Reproducible, incremental representation learning with Rosetta VAE

Miles Martinez
Electrical & Computer Engineering
Center for Cognitive Neuroscience
Duke University
miles.martinez@duke.edu

John Pearson
Biostatistics & Bioinformatics
Center for Cognitive Neuroscience
Electrical & Computer Engineering
Neurobiology
Psychology & Neuroscience
Duke University
john.pearson@duke.edu

1 Introduction

Variational autoencoders are among the most popular methods for distilling low-dimensional structure from high-dimensional data, making them increasingly valuable as tools for data exploration and scientific discovery. However, unlike typical machine learning problems in which a single model is trained once on a single large dataset, scientific workflows privilege learned features that are reproducible, portable across labs, and capable of incrementally adding new data. Ideally, methods used by different research groups should produce comparable results, even without sharing fully-trained models or entire data sets. Here, we address this challenge by introducing the Rosetta VAE (R-VAE), a method of distilling previously learned representations and retraining new models to reproduce and build on prior results. The R-VAE uses post hoc clustering over the latent space of a fully-trained model to identify a small number of Rosetta Points (input, latent pairs) to serve as anchors for training future models. An adjustable hyperparameter, ρ, balances fidelity to the previously learned latent space against accommodation of new data. We demonstrate that the R-VAE reconstructs data as well as the VAE and β-VAE, outperforms both methods in recovery of a target latent space in a sequential training setting, and dramatically increases consistency of the learned representation across training runs.

2 Related Work

Our approach is conceptually related to several strands of recent work. First, the notion of distilling a dataset by means of a small number of representative points is often studied under the heading of coresets [1, 7, 24, 3] or Bayesian coresets [11, 17, 5, 4]. Our method is simpler in that we use standard clustering to determine our data subset, an approach that leverages a large body of research on scalable clustering.

Second, this work intersects with recent results in identifiability for VAEs [12, 28, 31, 14]. Our approach, while providing fewer guarantees than these results, appears to work without these assumptions, since it replaces constraints on the encoding model class with a set of point constraints that approximately identify the latent space.

Third, the R-VAE shares with the VQ-VAE [29, 20] in both its hard and soft versions [27, 21, 9, 30, 6] the notion of a quantization of latent space. The major distinction between these approaches and ours is that the VQ-VAE and its variants employ quantization as a regularization strategy for avoiding posterior collapse, while here our focus is on issues of portability and reproducibility.

Finally, the portability problem as we describe it is closely related to those addressed by both continual learning and transfer learning. While in continual learning, the focus is often on learning new tasks,

Bayesian Deep Learning workshop, NeurIPS 2021.

several studies have used distillation, including coresets, as an intermediate step in this process (e.g., [25, 18, 23, 3]).

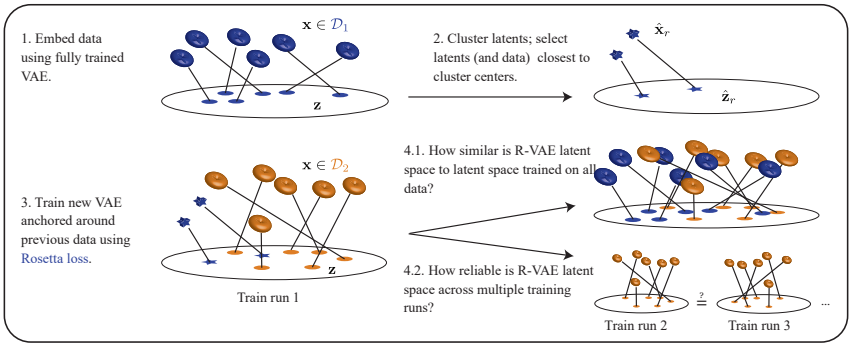

Figure 1: Schematic of Rosetta VAE training. **1)** Beginning with a trained VAE, data $\mathbf{x} \in \mathcal{D}_1$ are encoded to their latent representations $\mathbf{z}$. **2)** After clustering the latent representations, we identify Rosetta points $\hat{\mathbf{z}}_r$ closest to the cluster centroids and their preimages in the training set, $\hat{\mathbf{x}}_r$. **3)** For new data $\mathcal{D}_2$, the standard ELBO is augmented with a loss term (1) to enforce the constraint that the VAE preserves the Rosetta points $(\hat{\mathbf{x}}_r, \hat{\mathbf{z}}_r)$. **4.1)** For sequential training, we assess the similarity between the latent space inferred using the R-VAE and the latent space found by training jointly on all data. **4.2)** For reproducible training, we use Rosetta points from $\mathcal{D}_1$ to seed retraining on $\mathcal{D}_1$ and assess the reliability of the resulting embeddings across repeated training runs.

3 Experiments

Experiment structure and metrics

We focus on the two experimental settings, the sequential and reproducible training problems. In the former, we consider two scientists, Alice and Bob, who wish to embed their data in a shared latent space. In the portability setting, we ask how similar the latent space discovered by Bob — who has no access to Alice's data, or even Alice's trained model — can be made to the one found by joint training on both data sets combined. In the reproducibility case, we ask how similar latent embeddings of Alice's data can be made across repeated retrainings, as the variability of latent spaces learned over retraining has been previously demonstrated [16, 26, 12]. Our goal is therefore to enforce similarity between a previously learned latent space and a new latent space. We do so through the process outlined in Figure 1: First, we embed data using a fully trained VAE (trained on $\mathcal{D}_1$), then distill that latent space through standard clustering into a small set of **Rosetta Points** (latent, data pairs). We then train a new VAE anchored around our Rosetta points using our **Rosetta loss**:

$$\mathcal{L}_\rho = \mathcal{L}(\theta, \phi) - \rho \sum_{r=1}^{R} \left[\|\hat{\mathbf{x}}_r - \mathbf{m}(\hat{\mathbf{z}}_r)\|^2 + \|\hat{\mathbf{z}}_r - \boldsymbol{\mu}(\hat{\mathbf{x}}_r)\|^2 \right] \tag{1}$$

where the $\hat{\mathbf{z}}_r$ are chosen to be the closest points to the centroids found by clustering $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_1} q_{\phi_*}(\mathbf{z}|\mathbf{x})$ and $\mathcal{L}(\theta, \phi)$ is the standard ELBO. In the reproducibility case, $\mathcal{L}_\rho(\theta, \phi)$ is optimized over $\mathcal{D}_1$ again. In the sequential training case, $\mathcal{L}_\rho(\theta, \phi)$ is optimized over a separate dataset, $\mathcal{D}_2$, and Rosetta points from $\mathcal{D}_1$.

