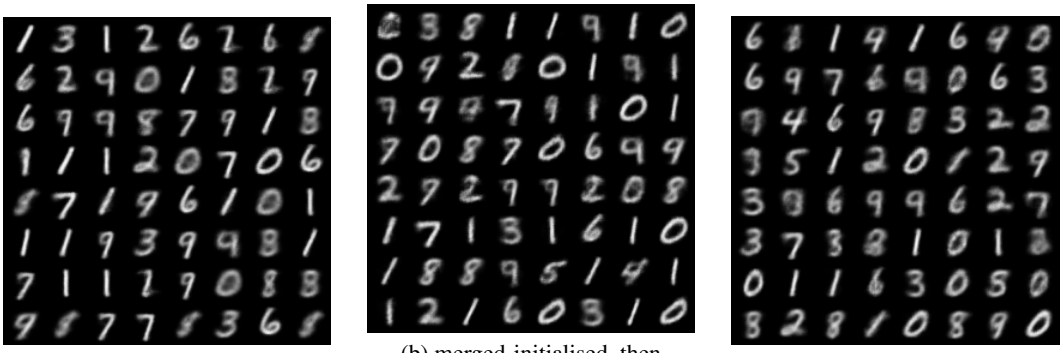
Our rule for reproductions. A ported baseline enters a comparison only after it has *reproduced the number its authors published, on their dataset, under their decisions* — the gate. Deviations are introduced only afterwards and are measured against the gated version. A third-party number is an acceptable gate only if it reproduces with that implementation's own code. Every run records its target and whether the gate passed.



(a) plain, β=1 (b) plain, σ_{dec}=0.3

(c) co-trained partition (d) resolution sweep, 64 px

Figure 5: CelebA-64 grids (seed 0 of each cell; calibrated σ except (b), fixed σ_{dec}=0.3). Rows, top to bottom: real test images; their reconstructions; samples from $\mathcal{N}(0, I)$; samples from the merged mixture (the aggregated posterior merged to the cell's K; for the co-trained cell the partition it trained with); samples from $\mathcal{N}(0, 0.6^2 I)$. The co-trained model (c) reconstructs as well as the plain one (a), but decoded from $\mathcal{N}(0, I)$ it gives near-identical average faces (pixel diversity of the row 7.3 against 20.5 for (a)), while sampling from its own partition gives varied faces (26.9): co-training moved the aggregated posterior away from the standard prior. These are ten images of one seed, a qualitative record, not a measurement. CelebA has no class labels, so these grids and the FID numbers of Table 19 are the whole qualitative record of the dataset.



(a) learned from scratch (b) merged-initialised, then learned (c) merged, frozen

Figure 6: Samples from the SQ-VAE-style model under the three codebook conditions (conv-MNIST, seed 200), drawn as $c \sim w$, $z \sim \mathcal{N}(m_c, S_c)$. The learned codebook (a) collapses onto few codes and its samples are dominated by ones and sevens, with several ambiguous glyphs; initialising from the merged mixture (b) helps but retains the bias; the frozen merged codebook (c) yields the widest spread of digit identities and the cleanest strokes. This is the visible counterpart of the utilisation numbers in Table 27 (21, 31 and 41 effective codes of 50).

The port. Our port mirrors the official Keras code term by term: the 784–500–500–2000–10 encoder with a mirrored decoder and Bernoulli likelihood; a K=10 diagonal Gaussian mixture prior whose weights, means and log-variances are learned inside the ELBO (their Eq. 12: reconstruction, the expected cross-entropy $\mathbb{E}_{q(c|x)}\mathbb{E}_{q(z|x)}[-\log p(z|c)]$, $\mathrm{KL}(q(c|x) \,\|\, \pi)$, and the entropy of $q(z|x)$); the responsibility $q(c|x) \propto \pi_c^d \mathcal{N}(z; \mu_c, \sigma_c^2)$ with the latent-dimension exponent on π_c that the

Table 31: The MNIST pixel score under two readings of the SAME Bernoulli-trained decoder, from identical posterior draws (k=16): the BCE pseudo-likelihood the paper reports (BCE) and the normalised continuous-Bernoulli density (CB). $\sum \log C$: the omitted normaliser, mean nats per test image; ρ: Spearman rank agreement of the two IWAE scores over the test split. AUROC and TNR at 95% TPR against FashionMNIST and KMNIST.

