What We Don't C
Representations for scientific discovery beyond VAEs.

Brian Rogers[1]* **Micah Bowles**[1,2] **Chris J. Lintott** [1] **Steve Croft**[1,3,4]

[1] Oxford Astrophysics
University of Oxford

[2] Ellison Institute of Technology

[3] Breakthrough Listen
University of California, Berkeley

[4] SETI Institute

Abstract

Accessing information in learned representations is critical for scientific discovery in high-dimensional domains. We introduce a novel method based on *latent flow matching* with classifier-free guidance that disentangles latent subspaces by explicitly separating information included in conditioning from information that remains in the residual representation. Across three experiments—a synthetic 2D Gaussian toy problem, colored MNIST, and the Galaxy10 astronomy dataset—we show that our method enables access to meaningful features of high dimensional data. Our results highlight a simple yet powerful mechanism for analyzing, controlling, and repurposing latent representations, providing a pathway toward using generative models for scientific exploration of *what we don't capture, consider, or catalog*.

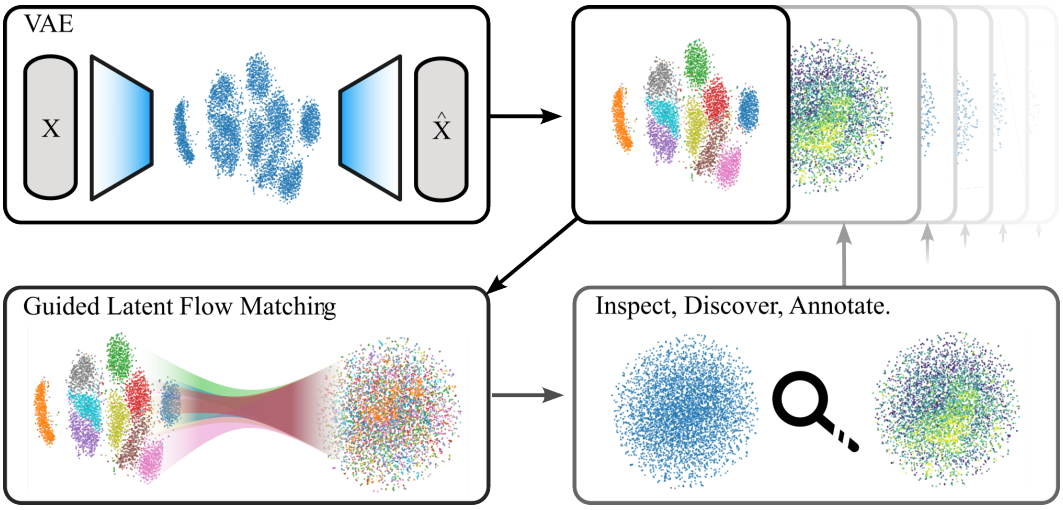


Figure 1: Using a VAE data manifold, aggregated labels can be used to remove recorded factors of variation from the latent space using a flow matching model. This enables access to features that are less apparent in the VAE manifold but are important in describing the underlying data. By disentangling learned manifolds from the information provided by labels, the resulting structure can be used to iteratively discover *What We Don't C*.

*brian.rogers@physics.ox.ac.uk

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

1 Introduction

Once described as a 'curse which has hung over the head of the physicist and astronomer for many a year'[1], high-dimensional spaces are difficult to deal with and understand. The search for embeddings that can provide access to the inherent structure of high-dimensional datasets has been motivated by many applications in the physical sciences [2–6]. Compression and disentanglement learning using VAEs [7, 8] and the β-VAE variant [9] are an especially interesting approach to dealing with the curse of dimensionality and increasing the interpretability of high-dimensional spaces.

It has been shown in β-VAEs that using $\beta >> 1$ can induce more disentangled representations [9–11]. However, there is a fundamental trade-off between the quality of samples from the decoder and the disentanglement of latents especially with large β values. Additionally, β-VAEs do not natively disentangle latent representations with respect to prior information about the data. Previous work often incorporates supervised classifiers into the model architecture in order to provide stimulus for conditional disentanglement [12–15]. These approaches require redefining and retraining the full model if new conditioning information is added, making them computationally expensive.

In this work, we introduce a method to disentangle known conditioning information in latent spaces using latent flow matching without requiring supervised classifiers or requiring full retraining of the underlying VAE. In this paper, we:

- Demonstrate that flow models can remove or retain discrete and continuous conditioning information using a flow model trained with conditioning dropout.
- Demonstrate the utility of the method for scientific applications by isolating, disentangling and visualizing concepts in a dataset of real galaxy images.
- Introduce a method to enable scientific discovery without conflating the most dominant signals that have already been thoroughly *captured*, *considered*, and *catalogued*, moving to uncover meaningful representations of *what we don't see*.