To assess the similarity of latent spaces across training, we introduce two new metrics. For the sequential training setting, as in [12], we calculate a normalized distortion that considers the two latent spaces the same if they differ only by a linear map. That is, for a given data set $\mathcal{D}$ and a pair of encoders q and q', we let $\boldsymbol{\mu}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\mathbf{z}$ (with a similar definition for $\boldsymbol{\mu}'$) and calculate the **latent space distortion** as

$$LSD = \min_{\mathbf{A},\mathbf{b}} \mathbb{E}_{\mathbf{x}\in\mathcal{D}} \|\mathbf{A} \cdot \boldsymbol{\mu}(\mathbf{x}) + \mathbf{b} - \boldsymbol{\mu}'(\mathbf{x})\|^2, \tag{2}$$

where $\mathbf{A}$ and $\mathbf{b}$ parameterize a linear map. For our sequential training experiments, q' is the embedding learned by joint training on the full data set $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$ and q is the encoder learned by training the R-VAE and comparison models on $\mathcal{D} = \mathcal{R}_1 \cup \mathcal{D}_2$. Note that this measure of distortion

uses only the mean of the embedding map, ignoring uncertainty in the mapping from $\mathbf{x}$ to $\mathbf{z}$. We choose not to use mean correlation coefficient as in [12], since we do not know the "true" latent space.

For reproducible training, we measure **retraining variability**, which we defined as the average volume of the covariance matrix across training runs for each data point. If we label encoders learned by sequential training runs $1 \dots M$ as $q_1 \dots q_M$ with conditional means $\boldsymbol{\mu}_1 \dots \boldsymbol{\mu}_M$, then

$$RV = \mathbb{E}_{\mathbf{x}\in\mathcal{D}} \log\det \mathbf{C}(\mathbf{x}), \qquad \mathbf{C}(\mathbf{x}) = \mathrm{cov}\left([\boldsymbol{\mu}_1(\mathbf{x}) \quad \boldsymbol{\mu}_2(\mathbf{x}) \quad \dots \quad \boldsymbol{\mu}_M(\mathbf{x})]\right). \tag{3}$$

This index then gives a (logged) volume measure of the ellipsoid containing the same data point's latent representation across runs, with lower numbers indicating more reliable embeddings. See appendix for datasets used and model training details.

3.1 Rosetta VAE consistently recovers the same latent space across retrainings

To test the ability of the Rosetta VAE training procedure to reproduce consistent latent representations of the same data across retrainings, we first trained a standard VAE to embed each of our example datasets. We then clustered the resulting (mean) latent representations using k-means with $k = 8$ for the Gaussians and $k = 64$ for MNIST and Birdsong, taking the closest data embedding to each cluster centroid ($\hat{\mathbf{z}}_r$) and its associated data point ($\hat{\mathbf{x}}_r$) as the Rosetta points ($\mathcal{R}_1$). We then used $\mathcal{R}_1$ and the original dataset to retrain a VAE, β-VAE, and R-VAE 10 times each and assessed the consistency of the embeddings across runs using (3). Note that, except for the R-VAE, only the $\hat{\mathbf{x}}_r$ from $\mathcal{R}_1$ were used. That is, we duplicated a small number of points from the original data set without upweighting them or fixing their embedding locations.

As visualized in Figure 2 and quantified in Table 1, the R-VAE produced much more consistent embeddings across training runs. In fact, this was true even when retraining the standard VAE using the same initial seed, due to GPU nondeterminism. Thus, retraining with the Rosetta VAE allowed us to reproduce the same latent space structure again and again by seeding only with a handful of latent data points and their embeddings.

Figure 2

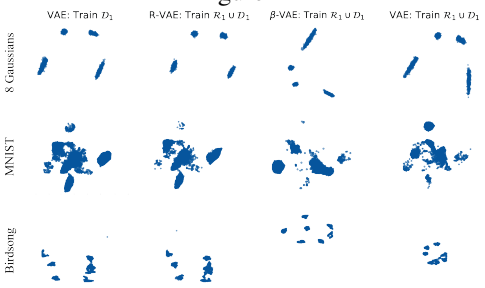

Table 1

		8 Gaussians	MNIST	Birdsong
RV($\mathcal{R}_1$)	VAE	0.00(0.193)	0.00 (5.861)	0.00(14.512)
	β-VAE	-1.151(0.970)	-4.934478(8.023)	-33.568 (27.799)
	VAE (same seed)	—	-48.828(9.352)	-9.123(11.649)
	R-VAE	**-17.247(1.496)**	**-95.393 (5.268)**	**-149.942(12.198)**
RV(rest of $\mathcal{D}_1$)	VAE	0.00(0.355)	0.00(14.254)	0.00(17.015)
	β-VAE	-1.204(1.003)	-5.671(10.069)	-40.019(22.808)
	VAE (same seed)	—	**-48.566(11.05)**	-7.542(12.147)
	R-VAE	**-16.425(1.840)**	**-46.241(12.593)**	**-82.440(23.285)**

Figure 2, Table 1: R-VAE reproduces the same latent space across training runs. (left panel) Learned latent representations of each dataset, projected to two dimensions by UMAP, in **(first column)** and as reproduced by retraining with Rosetta VAE, standard VAE, and β-VAE **(columns 2–4)**. $R = 8, 64, 64$ Rosetta points used for 8 Gaussians, MNIST, and birdsong respectively. In each case, R-VAE matches the target latent space, consistently across retrainings. (right panel) Medians and interquartile ranges of retraining variability metric for three example data sets, normalized by median vanilla VAE performance.

