Efficiently Moving Instead of Reweighting Collider Events with Machine Learning

Radha Mastandrea
Department of Physics, University of California, Berkeley
Berkeley, CA 94720
Physics Division, Lawrence Berkeley National Laboratory, Berkeley
Berkeley, CA 94720
rmastand@berkeley.edu

Benjamin Nachman
Department of Physics, University of California, Berkeley
Berkeley, CA 94720
Physics Division, Lawrence Berkeley National Laboratory, Berkeley
Berkeley, CA 94720
Berkeley Institute for Data Science, University of California, Berkeley
Berkeley, CA 94720
bpnachman@lbl.gov

Abstract

There are many cases in collider physics and elsewhere where a calibration dataset is used to predict the known physics and / or noise of a target region of phase space. This calibration dataset usually cannot be used out-of-the-box but must be tweaked, often with conditional importance weights, to be maximally realistic. Using resonant anomaly detection as an example, we compare a number of alternative approaches based on transporting events with normalizing flows instead of reweighting them. We find that the accuracy of the morphed calibration dataset depends on the degree to which the transport task is set up to carry out optimal transport, which motivates future research into this area.

1 Introduction

Many tasks in collider physics depend on the calibration of auxiliary datasets. Data from a well-understood region of phase space are chosen as a *reference* to model the known physics in a *target* region of phase space. When the reference and known physics in the target are identically distributed, differences between the reference and target would indicate the presence of new phenomena. However, the reference data may be a distorted version of the known physics in the target and so corrections need to be applied. The reference is chosen to be as similar as possible to the known physics in the target so that the corrections applied should be small.

Traditionally, this calibration task has been performed using importance weights estimated from ratios of histograms, either using data-driven approaches like the control region method or fully data-based approaches like the ABCD or Matrix Method techniques (see e.g. Ref. [1] for a review). These methods can be generalized to the unbinned (and high-dimensional) case with machine learning-based likelihood ratio estimation [2–4]. An alternative approach that may be more precise, especially when the reference and target probability densities have regions of non-overlapping support, is to morph the features themselves. Given a reference distribution $X_R \sim p_{X_R}$ and a target distribution $X_T \sim p_{X_T}$,

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

a mapping function $f : X_R \to X_T$ is chosen so that the probability density of the transformed reference $f(X_R)$ is as close a match as possible to the probability density of the target.

Optimal transport (OT) has been studied in collider physics to solve the calibration problem [5]. In this paper, we focus on the case of conditional morphing, where X_R and X_T are conditioned on an observable M (often a mass, thus the symbol), and we set $f(\cdot|M) : X_R|M \to X_T|M$. The reason for this setup is that the morphing function is typically learned in a background-dominated region and then interpolated or extrapolated in M to a signal-sensitive region. This setup was first explored in collider physics in Ref. [6] and uses normalizing flows (NFs) [7]. These NFs are not specifically tasked with making the morphing minimal. This our goal in this paper is to explore how minimal these moves are and examine if tweaks to the setup can bring these moves closer to the OT solution and thus improve model fidelity for downstream inference tasks. As in Ref. [6], we use anomaly detection as our numerical example.

This paper is organized as follows. In Sec. 2, we introduce the application of normalizing flows to the problem of resonant anomaly detection and describe our dataset and training procedure. Numerical results are presented in Sec. 3, and we discuss future directions in Ref. Sec. 4.

2 Methods

Resonant or *group* anomaly detection (see e.g. Ref. [8]) in collider physics begins with a resonant feature M. A potential signal will have $|M - M_0| \lesssim c$ (which defines the signal region) for some unknown M_0 and often knowable c. The value of M_0 is usually found through a scan. Additional features $X \in \mathbb{R}^N$ are chosen which can be used to distinguish signal from background. Weakly supervised methods learn a classifier acting on X that can distinguish signal region events in the target from events in the reference [2, 6, 9–13]. Approaches vary on how they construct the reference, but most use sideband information ($|M - M_0| \gtrsim c$) to some degree. The first use of NFs in this context was in Ref. [11], which used NFs as a generative model. In contrast, the idea of Ref. [6] is to use NFs as the morphing functions described in the previous section.