2 Method

For an overview of the methodology proposed in this work see Figure 1. We first train a VAE on a given high dimensional dataset. We then learn the distribution of latents from the VAE using flow matching [16, 17]. Flow matching provides an elegant way of transforming any pair of distributions between each other. We use the Gaussian CondOT [18] probability path as the foundation for the training objective of the vector field, u_t^ω, which is used to simulate the flow model

$$\mathcal{L}(\omega) = \mathbb{E}_\square \left[||u_t^\omega(tz + (1-t)\epsilon, y) - (z - \epsilon)||^2 \right] \tag{1}$$

where $\square = t \sim \mathcal{U}[0,1], z \sim q_\phi(z|x), \epsilon \sim \mathcal{N}(0, I_d)$ and y is available conditioning information [18]. Additionally, we use a classifier-free guidance approach during training [19]. By replacing the conditioning information, y, at a fixed probability with a null vector, $\varnothing$, we can approximate the unconditional distribution that generates the latent samples. For a more descriptive algorithm of the training procedure see A.1.

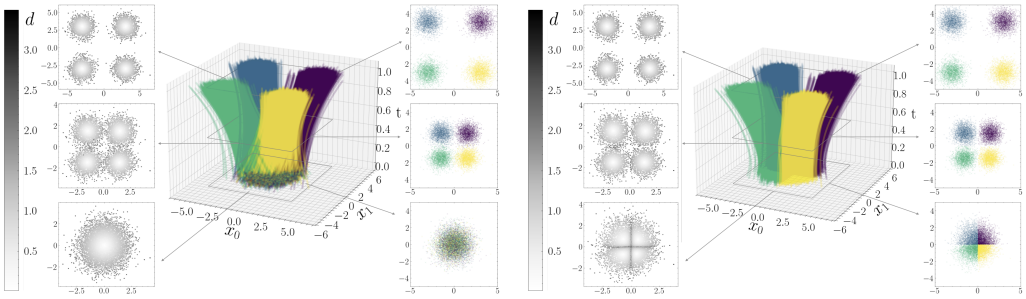
Once trained, the flow model defines a continuous and deterministic trajectory between the samples in the VAE space and the base distribution. By running the flow model backwards to $t = 0$, it is possible to find a point in the base that generates a particular VAE latent sample. Our aim is to preserve important information relevant to the VAE encoding but suppress already known information about the data. We hypothesize that due to equation 1, the KL constraint on the VAE latents, and the result of the data processing inequality: the flow model cannot add additional information during the simulation unless it is directly provided [20]. As a result, information that is contained in the VAE space should also appear in the base distribution unless explicitly conditioned on. This provides a more computationally feasible alternative to approaches such as [13, 15] which require additional terms in the loss function. Importantly, if new conditioning information is discovered or introduced, the full model requires re-training with an amended structure making this expensive for processes that seek to control and refine representations iteratively, for example: in scientific settings where one fundamentally wants to understand the factors of variation that describe some underlying data.

3 Results

3.1 2D Gaussians

We seed four synthetic isotropic Gaussians consisting of samples $\mathbf{x} \in \mathbb{R}^2$ as a first experiment. For this, we train a simple flow matching model to generate the target distribution. The velocity vector field is modeled using a simple multi-layer perceptron (MLP) which takes the position $\mathbf{x}$, a time step and the class information as input. Label dropout is used so that the data distribution can be generated either conditionally using the class index or unconditionally using a null value, $\varnothing = -1$.

Figure 2 shows the resulting flows, demonstrating how this approach can retain or remove class structure using either conditional (2a) or unconditional (2b) trajectories back to the base distribution. In the conditional flow in Figure 2a, latents become seemingly unstructured when considering class but preserve structure of the position of data points within a Gaussian as seen in the left slices coloured by Euclidean distance, d from the mean at $t = 1$. However, in the unconditional flow in Figure 2b, latents remain structured around the most prevalent feature - the classes, while also capturing position, but in a much more complex fashion than in Figure 2a.



(a) Conditional flow (b) Unconditional flow

Figure 2: Visualization of the 2D Gaussian experiment colored by Gaussian class index and Euclidean distance d of each sample to the center of its Gaussian at $t = 1$.