model	n	$\sum \log C$	ρ	FMNIST BCE	CB	TNR BCE	CB	KMNIST BCE	CB	TNR BCE	CB
conv, β=0.5	10	2077 $\pm$5	0.947 $\pm$.003	0.999 $\pm$.000	0.987 $\pm$.002	0.997 $\pm$.000	0.949 $\pm$.005	0.999 $\pm$.000	0.977 $\pm$.002	0.998 $\pm$.000	0.918 $\pm$.007
conv, β=3	10	1956 $\pm$14	0.932 $\pm$.003	0.999 $\pm$.000	0.990 $\pm$.001	0.994 $\pm$.000	0.956 $\pm$.004	0.999 $\pm$.000	0.970 $\pm$.002	0.997 $\pm$.000	0.901 $\pm$.006
MLP, β=0.5	10	1652 $\pm$4	0.926 $\pm$.003	1.000 $\pm$.000	0.999 $\pm$.000	0.999 $\pm$.000	0.995 $\pm$.001	1.000 $\pm$.000	0.994 $\pm$.001	0.999 $\pm$.000	0.975 $\pm$.004
MLP, β=3	10	1531 $\pm$3	0.917 $\pm$.002	0.999 $\pm$.000	0.998 $\pm$.000	0.994 $\pm$.001	0.987 $\pm$.004	0.999 $\pm$.000	0.992 $\pm$.001	0.994 $\pm$.001	0.957 $\pm$.006

original code produces (its ELBO uses $\log \pi_c$ once, so the original is internally inconsistent; we mirror it, since the published number comes from that code); 3000 epochs at batch 100; two Adam optimisers at 2×10^{-3}, decayed by 0.9 every 10 epochs with a 2×10^{-4} floor; training and scoring on the concatenated train and test sets (70,000 images), as both references do; initialisation from a layer-wise stacked autoencoder followed by a diagonal GMM fit on the embeddings. A second mirror reproduces the widely used PyTorch re-implementation instead: raw unnormalised π, exponent 1, 50 epochs of MSE pretraining, one Adam at 2×10^{-3} with a step decay of 0.95 every 10 epochs, batch 800, 300 epochs. Every setting of both mirrors is read from the saved sources, archived with checksums in the released code.

Results. Table 32 lists every run. The accuracy after pretraining and GMM initialisation — before a single VaDE epoch — already spans 0.61–0.88 depending on the seed, and the final accuracy tracks it: training adds $+0.02$ on average and *lowers* accuracy in 3 of 14 reference-mirror runs. No run of any implementation reaches the gate (0.9246, the published value minus two points).

implementation	schedule	runs	init acc.	final acc.	best
ours, original mirror, their shipped SAE weights	3000 ep, floor	1	0.727	0.811	0.811 @ 2950
ours, original mirror, our SAE	3000 ep, floor	1	0.735	0.802	0.804 @ 2910
ours, reference-port mirror	300 ep	14	0.72–0.88	0.68–0.87 (mean 0.785)	—
reference port, its own code as saved	300 ep	3	—	0.816 / 0.825 / 0.893	—
reference port, its own code as saved	3000 ep	3	—	0.820 / 0.823 / 0.907	—
published (Jiang et al., 2017)	3000 ep	—	—	0.9446	

Table 32: MNIST clustering accuracy (Hungarian matching, all 70,000 images) of every VaDE run. "init acc." is the accuracy of the GMM fitted to the pretrained embeddings before VaDE training. The reference port's own runs are its saved code executed unchanged apart from two import fixes; its prediction samples z, so its initial accuracy is not comparable and is omitted. No run passes the gate of 0.9246.

What was ruled out. *(i) The pretraining:* our stacked autoencoder initialises to the same accuracy as the authors' shipped weights (0.735 vs. 0.727) and ends within one point of them (0.802 vs. 0.811). *(ii) Our training step:* a line-by-line comparison against the saved reference sources finds every term, the optimiser, the schedule, the batch size and the data identical; the only differences (five GMM restarts instead of one; prediction at the posterior mean instead of a sampled z) favour our implementation. *(iii) The reference port's budget:* its own code run for 3000 instead of 300 epochs ends exactly where it stood at epoch 300, because its step decay has no floor and the learning rate is $\approx 4 \times 10^{-10}$ by the end; the original's schedule has the 2×10^{-4} floor, which is why 3000 epochs are meaningful there. *(iv) The data:* both references train and score on train plus test; so do we. *(v) The pretraining recipe that reproduces DEC* (Appendix H) applied to VaDE's pretraining leaves its GMM initialisation where it was (0.63 on two seeds against 0.61–0.73) and its final accuracy below the reproduced mean (0.67, 0.72).

