scientific reports

OPEN

Learning to simulate high energy particle collisions from unlabeled data

Jessica N. Howard[1]✉, Stephan Mandt[2], Daniel Whiteson[1] & Yibo Yang[2]

In many scientific fields which rely on statistical inference, simulations are often used to map from theoretical models to experimental data, allowing scientists to test model predictions against experimental results. Experimental data is often reconstructed from indirect measurements causing the aggregate transformation from theoretical models to experimental data to be poorly-described analytically. Instead, numerical simulations are used at great computational cost. We introduce Optimal-Transport-based Unfolding and Simulation (OTUS), a fast simulator based on unsupervised machine-learning that is capable of predicting experimental data from theoretical models. Without the aid of current simulation information, OTUS trains a probabilistic autoencoder to transform directly between theoretical models and experimental data. Identifying the probabilistic autoencoder's latent space with the space of theoretical models causes the decoder network to become a fast, predictive simulator with the potential to replace current, computationally-costly simulators. Here, we provide proof-of-principle results on two particle physics examples, *Z*-boson and top-quark decays, but stress that OTUS can be widely applied to other fields.

From measuring masses of particles to deducing the likelihood of life elsewhere in the Universe, a common goal in analyzing scientific data is statistical inference—drawing conclusions about values of a theoretical model's parameters, θ, given observed data, x. The likelihood model of observed data, $p(x|\theta)$, is a central ingredient in both frequentist and Bayesian approaches to statistical inference; however, it is typically intractable, due to the complexity of a full probabilistic description of the data generation process. One way to circumvent this difficulty is to *simulate* experimental data for a given value of the theoretical parameters, θ, from which a probability model of the likelihood, $p(x|\theta)$, can be constructed and used for downstream statistical inference regarding θ. This is known as simulation-based inference, and has found application across scientific disciplines ranging from particle physics to cosmology[1].

However, traditional approaches to simulation, which attempt to faithfully model complex physical phenomena, can be computationally expensive—a limitation we aim to overcome in this work. In simulation-based inference, experimental data arising from a physical system typically depend on an initial configuration of the system, z, that is unobserved, or belonging to a *latent* space, while the parameters θ govern the underlying mechanistic model. In many cases, the transformation from the latent state to experimental data is non-trivial, involving complex physical interactions that cannot be described analytically, but can be *simulated* numerically by Monte-Carlo algorithms. In particle physics, for example, the parameters θ govern theoretical models that describe fundamental particle interactions. These fundamental interactions produce secondary particles, z, which are not directly observable and often transform in flight before passing through layers of detectors whose indirect measurements, x, can help reconstruct their identities and momenta. The transformation from the unobserved latent space, particles produced in the initial interaction, to the experimental data, is stochastic, governed by quantum mechanical randomness, and has no analytical description.

Instead, Monte-Carlo-based numerical simulations of in-flight and detection processes generate samples of possible experimental data for a given latent space configuration [2–6]. This approach is computationally expensive[5,6] because it requires the propagation and simulation of every individual particle, each creating subsequent showers of thousands of derivative particles. Additionally, these simulations contain hundreds of parameters which must be extemporaneously tuned to give reasonable results in control regions of the data where the latent space has been well-established by results from previous experiments.

In particle physics, like many other fields in the physical sciences[7–10], the computational cost of numerical simulations has become a central bottleneck. A fast, interpretable, flexible, data-driven generative model which

[1]Department of Physics and Astronomy, UC Irvine, Irvine, CA, USA. [2]Department of Computer Science, UC Irvine, Irvine, CA, USA. ✉email: jnhoward@uci.edu

