

# UC Berkeley

## UC Berkeley Previously Published Works

### Title

Morphing one dataset into another with maximum likelihood estimation

### Permalink

<https://escholarship.org/uc/item/9q70q71w>

### Journal

Physical Review D, 108(9)

### ISSN

2470-0010

### Authors

Golling, Tobias

Klein, Samuel

Mastandrea, Radha

et al.

### Publication Date

2023-11-01

### DOI

10.1103/physrevd.108.096018

### Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

## Morphing one dataset into another with maximum likelihood estimation

Tobias Golling<sup>1,\*</sup>, Samuel Klein<sup>1,†</sup>, Radha Mastandrea<sup>2,3,‡</sup>, Benjamin Nachman<sup>3,4,§</sup> and John Andrew Raine<sup>1,||</sup>

<sup>1</sup>*Département de Physique Nucléaire et Corpusculaire, Université de Genève, Genève, Switzerland*

<sup>2</sup>*Department of Physics, University of California, Berkeley, California 94720, USA*

<sup>3</sup>*Physics Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

<sup>4</sup>*Berkeley Institute for Data Science, University of California, Berkeley, California 94720, USA*



(Received 21 September 2023; accepted 19 October 2023; published 21 November 2023)

Many components of data analysis in high-energy physics and beyond require morphing one dataset into another. This is commonly solved via reweighting, but there are advantages of preserving weights and shifting the data points instead. Normalizing flows are machine learning models that have shown impressive precision on a variety of particle physics tasks. Naively, normalizing flows cannot be used for morphing because they require knowledge of the probability density of the starting dataset; in most cases in particle physics, we can generate more examples, but we do not know densities explicitly. We propose a protocol called flows for flows for training normalizing flows to morph one dataset into another even if the underlying probability density of neither dataset is known explicitly. This enables a morphing strategy trained with maximum likelihood estimation, a setup that has been shown to be highly effective in related tasks. We study variations on this protocol to explore how far the data points are moved to statistically match the two datasets. Furthermore, we show how to condition the learned flows on particular features in order to create a morphing function for every value of the conditioning feature. For illustration, we demonstrate flows for flows on toy examples as well as a collider physics example involving dijet events.

DOI: [10.1103/PhysRevD.108.096018](https://doi.org/10.1103/PhysRevD.108.096018)

### I. INTRODUCTION

One common data analysis task in high-energy physics and beyond is to take a reference set of examples  $R$  and modify them to be statistically identical to a target set of examples  $T$ . In this setting, we do not have access to the probability density of  $x \in \mathbb{R}^N$  responsible for  $R$  or  $T$  (i.e.,  $p_T$  and  $p_R$ ), but we can sample from both by running an experiment or simulator. Examples of this task include shifting simulation to match data for detector calibrations, morphing experimental or simulated calibration data to match backgrounds in signal-sensitive regions of phase space for background estimation or anomaly detection, and tweaking simulated examples with one set of parameters to match another set for parameter inference.

A well-studied way to achieve dataset morphing is to assign importance weights  $w$  so that  $w(x) \approx p_T(x)/p_R(x)$ .

This likelihood ratio can be constructed using machine learning-based classifiers (see e.g., [1,2]) to readily accommodate  $N \gg 1$  without ever needing to estimate  $p_T$  or  $p_R$  directly. While highly effective, likelihood-ratio methods also have a number of fundamental challenges. With non-unity weights, the statistical power of a dataset is diluted. Furthermore, even small regions of nonoverlapping support between  $p_T$  and  $p_R$  can cause estimation strategies for  $w$  to fail.

A complementary strategy to importance weights is direct feature morphing. In this case, the goal is to find a map  $f: \mathbb{R}^N \rightarrow \mathbb{R}^N$  from the reference to the target space such that the probability density of  $f(x \sim p_R)$  matches  $p_T$ . Unlike the importance sampling scenario,  $f$  is not unique. The goal of this paper is to study how to construct  $f$  as a normalizing flow [3,4]—a type of invertible deep neural network most often used for density estimation or sample generation. Normalizing flows have proven to be highly effective generative models, which motivates their use as morphing functions. Traditionally, normalizing flows are trained in the setting where  $p_R$  is known explicitly (e.g., a Gaussian distribution). Here we explore how to use flows when neither  $p_R$  or  $p_T$  are known explicitly. We call our method flows for flows. This approach naturally allows for the morphing to be conditional on some feature, such as a mass variable [5–7]. Approaches similar to flows for flows have been performed for variational autoencoders [8] and recently, diffusion models [9].

\*tobias.golling@unige.ch

†samuel.klein@unige.ch

‡rmastand@berkeley.edu

§bpnachman@lbl.gov

||john.raine@unige.ch

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP<sup>3</sup>.

In many cases in physics,  $p_R$  is close to  $p_T$ , and so  $f$  should not be far from the identity map. For example,  $R$  might be a simulation of data  $T$ , or  $R$  might be close to  $T$  in phase space. In order to assess how well-suited normalizing flows are for this case, we also study how much  $x$  is moved via the morphing. An effective morphing map need not move the features minimally, but models that include this inductive bias may be more robust than those that do not. There is also a connection with optimal transport, which would be exciting to study in the future.

This paper is organized as follows. Section II briefly reviews normalizing flows and introduces all of the flows for flows variations we study. Next, Sec. III presents a simple application of the flows for flows variations on two-dimensional synthetic datasets. Sec. IV gives a more realistic application of the transport variations to sets of simulated particle collision data. We summarize the results and conclude in Sec. V.

## II. METHODS

### A. Normalizing flows as transfer functions

Normalizing flows are classically defined by a parametric diffeomorphism  $f_\phi$  and a base density  $p_\theta$  for which the density is known. Using the change of variables formula, the log likelihood (parametrized by both  $\theta$  and  $\phi$ ) of a data point  $x \sim p_D$  under a normalizing flow is given by

$$\log p_{\theta,\phi}(x) = \log p_\theta(f_\phi^{-1}(x)) - \log |\det(J_{f_\phi^{-1}(x)})|, \quad (1)$$

where  $J$  is the Jacobian of  $f_\phi$ . Training the model to maximize the likelihood of data samples results in a map  $f_\phi^{-1}$  between the data distribution  $p_D(x)$  and the base density  $p_\theta$ . As the base density should have a known distribution, it is usually taken to be a normal distribution of the same dimensionality as the data (which motivates the name ‘‘normalizing’’ flow).

At this point, we can introduce the first transfer method from a reference distribution  $p_R$  to a target distribution  $p_T$ , the base transfer. For this method, we train two normalizing flows with two different maps from the same base density. If  $f_{\phi_1}$  constitutes a map to the reference density  $p_R$  and  $f_{\phi_2}$  is a map to the target density  $p_T$ , then the composition  $f_{\phi_2}^{-1} \circ f_{\phi_1}$  is a transfer map  $f: R \rightarrow T$ . In other words, the transfer method routes from reference to target via some base density intermediary.

It is also possible to use a learned base density, such as another normalizing flow, instead of some known base distribution. This is our second method, unidirectional transfer. Given samples from two data distributions  $p_R$  and  $p_T$  of the same dimensionality, a map  $f_\gamma: R \rightarrow T$  between these distributions can be found by estimating a density  $p_{\phi,R}$  for  $R$  to use as the base density in the construction of another normalizing flow. In practice, this involves first training a normalizing flow to learn the density  $p_R$  by constructing the map  $f_\phi^{-1}$  from a base density  $p_\theta$  to  $p_R$ .

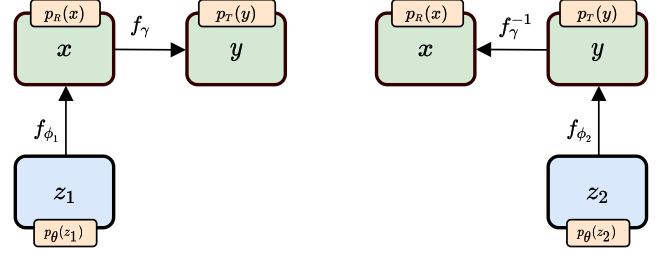


FIG. 1. A schematic of the flows for flows architecture.