3.2 Rosetta VAE approximates full data latent spaces under sequential training

To assess the ability of the Rosetta VAE procedure to capture the full data latent space under sequential training, we partitioned each of our test data sets into halves as specified in Appendix A: $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$. We then performed VAE training as described above on $\mathcal{D}_1$, distilled the mean latent embeddings of these data via k-means clustering ($k = 8$ for the 8 Gaussians, $k = 32$ for Birdsong, $k = 64$ for MNIST, and $k = 128$ for 3D Chairs) to produce a set of Rosetta points $\mathcal{R}_1$. These points were then used to train an R-VAE model on $\mathcal{R}_1 \cup \mathcal{D}_2$ using the Rosetta Loss (1). We then assessed the similarity of the latent spaces found by this procedure to the latent space trained on $\mathcal{D}_1 \cup \mathcal{D}_2$ via the distortion

Table 2: Medians and interquartile ranges of distortion for four example data sets, normalized by median vanilla VAE performance.

	Latent Space Distortion					
	$\mathcal{D}_1$			$\mathcal{D}_2$		
	VAE	β-VAE	R-VAE	VAE	β-VAE	R-VAE
8 Gaussians	0.00(0.216)	**-0.056(0.066)**	**-0.049(0.057)**	0.00(0.304)	-0.060(0.336)	**-0.303(0.056)**
MNIST	0.00 (0.049)	0.989(0.656)	**-0.749(0.220)**	0.00(0.431)	0.913(0.754)	**-0.982(0.069)**
Birdsong	0.00 (0.541)	-0.379(0.326)	**-0.512(0.116)**	0.00 (0.376)	**-0.588(0.305)**	-0.153(0.529)
3D Chairs	**0.00(0.190)**	3.509(0.436)	**-0.039(0.265)**	**0.00(0.213)**	3.565(0.440)	**-0.033 (0.289)**

measure (2), which we report separately for both $\mathcal{D}_1$ and $\mathcal{D}_2$ in Table 2 (using the same linear map for both). See Appendices G and H for visualization of embeddings in each case.

We further examined the latent embeddings by asking how distorted latent maps were across the models and training runs, assessed by the similarity of the learned $\mathbf{A}$ matrix in (2) to the identity (Appendix I). Overall, the R-VAE required linear maps that were closer to the identity than the VAE and $\beta-$VAE, indicating less non-uniform stretching and compression of the learned space relative to the joint training template. The biases $\mathbf{b}$ are likewise small (Appendix J).

4 Discussion

The issues of reproducibility and portability, while central to the scientific enterprise, have been sparsely addressed in the neural networks literature. Our Rosetta VAE provides a simple, intuitive prescription for increasing both. By adjusting a single parameter, we can trade off fidelity to previously learned latent spaces against accommodation of distribution shifts driven by new data. Perhaps surprisingly, latent representations for *all* VAE models we tested showed striking reproducibility under sequential training, much more so than might be expected from representation learning results in, e.g., [16]. The important qualification to this result, of course, is that when we assessed similarity between latent representations in the sequential training case, we calculated distortion modulo an overall linear transformation, similar to [12]. This finding suggests that what might seem like large differences in learned representations in previous studies may simply be the result of linearly transformed latent variables. Of course, in the absence of a joint training template (Figure 4, first column), this is impossible to assess.

Our Rosetta loss, which simply fixes the embedding locations of a small number of data points, is both simple to implement and conceptually intuitive: by "tacking down" our Rosetta points, we effectively remove symmetries in the latent space that prevent identifiability. In this sense, our $\mathcal{R}_1$ points are reminiscent of the exogenous covariates $\mathbf{u}$ that make identifiability possible in [12]. Moreover, our results do not depend sensitively on either the number of Rosetta points nor the architecture of the model from which they are derived (Appendices D, E, F), suggesting a robust, practical method for VAE reproducibility. However, as noted above, a limitation of this work is that we do not solve the coreset problem except heuristically and so offer none of the convergence guarantees of the coreset or identifiability literature (e.g., [7, 24, 11, 12]).

It is also important to note that, as a data distillation method, our work potentially compounds problems of bias and underrepresentation in existing datasets. That is, in selecting Rosetta points near areas of high density in latent space, it is most likely to preserve the most typical points in initial training sets most faithfully. As such, care must be taken not to exacerbate bias in sensitive applications, and more studies will be needed to assess its potential for harm. Conversely, our results on sequential training suggest that previously trained models may be augmented by new data without appreciable distortion of the latent space, which may help to redress problems in some models that result from gaps in training data.

Finally, our results confirm the practical utility of models like the β-VAE [10] and regularization in general, in producing robust learning of latent spaces. While our R-VAE strongly outperformed both standard and β-VAEs in reproducing the same latent space across training runs, performance was more variable on sequential training. In cases like the 3D Chairs dataset, which we partitioned randomly, most models performed well, while the β-VAE outperformed R-VAE on $\mathcal{D}_2$ for the

birdsong data, suggesting that dataset structure and the distribution shifts associated with adding new data to existing models may play a role. Further work is needed to combine the symmetry-breaking benefits of methods like the R-VAE with the overall regularization offered by β-VAE and related models.

Acknowledgments and Disclosure of Funding

We thank the Mooney Lab for birdsong data and Jack Goffinet for helpful conversations on data preprocessing and visualization. This work was funded by BRAIN grant R01-NS118424.

References

[1] Pankaj K Agarwal, Sariel Har-Peled, Kasturi R Varadarajan, et al. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

[2] Mathieu Aubry, Daniel Maturana, Alexei Efros, Bryan Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014.

[3] Zalán Borsos, Mojmír Mutnỳ, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *arXiv preprint arXiv:2006.03875*, 2020.

[4] Trevor Campbell and Boyan Beronov. Sparse variational inference: Bayesian coresets from scratch. In *Advances in Neural Information Processing Systems*, pages 11461–11472, 2019.

[5] Trevor Campbell and Tamara Broderick. Automated scalable bayesian inference via hilbert coresets. *The Journal of Machine Learning Research*, 20(1):551–588, 2019.

[6] Amir Dib. Quantized variational inference. *Advances in Neural Information Processing Systems*, 33, 2020.

[7] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578, 2011.

[8] Jack Goffinet, Samuel Brudner, Richard Mooney, and John Pearson. Low-dimensional learned feature spaces quantify individual and group differences in vocal repertoires. *eLife*, 10, May 2021. doi: 10.7554/elife.67855. URL `https://doi.org/10.7554/elife.67855`.