3.2 Colored MNIST

We first train a β-VAE on colored MNIST [21] with minimal weighting on the KL penalty for improved generation quality. Then we train a flow matching model on the latent space of the VAE using a simple MLP to parameterize the velocity vector field. This model is conditioned on the digit class and the maximum red and green values of the colored digit whilst blue is withheld from the conditioning. Label dropout is used to estimate the unconditional velocity field.

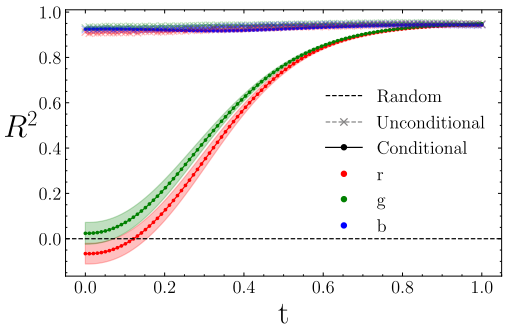
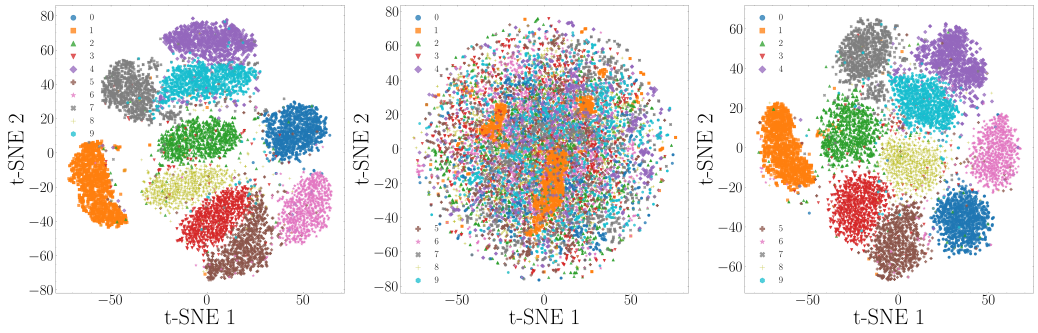


Figure 3: R^2 scores of linear regression model trained to predict the r, g, and b values throughout the conditional and unconditional flow. Note that b is withheld and is consistently recovered throughout both flows.

Figure 4a shows the latent space produced by the trained VAE using t-SNE for visualization [22] and colored by the digit type. The latent space shows clear structure concerning the digit type. With the trained flow model, we find the latent values at $t = 0$ for the samples from the VAE. Figure 4c shows what the latent values look like at $t = 0$ when generated unconditionally: the unconditional flow shown in 4c retains class structure seen in the VAE latent space. The class structure seen in Figures 4a and 4c is highly suppressed in the low dimensional projection of the conditioned space, Figure 4b. From this, we conclude that the conditioning in the flow removes the need for the conditions to be structurally present in the resulting manifold. To verify this intuition that information not in the conditioning is maintained in the conditional representation, we linearly probe both flows



(a) VAE ($t = 1$) (b) Conditional ($t = 0$) (c) Unconditional ($t = 0$)

Figure 4: t-SNE projections of cMNIST embeddings from three points in the flow. The reverse conditional flow removes the most visible class structure found in the VAE latents while the unconditional flow preserves the dominant class specific structure of the VAE latents.

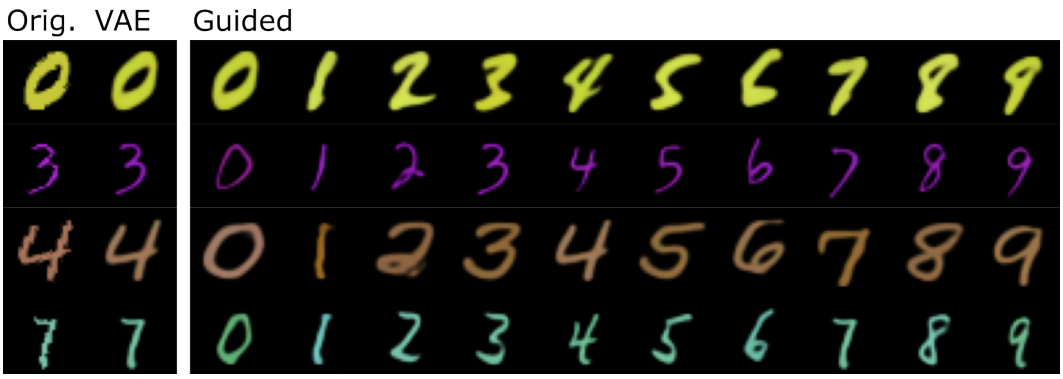


Figure 5: Style transfer in colored MNIST: the conditional $t = 0$ embeddings are then used with a different conditioning digit to produce stylistically similar digits in the VAE space. This demonstrates that stylistic features are captured and disentangled by the conditional distribution.

throughout, regressing to the maximum value of blue in the digit. Figure 3 shows the R^2 performance at points throughout the flow. The ability of the conditional representation to represent red and green degrades to random, while for blue, which is not included in the conditioning information, the probe successfully retrieves its value throughout. See Appendix A.3 for more details.