What remains, and how we compare. The outcome is set by the initialisation: which clusters the GMM finds in the pretrained embedding decides the final accuracy within a few points, and VaDE training does not repair a poor draw — a spread of 0.68–0.91 across 22 runs of two implementations (2 of our original mirror, 14 of our reference-port mirror, 6 of the reference port's own code). That sensitivity is our observation, not the authors': Jiang et al. (2017) state that a few epochs of pretraining give a good initialisation and that VaDE is *not* sensitive to hyper-parameters after it. We

Table 33: Reproducing DEC on MNIST (Hungarian accuracy, all 70,000 images unless noted; the reference's own convention is the 60k training images, its "test" column the 10k test images). "init" is the accuracy of k-means on the pretrained embedding before the KL phase; the gate is the published 84.30% minus two points, on the training-convention column.

run	n	init	final	test	gate
reference (pt-dec/pt-sdae) as saved	5	0.826 $\pm.008$	0.859 $\pm.026$	0.862 $\pm.028$	4/5
our port, the paper's recipe	2	0.773 $\pm.012$	0.808 $\pm.001$	0.811 $\pm.002$	0/2
reference SAE $\rightarrow$ our KL phase	2	0.817 $\pm.026$	0.867 $\pm.013$	0.870 $\pm.017$	2/2
our SAE $\rightarrow$ reference KL phase	2	0.773 $\pm.012$	0.812 $\pm.006$	0.815 $\pm.005$	0/2
our port + finetune dropout (declared deviation)	2	0.820 $\pm.017$	0.864 $\pm.027$	0.868 $\pm.027$	2/2

did not obtain a run above 0.91 from any implementation. Every number of our own is a mean over seeds of the run's selected checkpoint, never a best seed, and we hold VaDE to the same standard: its comparator is the *reproduced* mean over seeds of the best implementation we have — the reference port's own code, whose 300- and 3000-epoch runs agree — which is 0.85; the original paper's 94.46% is a point estimate without a stated seed protocol that did not reproduce from either implementation. On the same metric our co-trained conv-MNIST partition at $K_{\text{target}}{=}1000$ reaches 0.771 ± 0.061 over ten seeds, 8 points below reproduced VaDE; three differences keep this from being a controlled comparison (VaDE: all $70{,}000$ images, an MLP with a 10-dimensional latent, a mixture fitted inside the ELBO for 3000 epochs; ours: the $15{,}000$-image subsample, a convolutional encoder with a 32-dimensional latent, a partition merged from a 30-epoch run), so the head-to-head under one protocol remains future work. The port, both mirrors, the runner for the reference code, every result file and the archived sources are released.

H Reproducing DEC

DEC (Xie et al., 2016) is the non-generative clustering reference of our comparison. We followed the order fixed for every port (mirror, gate, then deviate measurably): first the authors' recipe run from a public reference implementation as saved (`vlukiyanov/pt-dec` and `pt-sdae`, archived in the released code), then our own term-by-term port of the paper's text: the 784–500–500–2000–10 stacked denoising autoencoder, $50{,}000$ iterations per layer with 20% dropout, $100{,}000$ finetune iterations, SGD 0.1 with momentum 0.9 divided by ten every $20{,}000$ iterations, k-means with 20 restarts, the soft assignment of their Eq. 1, the target distribution of Eq. 3 over all points at every interval, the 0.1% stopping rule. The gate is the published 84.30% minus two points. Table 33 is the record.

The reference implementation reproduces the paper (0.859 ± 0.026 over five seeds, four of five above the gate); the port of the paper's text does not (0.808, 0.807). A swap experiment locates the gap entirely in the pretraining: the reference's stacked autoencoder followed by our KL phase passes (0.875, 0.858), our autoencoder followed by the reference's KL phase does not (0.808, 0.817), and both KL phases add the same $+0.04$ to whatever initialisation they receive. Table 34 ablates the reference's pretraining one knob at a time on our code.