Training of the two normalizing flows (the first for the base density, the second for the transport) is done by maximizing the log-likelihood of the data under the densities defined by the change of variables formula and given by

$$\begin{aligned} & \max_{\gamma} \mathbb{E}_{y \sim p_T} [\log p_{\theta,\phi,\gamma}(y)] \\ & = \max_{\gamma} \mathbb{E}_{y \sim p_T} [\log p_{\theta,\phi}(f_\gamma^{-1}(y)) - \log |\det(J_{f_\gamma^{-1}(y)})|], \\ & \max_{\phi} \mathbb{E}_{x \sim p_R} [\log p_{\theta,\phi}(x)] \\ & = \max_{\phi} \mathbb{E}_{x \sim p_R} [\log p_\theta(f_\phi^{-1}(x)) - \log |\det(J_{f_\phi^{-1}(x)})|]. \end{aligned}$$

A direct extension of the unidirectional training method is given by defining densities on both the reference and the target distributions,  $p_{\theta_1,R}$  and  $p_{\theta_2,T}$  which allows both  $f_\gamma$  and  $f_\gamma^{-1}$  to be explicitly used by training in both directions, from  $R$  to  $T$  and  $T$  to  $R$ . This comprises our third transfer method, flows for flows. A benefit of training in both directions is that the dependence of  $f_\gamma$  on the defined and learned densities  $p_{\theta_1,R}$  and  $p_{\theta_2,T}$  is reduced. A schematic of the flows for flows architecture is shown in Fig. 1.

The invertible network  $f_\gamma$  that is used to map between the two distributions may not have semantic meaning on its own, as some invertible neural networks are known to be universal function approximators. This map can become interpretable if it is subject to additional constraints. In this work, we investigate two physically motivated modifications to the flow training procedure; movement penalty, where we add an L1 loss term to the flow training, and identity initialization, where we initialize the flow architecture to the identity function. The L1 variation directly penalizes the average absolute value of the distance moved, while the idea for the identity initialization is that the model will converge on the

TABLE I. We consider five transfer methods from a reference dataset to a target dataset with unknown distributions  $p_R$  and  $p_T$ .

Method name	Training heuristic
Base transfer	$p_R \rightarrow \mathcal{N}(0, 1) \rightarrow p_T$
Unidirectional transfer	$p_R \rightarrow p_T$
Flows for flows	$p_R \leftrightarrow p_T$
Movement penalty	$p_R \overset{+L1}{\leftrightarrow} p_T$
Identity initialization	$p_R(\mathbb{I} + \epsilon \leftrightarrow) p_T$

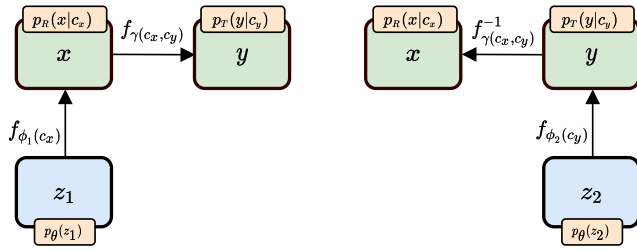


FIG. 2. Schematic of a conditional flows for flows architecture.

first best solution that gives close to no movement. All five transfer methods introduced in this section are summarized in Table I.

This entire setup can be made conditional by making the parameters of the invertible neural network dependent on some selected parameter (i.e., the “condition”). The log-likelihood for a normalizing flow conditioned on some variables  $c$  is defined by

$$\log p_{\theta, \phi}(x|c) = \log p_{\theta}(f_{\phi}^{-1}(x)|c) - \log |\det(J_{f_{\phi}^{-1}}(x))|, \quad (2)$$

where the base density can also be conditionally dependent on  $c$ . In the case of conditional distributions with continuous conditions, the distributions on data  $p_D(x|c)$  will often change smoothly as a function of the condition. For these situations, a flow that is explicitly parameterized by a well-motivated choice of conditioning variable may have a cleaner physical interpretation. We provide an example of such a flow for our application to particle collision datasets in Sec. IV. In particular, conditional flows have been used often in high-energy physics to develop “bump hunt” algorithms to search for new particles [5–7, 10–12]. In such studies, the resulting flows perform well when interpolated to values of the conditioning variable not used in training.

A schematic of a conditional flows for flows model is shown in Fig. 2, where the conditioning function  $f_{\gamma(c_x, c_y)}$  can also take more restrictive forms, such as  $f_{\gamma(c_x - c_y)}$  to ensure that the learned map is simple [5, 7]. Furthermore, the two conditional base distributions can be identical such

that  $\phi_1 = \phi_2$ . Alternatively, the base distributions can be different and instead a shared condition can be used such that  $c = c_x = c_y$ .

## B. Network architecture

Throughout this work, we use two different flow architectures, one for the “standard” normalizing flow architecture (i.e., learning transformations from standard normal distributions to arbitrary distributions) and one for the flows for flows architecture (i.e., learning transformations between two nontrivial distributions).

For the former architecture type, the invertible neural networks are constructed from rational quadratic splines with four autoregressive (AR) layers [13]. Each spline transformation has eight bins and the parameters of the spline are defined using masked AR networks with two blocks and 128 nodes as defined in the `NFLOWS` package [14]. For the latter architecture type, we use eight AR layers with splines of eight bins from three masked AR blocks of 128 nodes. This slightly more complex architecture is found to give better performance for the large shifts between the toy distributions that we consider. However, in cases where the reference and the target distributions are similar to each other, the architecture of the flows for flows model could in principle be simplified for faster training time while maintaining good performance.

An initial learning rate of  $10^{-4}$  is annealed to zero following a cosine schedule [15] over 60 epochs for the first flow type and 64 epochs for the second flow type. All trainings use a batch size of 128 and the norm of the gradients is clipped to five. For the toy distribution analyses in Sec. III, the training datasets all contain  $10^6$  samples.

## III. TOY EXAMPLE RESULTS

In this section, we explore the performance of the five transfer methods for learning a mapping between nontrivial two-dimensional distributions. In general, we consider both the accuracy of the transform, i.e., does the transfer method learn to successfully morph between the reference and the

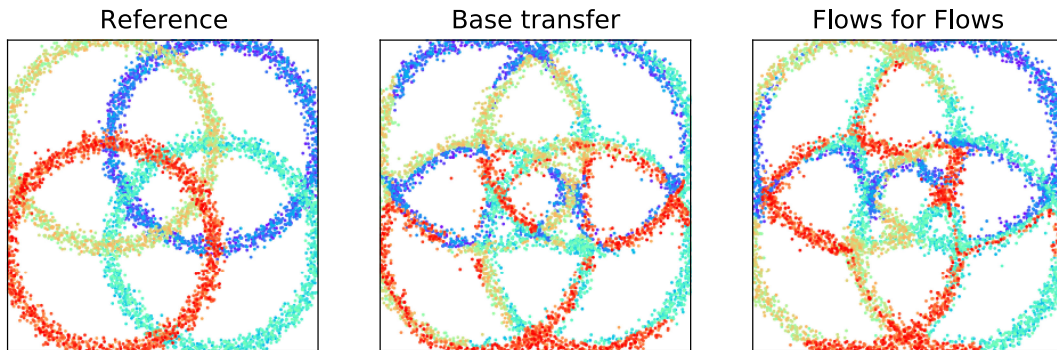


FIG. 3. Transport task between two instantiations of the same distribution. The first column shows the reference distribution; the second column shows the base transfer method acting on the reference distribution; the third column shows the flows for flows method. Individual samples have been color coded so as to make clear their paths assigned by the transport method.

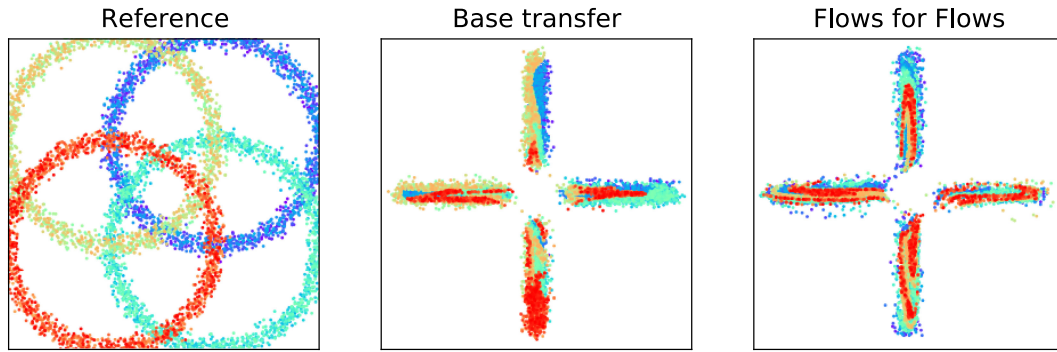


FIG. 4. Transport task between two different distributions. Individual samples have been color coded so as to make clear their paths assigned by the transport method.