We also demonstrate that the information content of the latent representations is meaningful. We first move backwards in the flow using the digit class as a conditioning variable to arrive at $t = 0$. We then use this point to initialize the flow forwards using a different class. The results, shown in Figure 5, demonstrates that we can effectively transfer style attributes.

3.3 Galaxy10

We now explore how techniques from previous sections are applicable to the Galaxy 10 DECaLS dataset (§ A.4). Galaxies contained in the data are separated into ten broad classes.

To demonstrate how class and concepts are disentangled during the latent flow, we first select classes from the data and project to the base distribution. Afterwards, we conditionally flow from the base towards $t = 1$ using the 'round' class as conditioning information. This produces an image that has the structural features of the central galaxy removed, while retaining the color and brightness of the central point, as well as the features of the remaining field of view of the image. Figure 6 shows the residual of the conditionally generated image and the original reconstruction. Through this, we see an effective isolation of the component that the class encodes in that image. This demonstrates that even in higher dimensions, we can disentangle complex features in real data, potentially enabling novel galactic science for modern surveys [e.g. LSST 23].

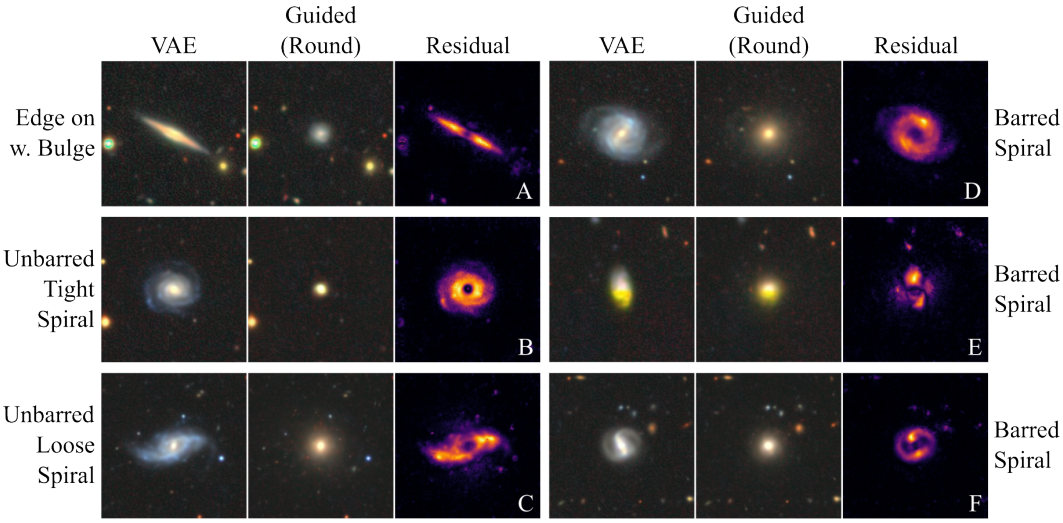


Figure 6: Six samples of feature isolation for Galaxy10. Galaxies, their conditionally generated 'round' versions and the residuals are shown. Features associated with the original galaxies are clearly separated from the remaining features of the image.

4 Conclusion

Firstly, we demonstrated with a toy Gaussian problem how flow models can produce *meaningful representations* of the data without the coarse-grained categorical data becoming the primary feature of the latent space. Our colored MNIST experiment shows this in a more complex setting where we selectively retrieve color from latents *only* if we do not condition on them. We highlight that the latents must contain information on the style of a given digit by conditional generation. Finally, we bring the method to astrophysical data, where we show that our model can disentangle important class features in galaxy images. This work is an early demonstration of a novel approach to representation learning - one which aims to support researchers in exploring what information they haven't yet captured, either because they didn't think of it, or simply could not access easily.

Acknowledgments and Disclosure of Funding

M.B. gratefully acknowledges the support provided by Schmidt Sciences. B.R. and S.C. gratefully acknowledge funding from Breakthrough Listen. The Breakthrough Prize Foundation funds the Breakthrough Initiatives, which manage Breakthrough Listen.