One knob carries the whole gap: keeping the 20% input corruption through the end-to-end finetune (A4) lifts our autoencoder to the reference's level on its own (0.808, 0.832 against 0.831, 0.820), and the full reference recipe adds nothing beyond it. The paper states the opposite — "The entire deep autoencoder is further finetuned for 100000 iterations without dropout" (their Sec. 5.1) — so the published 84.30% is obtained, by the reference code and by our port alike, only with a finetune the paper says it does not do. With that one deviation declared, our port passes the gate (0.845, 0.883; last row of Table 33), and it is the version we compare against. At the paper's learning rate of 0.1 the unrolled stack is fragile — six of sixteen ablation arms lost a finite loss within the first 4–13 finetune iterations or in the bottleneck layer, and two more ended as dead networks — and the finetune-time corruption is also what keeps the reference stable. One deviation of both implementations from the paper's text remains: the paper initialises weights from $\mathcal{N}(0, 0.01)$, the reference uses Xavier and our port PyTorch's default; A3 shows Xavier alone is not the knob.

Table 34: DEC pretraining ablation: Hungarian accuracy after stacked pretraining and k-means (no KL phase), all 70,000 MNIST images, seeds 1 and 2 – one knob of the reference implementation per arm, applied to our port. Our port as written: 0.765 / 0.782; the reference's own SAE: 0.831 / 0.820. "diverged": a non-finite loss stopped the pretraining at the named stage; "dead": the finetune ended with a reconstruction MSE of 1.7 and a collapsed embedding.

arm	seed 1	seed 2
A0: our SAE as written (the paper's wording)	0.765	0.782
A1: dropout on the code, none at the bottleneck	diverged (finetune)	diverged (finetune)
A2: linear reconstruction of hidden levels	diverged (finetune)	diverged (layer 4)
A3: Xavier init, zero bias	0.150 (dead)	diverged (layer 4)
A4: 20% input dropout kept during the finetune	0.808	0.832
A5: the reference's length (∼70k/117k, /10 every 23k)	diverged (finetune)	0.119 (dead)
A6: train split only (60k)	diverged (finetune)	0.822
ALL: A1–A5 together (the reference's recipe)	0.823	0.828

Table 35: DEC under our protocol on MNIST: the gated port's k-means initialisation and KL self-training run on OUR encoders (the same networks, data, seeds and test-split evaluation as the cells; pretraining = the cell's own 30-epoch ELBO training at β=1), next to what the cells report on the same conditions. NMI and Hungarian accuracy at K=10 on the 10k test images, mean ± std over seeds. The plain rows are the 20-epoch cells of Table 5; the co-trained rows are the 30-epoch cells of the budget study.

condition	method	n	NMI	acc.
conv / MNIST	our VAE + k-means (DEC init)	10	0.700 ±.021	0.796 ±.032
conv / MNIST	DEC on our VAE encoder	10	0.861 ±.024	0.895 ±.036
conv / MNIST	plain, merged-GMM (β=1)	10	0.594 ±.030	0.605 ±.047
conv / MNIST	plain, EM-GMM (β=1)	10	0.641 ±.022	0.702 ±.032
conv / MNIST	co-trained partition (β=3, K_{target}=50)	10	0.696 ±.032	0.703 ±.039
conv / MNIST	co-trained partition (K_{target}=1000)	10	0.740 ±.026	0.771 ±.061
MLP / MNIST	our VAE + k-means (DEC init)	10	0.632 ±.028	0.728 ±.032
MLP / MNIST	DEC on our VAE encoder	10	0.777 ±.030	0.807 ±.034
MLP / MNIST	plain, merged-GMM (β=1)	10	0.564 ±.030	0.603 ±.047
MLP / MNIST	plain, EM-GMM (β=1)	10	0.501 ±.119	0.574 ±.129
MLP / MNIST	co-trained partition (β=3, K_{target}=50)	10	0.588 ±.026	0.610 ±.046

DEC under our protocol. With the port gated, the third step — change only what the protocol changes — runs DEC's mechanism on our encoders: the pretraining is the cell's own 30-epoch ELBO training (the same network, likelihood, latent size, rate and seed as the cell it is compared with; the gated port's declared deviation concerns its stacked autoencoder and has no counterpart here), the embedding is the posterior mean $\mu(x)$, and the port's own code then runs unchanged: k-means with 20 restarts on the training means, the KL self-training with SGD 0.01, a fresh solver per interval and the 0.1% stopping rule (the KL-phase rate is DEC's decision, mirrored, not swept: the phase has no validation loss to select on). Table 35 gives the result on the two MNIST conditions, ten seeds each, evaluated on the test split exactly as our cells are.