Normalizing flows are neural networks that can approximate the density of complex data through a composition of relatively simpler functions with a tractable Jacobian: $L = \log p(z) + \sum \log J_i$, where L is the loss, $p(z)$ is the base distribution, and J_i is the Jacobian when transforming from the i^{th} function to the $(i+1)^{\text{th}}$ function in the composition. We consider multiple ways of training the flow:

1. **Double Base**. We learn two conditional NFs, one that maps from a standard normal $z \sim \mathcal{N}(0,1)^N$ to the target $f_T(\cdot|M) : z|M \to X_T|M$ and one from a standard normal to the reference $f_R(\cdot|M) : z|M \to X_R|M$. The morphing function is then $f_T \circ f_R^{-1}$.
2. **Base to Data**. We learn a conditional NF for the reference $f_R(\cdot|M) : z|M \to X_R|M$ and then use this as the base density to learn a second flow from the reference to the target $f(\cdot|M) : X_R|M \to X_T|M$.
3. **Identity Initialization.** Same as (2), but we initialize the mapping at the identity function. This is done by first learning the mapping $f(\cdot|M) : X_R|M \to X_R|M$, then using transfer learning to adapt this flow to map X_R to X_T.
4. **Movement penalty.** Same as (2), but we add a term to the loss function that explicitly penalizes movement in the parameter space, $L_2 = \alpha \|X_R - f(X_R|M)\|^2$ for a hyperparameter α.

To evaluate the performance of the models, we study how far the events were moved, $|X_R - f(X_R|M)|$ as well as the fidelity of the move, quantified by the (ideally poor) performance of a post-hoc classifier trained to distinguish the interpolated reference from the target in the signal region.

2.1 Dataset

We use the LHC 2020 Olympics R&D dataset [14, 15] which consists of 1M background (Standard Model) events and 100k signal/anomaly events. Our setup is nearly the same as in many previous resonant anomaly detection studies with the same dataset [2, 6, 9–13]. The events naturally live in a high-dimensional space (each containing hundreds of particles with a momentum), but we consider a

five-dimensional compressed version that has been extensively studied in collider analyses. These five features, along with the resonant feature, are displayed in Fig. 1. The reference dataset consists of 1M events from one simulator while the target consists of 1M background events from a different simulator. As we are investigating the background estimation, no anomalous events are part of the training or testing in this paper.

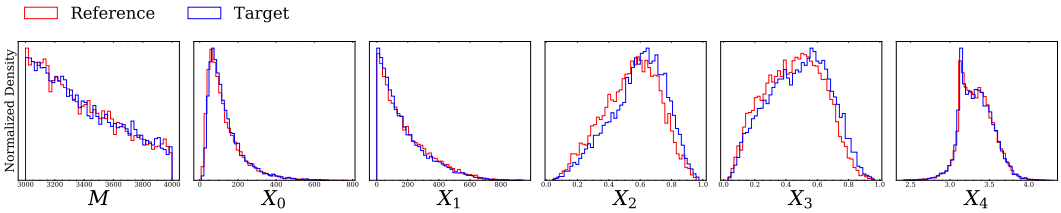


Figure 1: Reference and target distributions used in this study The feature space is comprised of the resonant feature M and five other features m_{J_1}, Δm_{JJ}, $\tau_{J_1}^{21}$, $\tau_{J_2}^{21}$, and ΔR_{JJ}. A description of these observables can be found in [16]. The signal region is defined by $|M - M_0| < c$ for $M_0 = 3500$ and $c = 100$.

2.2 Training procedure

To generate flows that map from $z \sim \mathcal{N}(0,1)^N$ to a reference / target distribution, we use the highly expressive MADE autoregressive (AR) module [17]. We use 8 modules consisting of 8 stacked piecewise rational quadratic transformations, interleaved with a reverse permutation of all dimensions. Each masked linear layer of the MADE modules has dimensions (64, 64). The resonant feature M is embedded in a linear network of size (1, 64). We train for 60 epochs with a batch size of 128, a a cosine annealing learning rate initialized at 10^{-4}, and a weight decay of 10^{-4}.