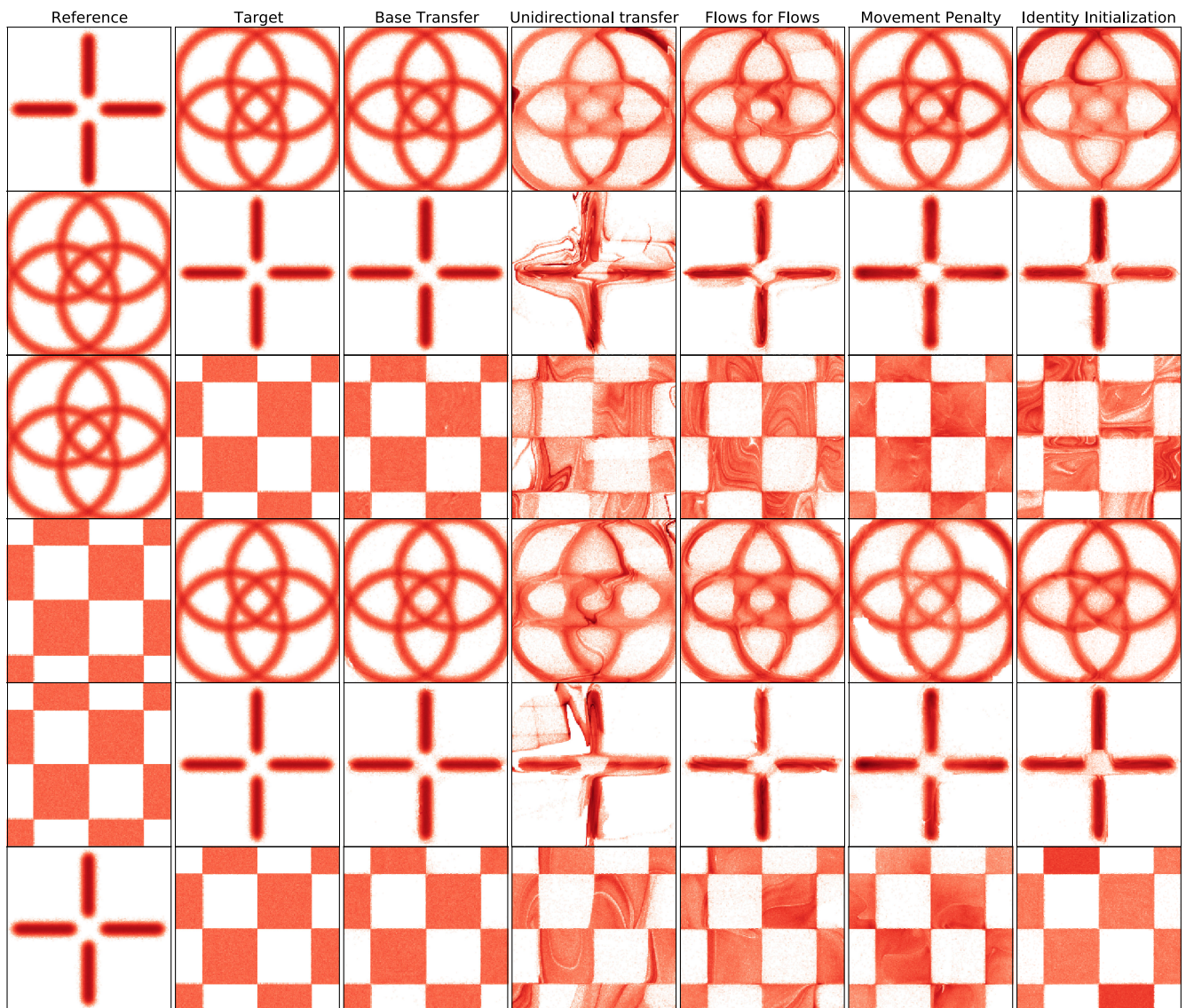


FIG. 5. Transport tasks between various choices of nonidentical reference and target toy distributions. The color bar has been set to scale logarithmically, which can emphasize out-of-distribution points.

target distribution, and the efficiency of the transform, i.e., does the method learn a morphing that is logical, and not unnecessarily circuitous.

### A. Base transfer vs flows for flows

In Fig. 3 we show a transport task between two datasets drawn from a toy distribution of four overlapping circles. Here we are in some sense trying to learn the identity mapping. We compare the action of the base transfer, which can be seen as the “default” method of mapping between two nontrivial distributions, against the flows for flows method. Both methods are able to successfully map the overall shape of the reference to the target distribution. However, the base transfer method tends not to keep points in the same circle when mapping them from reference to target, while the flows for flows method is more successful at keeping larger portions of each ring together.

In Fig. 4, we show a transport task between two different distributions, from four overlapping circles to a four-pointed star. As before, both the base transfer and flows for flows methods are able to morph the shape of the reference distribution into the shape of the target distribution. Interestingly, the flows for flows method appears to distribute points from each of the four circles more equally among each point of the star.

### B. Evaluating multiple transfer methods

In Fig. 5, we evaluate just the shape-morphing ability of the transfer methods. We consider six reference-target pairings, where the reference and target distributions are different,<sup>1</sup> and show the action of the base transfer, unidirectional transfer, flows for flows, movement penalty, and identity initialization methods on the reference distribution. We consider transports between three toy distribution types; four overlapping circles, a four-pointed star, and a checkerboard pattern. All of the transfer methods considered are able to successfully learn to map from reference to target, except for the unidirectional transfer, which exhibits a large amount of smearing in the final distribution. Overall, the base transfer, movement penalty, and identity initialization methods show the cleanest final-state distributions.

Another useful metric is the distance traveled by a sample that is mapped under a flow action. For many physical applications, a map that moves data the least is ideal, but we have only explicitly added an L1 loss term to the movement penalty method. Therefore, it is interesting to consider how far, on average, all the methods move the features.

In Fig. 6, we show a histogram of the distances traveled, amassing all of the six transfer tasks shown in the rows of

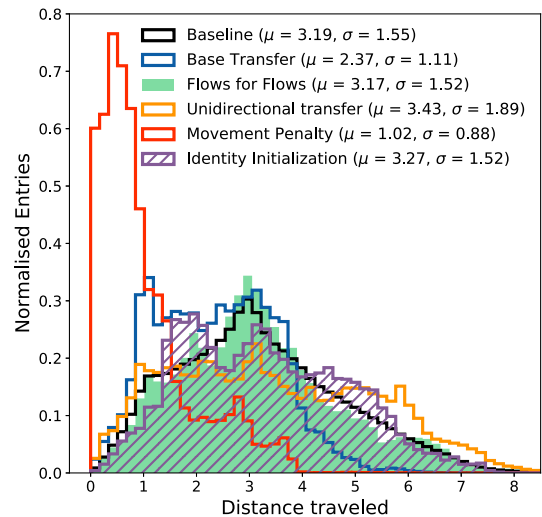


FIG. 6. Distances traveled in parameter space between two nonidentical toy distributions. Each histogram compiles data from six transfer tasks, corresponding to the rows of Fig. 5. The “baseline” method shows the distances between two random, unrelated instantiations of each reference; target distribution pairing. The maximum possible distance travelable in parameter space is 11.31.

Fig. 5 so as to equalize over many types of starting and target shapes. The movement penalty method performs best, producing the shortest distances traveled from reference to target by a large margin compared with the other methods. Interestingly, the flows for flows and identity initialization methods have larger mean distances traveled than the base transfer method as well as larger standard deviations. This is somewhat counterintuitive given that the base transfer method does not explicitly link the reference and target distributions during the training procedure, but it may reflect the somewhat contrived nature of the toy examples (especially in light of the more intuitive results for the science datasets shown in Fig. 9). All methods except the unidirectional transfer perform more optimally than or on par with the expected baseline, which comes from computing the distances between two random, unrelated instantiations of each reference-target distribution pairing.

## IV. APPLICATION: CALIBRATING COLLIDER DATASETS

We now move to a physical example: mapping between distributions of scientific observables. Many analyses of collider data are geared towards finding evidence of new physics processes. One powerful search strategy is to compare a set of detected data with an auxiliary dataset, where the auxiliary dataset is known to contain Standard Model-only physics. Any nontrivial difference between the detected and the auxiliary datasets could then be taken as evidence for the existence of new physical phenomena.

<sup>1</sup>For results corresponding to transports between identical distributions, see Appendix A.