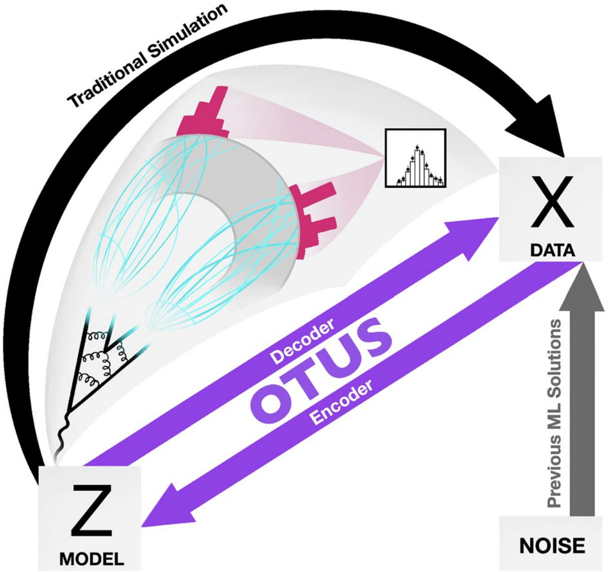

Figure 1. Schematic of the problem and the solution. Current simulations map from a physical latent space, $\mathcal{Z}$, to data space, $\mathcal{X}$, attempting to mimic the real physical processes at every step. This results in a computationally intensive simulation. Previous Machine Learning (ML) solutions can reproduce the distributions in $\mathcal{X}$ but are not conditioned on the information in $\mathcal{Z}$; instead they map from unphysical noise to $\mathcal{X}$, which limits their scope. We introduce a new method which provides the best of both worlds. OTUS provides a simulation $\mathcal{Z} \rightarrow \mathcal{X}$ (Decoder) which is conditioned on $\mathcal{Z}$ yet is computationally efficient. Advantageously, it also inadvertently provides an equivalently fast unfolding mapping from $\mathcal{X} \rightarrow \mathcal{Z}$ (Encoder).

can transform between the latent space and the experimental data would be significant for these fields. Recent advances in the flexibility and capability of machine learning (ML) models have allowed for their application as computationally inexpensive simulators[11–19]. Applications of these techniques have made progress towards this goal but fall short in crucial ways. For example, approaches leveraging Generative Adversarial Networks (GANs) are able to mimic experimental data for fixed distributions in the latent space[11–13], but are unable to generate predictions for new values of latent variables, a crucial requirement for a simulator. Other efforts condition on latent variables[14] but require training with labeled pairs generated by slow Monte-Carlo generators, incurring some of the computational cost they seek to avoid.

We lay the foundations and provide a proof-of-principle demonstration for Optimal-Transport-based Unfolding and Simulation (OTUS). We use unsupervised learning to build a flexible description of the transformation from latent space, $\mathcal{Z}$, to experimental data space, $\mathcal{X}$, relying on theoretical priors, $p(z)$, where $z \in \mathcal{Z}$ and a set of samples of experimental data $\{x \in \mathcal{X}\}$ but, crucially, no labeled pairs, (z, x). Our model applies a type of probabilistic autoencoder[20,21], which learns two mappings: encoder (data → latent, $p_E(z \mid x)$) and decoder (latent → data, $p_D(x \mid z)$). Typical probabilistic autoencoders (i.e. variational autoencoders (VAEs)[13,22]) use a simple, unphysical latent space, $\mathcal{Y}$, for computational tractability during learning. However, this causes VAEs to suffer from the same weakness as GANs: doomed to mimic the data distribution, $p(x)$, for a fixed physical latent space, $p(z)$, unless the model compromises to requiring expensive simulated pairs (e.g. a conditional VAE approach[23]). OTUS's innovation is to align the probabilistic autoencoder's latent space, $\mathcal{Y}$, with that of our inference task, $\mathcal{Z}$. With this change, our decoder becomes a computationally inexpensive, conditional simulator mapping $\mathcal{Z} \rightarrow \mathcal{X}$ as well as a tractable transfer function, $p_D(x \mid z)$. See Fig. 1 for a visual description.

For VAEs, identifying $\mathcal{Y}$ with $\mathcal{Z}$ is difficult because the training objective requires the ability to explicitly compute the latent space prior, $p(y)$ for $y \in \mathcal{Y}$. In particle physics, such explicit computations are intractable. We therefore turn to a new form of probabilistic autoencoder: the Sliced Wasserstein Autoencoder (SWAE)[20,21], which alleviates this, and other, issues by reformulating the objective using the Sliced Wasserstein distance and other ideas from optimal transport theory. This reformulation lets us identify $\mathcal{Y}$ with $\mathcal{Z}$ and also allows the encoder and decoder network mappings to be inherently stochastic.

We suggest that an SWAE[20] can be used to achieve the broad goal of simulators: learning the mapping from the physical latent space to experimental data directly from samples of experimental data $\{x \sim p(x)\}$ and theoretical priors $\{z \sim p(z)\}$ in control regions. The resulting decoder ($\mathcal{Z} \rightarrow \mathcal{X}$) can be applied as a simulator, generating samples of experimental data from latent variables in a fraction of the time, and probed and visualized to ensure a physically meaningful transformation. Additionally, the decoder's numerically tractable detector response function, $p_D(x \mid z)$, would be useful in other applications, such as direct calculation of likelihood ratios via integration[24]. The encoder network's $\mathcal{X} \rightarrow \mathcal{Z}$ mapping can also be used in unfolding studies[25,26]. Lastly, the mathematical attributes of the SW distance allow for the inclusion of informed constraints on the mappings.