To generate flows that map between the reference and target distributions (which are relatively similar to each other), we use a more lightweight architecture consisting of a coupling model with piecewise rational quadratic transformations. These flows contains 2 stacked layers interleaved with a reverse permutation, with each masked linear layer having dimensions (16, 16). We train for 40 epochs with a batch size of 256, a cosine annealing learning rate initialized at 4×10^{-4}, and a weight decay of 10^{-4}.

The training dataset is comprised from the sidebands of the resonant feature M, which ensures that the signal region is kept blinded until testing. All flows are constructed using the NFLOWS package [18], trained using PYTORCH [19], and optimized using ADAM [20]. All hyperparameters are optimized through grid search. For each training method, we repeat the flow training 5 times with a different random seed.

3 Results

We evaluate the learned mappings from reference X_R to target X_T for the four training procedures proposed in Sec. 2 in two ways: (1) we observe how far in parameter space points from the reference travel as they are mapped to the target distribution, and (2) we evaluate the fidelity of the mapping through training a classifier to discriminate X_T from $f(X_R|M)$.

The mappings are evaluated in the signal region ($|M - M_0| < 100$), in training sidebands regions ($100 < |M - M_0| < 300$), and additionally in outer sidebands regions ($300 < |M - M_0| < 500$). This allows us to gauge the performance of the mapping in both interpolation and extrapolation tasks.

3.1 Distanced moved in parameter space

In Fig. 2, we show the distances traveled in feature space for each data point X_R that is mapped to the target distribution by $f(X_R|M)$. In Sec. A, we show the distances traveled for features X_2 and X_3, which according to Fig. 1 are the ones that are most different between the reference and target distributions.

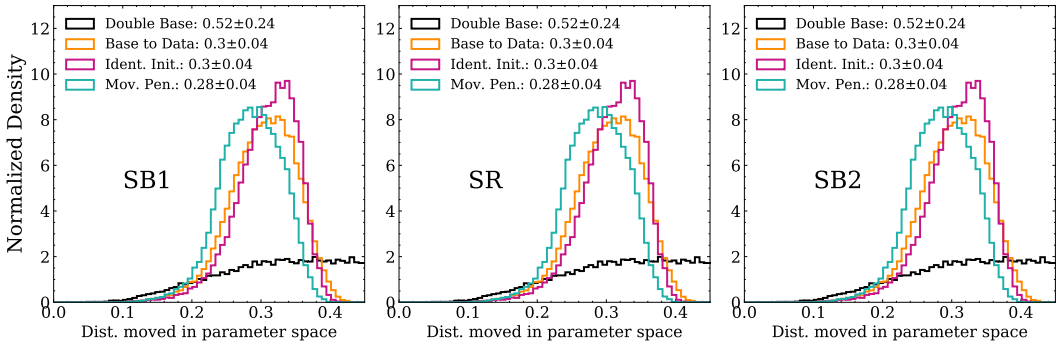


Figure 2: Distance traveled in the full parameter space under a mapping from reference X_R to target X_T. The data is preprocessed so feature lies in the interval (-3, 3), so the maximum possible distance traveled is ~ 13. We provide the mean distances traveled $\pm 1\sigma$ errors.

Band	Double Base	Base to Data	Identity Init.	L_2 ($\alpha = 10^{-2}$)
OB1	0.630 ± 0.024	0.511 ± 0.003	0.508 ± 0.004	0.507 ± 0.002
SB1	0.501 ± 0.000	0.502 ± 0.001	0.501 ± 0.000	0.502 ± 0.001
SR	0.553 ± 0.011	0.503 ± 0.001	0.503 ± 0.001	0.503 ± 0.000
SB2	0.501 ± 0.000	0.503 ± 0.001	0.503 ± 0.001	0.502 ± 0.001
OB2	0.594 ± 0.030	0.506 ± 0.002	0.507 ± 0.004	0.507 ± 0.003

Table 1: ROC scores for a binary classifier trained to discriminate mapped reference $f(X_R|M)$ from target X_T . The scores are calculated separately for different regions of the resonant parameter. (OB1, OB2) = (low, high) mass outer bands; (SB1, SB2) = (low, high) mass sidebands; SR = signal region. We provide the mean ROC scores $\pm 1\sigma$ errors.