References

[1] Richard Bellman. *Dynamic Programming*. Dover Publications, 1957. ISBN 9780486428093.

[2] Stephen K. N. Portillo, John K. Parejko, Jorge R. Vergara, and Andrew J. Connolly. Dimensionality Reduction of SDSS Spectra with Variational Autoencoders. , 160(1):45, July 2020. doi: 10.3847/1538-3881/ab9644.

[3] Mike Walmsley, Micah Bowles, Anna M. M. Scaife, Jason Shingirai Makechemu, Alexander J. Gordon, Annette M. N. Ferguson, Robert G. Mann, James Pearson, Jürgen J. Popp, Jo Bovy, Josh Speagle, Hugh Dickinson, Lucy Fortson, Tobias Géron, Sandor Kruk, Chris J. Lintott, Kameswara Mantha, Devina Mohan, David O'Ryan, and Inigo V. Slijepevic. Scaling Laws for Galaxy Images. *arXiv e-prints*, art. arXiv:2404.02973, April 2024. doi: 10.48550/arXiv.2404.02973.

[4] Carlos X. Hernández, Hannah K. Wayment-Steele, Mohammad M. Sultan, Brooke E. Husic, and Vijay S. Pande. Variational encoding of complex dynamics. *Phys. Rev. E*, 97:062412, Jun 2018. doi: 10.1103/PhysRevE.97.062412. URL `https://link.aps.org/doi/10.1103/PhysRevE.97.062412`.

[5] Sergei V. Kalinin, Ondrej Dyck, Stephen Jesse, and Maxim Ziatdinov. Exploring order parameters and dynamic processes in disordered systems via variational autoencoders. *Science Advances*, 7(17):eabd5084, 2021. doi: 10.1126/sciadv.abd5084.

[6] Ibomoiye Domor Mienye and Theo G. Swart. Deep autoencoder neural networks: A comprehensive review and new perspectives. *Archives of Computational Methods in Engineering*, 2025. ISSN 1134-3060. doi: 10.1007/s11831-025-10260-5.

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

[8] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014. URL https://api.semanticscholar.org/CorpusID:16895865.

[9] 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*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

[10] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae, 2018. URL https://arxiv.org/abs/1804.03599.

[11] Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders, 2019. URL https://arxiv.org/abs/1802.04942.

[12] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf.

[13] N. Siddharth, Brooks Paige, Jan-Willem van de Meent, Alban Desmaison, Noah D. Goodman, Pushmeet Kohli, Frank Wood, and Philip H. S. Torr. Learning disentangled representations with semi-supervised deep generative models, 2017. URL https://arxiv.org/abs/1706.00400.

[14] Andrei Atanov, Alexandra Volokhova, Arsenii Ashukha, Ivan Sosnovik, and Dmitry Vetrov. Semi-conditional normalizing flows for semi-supervised learning, 2020. URL https://arxiv.org/abs/1905.00505.

[15] Brian Cheung, Jesse A. Livezey, Arjun K. Bansal, and Bruno A. Olshausen. Discovering hidden factors of variation in deep networks, 2015. URL https://arxiv.org/abs/1412.6583.

[16] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

[17] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL https://arxiv.org/abs/2412.06264.

[18] Peter Holderrieth and Ezra Erives. An Introduction to Flow Matching and Diffusion Models. *arXiv e-prints*, art. arXiv:2506.02070, June 2025. doi: 10.48550/arXiv.2506.02070.