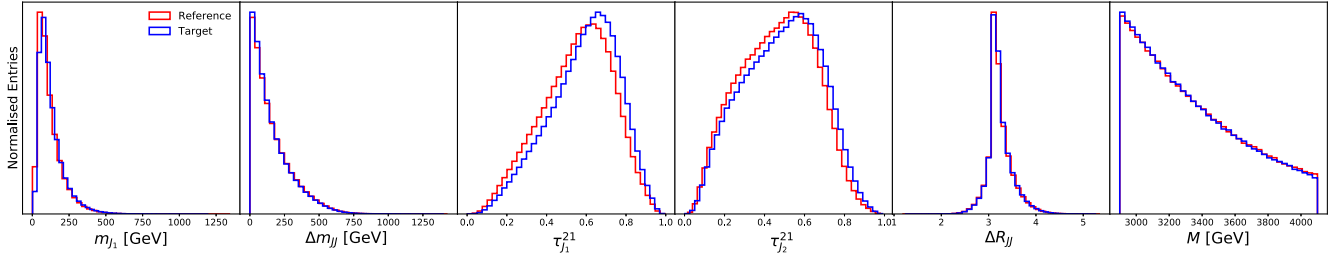


FIG. 7. Reference and target distributions used in the application of the flows for flows procedure to scientific datasets. The feature space is comprised of the resonant feature  $M$  and five other features  $m_{J_1}$ ,  $\Delta m_{JJ}$ ,  $\tau_{J_1}^{21}$ ,  $\tau_{J_2}^{21}$ , and  $\Delta R_{JJ}$ . A description of these observables can be found in [23]. The signal region is defined by  $|M - M_0| < c$  for  $M_0 = 3.5$  TeV and  $c = 200$  GeV.

The above analysis procedure is contingent upon the auxiliary dataset being a high-fidelity representation of Standard Model physics. However, such an assumption is not true for many datasets that would be, at first glance, ideal candidates for the auxiliary dataset, such as simulation of Standard Model processes or detected data from adjacent regions of phase space. Therefore it is necessary to calibrate the auxiliary dataset such that it becomes ideal. Historically, 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 alternatives. Recently, machine learning has enabled these approaches to be extended to the case of many dimensions and/or no binning—see e.g. Ref. [16] for a review.

With the flows for flows method, we can consider yet another calibration approach; to create an ideal auxiliary dataset (the target) by morphing the features from a less-ideal, imperfect auxiliary dataset (the reference). When the imperfect auxiliary dataset is chosen to be close to the ideal reference dataset, as would be true of the candidates listed in the previous paragraph, then the flows for flows method should simply be a perturbation on the identity map.<sup>2</sup>

### A. Analysis procedure and dataset

We focus on the problem of resonant anomaly detection, which assumes that given a resonant feature  $M$ , a potential new particle will have  $|M - M_0| \lesssim s$  (which defines the signal region) for some unknown  $M_0$  and often knowable  $s$  [17]. The value of  $M_0$ , which corresponds to the mass of the new particle, can be derived from theoretical assumptions on the model of new physics or can be found through a scan. Additional features  $X \in \mathbb{R}^N$  are chosen which can be used to distinguish the signal (the new particle) from background (Standard Model-like collisions), which can

be done by comparing detected data with reference data within the signal region.

For our datasets, we use the LHC 2020 Olympics dataset [18,19] which consists of a large number ( $\sim 10^6$ ) Standard Model simulation events. The events naturally live in a high-dimensional space, as each contains hundreds of particles with momenta in the  $x$ ,  $y$ , and  $z$  directions. To reduce their dimensionality, the events are clustered into collimated sprays of particles called *jets* using the `FastJet` [20,21] package with the anti- $k_t$  algorithm [22] ( $R = 1$ ). From these jets, we can pull a compressed feature space of only five dimensions; this set of features has been extensively studied in collider analyses. The jet features, along with the resonant feature  $M$ , are displayed in Fig. 7. We take the band  $M \in [3.3, 3.7]$  TeV as our signal region.

The LHC Olympics dataset contains two sets of Standard Model data generated from the different simulation toolkits `PYTHIA 8.219` [24,25] and `HERWIG++` [26]. We use the former as a stand-in for detected collider data. The latter is used as the reference dataset, the less than ideal auxiliary dataset that is calibrated through the flows for flows method to form the ideal auxiliary, target dataset.

To construct the ideal auxiliary dataset, we train a flow to learn the mapping between the reference dataset and the target data outside of the signal region, so as to keep the signal region blinded. Once trained, the flow can then be applied to the nonideal auxiliary dataset within the signal region, thus constructing the ideal auxiliary dataset. We use the same architectures as in Sec. II B, with the modification that we *condition* the transport flows on the mass feature  $M$ . This conditioning is motivated by the fact that the flow is trained outside the signal region and applied within the signal region, which is defined exactly by the variable  $M$ .

### B. Results

In Fig. 8, we provide the distributions of the flow-moved reference dataset to the target dataset, as well as the ratios to the target, outside of the signal region. As is clear from Fig. 7, the reference and target datasets are far more similar in this calibration example than they were in the toy examples. Therefore for the movement penalty method,

<sup>2</sup>This procedure is the underlying motivation for the flow-enhanced Transportation for anomaly detection method [6]. Equally, the ideal and reference datasets could be defined using the same dataset but selecting different conditional values. This method of flows for flows is used to train the constructing unobserved regions with maximum likelihood estimation method [7] and is studied for toy datasets in Appendix B.

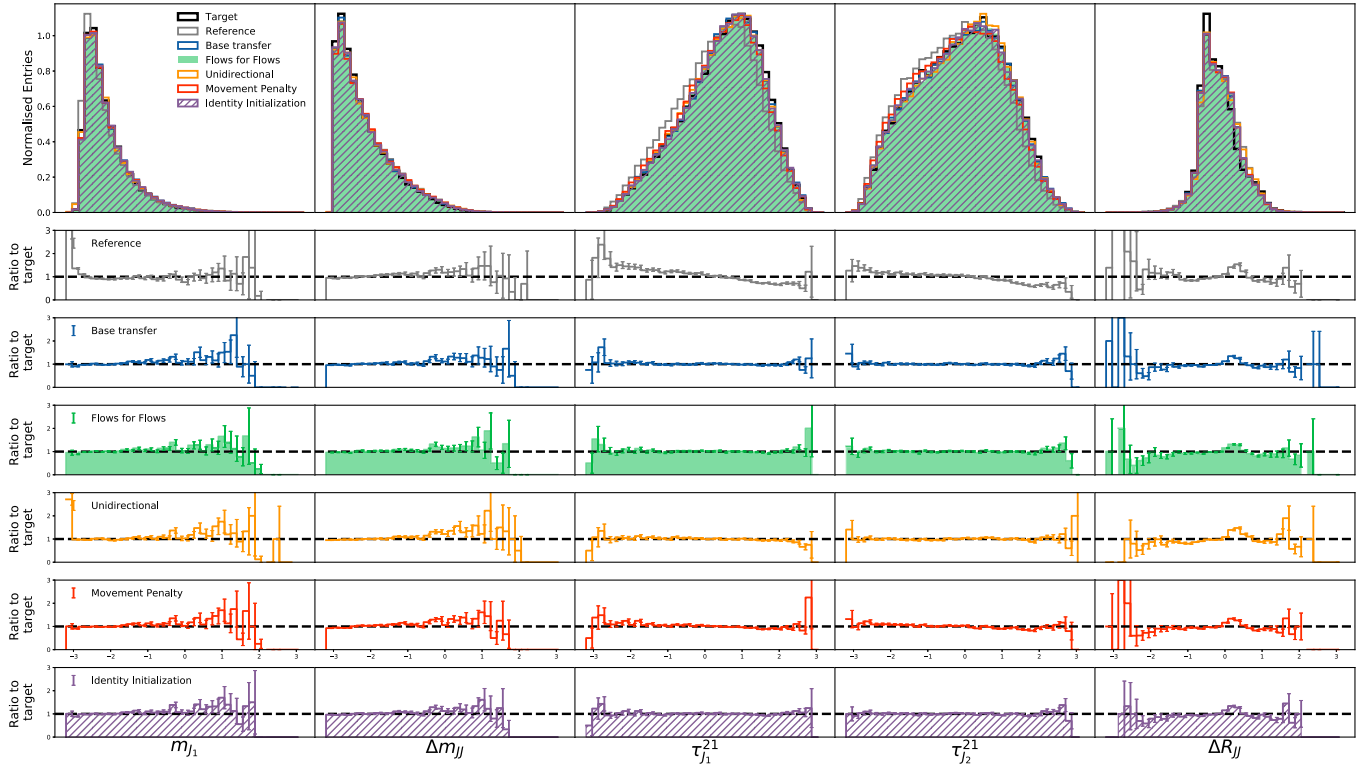


FIG. 8. Distributions of and ratios of the flow-transported reference (less than ideal auxiliary) dataset to the target (ideal auxiliary) dataset. Ratios are taken over each of the five marginal distributions in the parameter space; errorbars represent Poisson uncertainties in bin counts. All data is taken outside of the signal region. All features have been individually minmaxscaled to the range  $[-3, 3]$  to optimize flow training.

it was necessary to scan over the strength of the L1 term added to the training loss in order to achieve good performance; we found that we needed to reduce the strength by a factor of 20, as compared with what was used for the toy distributions. In fact, all five transfer methods (base transfer, unidirectional transfer, flows for flows, movement penalty, and identity initialization) perform comparably, and all five methods are able to successfully transform the reference dataset such that the five marginal feature distributions greatly resemble those of the target.