In this work, we first present background on the problem, the objective, and discuss related work. In Proposed Solution, we present the foundations for the OTUS method and discuss steps toward scaling OTUS to a full simulation capable of replacing current Monte-Carlo methods in particle physics analyses. In Results, we give initial proof-of-principle demonstrations on Z-boson and semileptonic top-quark decays. In "Methodology", we discuss the details of our methods. We then conclude by discussing directions for future work and also briefly discuss how OTUS might be applied to problems in other scientific fields.

Theoretical background

The primary statistical task in particle physics, as in many areas of science, is inferring the value of a model parameter, θ, based on a set of experimental data, $\{x\}$. For example, physicists inferred the mass of the Higgs boson from Large Hadron Collider data[27,28]. Inference about θ requires a statistical model, $p(x \mid \theta)$, which can be used to calculate the probability to make an observation, x, given a parameter value, θ. Unfortunately, such analytical expressions are unavailable due to the indirect nature of observations and the complexity of detectors. Previous solutions to this problem have relied on numerical Monte-Carlo-based simulations[2–4].

Fundamental particle interactions, like the decay of a Higgs boson, produce a set of particles which define an unobserved latent space, $\mathcal{Z}$. The statistical model $p(z \mid \theta)$ is usually well-understood and can often be expressed analytically or approximated numerically. However, experimenters only have access to samples of experimental data, $\{x\}$. Therefore, calculating $p(x \mid \theta)$ requires integrating over the unobserved $\{z \sim p(z \mid \theta)\}$; namely, $p(x \mid \theta) = \int dz\, p(x \mid z)\, p(z \mid \theta)$.

The transfer function, $p(x \mid z)$, represents the multi-staged transformation from the unobserved latent space, $\mathcal{Z}$, to the experimental data space, $\mathcal{X}$. As latent space particles travel they may decay, interact, or radiate to produce subsequent showers of hundreds of secondary particles. These particles then pass through the detector, comprising many layers and millions of sensors resulting in a high-dimensional response of order $\mathcal{O}(10^8)$. Finally, the full set of detector measurements are used to reconstruct an estimate of the identities and momenta of the original unobserved particles in the latent space. For the vast majority of analyses this final, experimental data space, $\mathcal{X}$, has a similar dimensionality [The dimensionality is not necessarily equal due to the imperfect nature of the detection process. For example, $\mathcal{Z}$ may represent four quarks but $\mathcal{X}$ may only contain three jets.] to that of $\mathcal{Z}$, usually $\mathcal{O}(10^1)$. However, the complex, stochastic, and high-dimensional nature of the transformation makes it practically impossible to construct a closed-form expression for the transfer function $p(x \mid z)$. Instead, particle physicists use simulations as a proxy for the true transfer function.

To arrive at $p(x \mid \theta)$, samples of $\{z \sim p(z \mid \theta)\}$ are transformed via simulations into effective samples of $\{x \sim p(x \mid \theta)\}$, approximating the integral above. Current state-of-the-art simulations strive to faithfully model the details of particle propagation and decay via Monte-Carlo techniques. This approach is computationally expensive and limited by our poor understanding of the processes involved. Ad-hoc parameterizations often fill gaps in our knowledge but introduce arbitrary parameters which must be tuned to give realistic results using data from control regions, where the underlying $p(z \mid \theta)$ is well-established from previous experiments, freeing $p(x \mid \theta)$ of surprises. Examples of control regions include decays of heavy bosons (e.g. Z) or the top quark (t).

The computational cost of current simulations is the dominant source of systematic uncertainties and the largest bottleneck in testing new models of particle physics[29]. A computationally-inexpensive, flexible simulator which can map from $\mathcal{Z}$ to $\mathcal{X}$ such that it effectively approximates $p(x \mid z)$ would be a breakthrough.

Objective and related work

The development of OTUS was guided by the goals of the simulation task and the information available for training. Specifically, the simulator has access to samples from model priors, $p(z \mid \theta)_{\text{control}}$, and experimental data samples, $\{x_{\text{control}}\}$. Critically, $\{x_{\text{control}}\}$ samples come from experiments, where the true $\{z_{\text{control}}\}$ are unknown, such that no $(z_{\text{control}}, x_{\text{control}})$ pairs exist. Instead, the distribution of $\{z_{\text{control}}\}$ are known to follow $p(z \mid \theta)_{\text{control}}$ and the distribution of $\{x_{\text{control}}\}$ is observed.