The Double Base method produces a vastly larger range of distances traveled than the other three methods, which is to be expected as the training procedure contains no explicit link between the reference and target distributions. The Movement Penalty method has the smallest mean distance traveled in the entire and individual feature space(s), although by a narrow margin.

3.2 Fidelity of transform

In Table 1, we provide the ROC scores for a binary classification neural net trained to discriminate $f(X_R|M)$ from X_T. The classifiers are dense networks of three hidden layers and 32 nodes, all with RELU activation. We train the binary classifiers for 20 epochs with a batch size of 128 and a cosine annealing learning rate initialized at 10^{-3},

For the purpose of this report, we define a “random" classifier to have a ROC score < 0.51. A successful mapping between the reference and target distributions would result in the trained binary classifier performing no better than random.

All four training procedures show good fidelity in the sidebands regions (recall that this comprises the training dataset region). All methods except the Double Base method also show good fidelity when evaluated in the signal region. All methods exhibit a drop in performance when evaluated in the outer bands regions, but only the Double Base method drops enough to lose fidelity.

4 Future areas of study

In this work, we have explored modifications to the training procedure for a normalizing flow tasked with learning a mapping between a reference and a target dataset. Our explored application uses reference and target datasets that are similar, as might be expected in the collider physics calibration problem of modeling known physics in a specific region of phase space.

The Double Base method is clearly suboptimal in terms of optimal transport and mapping fidelity, likely due to the fact that it transports between the reference and the target through an uncorrelated

standard normal distribution. The other three methods (Base to Data, Identity Initialization, and Movement Penalty) show comparable performances in both metrics, and in fact all meet the requirement of being a faithful mapping due to the indiscriminability of X_T from $f(X_R|M)$. In this situation where the reference and target sets are similar, using the Base to Data method is acceptable. However, this method may not be adequate for more complex (transformations between) datasets.

As an avenue for future investigations, we may consider exchanging a normalizing flow for a continuous normalizing flow (CNF) [21]. Such a flow restricts transformations to be continuous, such that each point can be assigned a trajectory with a velocity vector. Building upon this, the OT-Flow method [22] adds to the CNF loss both an L_2 movement penalty and a penalty that encourages the mapping to transport points along the minimum of some potential function. Such alternatives might be explored for situations when the reference and target distributions are significantly different.

5 Potential Broader Impacts

In this work, we have explored methods to augment the training of normalizing flows that learn mappings between probability distributions derived from LHC-like particle collision datasets. However, these modifications could be used to improve the performance for any normalizing flow-like architecture, regardless of the physical origin of the reference and target datasets. Our work could then be beneficial to any physical problem that relies on the creation or transformations of probability distributions. Since our work serves as a method to improve an existing architecture, rather than to define a new analysis procedure, it is our belief that this work does not present any foreseeable societal consequence.

Code availability

The code for all experiments in this report can be found at https://github.com/rmastand/FETA.

Acknowledgments and Disclosure of Funding

We thank Samuel Klein for his useful discussions relating to these experiments. B.N. and R.M. were supported by the Department of Energy, Office of Science under contract number DE-AC02-05CH11231. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 2146752. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman, and D. Shih, "Machine Learning in the Search for New Fundamental Physics," Dec. 2021. arXiv: 2112.03769 [hep-ph].

[2] A. Andreassen, B. Nachman, and D. Shih, "Simulation Assisted Likelihood-free Anomaly Detection," *Phys. Rev. D*, vol. 101, no. 9, p. 095 004, 2020. DOI: 10.1103/PhysRevD.101.095004. arXiv: 2001.05001 [hep-ph].