[9] Gustav Eje Henter, Jaime Lorenzo-Trueba, Xin Wang, and Junichi Yamagishi. Deep encoder-decoder models for unsupervised learning of controllable speech synthesis. *arXiv preprint arXiv:1807.11470*, 2018.

[10] 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. *ICLR*, 2017.

[11] Jonathan Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable bayesian logistic regression. In *Advances in Neural Information Processing Systems*, pages 4080–4088, 2016.

[12] Ilyes Khemakhem, Diederik P. Kingma, Monti, Ricardo Pio, and Aapo Hyvärinen. Variational autoencoders and nonlinear ica: A unifying framework. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 2207–2217, 2020.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of 3rd International Conference for Learning Representaitons*, 2015.

[14] Abhishek Kumar and Ben Poole. On implicit regularization in β-vaes. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5480–5490, 2020.

[15] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[16] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[17] Simon Mak, V Roshan Joseph, et al. Support points. *The Annals of Statistics*, 46(6A):2562–2592, 2018.

[18] Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *Proceedings of the 7th International Conference on Learning Representations*, 2018.

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. URL `http://arxiv.org/abs/1912.01703`.

[20] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pages 14866–14876, 2019.

[21] Aurko Roy, Ashish Vaswani, Arvind Neelakantan, and Niki Parmar. Theory and experiments on vector quantized autoencoders. *arXiv preprint arXiv:1805.11063*, 2018.

[22] Tim Sainburg, Marvin Thielk, and Timothy Q Gentner. Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLoS computational biology*, 16(10): e1008228, 2020.

[23] Jonathan Schwarz, Jelena Luketina, Wojciech M. Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[24] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[25] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

[26] Harshvardhan Sikka, Weishun Zhong, Jun Yin, and Cengiz Pehlevan. A closer look at disentangling in β-vae, 2019.

[27] Casper Kaae Sønderby, Ben Poole, and Andriy Mnih. Continuous relaxation training of discrete latent variable image models. In *NIPS Bayesian Deep Learning Workshop*, 2017.

[28] Peter Sorrenson, Carsten Rother, and Ullrich Köthe. Disentanglement by nonlinear ica with general incompressible-flow networks (gin). *arXiv preprint arXiv:2001.04872*, 2020.

[29] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[30] Hanwei Wu and Markus Flierl. Vector quantization-based regularization for autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6380–6387, 2020.

[31] Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-vae. *arXiv preprint arXiv:2011.04798*, 2020.

A Datasets

For our experiments, we used four data sets of varying complexity: 1) a toy data set comprising 8 Gaussians in two dimensions, 2) MNIST (from LeCun et al. [15], licensed under Creative Commons Attribution-Share Alike 3.0 license), 3) 3D Chairs (from Aubry et al. [2]), and 4) spectrograms representing syllables of zebra finch birdsong (from Goffinet et al. [8], licensed under Creative Commons CC0 1.0 Universal). The last of these allows us to consider real data of a type that are known to exhibit strong clustering in latent space [22, 8]. Moreover, they are of scientific interest because studies of birdsong both within and across individuals require multiple animals, and so latent space modeling should produce structures that are both reproducible and portable across labs. In each case, we partitioned the total data into sets $\mathcal{D}_1$ and $\mathcal{D}_2$ as follows: 1) four Gaussians each, separated by a half-plane; 2) Digits 0–4 and 5–9; 3) a random division of the data set into halves; 4) distinct sets of birds, with each individual singing only minor variants of a single song.

B Model Training

After division into $\mathcal{D}_1$ and $\mathcal{D}_2$ data sets as described above, data were further divided within each partition into 60/40 train/validation split. Hyperparameters β and ρ were selected by training models for 20 epochs at β = [0:2.5:25], ρ = [0:0.75:15]. The β and ρ with the best validation performance after this initial training were selected and used for experiments. For R-VAE training, ρ was weighted by the ratio of the number of Rosetta Points to batch size, and 1 was applied to the Rosetta Points alongside every batch of new data. Models were optimized using Adam [13] with learning rate 1e-3. Stopping criterion was determined by letting joint data models train until loss plateaued and then using that same number of epochs for each of the second-phase comparison models. For 8 Gaussians and MNIST we used 200 epochs, for all others we used 300 epochs. Appendix C contains details of model architecture. All models were created and trained using PyTorch [19] (licensed under BSD), code for all experiments in paper can be found in the supplemental material. Experiments on average required 20 hours to run on an RTX3070 GPU.