[19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

[20] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

[21] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

[22] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

[23] Željko Ivezić, Steven M. Kahn, J. Anthony Tyson, Bob Abel, Emily Acosta, Robyn Allsman, David Alonso, Yusra AlSayyad, Scott F. Anderson, John Andrew, James Roger P. Angel, George Z. Angeli, Reza Ansari, Pierre Antilogus, Constanza Araujo, Robert Armstrong, Kirk T. Arndt, Pierre Astier, Éric Aubourg, Nicole Auza, Tim S. Axelrod, Deborah J. Bard, Jeff D. Barr, Aurelian Barrau, James G. Bartlett, Amanda E. Bauer, Brian J. Bauman, Sylvain Baumont, Ellen Bechtol, Keith Bechtol, Andrew C. Becker, Jacek Becla, Cristina Beldica, Steve Bellavia, Federica B. Bianco, Rahul Biswas, Guillaume Blanc, Jonathan Blazek, Roger D. Blandford, Josh S. Bloom, Joanne Bogart, Tim W. Bond, Michael T. Booth, Anders W. Borgland, Kirk Borne, James F. Bosch, Dominique Boutigny, Craig A. Brackett, Andrew Bradshaw, William Nielsen Brandt, Michael E. Brown, James S. Bullock, Patricia Burchat, David L. Burke, Gianpietro Cagnoli, Daniel Calabrese, Shawn Callahan, Alice L. Callen, Jeffrey L. Carlin, Erin L. Carlson, Srinivasan Chandrasekharan, Glenaver Charles-Emerson, Steve Chesley, Elliott C. Cheu, Hsin-Fang Chiang, James Chiang, Carol Chirino, Derek Chow, David R. Ciardi, Charles F. Claver, Johann Cohen-Tanugi, Joseph J. Cockrum, Rebecca Coles, Andrew J. Connolly, Kem H. Cook, Asantha Cooray, Kevin R. Covey, Chris Cribbs, Wei Cui, Roc Cutri, Philip N. Daly, Scott F. Daniel, Felipe Daruich, Guillaume Daubard, Greg Daues, William Dawson, Francisco Delgado, Alfred Dellapenna, Robert de Peyster, Miguel de Val-Borro, Seth W. Digel, Peter Doherty, Richard Dubois, Gregory P. Dubois-Felsmann, Josef Durech, Frossie Economou, Tim Eifler, Michael Eracleous, Benjamin L. Emmons, Angelo Fausti Neto, Henry Ferguson, Enrique Figueroa, Merlin Fisher-Levine, Warren Focke, Michael D. Foss, James Frank, Michael D. Freemon, Emmanuel Gangler, Eric Gawiser, John C. Geary, Perry Gee, Marla Geha, Charles J. B. Gessner, Robert R. Gibson, D. Kirk Gilmore, Thomas Glanzman, William Glick, Tatiana Goldina, Daniel A. Goldstein, Iain Goodenow, Melissa L. Graham, William J. Gressler, Philippe Gris, Leanne P. Guy, Augustin Guyonnet, Gunther Haller, Ron Harris, Patrick A. Hascall, Justine Haupt, Fabio Hernandez, Sven Herrmann, Edward Hileman, Joshua Hoblitt, John A. Hodgson, Craig Hogan, James D. Howard, Dajun Huang, Michael E. Huffer, Patrick Ingraham, Walter R. Innes, Suzanne H. Jacoby, Bhuvnesh Jain, Fabrice Jammes, M. James Jee, Tim Jenness, Garrett Jernigan, Darko Jevremović, Kenneth Johns, Anthony S. Johnson, Margaret W. G. Johnson, R. Lynne Jones, Claire Juramy-Gilles, Mario Jurić, Jason S. Kalirai, Nitya J. Kallivayalil, Bryce Kalmbach, Jeffrey P. Kantor, Pierre Karst, Mansi M. Kasliwal, Heather Kelly, Richard Kessler, Veronica Kinnison, David Kirkby, Lloyd Knox, Ivan V. Kotov, Victor L. Krabbendam, K. Simon Krughoff, Petr Kubánek, John Kuczewski, Shri Kulkarni, John Ku, Nadine R. Kurita, Craig S. Lage, Ron Lambert, Travis Lange, J. Brian Langton, Laurent Le Guillou, Deborah Levine, Ming Liang, Kian-Tat Lim, Chris J. Lintott, Kevin E. Long, Margaux Lopez, Paul J. Lotz, Robert H. Lupton, Nate B. Lust, Lauren A. MacArthur, Ashish Mahabal, Rachel Mandelbaum, Thomas W. Markiewicz, Darren S. Marsh, Philip J. Marshall, Stuart Marshall, Morgan May, Robert McKercher, Michelle McQueen, Joshua Meyers, Myriam Migliore, Michelle Miller, and David J. Mills. LSST: From Science Drivers to Reference Design and Anticipated Data Products. , 873(2):111, March 2019. doi: 10.3847/1538-4357/ab042c.

[24] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017. URL `https://arxiv.org/abs/1709.07871`.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[26] Chris Lintott, Kevin Schawinski, Steven Bamford, Anže Slosar, Kate Land, Daniel Thomas, Edd Edmondson, Karen Masters, Robert C. Nichol, M. Jordan Raddick, Alex Szalay, Dan Andreescu, Phil Murray, and Jan Vandenberg. Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies. , 410(1):166–178, January 2011. doi: 10.1111/j.1365-2966.2010.17432.x.

[27] Henry W. Leung and Jo Bovy. Deep learning of multi-element abundances from high-resolution spectroscopic data. , 483(3):3255–3277, March 2019. doi: 10.1093/mnras/sty3217.