DEC is the strongest clusterer on our encoders by a wide margin: 0.861 ± 0.024 NMI on conv-MNIST against 0.740 for the co-trained partition at its largest budget (p=10^{-5}, Mann–Whitney U, 10 vs. 20 seeds) and 0.696 at K_{target}=50 (p=0.0002); 0.777 against 0.588 on MLP-MNIST (p=0.0002). The self-training adds +0.16 / +0.14 NMI over its own k-means initialisation on every seed, and pays for it with the generative model: the KL phase moves the encoder until the frozen decoder's test MSE rises from 0.005 to 0.08 per pixel (conv; 23 dB to 11 dB), so the DEC encoder is a clustering map and no longer a VAE — the trade our partitions do not make, and the reason the comparison is a reference and not a competitor. The initialisation row is a control: plain k-means on the posterior means of a β=1 VAE reaches 0.700 NMI on conv-MNIST, above the same cells' merged-GMM (0.594, p=0.0002) and EM (0.641, p=0.0004) at the same K=10, and level with the

co-trained partition at K_{target}=50 (0.696, p=0.47); on MLP-MNIST 0.632 against 0.564 / 0.501 / 0.588. The plain rows come from 20-epoch cells and the k-means row from 30-epoch ones, so the gap is not a paired measurement, but its size is well beyond that difference: on MNIST the hard Euclidean partition of the means is a better K=10 clustering than either Gaussian mixture of the full posteriors, and the NMI advantage of the merge over EM (Table 11) is an advantage among mixture fits, not over clustering as such. The MLP-MNIST number (0.807 accuracy) is also the direct read against the gated port's 0.848 / 0.887 on the test split: our VAE encoder starts DEC lower than the reference's denoising autoencoder does (0.73 vs. 0.81–0.83 after k-means) and the KL phase adds a similar amount to both.

I REPRODUCIBILITY

What is released. We release the complete source for this paper: the training entry points, the evaluation and study drivers, the aggregation scripts, and the script that generates every table of this appendix directly from the per-cell result files, so that the numbers in the paper cannot drift from the data they came from. Alongside the code we release the trained checkpoints, the generated sample grids and figures, and the per-cell result files behind every number reported here (small JSON records, version-controlled with the source); checkpoints and cached latent tensors are distributed as a separate artifact bundle.

Seeding and determinism. A *cell* is one independent training run identified by its seed. One seed per run seeds every random number generator the run touches — the model initialisation, the 15 000-image subsample, the batch order, and the sampling draws used at evaluation — and evaluation uses the same seed as training, so seed-to-seed spread includes the evaluation's own variance. Studies run seeds 0–9. Two consequences matter for the statistics: the same seed selects the same subsample, so a matched control trains on exactly the images its comparator trained on; and within one seed the co-trained partition, the post-hoc merge and the EM fit are all evaluated on the *same* trained encoder, which is what makes the paired Wilcoxon tests throughout the paper valid. Every result record carries the git revision, the device, the library versions, the split fingerprints (content hashes of the decoded images for CelebA; dataset, split and size for the torchvision datasets), the learning rate and its provenance, and whether deterministic kernels were enabled. No claim in this paper rests on a single run.

Results are append-only records. Every cell writes one result file, named by machine, study, timestamp, architecture, dataset, hyper-parameter and seed; completed cells are skipped on restart, so every study is resumable and its seed count can be grown later without recomputing finished cells, and no result is ever overwritten. A run that diverges leaves a marker in place of its result, so a missing cell is visible rather than silently absent.