C Model Architectures

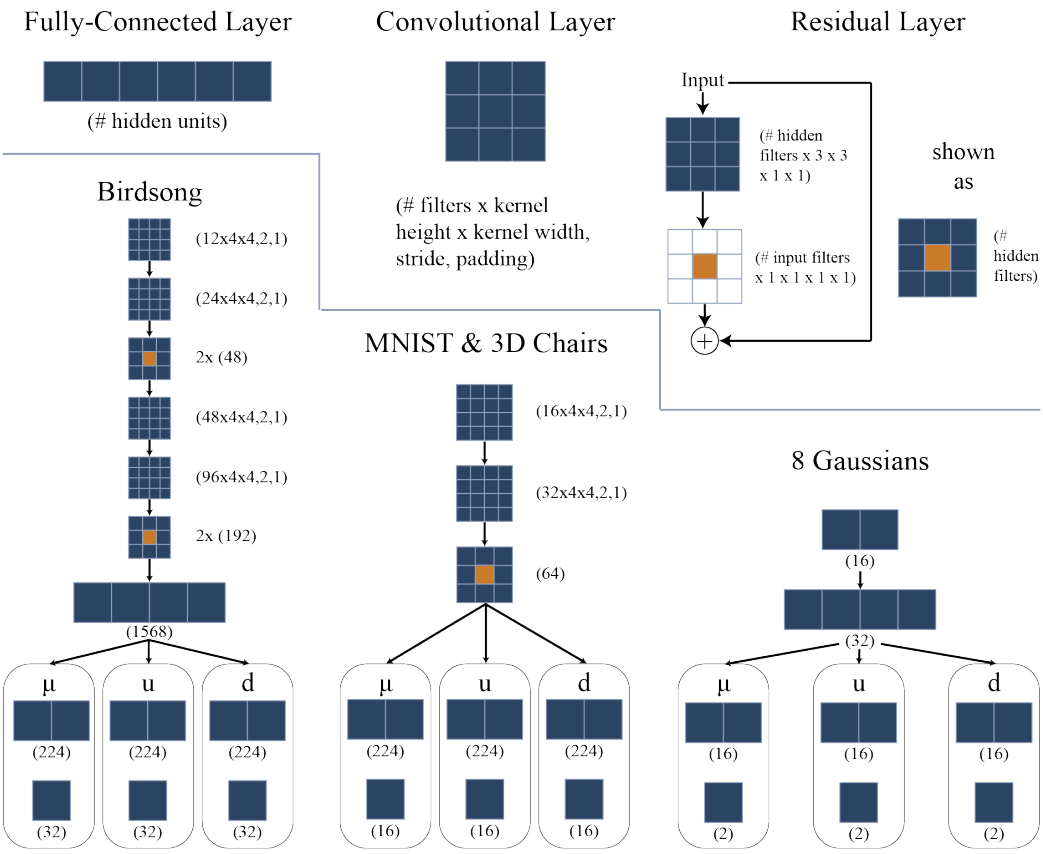

Figure 3: **Architectures used in main text. Top row**: types of layers used in the neural networks in the paper. Fully connected, convolutional, and residual layers were used. For fully connected layers, parameters used are presented underneath in parentheses; for convolutional and residual layers, parameters are shown beside the layer in parentheses. All residual layers had the same size hidden filters, and only differed in the number of hidden filters. **Bottom Row**: Encoder architectures used for each dataset in the paper, ordered by decreasing model complexity from left to right. MNIST & 3d Chairs used the same model architectures, and differed only in input size. The input for birdsong was 128x128 spectrograms, for 3D Chairs 64x64 grayscale images, for MNIST 28x28 grayscale images, and for the 8 Gaussians 5-dimensional vectors (two (x,y) position dimensions + 5 dimensional gaussian noise). Output network μ parameterizes the mean $\mathbb{E}\mathbf{z}$, and networks u and d parameterize the (flattened) lower triangle and diagonal of the Cholesky decomposition of $\mathrm{cov}[\mathbf{z}]$, respectively. Only encoders are presented for space, all decoders were reverse of encoders, with convolutional layers replaced with transposed convolutional layers.

D Rosetta-VAE dependence on number of Rosetta Points

Rosetta-VAE consistency did not depend strongly on the number of Rosetta Points. Experiments here followed the same form as those in the main text, with **reproducibility trained models** referring to the procedure in section 3.1 and **sequentially trained models** to the procedure in section 3.2. Tables 3,4,5 show an extremely limited effect of the number of RPs, with even 4 RPs greatly improving the consistency of learned embeddings over VAE and β-VAE. The sequentially trained case shows slightly more reliance on number of RPs, but still remains consistent across a range of numbers of RPs.

Table 3: Medians and interquartile ranges of RV for reproducibility trained 8 Gaussians, normalized by median vanilla VAE performance.

	RV			
	2 RPs	4 RPs	8 RPs	16 RPs
VAE	0.00 (0.933)	0.00(0.316)	0.00(0.355)	0.00(0.511)
β-VAE	**-1.225 (1.068)**	-1.021 (0.504)	-1.204 (1.00)	-0.934 (0.348)
R-VAE	**-1.412(4.191)**	**-14.656 (6.642)**	**-16.425 (1.840)**	**-17.116 (1.649)**

Table 4: Medians and interquartile ranges of RV for reproducibility trained MNIST, normalized by median vanilla VAE performance.

	RV	
	32 RPs	64 RPs
VAE	0.00 (9.557)	0.00(9.111)
β-VAE	-5.776 (10.243)	-5.671 (10.069)
R-VAE	**-32.083 (15.603)**	**-46.241 (12.593)**

Table 5: Medians and interquartile ranges of RV for reproducibility trained birdsong, normalized by median vanilla VAE performance.

	RV				
	4 RPs	8 RPs	16 RPs	32 RPs	64 RPs
VAE	0.00(14.254)	0.00(17.015)	0.00(15.805)	0.00(18.083)	0.00(16.672)
β-VAE	-46.503 (26.498)	-40.019 (22.808)	-38.892 (24.572)	-31.955 (27.672)	-41.852 (23.307)
R-VAE	**-75.852 (20.358)**	**-82.440 (23.285)**	**-74.166 (27.809)**	**-85.790 (32.503)**	**-95.529 (28.815)**

Table 6: Medians and interquartile ranges of LSD for sequentially trained 8 Gaussians, normalized by median vanilla VAE performance.

	LSD				
	2 RPs	4 RPs	8 RPs	16 RPs	$\mathcal{D}_1$
VAE	**0.00 (0.115)**	**0.00(0.024)**	0.00(0.216)	0.00(0.283)	0.00 (0.301)
β-VAE	0.143 (0.293)	**0.00 (0.052)**	**-0.056 (0.066)**	**-0.042 (0.044)**	**-0.046 (0.042)**
R-VAE	0.435 (0.355)	**0.015 (0.056)**	**-0.049 (0.057)**	**-0.029 (0.044)**	-0.007 (0.025)

Table 7: Medians and interquartile ranges of LSD for sequentially trained MNIST, normalized by median vanilla VAE performance.

	LSD		
	32 RPs	64 RPs	$\mathcal{D}_1$
VAE	0.00 (0.635)	0.00(0.487)	**0.00 (0.553)**
β-VAE	0.904 (0.289)	0.989 (0.656)	1.217 (0.386)
R-VAE	**-0.725 (0.113)**	**-0.749 (0.220)**	0.770 (0.110)

Table 8: Medians and interquartile ranges of LSD for sequentially trained birdsong, normalized by median vanilla VAE performance.

	LSD	
	32 RPs	64 RPs
VAE	0.00(0.541)	0.00(0.394)
β-VAE	-0.379 (0.326)	**-0.588 (0.305)**
R-VAE	**-0.512 (0.116)**	-0.153 (0.529)

E Rosetta-VAE is agnostic to Rosetta Point selection method

Here, we compare R-VAE performance when different clustering methods are used to select the Rosetta points. We compare K-means, as used in the main text, to agglomerative clustering, Gaussian mixture model clustering, and random selection of embeddings. Data reported are from sequentially trained R-VAEs.