[28] M. Walmsley. Galaxy 10 DECaLS: Dataset on hugging face. https://huggingface.co/datasets/mwalmsley/galaxy10_decals.

[29] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv e-prints*, art. arXiv:1505.04597, May 2015. doi: 10.48550/arXiv.1505.04597.

[31] Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

[32] Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. *arXiv preprint 2307.03672*, 2023.

A Experiment Details

A.1 Additional training details

A more detailed description to training is presented here [18].

Algorithm 1: Classifier-free guidance training for flow matching using the Gaussian CondOT path $p_t(z_t|z) = \mathcal{N}(z_t; tz, (1-t)^2 I_d)$

Require: Paired dataset $(x, y) \sim p_{data}$, frozen VAE encoder f, neural network to parameterize the velocity vector field u_t^ω

for *each mini-batch of data* **do**
- Sample a data example (x, y) from the dataset
- Encode x using the VAE such that $z \leftarrow f(x)$
- Sample a random time $t \sim \mathrm{Unif}[0, 1]$
- Sample noise $\epsilon \sim \mathcal{N}(0, I_d)$
- Set $z_t \leftarrow tz + (1-t)\epsilon$
- With probability p drop label: $y \leftarrow \varnothing$
- Compute loss

$$\mathcal{L}(\omega) = \left\| u_t^\omega(z_t \mid y) - (z - \epsilon) \right\|^2$$

- Update the model parameters ω via gradient descent on $\mathcal{L}(\omega)$.

end for

A.2 2d Gaussians

The four Gaussians were each generated with covariances $\Sigma = 0.5\mathbb{I}$ and respective means from the set $\{(\pm 3, \pm 3)^\intercal, (\pm 3, \mp 3)^\intercal\} \in \mathbb{R}^2$. The model that is fit to the flow is an MLP consisting of four linear layers with ELU activations in-between, resulting in 8.5k trainable parameters. We used a label dropout probability of $p = 0.2$.

A.3 cMNIST

We augment the MNIST dataset [21] with color information to provide a highly controllable experiment in order to verify the intuitions we have built from the 2D Gaussian case in section 3.1. The color information is determined such that

$$c_i \sim \mathcal{U}(0.05, 0.95) \tag{2}$$

for $i = 1, 2, 3$ (i.e. the R, G and B channels of the image where an image I, with height H and width W, is $I \in [0, 1]^{H \times W \times 3}$). We also use the following data augmentations:

- Rotations: $\theta \sim \mathcal{N}(0°, 10°)$
- Scale: $s \sim \mathcal{N}(1, 0.1)$

The β-VAE consists of CNN encoder and decoder. The encoder and decoders are inspired by the VGG architecture (convolutions followed by batch norms, ReLU, and maxpooling to step down in scale). The decoder uses upsampling instead of max pooling. There are also linear projections with ReLU activations from the CNN encoder to the latent space and a linear projection from the latent space to the CNN decoder. This model contains 23.4 M parameters.

The model is trained using $\beta = 1 \times 10^{-6}$. The flow model consists of a simple MLP of four hidden layers with GELU activations taking the latent vector and time as inputs. This model consists of 171 k parameters. The intermediate linear layers are modulated with a projection of the class information. For this, the digit is embedded and concatenated with the maximum red and green value. We then project the resulting vector using a linear layer to output scale and shift terms for the intermediate states of the network after linear layers in a method similar to [24]. Conditioning dropout is used with a probability of $p = 0.1$.

For the linear regression model in Figure 3 we draw 512 samples at random from set of embedding data. We then compute the flow for each sample and return the chain of representations. We then train the linear regression model [25] on each time step of the chain for all representations and compute the R^2 on a test set. We repeat the sampling of the training set and report the mean and standard deviation in the shaded area of Figure 3.

A.4 Galaxy10 DECaLS

The Galaxy10 DECaLS Dataset contains 17 736, 256 x 256 sized colored galaxy images in the g, r and z band which have been scaled for clarity to RGB PNGs. These are separated into ten broad classes including: disturbed, merging, round smooth, in-between round smooth, cigar round smooth, barred spiral, unbarred tight spiral, unbarred loose spiral, edge-on without bulge and edge-on with bulge. The labels were originally provided by volunteers from the Galaxy Zoo project [26], the collection was compiled by [27] and the data loader was made available by [28].