Reproduction checked, not assumed. Determinism is tested, not only configured. A complete re-run of the CelebA campaign, with a shorter schedule, returned validation-best checkpoints whose best epoch, PSNR and per-dimension KL match the earlier run's to the last bit (the best epochs fall before the schedules diverge); the tempered-prior sampling study re-run on them returned every recorded value exactly, and the OOD evaluation re-run at a later code revision returned an identical output file up to the recorded revision. The pipeline is covered by an automated test suite — unit tests of every measured quantity, each of which has one shared implementation; consistency tests across the studies; a stubbed end-to-end chain; and checks of each campaign driver's printed plan — that runs on every commit and again on the GPU servers before a campaign starts. Runtime dependencies are declared in `requirements.txt` and `pyproject.toml` (a test keeps the two identical and fails on any undeclared import), and the repository's README documents installation, data download and how to run each campaign, including a dry-run mode that prints a driver's exact commands.

Practices that spare others our reproduction problems. Three practices came out of the baseline reproductions. Reference implementations are archived verbatim with checksums, and a port must first reproduce the published number under the authors' own decisions (the *gate*) before anything is changed (Appendices G and H); where the paper's text and its released code disagree — VaDE's responsibility exponent, DEC's finetune-time dropout — the record states both and which

one produced the published number. Every port records its target and its gate verdict in the result file. Decoder likelihoods are calibrated and stated per model, and the likelihood a pixel score is computed under is part of the score's name, because the same network trained and scored under two likelihoods gives opposite verdicts (Table 16). And every campaign renders sample grids from its checkpoints (Appendix E), because a model whose decoder has collapsed still produces every number.

Environment and compute. Every result record carries its library versions (Python 3.12 with PyTorch 2.11 on the H100s, Python 3.11 with PyTorch 2.6 on the consumer GPU); the code runs on CUDA, on Apple-silicon MPS and on CPU. All timing claims were measured on an H100, and the multi-seed grids were trained on one or two H100s; the clustering baselines ran on a single consumer GPU. The grids of Table 4 amount to 2,350 independent training runs. A convolutional co-trained cell takes roughly 15–25 minutes on one H100, so a co-trained grid of 40 cells fits in an overnight run on two GPUs, while the full plain-VAE sweep of 270 cells takes several hours.

J USE OF GENERATIVE AI

This section details the disclosure at the end of the main text.

What is the authors' own. The object and the method — compressing a VAE's aggregated posterior by moment-preserving hierarchical merging, co-training the partition with the encoder, and using the merged mixture as a prior and as a codebook — together with the pre-registered hypothesis and the experimental protocol are the authors' own and were not generated by an AI tool.

Code. LLM-based coding assistants wrote a large part of the experiment code, the campaign drivers and the tests, to the authors' specifications, and suggested experimental parameters and controls. Every change runs through the test suite of Appendix I on each commit and again on the GPU servers before a campaign starts; tests of new code are mutation-checked, i.e. a deliberately introduced bug must make a test fail before the test is accepted; campaign drivers print their plan before anything runs; and intermediate results were printed and inspected. The code is released so that anyone can check it.

Experiment design and interpretation. LLMs gave feedback on the methodology, proposed follow-up hypotheses and controls — for example the checkpoint-matched control of the co-trained CelebA comparison (Appendix D.4) — and drafted summaries and interpretations of the results. The rules the experiments follow, every approval of a learning rate or training length, and every decision about what to report were made by the authors; every summary and interpretation was checked against the raw result records, and design questions were also discussed with colleagues.

Literature and citations. LLMs were used to search for and summarize literature and to identify gaps in it. Every citation was then checked against the archived full text of the cited work: each attributed claim, quote and number was compared with its source, and the misattributions this found were corrected.

Writing. LLMs drafted parts of the paper, edited it for readability, suggested its structure and title, and formatted the references. Every appendix table is generated by code from the result records, and numbers quoted in the text were checked against the same records, including by automated LLM-based consistency passes whose every finding was re-verified against the data before anything was changed. The authors have read the final text and take responsibility for it.