In Fig. 9, we show a histogram of the distances traveled for each data point due to the flow action. Distributions for distance traveled in each individual dimension of feature space are given in Fig. 10. Since the reference and target distributions are so similar, the base transfer methods leads to a highly nonminimal transport path. While the unidirectional method performs well, it shows a longer tail in distance traveled that may represent a less than ideal mapping. The flows for flows and identity initialization methods perform comparably with relatively little distance traveled, while movement penalty appears to have found a nearly minimal path.

Based on the closeness of the distributions of the reference and target in Fig. 7, we might hope for a mapping that morphs features  $m_{J_1}$ ,  $\Delta m_{JJ}$ , and  $\Delta R_{JJ}$  almost not at all,

and features  $\tau_{J_1}^{21}$  and  $\tau_{J_2}^{21}$  very minimally. Indeed, this is exactly the behavior we see in Fig. 10 for the movement penalty method (and, to a lesser extent, for the flows for flows and identity initialization methods).

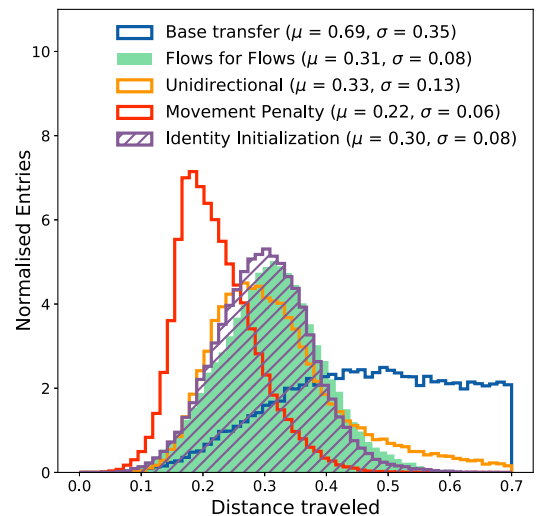


FIG. 9. Distances traveled in five-dimensional parameter space between the reference (less-than ideal auxiliary) dataset to the target (ideal auxiliary) dataset, outside of the signal region. The maximum possible distance travelable in parameter space (for the minmaxscaled features) is 13.41.



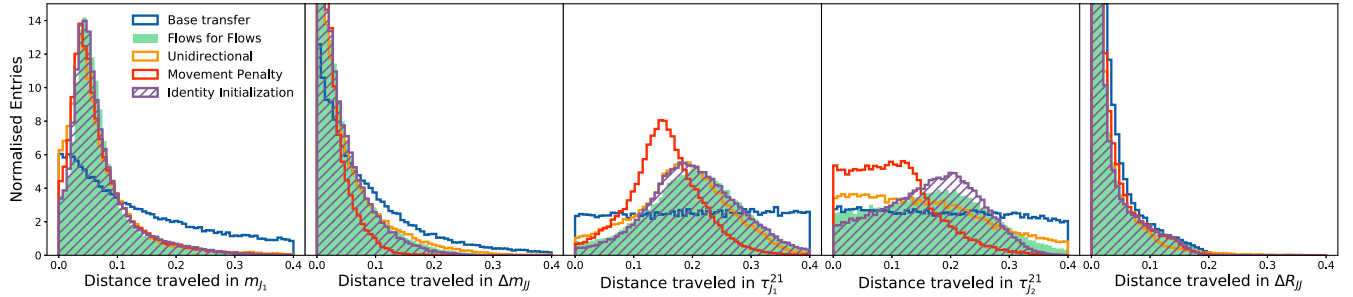


FIG. 10. Distances traveled in each dimension of the five-dimensional parameter space between the reference (less than ideal auxiliary) dataset to the target (ideal auxiliary) dataset, outside of the signal region. The maximum possible distance travelable in each dimension of parameter space (for the minmaxscaled features) is 6.

## V. CONCLUSIONS AND FUTURE WORK

In this work, we have explored a number of ways to use normalizing flows to create mappings between nontrivial reference and target datasets of the same dimensionality. Our aim is to consider methods that go above and beyond the “naive” base transfer method, which uses standard normalizing flows that map from reference to target via a base density intermediary. In particular, we have introduced the flows for flows method, which uses two normalizing flows to parametrize the probability densities of both the reference and the target and trains both with exact maximum likelihoods.

We have evaluated five transfer methods; base transfer, unidirectional transfer, flows for flows, movement penalty, and identity initialization. We have attempted to evaluate each method on two facets; the accuracy of the transport between reference and target, and the efficiency of the transport (i.e., how far away are points transported by the mapping). When the reference and target are fully unrelated (such as for the toy examples in Sec. III), the flows for flows method is comparable with the naive base transfer method both for accuracy and extent. When the reference and target sets are similar, or obviously related in some way (such as for the particle physics calibration application in Sec. IV), the flows for flows method is far preferable to the base transfer method. These results imply that the flows for flows method should be used over the base transfer method, as it can always provide both an accurate and efficient transport. However, the highest performing (and thus our recommended) methods of transport are either the movement penalty or identity initialization methods, depending on the specific application.

There are many avenues for further modifications of the flows for flows method, or other ways to construct flow-based mapping functions in general. One interesting avenue involves physically-motivated extensions of normalizing flows: continuous normalizing flows (CNF) [27] constrain the flow mappings such that they can be assigned velocity vectors, and convex potential (CP) flows [28] constrain the map to be the gradient of a convex potential. One can explicitly enforce optimal transports with OT-Flows [29], which add to the CNF loss both an L2 movement penalty

and a penalty that encourages the mapping to transport points along the minimum of some potential function. While such modifications may not be necessary when the reference and target distributions are very similar, they could be explored for situations when the reference and target distributions are significantly different.

Looking towards applications that are different to that explored in this work, it would be interesting to use the flows for flows methods in high-dimensional physical analyses. While our science example used a six-dimensional, high-level subspace of dijet production variables, the fact that flow-based models are well-suited for modeling higher-dimensional datasets means that they could be used on lower-level event variables, such as particle 4-momenta. While the standard flow architecture is not permutation-invariant variants exist that can account for this [30,31]. Further, the flows for flows method could conceivably be used for tasks such as jet-flavor tagging, where some input variables are integers. Here, similar to other applications with standard normalizing flows, a dequantization step [32] would first need to be applied to such variables for training and evaluation, then reversed after inference.

The flows for flows package can be found at <https://github.com/jrairie/flows4flows>. J. R. and S. K. contributed equally to its creation.

## ACKNOWLEDGMENTS

T. G., S. K., and J. R. would like to acknowledge funding through the SNSF Sinergia grant called Robust Deep Density Models for High-Energy Particle Physics and Solar Flare Analysis (RODEM) with Funding No. CRSII5\_193716. B. N. and R. M. are supported by the U.S. Department of Energy (DOE), Office of Science under Contract No. DE-AC02-05CH11231. R. M. is additionally 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.

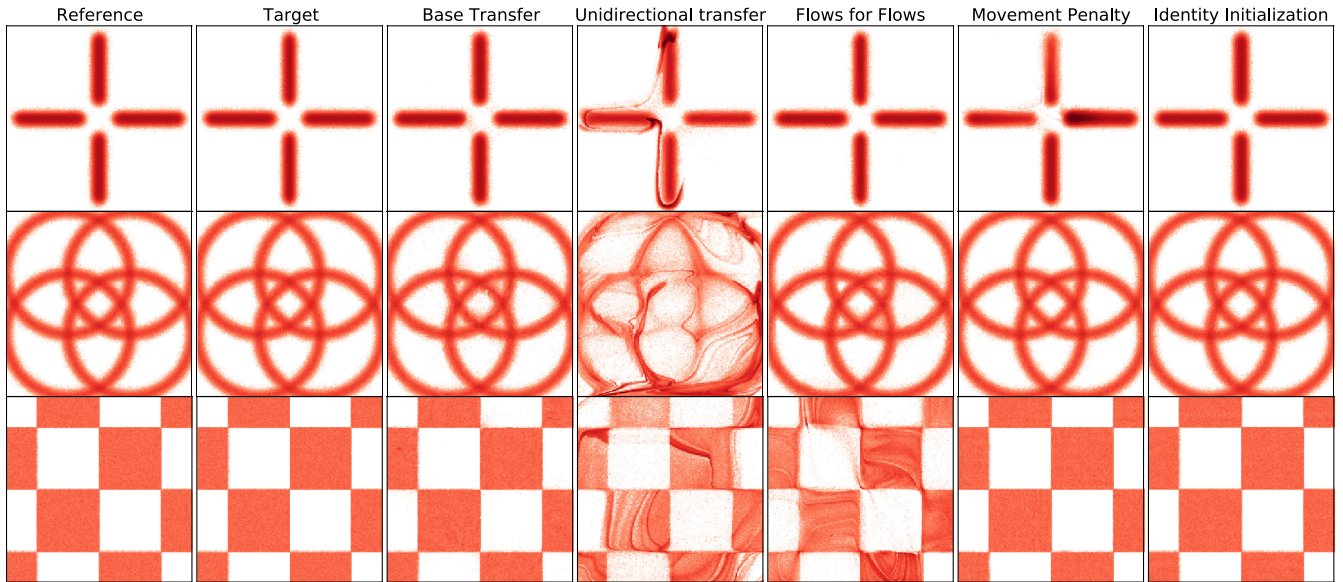


FIG. 11. Transport tasks between various choices of identical reference and target toy distributions. The color bar has been set to scale logarithmically, which can emphasize out-of-distribution points.