The simulator should learn a stochastic transformation $\mathcal{Z} \to \mathcal{X}$ such that samples $\{z\}$ drawn from $p(z \mid \theta)_{\text{control}}$ can be transformed into samples $\{x\}$ whose distribution matches that of the experimental data $\{x_{\text{control}}\}$. Additionally, these control regions should be robust so that the simulator can approximate $p(x \mid \theta)$ for different, but related, values of θ. Traditional Monte-Carlo simulators such as GEANT4[2] face related challenges.

The flexibility of ML models at learning difficult functions across a wide array of contexts suggests that these tools could be used to develop a fast simulator. The objectives described above translate to four constraints on the class of ML model and methods of learning. Generating samples of $\{x \in \mathcal{X}\}$ requires a (1) generative ML method. For $z \in \mathcal{Z}$, the simulator maps $z \to x$ such that the output x depends on the input z, meaning the mapping is (2) conditional. The problem's inherent and unknown randomness prevents us from assuming any particular density model, suggesting that our simulator should preferably be (3) inherently stochastic. The lack of (z, x) pairs mandates an (4) unsupervised training scheme. Additionally, the chosen method should produce a simulation mapping ($\mathcal{Z} \to \mathcal{X}$) which is inspectable and physically interpretable.

Generative ML models can produce realistic samples of data in many settings, including natural images. Generative Adversarial Networks (GANs) transform noise into artificial data samples and have been adapted to particle physics simulation tasks for both high-level and raw detector data, which can resemble images[11–14,17]. However, while GANs have successfully mimicked existing datasets, $\{x\}$, for a fixed set of $\{z\}$, they have not learned the general transformation $z \to x$ prescribed by $p(x \mid z)$, and so cannot generate fresh samples $\{x'\}$ for a new set of $\{z'\}$, thus failing condition (2). Other GAN-based approaches[14] condition the generation of $\{x\}$ on values of $\{z\}$, but in the process use labeled pairs (x, z), which are only obtained from other simulators, rather than from experiments, thus failing condition (4). Relying on simulated (x, z) pairs incurs the computational cost we seek to avoid, and limits the role of these fast simulators to supplementing traditional simulators, rather than replacing them.

An alternative class of unsupervised, generative ML models are variational autoencoders (VAEs). While GANs leverage an adversarial training scheme, VAEs instead optimize a variational bound on the data's likelihood by constructing an intermediate latent space, $\mathcal{Y}$, which is distributed according to a prior, $p(y)$[30]. An encoder ($\mathcal{X} \rightarrow \mathcal{Y}$) network transforms $x \rightarrow \tilde{y}$, where the ~ distinguishes a mapped sample from those drawn from $p(y)$. Similarly, a decoder ($\mathcal{Y} \rightarrow \mathcal{X}$) network transforms a sample produced by the encoder back to the data space, $\tilde{y} \rightarrow \tilde{x}$. The autoencoder structure is the combined encoder–decoder chain, $x \rightarrow \tilde{y} \rightarrow \tilde{x}$. During training, the distribution of the encoder output, $p_E(y \mid x)$, is constrained to match the latent space prior, $p(y)$, via a latent loss term which measures the distance between the distributions. At the same time, the output of the autoencoder, $\tilde{x}$, is constrained to match the input, x, which are compared pairwise. New samples from $\mathcal{X}$ following the distribution of the data, $p(x)$, can then be produced by decoding samples, $\{y\}$, drawn from $p(y)$, via $y \rightarrow \tilde{x}'$.

The form of $p(y)$ is usually independent of the nature of the problem's underlying theoretical model, and is often chosen to be a multi-dimensional Gaussian for simplicity. This choice provides sufficient expressive power even for complex datasets (i.e. natural images). However, in the particle physics community, optimizing the encoding mapping to match this latent space is seen as an extra, unnecessary hurdle in training[12]. Therefore, GANs have been largely favored over VAEs in the pursuit of a fast particle physics simulator. Some studies investigated VAEs in this context, but retained the unphysical form of $p(y)$ (i.e. multi-dimensional Gaussian)[13,15,16], preventing them from being conditional generators, failing requirement (2).

Proposed solution

Our approach: OTUS.