K ADDITIONAL DETAILS

K.1 MERGE CONVENTIONS, MEMORY BUDGET, AND PROTOCOL DETAILS

Given the latents, the sequence of merges is fixed, with no initialisation and no seed — under three conventions that are part of that statement. *(i) Ties are broken by component index:* a row's nearest neighbour is the lowest-indexed minimiser, and a merged component is stored at the index of one

member of its pair (the other is retired), so indices never move and a tie resolves identically in every run. *(ii) The path is a total order, and a target K is a prefix of it.* Within a step the selected pairs are vertex-disjoint, so the order in which they execute does not change the mixture at the end of the step; it matters only when a step is cut because it would overshoot the target. We order every step's pairs by their merge distance (index breaks exact ties), so the sequence of merges from N to 1 is a total order — by step, then by distance — and the mixture at any K is the first $N - K$ merges of it; a cut step executes its closest pairs first, and separate runs to different targets are nested. *(iii) Exact ties are decided in the last bit.* Real latents are continuous, so exact ties practically never occur; where one does, a one-unit-in-the-last-place difference in the component weights can flip the tie-break, after which the merge sequences diverge, so determinism is a statement about a fixed numerical environment. Merging is driven by a target K rather than a distance threshold (thresholds calibrated on low-dimensional synthetic data do not transfer to a d-dimensional latent), and the divergence to the original mixture is monitored sparsely along the path.

The matrix D is $N \times N$: about 14 GB at N=60k in single precision and 106 GB at CelebA's 162,770 training latents. Above a configurable matrix budget (20 GB by default) each component keeps only its m=64 nearest candidates, built in one blocked pass that never holds the matrix, together with a *certified radius*: the distance below which the list is known to be complete. The mutual-nearest selection is carried out on these lists, but a listed candidate is believed only if it lies strictly inside the radius; otherwise the row is recomputed exactly against all free components. The radius is what makes the lists safe: a component that merges moves, and may re-enter a list far beyond the distance up to which that list was ever complete. Because the active count only shrinks, the survivors fit the budget within a few steps. The switch happens at the power-of-two component count whose matrix approximately fills the budget (65,536 at 20 GB); the matrix is then materialised among the survivors in the same index order, so that tie-breaks are unchanged. Both phases select the same pairs from the same distances, so the merge sequence is that of a pure full-matrix run, verified by exact comparison at sizes where both fit.

Model interface, in full. Any model exposing, per item, one continuous Gaussian posterior $\mathcal{N}(\mu, \operatorname{diag}\sigma^2)$ over a flat latent is compatible; the condition is on the encoder, not on the model family. It holds as stated for MLP/convolutional VAEs, β-VAE (Higgins et al., 2017), the calibrated-decoder σ-VAE (Rybkin et al., 2021), a VAE trained with the IWAE objective (Burda et al., 2016) (its encoder is unchanged), and the single-level VampPrior VAE (Tomczak & Welling, 2018); for a two-stage VAE (Dai & Wipf, 2019) it applies to the first stage's posterior over x (the second stage models that latent, not the data). VampPrior's hierarchical variant, flow posteriors, hierarchical-spatial latents, and discrete (VQ) latents are out of scope.

The prior comparison, in full. The four priors are sampled on the same frozen decoder and compared by (i) a pixel-space two-sample test against held-out real images (leave-one-out 1NN accuracy; 0.5 is indistinguishable); (ii) a direct prior-hole score, the distance of a sampled z to the nearest training latent, not reported for the tempered prior (a shrunk prior moves its samples relative to the data in a way the nearest-latent distance does not summarise); (iii) sample diversity in pixel space; (iv) the k-nearest-neighbour precision and recall of Kynkäänniemi et al. (2019) (k=3) in the 2048-dimensional pool features of the clean-fid (Parmar et al., 2022) Inception-v3 (the extractor and preprocessing of our FID protocol; their reference implementation uses VGG-16, with Inception-v3 as their reported alternative), on 2,000 real and 2,000 generated images per prior and seed, matched across arms — finite-sample manifold estimates that separate rendering quality (precision) from coverage (recall), which we use because a pairwise pixel distance shrinks with blur and cannot tell the two apart; and (v) each prior's placement in latent space, the radius $\|z\|$ of its samples and their log-density under the whole aggregated posterior, next to the same two numbers for the training latents. The tempered temperatures are those published models are shown at (t=0.6 for CelebA-64 and 0.7 for CIFAR-10 in NVAE; 0.7 elsewhere); NVAE tempers the conditional Gaussians of a hierarchy and re-calibrates its batch-norm statistics, so the flat analogue is a heuristic transfer.