Table 9: Medians and interquartile ranges of LSD & RV for different RP selection methods in the sequential training setting, normalized by median R-VAE performance using K-Means clustering.

		$\mathcal{D}_1$	$\mathcal{D}_2$
LSD	Agglomerative Clustering	**0.834 (2.209)**	**0.251 (2.115)**
	Gaussian Mixture Model	**0.933 (2.204)**	**0.255 (2.130)**
	K-Means	**0.00 (1.616)**	**0.00 (2.025)**
	Random Selection	**0.073 (1.698)**	12.063 (8.485)
RV	Agglomerative Clustering	43.024 (34.094)	17.636 (29.159)
	Gaussian Mixture Model	29.215 (24.532)	**-6.842 (20.999)**
	K-Means	**0.00 (18.415)**	**0.00 (31.612)**
	Random Selection	**9.356 (17.569)**	**4.367 (19.808)**

F Rosetta-VAE is model agnostic

R-VAEs were trained on birdsong using three different architectures. The first was used as a reference and had the same architecture as models used in the main text (and as the template model). Our "complex" model had an additional residual layer with 96 hidden units inserted between the third and fourth convolutional layers (see figure 3) while our "simple" model had the first two existing residual layers removed. All models demonstrated similar performance in both metrics. Data reported are from sequentially trained R-VAEs.

Table 10: Medians and interquartile ranges of LSD & RV for sequentially trained birdsong, normalized by performance of the R-VAE with the architecture of the template model.

	$\mathcal{D}_1$			$\mathcal{D}_2$		
	Simple	Complex	Same Arch	Simple	Complex	Same Arch
LSD	0.175 (0.234)	0.191 (0.273)	**0.00 (0.312)**	**-0.102 (0.147)**	0.119 (0.171)	**0.00 (0.293)**
RV	**1.485 (19.221)**	**3.912 (19.155)**	**0.00 (18.415)**	**-4.725 (27.337)**	**2.927 (30.421)**	**0.00 (31.612)**

G Sequentially trained latent spaces corrected for linear transformation

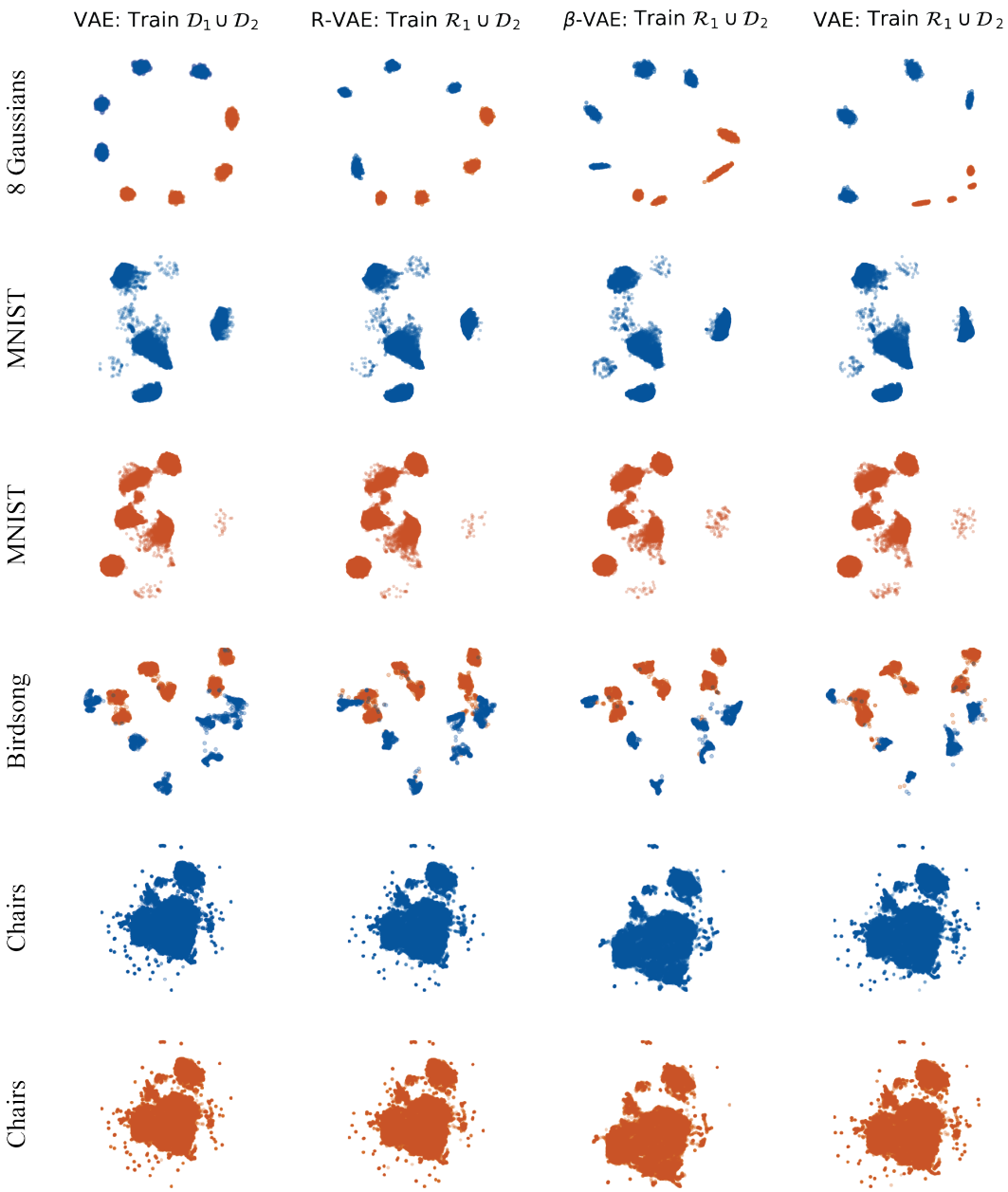

Figure 4: **Learned representations with linear correction. First column:**Learned latent representations of each dataset, through joint training, projected to two dimensions by UMAP. **Second column:** Representations learned through training on $\mathcal{D}_1$ (used to find RPs). **Columns 2–4:** Latent spaces as reproduced by retraining with Rosetta VAE, standard VAE, and β-VAE . $R = 8, 64, 32, 128$ Rosetta points used for 8 Gaussians, MNIST, birdsong, and 3D Chairs respectively. In each case, the recovered latent space looks similar to the original target, suggesting rough linear equivalence between all learned models.