A β-VAE with $\beta = 1 \times 10^{-6}$ was trained on the images to produce a $4 \times 32 \times 32$ latent representation. We used the diffusers [29] implementation of a variational autoencoder with four down sampling blocks using the "DownEncoderBlock2D" in the encoder, each outputting 32, 64, 128 and 256 channels respectively. The decoder follows a symmetric structure using "UpDecoderBlock2D". Each block in the VAE has four layers. This model accounts for 20.3M parameters.

We use a class conditional UNet from the TorchCFM package [30–32] to parameterize the velocity field using the Galaxy10 classes as the conditioning signal. The UNet has four layers with 64, 128, 128 and 128 channels. Each downsampling of the UNet has a single residual block. This totals 6.1M parameters. Conditioning dropout is used with a probability of $p = 0.1$.

B Limitations

This work is in early stages and so it should be noted that there are a number of limitations. These include:

- Solutions to ODEs used for the flow model training and inference have an inherent source of error. At this stage no quantification has been undertaken on how this may impact the

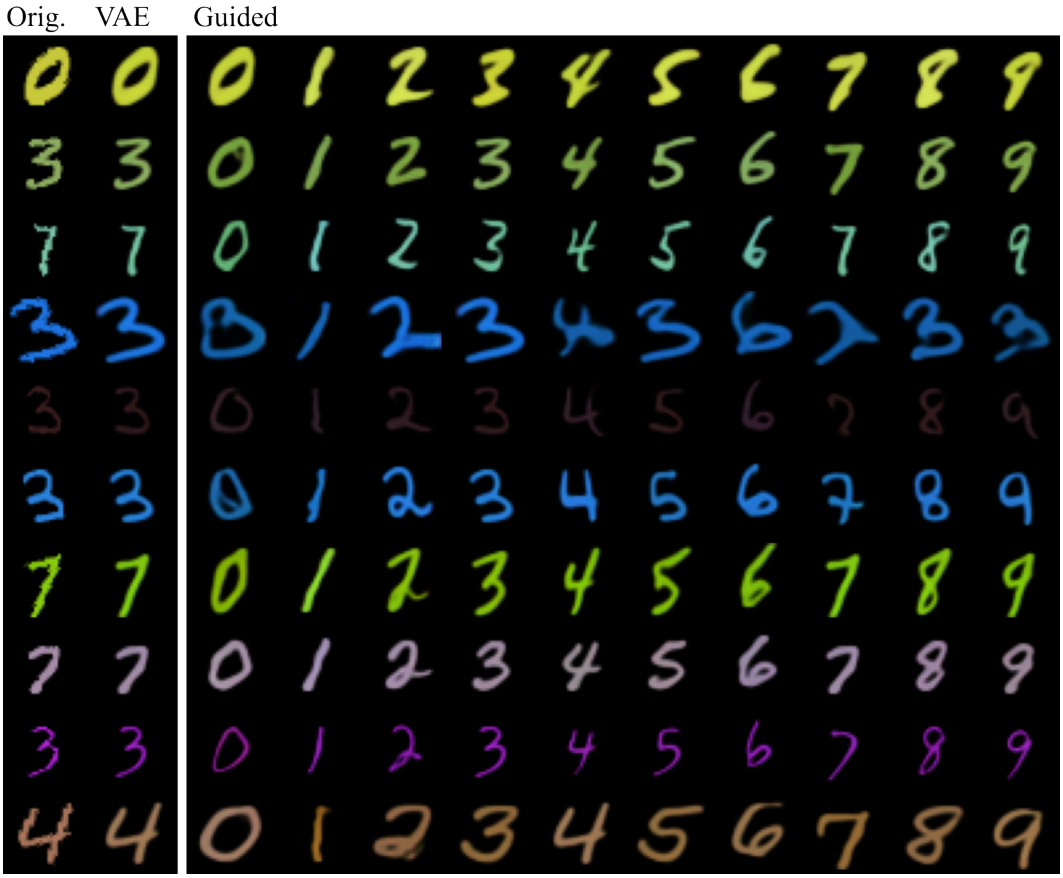


Figure 7: Style transfer in colored MNIST: the conditional $t = 0$ embeddings are then used with a different conditioning digit to produce stylistically similar digits in the VAE space. This demonstrates that non-classification features are captured by the conditional distribution.

representations during the chain, especially with the loss of information in the conditioning variables.

- Computational resources were limited in the development of this paper, and therefore a full investigation into the hyper-parameters associated with the flow model training has not been undertaken. This is especially true of the dropout frequency used in training and how it may impact the quality of the unconditional distributions.
- It is unclear which conditioning mechanisms are the most appropriate and efficient in approximating the conditional velocity field.