The matched control. Plain cells train on the full training split and the co-trained cells on the 15k subsample (on CelebA both on the full split); because that difference favours the plain pipeline, the matched control retrains the plain pipeline on exactly the same 15k images (same seed, same indices) and scores it with the same evaluator, and matched numbers are reported wherever the two pipelines are compared (Appendix D.8).

K.2 Training budget and step size

Our cells are trained shorter than most of the work we compare against, and with a smaller nominal step per example (the exceptions are two CelebA ladder rungs, below). We measure budgets in *samples seen* (epochs $\times$ N_{train}), because optimiser steps are not comparable across batch sizes (a larger batch needs fewer of them). Our plain CelebA cells see 4.1M samples (25 epochs on the full $162{,}770$-image split); the resolution cells see 5.7–7.3M, the widest ladder rung 3.3M, and the two ladder rungs with the plain convolutional network 15.5–16.3M. Published budgets are at most 11.4M for Ghosh et al. (2020) (their stated maximum of 70 epochs on the official CelebA split, which we also use), 14.6M for NVAE at the same resolution (90 epochs at their CelebA-64 setting, their Table 6; Vahdat & Kautz, 2020), and 1.9M for the calibrated-decoder models of Rybkin et al. (2021) ($40{,}000$ steps at batch 48, their Appendix B); our closest architectural match on CIFAR-10, the flat convolutional VAE of Xiao et al. (2020), sees 10M (200 epochs at batch 64, their Appendix A.2). The co-trained CelebA cells are matched to the plain $\beta{=}1$ cells they are compared with (the full split, 25 epochs, 4.1M samples). Separately, our cells at learning rate 3×10^{-4} (every CelebA cell except ladder rungs A, B, E and the 16-px cells), a rate chosen at batch 256, take a nominal step per example of 1.2×10^{-6} against Ghosh et al.'s 10×10^{-6}; scaled to our batch, their setting would be $\approx 2.6 \times 10^{-3}$. This is a heuristic comparison (lr/batch), since Adam's effective step does not scale linearly with the batch. The two gaps (fewer samples, smaller steps) compound, except on ladder rungs A and B, which see more samples than RAE and NVAE; rung B, at 3×10^{-3}, also takes a larger step per example than RAE. This bounds the *positive* claims most — a reconstruction-quality or FID number from a model trained this much less is not comparable to a published one — and bounds the negative result least, since the capacity ablation of Appendix D.10 shows the refutation *strengthens* as reconstruction improves.

K.3 Limitations, in full

The latent basis is rotation-equivalent, so we merge diagonal-joint only, and the method requires a continuous Gaussian posterior. *Training budget.* Our cells are trained shorter than the work we compare against and with a smaller step per example: our plain CelebA cells see 4.1M samples against at most 11.4M for Ghosh et al. (2020) and 14.6M for NVAE (90 epochs, their Table 6; Vahdat & Kautz, 2020), with two ladder rungs as the exceptions (Appendix K.2). This bounds the *positive* claims more than the negative result, which our capacity ablation shows *strengthening* as reconstruction improves. *Two comparison gaps.* (i) Our OOD baselines are uncorrected; of the corrected family (Ren et al., 2019; Serrà et al., 2020; Xiao et al., 2020; Choi et al., 2018; Havtorn et al., 2021) we report only the input-complexity analysis of one CelebA cell (Section 5), so our claim is that the compressed aggregated posterior is not a usable density *on its own*. (ii) The clustering comparison is against EM and post-hoc merging on the same encoder, not against VaDE, GMVAE or DEC, so the co-training result is an improvement over its own ablations, not over the state of the art. Outside that comparison, VaDE, reproduced under our seed-averaging standard, reaches 0.85 MNIST clustering accuracy against our 0.77 under a different protocol (Appendix G). DEC, gated only with a finetune-time dropout its paper says it does not use (Appendix H), clusters MNIST under our protocol far better than any partition of ours (0.861 vs. 0.740 NMI), at the price of the generative model (decoder test MSE up $16\times$); its k-means initialisation alone already matches our co-trained partition and beats the merged and EM mixtures at $K{=}10$, so the merge's NMI advantage is over EM, not over clustering. A named, unresolved confound: merging uses the posterior scales σ_i, EM only the means.s