H Sequentially trained latent spaces uncorrected for linear transformation

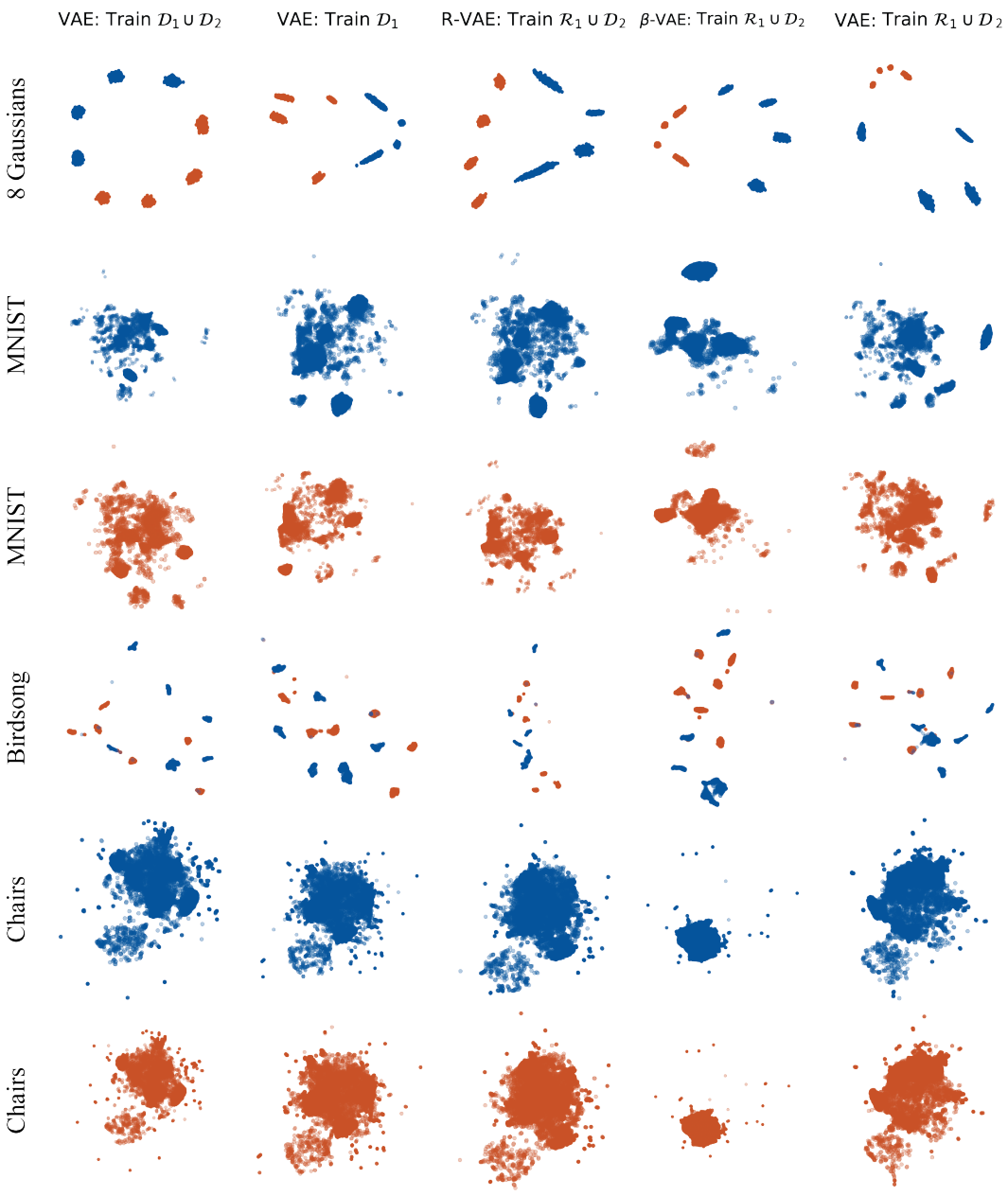

Figure 5: **Learned representations without linear correction. First column:**Learned latent representations of each dataset, through joint training, projected to two dimensions by UMAP. **Second column:** Representations learned through training on $\mathcal{D}_1$ (used to find RPs). **Columns 2–4:** Latent spaces as reproduced by retraining with Rosetta VAE, standard VAE, and β-VAE . $R = 8, 64, 32, 128$ Rosetta points used for 8 Gaussians, MNIST, birdsong, and 3D Chairs respectively. For 8 Gaussians, the linear warping of the recovered latent spaces is obvious, while for MNIST and 3D Chairs, the R-VAE produces a less warpred version of the joint latent space than other models. For birdsong, the β-VAE appears least warped, as suggested by Table 2 and Figure 6.

I Analysis of linear map in sequential case

To examine the latent embeddings further, we asked how distorted learned maps were across the various models and training runs by asking how close the learned $\mathbf{A}$ matrix in (2) is to the identity. Models that best recapitulate the latent space should have $\mathbf{A} \approx \mathbb{1}$. To investigate this, we performed a polar decomposition, $\mathbf{A} = \mathbf{UP}$, with $\mathbf{U}$ an orthogonal matrix and $\mathbf{P}$ symmetric positive semi-definite. In Figure 6, we plot the eigenvalue spectra for $\mathbf{P}$ for each model type. In all but the 8 Gaussians case, the spectra exhibit an abrupt drop, identifying an effective dimensionality for the data set. Moreover, in each case, the R-VAE exhibits the flattest spectrum, indicating less non-uniform stretching and compression of the learned space relative to the joint training template. We further investigate this by defining $\tilde{\mathbf{A}} = \mathbf{A}/\|\mathbf{A}\|_\infty$ to be the linear transformation with maximum rescaling of 1. That is, if the two latent spaces are the same up to a global rescaling, we should expect $\tilde{\mathbf{A}} \approx \mathbb{1}$. In the bottom row of Figure 6, we show the quality of this approximation across different models and training runs. Just as in Table 2, the R-VAE shows lower distortion than both the VAE and β-VAE with the exception of birdsong, where the results are comparable.