### APPENDIX A: TOY RESULTS: LEARNING THE IDENTITY

In Sec. III, we focused on mappings between a non-identical reference and target distribution. However, it is possible to get a more meaningful interpretation of a given transport method by considering an identical reference and target; in other words, when we ask the flow to learn the identity mapping. In this situation, it would be desirable for the flow to learn to transport data points as little as possible.

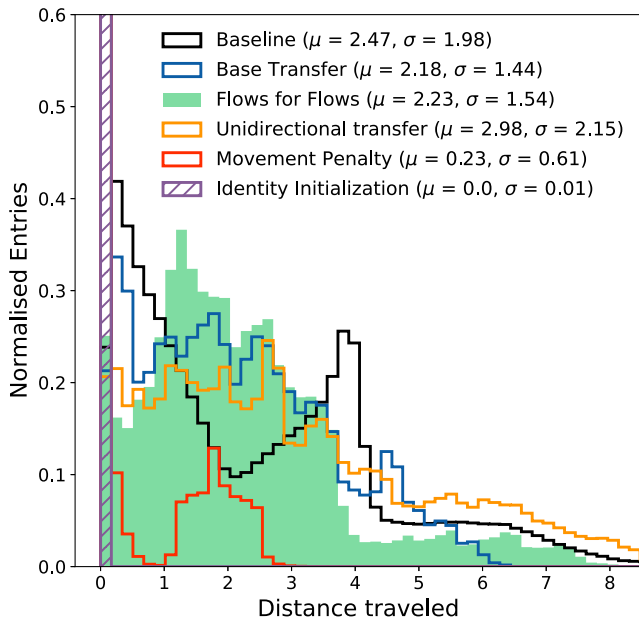


FIG. 12. Distances traveled in parameter space between two identical toy distributions. Each histogram contains data from three transfer tasks, corresponding to the rows of Fig. 11.

In Fig. 11, we plot the transport results between three identical reference-target pairings (four overlapping circles, a four-pointed star, and a checkerboard pattern). All of the transfer methods shown are able to successfully learn to map from reference to target, except the unidirectional transfer, which fails glaringly. The base transfer, movement penalty, and identity initialization methods show the cleanest final-state distributions.

In Fig. 12, we provide a compiled histogram of the distances traveled for the three transfer tasks shown in the rows of Fig. 11. In this case, the identity initialization method performs ideally, leaving the reference distribution virtually unchanged. The movement penalty method also performs well. The base transfer and flows for flows method are comparable and perform about as well as, if not slightly better than, the baseline (as was the case for nonidentical reference and target distributions); the unidirectional method performs suboptimally.

### APPENDIX B: CONDITIONAL TOY DISTRIBUTIONS

In this appendix, we provide plots similar to those in Sec. III but for conditional normalizing flows (as given in Sec. II A). We extend the two-dimensional toy distributions that have been studied so far by introducing rotations. A flows for flows model can then learn to move points sampled at one value of the condition such that they follow the distribution defined by another value of the condition.

To train the flows for flows models for the conditional toy datasets; we first generate unconditional toy distributions. We then rotate each data point in the distribution  $x_i$  by a random angle  $\theta_i \in [7.5^\circ, 15^\circ, 22.5^\circ, 30^\circ, 37.5^\circ, 45^\circ]$ .

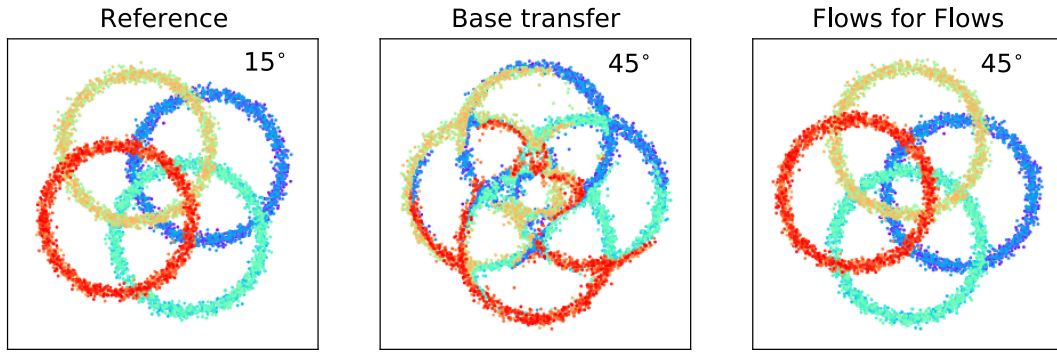


FIG. 13. Transport task between a choice of identical reference and target toy distributions. Individual samples have been color coded so as to make clear their paths assigned by the transport method. The conditioning rotation of each distribution is given in the top-right corner.

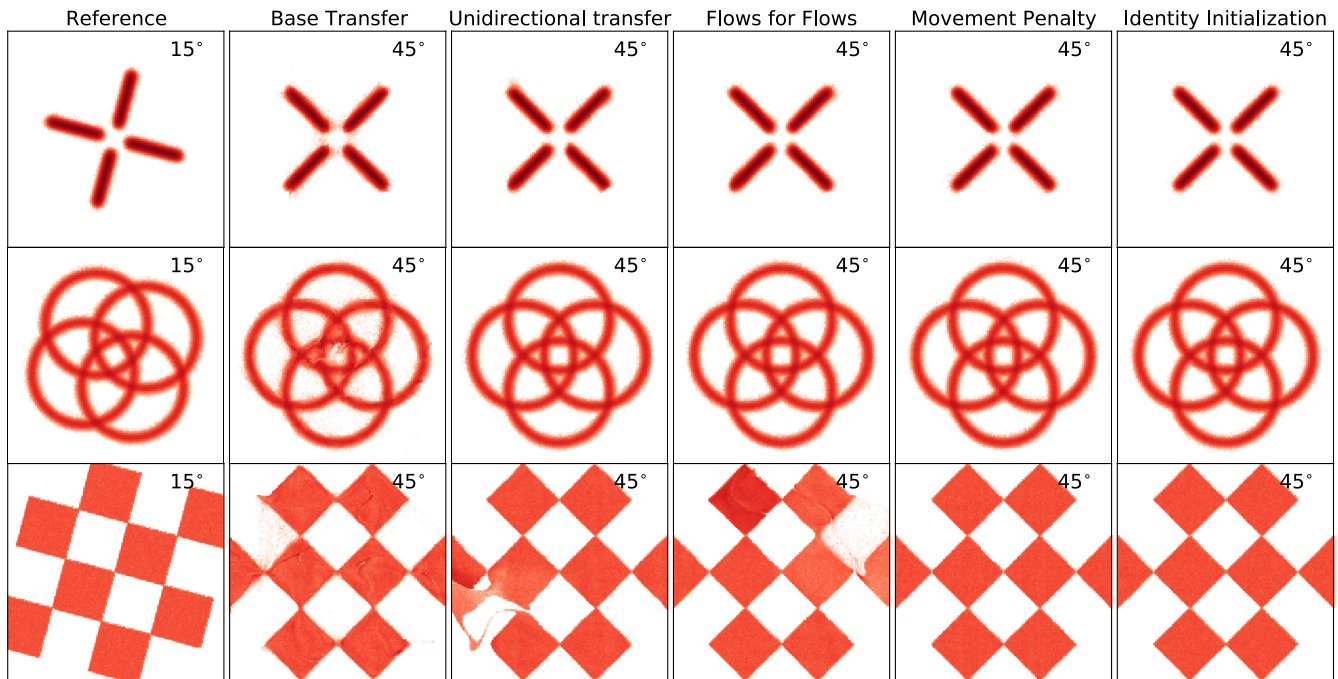


FIG. 14. Transport tasks between various choices of identical reference and target toy distributions. The color bar has been set to scale logarithmically, which can emphasize out-of-distribution points.