[3] G. Aad *et al.*, "Search for resonant pair production of Higgs bosons in the $b\bar{b}b\bar{b}$ final state using pp collisions at $\sqrt{s}$ = 13 TeV with the ATLAS detector," *Phys. Rev. D*, vol. 105, no. 9, p. 092 002, 2022. DOI: 10.1103/PhysRevD.105.092002. arXiv: 2202.07288 [hep-ex].

[4] "Anomaly detection search for new resonances decaying into a Higgs boson and a generic new particle X in hadronic final states using $\sqrt{s}$ = 13 TeV pp collisions with the ATLAS detector," 2022.

[5] C. Pollard and P. Windischhofer, "Transport away your problems: Calibrating stochastic simulations with optimal transport," Jul. 2021. arXiv: 2107.08648 [physics.data-an].

[6] J. A. Raine, S. Klein, D. Sengupta, and T. Golling, "CURTAINs for your Sliding Window: Constructing Unobserved Regions by Transforming Adjacent Intervals," Mar. 2022. arXiv: 2203.09470 [hep-ph].

[7] D. J. Rezende and S. Mohamed, *Variational inference with normalizing flows*, 2015. DOI: 10.48550/ARXIV.1505.05770. [Online]. Available: https://arxiv.org/abs/1505.05770.

[8] G. Kasieczka, B. Nachman, and D. Shih, "New Methods and Datasets for Group Anomaly Detection From Fundamental Physics," Jul. 2021. arXiv: 2107.02821 [stat.ML].

[9] J. H. Collins, K. Howe, and B. Nachman, "Anomaly Detection for Resonant New Physics with Machine Learning," *Phys. Rev. Lett.*, vol. 121, no. 24, p. 241 803, 2018. DOI: 10.1103/PhysRevLett.121.241803. arXiv: 1805.02664 [hep-ph].

[10] J. H. Collins, K. Howe, and B. Nachman, "Extending the search for new resonances with machine learning," *Phys. Rev.*, vol. D99, no. 1, p. 014 038, 2019. DOI: 10.1103/PhysRevD.99.014038. arXiv: 1902.02634 [hep-ph].

[11] B. Nachman and D. Shih, "Anomaly Detection with Density Estimation," *Phys. Rev. D*, vol. 101, p. 075 042, 2020. DOI: 10.1103/PhysRevD.101.075042. arXiv: 2001.04990 [hep-ph].

[12] K. Benkendorfer, L. L. Pottier, and B. Nachman, "Simulation-Assisted Decorrelation for Resonant Anomaly Detection," Sep. 2020. arXiv: 2009.02205 [hep-ph].

[13] A. Hallin *et al.*, "Classifying Anomalies THrough Outer Density Estimation (CATHODE)," Sep. 2021. arXiv: 2109.00546 [hep-ph].

[14] G. Kasieczka, B. Nachman, and D. Shih, *Official Datasets for LHC Olympics 2020 Anomaly Detection Challenge (Version v6) [Data set]*. https://doi.org/10.5281/zenodo.4536624, 2019.

[15] G. Kasieczka *et al.*, "The LHC Olympics 2020: A Community Challenge for Anomaly Detection in High Energy Physics," Jan. 2021. arXiv: 2101.08320 [hep-ph].

[16] J. Thaler and K. Van Tilburg, "Identifying Boosted Objects with N-subjettiness," *JHEP*, vol. 03, p. 015, 2011. DOI: 10.1007/JHEP03(2011)015. arXiv: 1011.2268 [hep-ph].

[17] M. Germain, K. Gregor, I. Murray, and H. Larochelle, "Made: Masked autoencoder for distribution estimation," 2015. DOI: 10.48550/ARXIV.1502.03509. [Online]. Available: https://arxiv.org/abs/1502.03509.

[18] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, *nflows: Normalizing flows in PyTorch*, version v0.14, Nov. 2020. DOI: 10.5281/zenodo.4296287. [Online]. Available: https://doi.org/10.5281/zenodo.4296287.