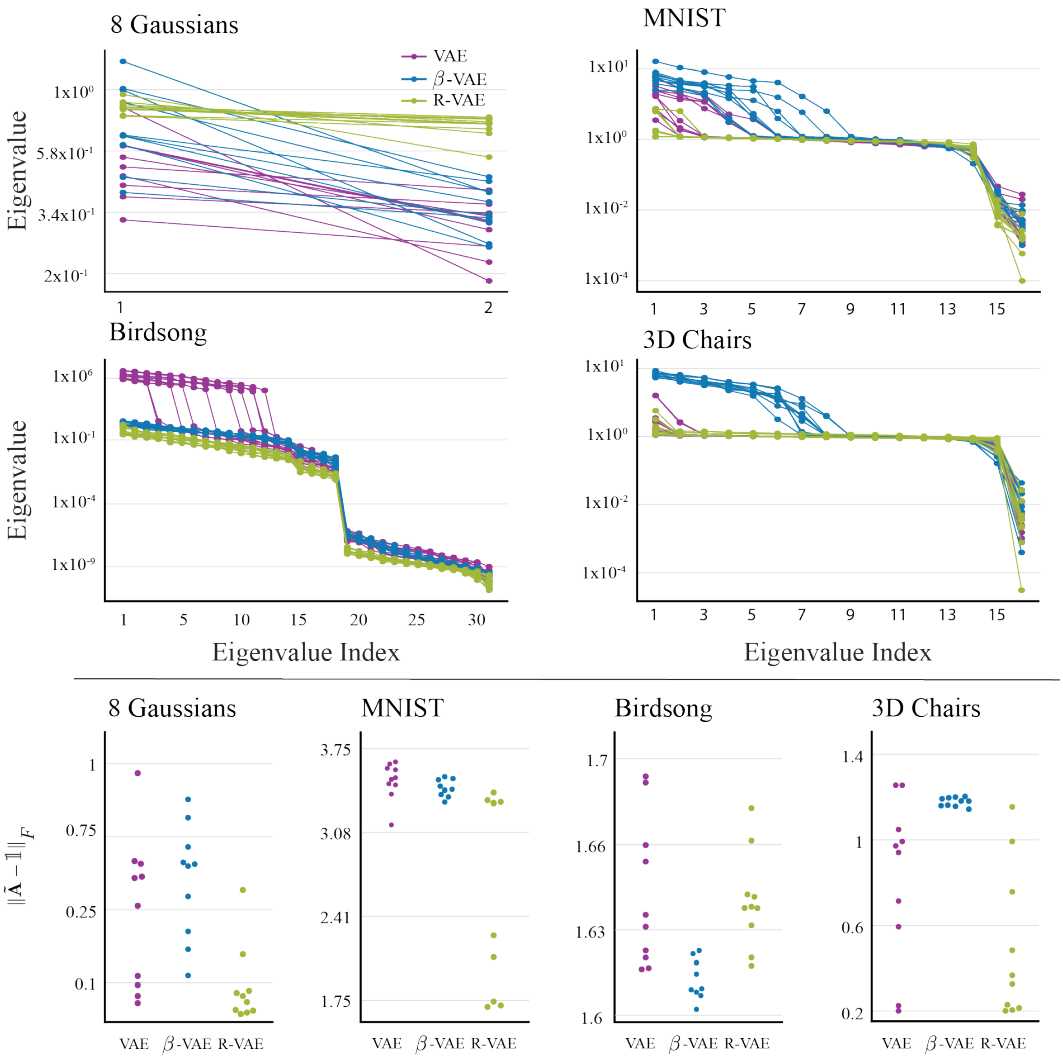

Figure 6: **Linear transformations between latent spaces are simplest for the R-VAE. First two rows:** Plots of the eigenvalues of the positive semidefinite matrix from polar decomposition of $\mathbf{A}$. In each case, the R-VAE exhibits the flattest spectrum, suggesting only a global rescaling without skew. **Bottom row:** Norm of the difference between the identity matrix and a version of $\mathbf{A}$ rescaled to have maximum singular value 1. The R-VAE has smaller values, indicating that less linear transformation is needed to align latent spaces found by sequential training.

J Learned biases of linear transformation are small

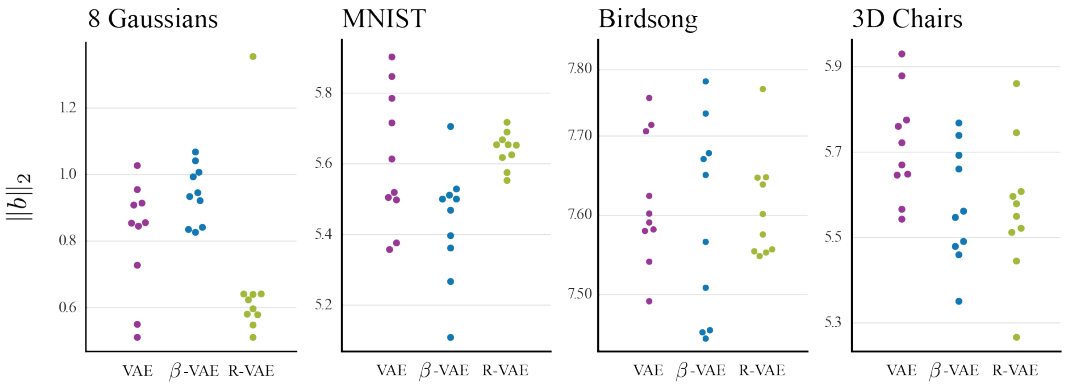

Figure 7: **Biases of linear transformations between latent spaces are small.** Norm of the learned bias vector for the linear transformation in (2). The bias learned by for R-VAE tends to be smaller than learned by VAE and β-VAE, indicating that less linear transformation is needed to align latent spaces found by sequential training, although all biases are small.