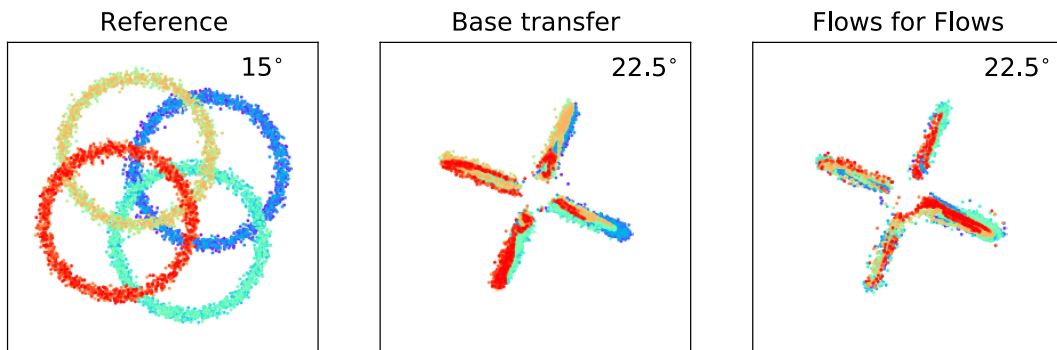


FIG. 15. Transport task between a nonidentical choice of reference and target toy distributions. Individual samples have been color coded so as to make clear their paths assigned by the transport method.

For a given reference-target data point pairing  $x_R, x_T$  used to train the flow, the condition is then set to be the difference between the training data's rotation angles,  $\theta_R - \theta_T$ . In all other ways, the training procedure is the same as that of the unconditional toy distributions.

The results for the conditional toy distribution transportation tasks tell a similar story as before, in Sec. III. For a transport task between identical toy distributions with different conditioning angles (shown in Fig. 13), the flows for flows method appears to minimally and logically

transport individual samples, in contrast to the base transport method. When looking more broadly at just the transport accuracy (shown in Fig. 14), the flows for flows-based methods (i.e., flows for flows, movement penalty, and identity initialization) are visibly better than the base transfer and unidirectional transfer.

For a transport task between nonidentical toy distributions (shown in Fig. 15), the flows for flows method performs far more similarly to the base transfer method with respect to dispersing samples from the rings to the

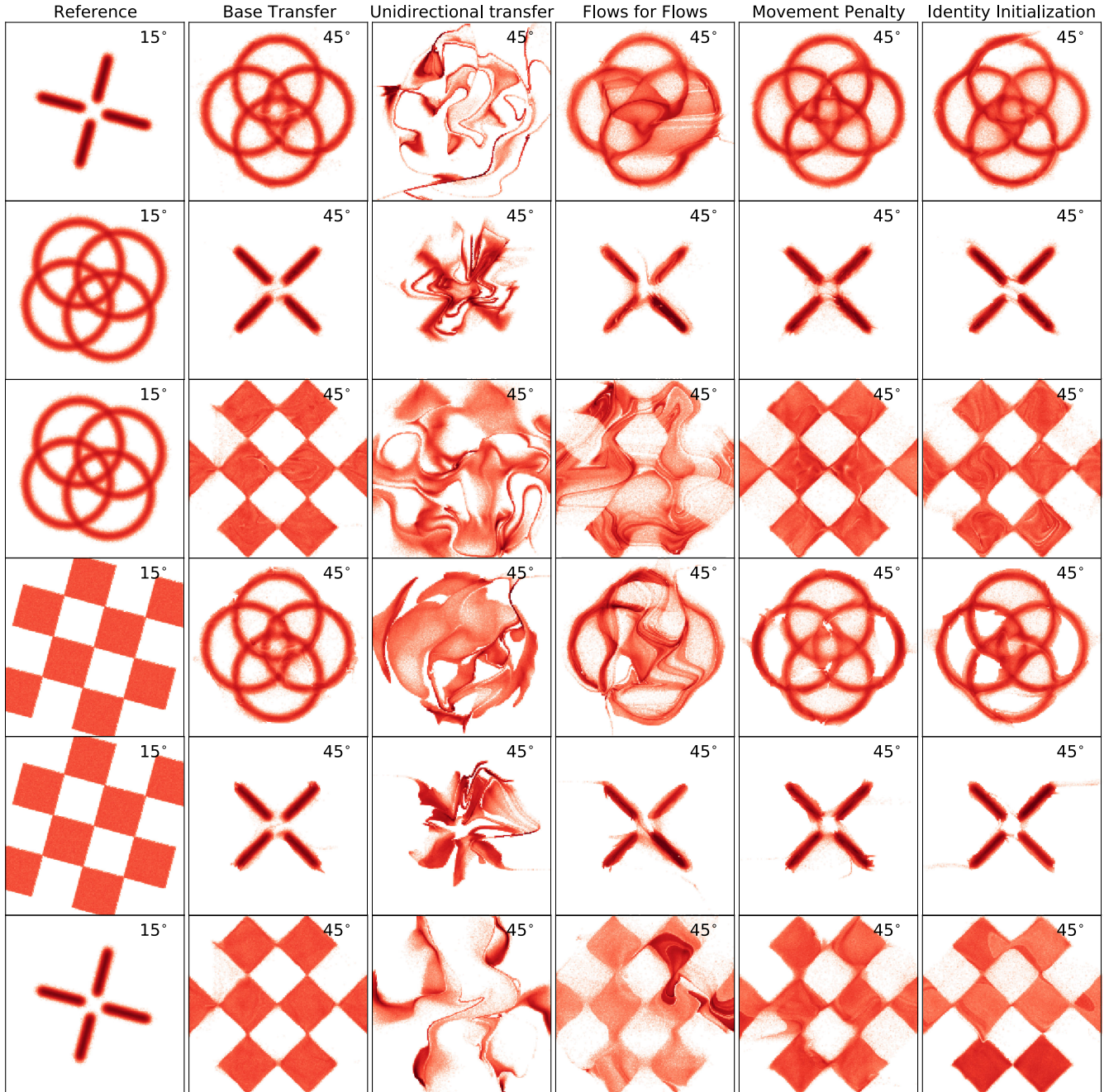


FIG. 16. Transport tasks between various choices of identical reference and target toy distributions. The color bar has been set to scale logarithmically, which can emphasize out-of-distribution points.