[19] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[20] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2014. DOI: 10.48550/ARXIV.1412.6980. [Online]. Available: https://arxiv.org/abs/1412.6980.

[21] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, *Neural ordinary differential equations*, 2018. DOI: 10.48550/ARXIV.1806.07366. [Online]. Available: https://arxiv.org/abs/1806.07366.

[22] D. Onken, S. W. Fung, X. Li, and L. Ruthotto, "Ot-flow: Fast and accurate continuous normalizing flows via optimal transport," *CoRR*, vol. abs/2006.00104, 2020. arXiv: 2006.00104. [Online]. Available: https://arxiv.org/abs/2006.00104.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 2.2 and additionally Sec. 3.2.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] No resources external to the authors' institutions were used.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Sec. 2.2.

(b) Did you mention the license of the assets? [Yes]

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

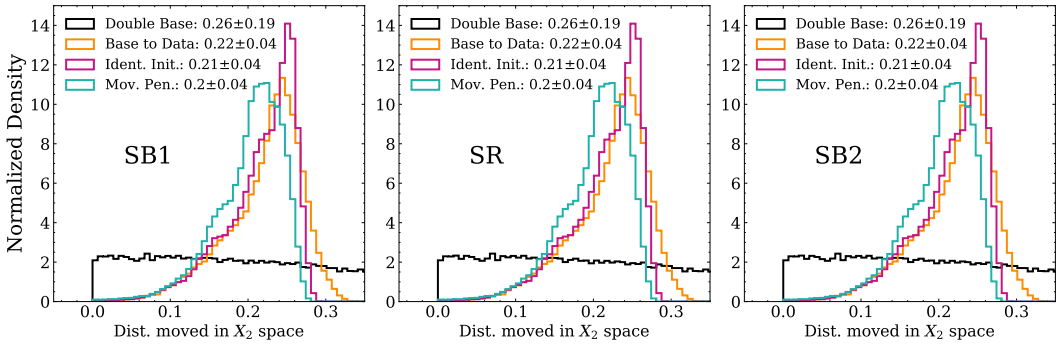
(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

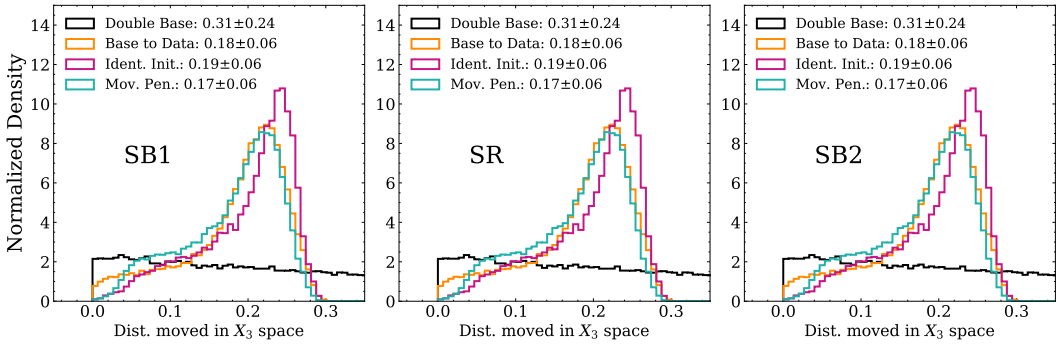
(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

Here, we provide supplementary plots for the distances traveled in X_2 and X_3 space for each data point X_R that is mapped to the target distribution by $f(X_R|M)$. The other three features did not show appreciable movement after training, likely due to the similarity of their distributions between the reference and the target datasets.



(a) Feature X_2, corresponding to $\tau_{J_1}^{21}$.



(b) Feature X_3, corresponding to $\tau_{J_2}^{21}$.

Figure 3: Distance traveled in a single dimension of the parameter space under a mapping from reference X_R to target X_T. The maximum possible distance traveled is 6. We provide the mean distances traveled $\pm 1\sigma$ errors.