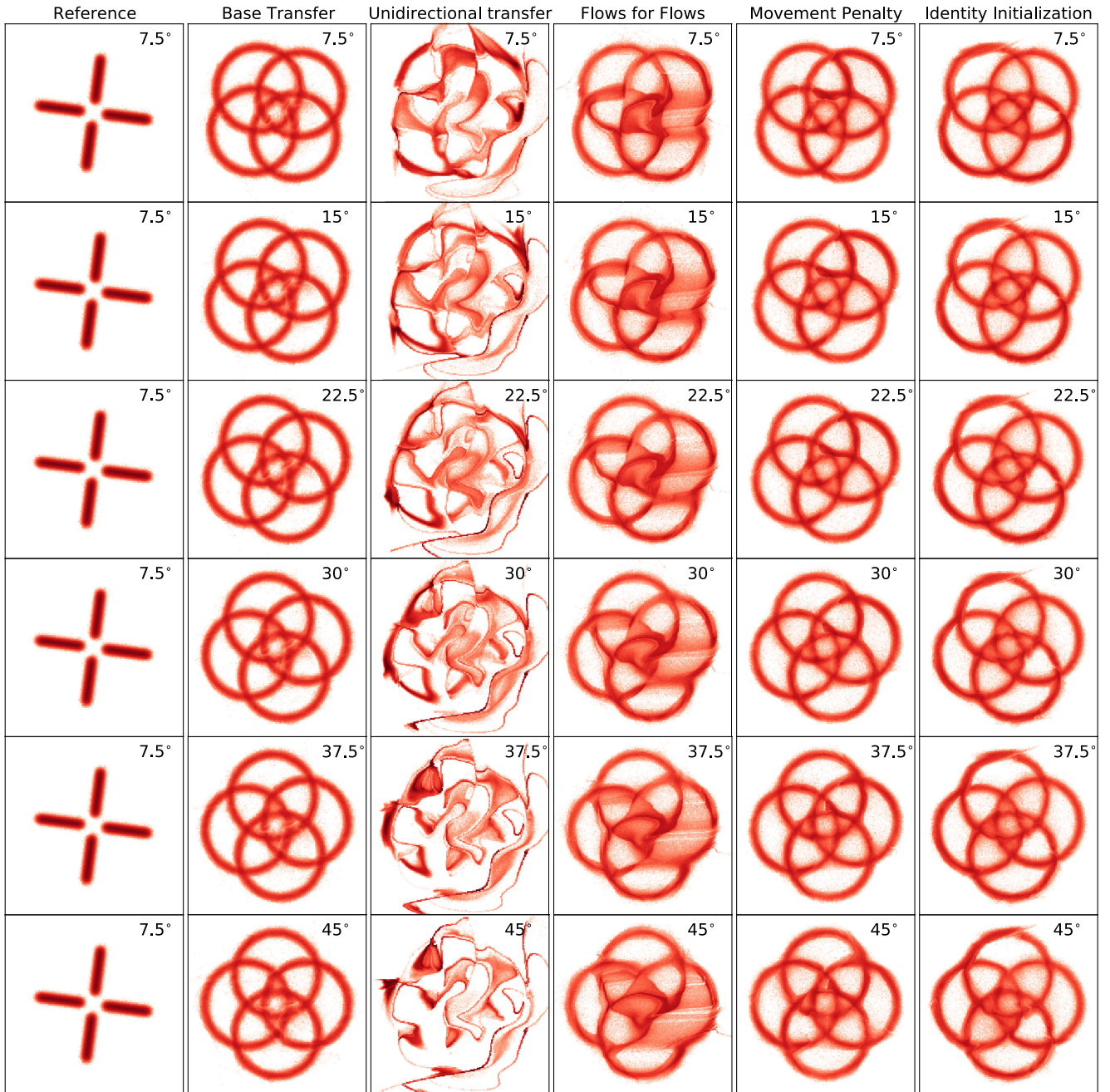


FIG. 17. Transport tasks between the fourcircles and star distributions at a variety of conditioning angles. The colorbar has been set to scale logarithmically, which can emphasize out-of-distribution points.

arms of the star. When considering just the distribution shapes (shown in Fig. 16), the unidirectional transfer performs extremely poorly, the flows for flows method performs next best, the movement penalty and identity initialization methods perform well, and the base transfer method shows the cleanest final state. Again, given the more intuitive performance rankings for the scientific dataset, it is likely that the superior performance of the

base transfer method is more a reflection of the contrived nature of the transport tasks between these toy distributions chosen.

In Fig. 17, we show again a series of transport tasks between two nonidentical toy distributions, but for a selection of conditioning angles. The choice of angle does not appear to affect the performance of each individual transport method.

- [1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics (Springer New York Inc., New York, NY, USA, 2001).
- [2] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density Ratio Estimation in Machine Learning* (Cambridge University Press, Cambridge, England, 2012).
- [3] E. G. Tabak and C. V. Turner, A family of nonparametric density estimation algorithms, *Commun. Pure Appl. Math.* **66**, 145 (2013).
- [4] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference [arXiv:1912.02762](https://arxiv.org/abs/1912.02762).
- [5] J. A. Raine, S. Klein, D. Sengupta, and T. Golling, CUR-TAINs for your sliding window: Constructing unobserved regions by transforming adjacent intervals, *Front. Big Data* **6**, 899345 (2023).
- [6] T. Golling, S. Klein, R. Mastandrea, and B. Nachman, Flow-enhanced transportation for anomaly detection, *Phys. Rev. D* **107**, 096025 (2023).
- [7] D. Sengupta, S. Klein, J. A. Raine, and T. Golling, CUR-TAINs flows for flows: Constructing unobserved regions with maximum likelihood estimation, [arXiv:2305.04646](https://arxiv.org/abs/2305.04646).
- [8] J. N. Howard, S. Mandt, D. Whiteson, and Y. Yang, Learning to simulate high energy particle collisions from unlabeled data, *Sci. Rep.* **12**, 7567 (2022).
- [9] S. Diefenbacher, V. Mikuni, and B. Nachman, Refining fast calorimeter simulations with a Schrödinger bridge, [arXiv:2308.12339](https://arxiv.org/abs/2308.12339).
- [10] B. Nachman and D. Shih, Anomaly detection with density estimation, *Phys. Rev. D* **101**, 075042 (2020).
- [11] G. Stein, U. Seljak, and B. Dai, Unsupervised in-distribution anomaly detection of new physics through conditional density estimation, in *Proceedings of the 34th Conference on Neural Information Processing Systems* (2020), [arXiv:2012.11638](https://arxiv.org/abs/2012.11638).
- [12] A. Hallin, J. Isaacson, G. Kasieczka, C. Krause, B. Nachman, T. Quadfasel, M. Schlaffer, D. Shih, and M. Sommerhalder, Classifying anomalies through outer density estimation (cathode), *Phys. Rev. D* **106**, 055006 (2022).
- [13] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, Neural spline flows, *Adv. Neural Inf. Process. Syst.* **32** (2019), [https://proceedings.neurips.cc/paper\\_files/paper/2019/hash/7ac71d433f282034e088473244df8c02-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2019/hash/7ac71d433f282034e088473244df8c02-Abstract.html).
- [14] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, NFLOWS: Normalizing flows in PyTorch, [Zenodo, 10.5281/zenodo.4296287](https://zenodo.org/record/4296287) (2020).
- [15] I. Loshchilov and F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, [arXiv:1608.03983](https://arxiv.org/abs/1608.03983).
- [16] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman, and D. Shih, Machine learning in the search for new fundamental physics, *Nat. Rev. Phys.* **4**, 399 (2021).
- [17] G. Kasieczka, B. Nachman, and D. Shih, New methods and datasets for group anomaly detection from fundamental physics, in *Proceedings of the Conference on Knowledge Discovery and Data Mining* (2021), [arXiv:2107.02821](https://arxiv.org/abs/2107.02821).
- [18] G. Kasieczka, B. Nachman, and D. Shih, Official datasets for the LHC Olympics 2020 anomaly detection challenge (Version v6) [Data set], [10.5281/zenodo.4536624](https://zenodo.org/record/4536624) (2019).
- [19] G. Kasieczka *et al.*, The LHC Olympics 2020 a community challenge for anomaly detection in high energy physics, *Rep. Prog. Phys.* **84**, 124201 (2021).
- [20] M. Cacciari, G. P. Salam, and G. Soyez, FastJet user manual, *Eur. Phys. J. C* **72**, 1896 (2012).
- [21] M. Cacciari and G. P. Salam, Dispelling the  $N^3$  myth for the  $k_t$  jet-finder, *Phys. Lett. B* **641**, 57 (2006).
- [22] M. Cacciari, G. P. Salam, and G. Soyez, The anti- $k_t$  jet clustering algorithm, *J. High Energy Phys.* **04** (2008) 063.
- [23] J. Thaler and K. Van Tilburg, Identifying boosted objects with N-subjettiness, *J. High Energy Phys.* **03** (2011) 015.
- [24] T. Sjöstrand, S. Mrenna, and P. Z. Skands, PYTHIA 6.4 physics and manual, *J. High Energy Phys.* **05** (2006) 026.
- [25] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, An introduction to PYTHIA 8.2, *Comput. Phys. Commun.* **191**, 159 (2015).
- [26] M. Bähr, S. Gieseke, M. A. Gigg, D. Grellscheid, K. Hamilton, O. Latunde-Dada, S. Plätzer, P. Richardson, M. H. Seymour, A. Sherstnev, and B. R. Webber, Herwig ++ physics and manual, *Eur. Phys. J. C* **58**, 639 (2008).
- [27] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, Neural ordinary differential equations (2018).
- [28] C.-W. Huang, R. T. Q. Chen, C. Tsirigotis, and A. Courville, Convex potential flows: Universal probability distributions with optimal transport and convex optimization, in *International Conference on Learning Representations* (2020), <https://openreview.net/forum?id=te7PVH1sPxJ>.
- [29] D. Onken, S. W. Fung, X. Li, and L. Ruthotto, Ot-flow: Fast and accurate continuous normalizing flows via optimal transport, *CoRR* abs/2006.00104 (2020), [arXiv:2006.00104](https://arxiv.org/abs/2006.00104).
- [30] D. Nielsen, P. Jaini, E. Hoogeboom, O. Winther, and M. Welling, Survae flows: Surjections to bridge the gap between VAEs and flows, *Adv. Neural Inf. Process. Syst.* **33**, 12685 (2020), [https://proceedings.neurips.cc/paper\\_files/paper/2020/hash/9578a63fbc545bd82cc5bbe749636af1-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2020/hash/9578a63fbc545bd82cc5bbe749636af1-Abstract.html).
- [31] R. Verheyen, Event generation and density estimation with surjective normalizing flows, *SciPost Phys.* **13**, 047 (2022).
- [32] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, Flow ++: Improving flow-based generative models with variational dequantization and architecture design, in *International Conference on Machine Learning* (PMLR, 2019), pp. 2722–2730.