In this work, we aim to align the probabilistic autoencoder's latent space, $\mathcal{Y}$, with that of our inference task, $\mathcal{Z}$. This will allow us to learn a conditional simulation mapping from our theoretical model latent space to our data space, $\mathcal{Z} \rightarrow \mathcal{X}$. Therefore, we construct a probabilistic autoencoder where the latent space prior, $p(y)$, is identical to the physical latent space, $p(y) \equiv p(z) = p(z \mid \theta)$, for the choice of particular parameters, θ. The decoder then learns $p_D(x \mid z)$ providing precisely the desired conditional transformation, $z \rightarrow x$. Additionally, $p_D(x \mid z)$ can act as a tractable transfer function in approaches which estimate $p(x \mid \theta)$ via direct integration[24]. The encoder's learned $p_E(z \mid x)$ is of similar interest in unfolding applications[25,26].

This is not possible with VAEs because optimizing the variational objective requires explicit computation of the densities $p(y), p_E(y \mid x)$, and $p_D(x \mid y)$. Therefore, $p(y)$ is often assumed to be a standard isotropic Gaussian for its simplicity and potential for uncovering independent latent factors of the data generation process. However, in particle physics the true prior, $p(z)$, which is governed by quantum field theory, is highly non-Gaussian and computing its density explicitly requires an expensive numerical procedure. Similarly, as we have little knowledge about the true underlying stochastic transforms, assuming any particular parametric density model for $p_E(y \mid x)$ or $p_D(x \mid y)$, like a multivariate Gaussian, would be inappropriate and overly restrictive. These concerns led us to use inherently stochastic (i.e. implicit) models for $p(z), p_E(z \mid x)$, and $p_D(x \mid z)$ that are fully sample-driven.

Additionally, the VAE objective's use of KL-divergence introduces technical disadvantages. The KL-divergence, $D_{\mathrm{KL}}(\cdot \| \cdot)$, is not a true distance metric, and will diverge for non-overlapping distributions often leading to unusable gradients during training[20,31]. Moreover, the specific use of $D_{\mathrm{KL}}(p_E(z \mid x) \| p(z))$ within the VAE loss forces $p_E(z \mid x)$ to match $p(z)$ for every value of $x \sim p(x)$[21]. This term must be carefully tuned (e.g. with a β-VAE approach[13,32]) to avoid the undesirable effect of the encoder mapping different parts of $\mathcal{X}$ to the same overlapping region in $\mathcal{Z}$, which can be particularly problematic if $\mathcal{Z}$ represents a physically meaningful latent space.

We resolve these issues by applying an emerging class of probabilistic autoencoders, based instead on the Wasserstein distance, which is a well-behaved distance metric between arbitrary probability distributions rooted in concepts from optimal transport theory[20,21].

The original Wasserstein Autoencoder (WAE)[21] loss function is

$$\mathcal{L}_{\mathrm{WAE}}(p(x), p_D(x \mid z), p_E(z \mid x)) = \underset{1A}{\mathbb{E}_{x \sim p(x)} \mathbb{E}_{p_E(z \mid x)} \mathbb{E}_{\tilde{x} \sim p_D(x \mid z)}[c(x, \tilde{x})]} + \lambda \underset{1B}{d_z(p_E(z), p(z))}, \tag{1}$$

where $\mathbb{E}$ denotes the expectation operator and $c(\cdot, \cdot)$ is a cost metric. For the optimal $p_E(z \mid x)$, $\mathcal{L}_{\mathrm{WAE}}$ becomes an upper bound on the Wasserstein distance between the true data distribution, $p(x)$, and the decoder's learned distribution, $p_D(x) = \int dz p_D(x \mid z) p(z)$; the bound is tight for deterministic decoders.

Term A of Eq. (1) constrains the output of the encoder–decoder mapping, $\tilde{x}$, to match the input, x, while term B of Eq. (1) constrains the encoder mapping. The hyperparameter λ provides a relative weighting between the two terms. The difference between the marginal encoding distribution, $p_E(z) = \int dx p_E(z \mid x) p(x)$, and the latent prior, $p(z)$, is measured by $d_z(\cdot, \cdot)$. [Comparing $p_E(z)$ and $p(z)$ rather than $p_E(z \mid x)$ and $p(z)$ is the crucial innovation which allows different parts of $\mathcal{Z}$ to remain disjoint.] Unfortunately, the originally proposed options for $d_z(\cdot, \cdot)$[21] had undesirable features which made them ill-suited for this particle physics problem (see "Model choice").

The more recent Sliced Wasserstein Autoencoder (SWAE)[20] uses the Sliced Wasserstein (SW) distance as the $d_z(\cdot, \cdot)$ metric. The SW distance, $d_{\mathrm{SW}}(\cdot, \cdot)$, is a rigorous approximation to the Wasserstein distance, $d_W(\cdot, \cdot)$. The SWAE completely grounds the loss function in optimal transport theory as each term and the total loss can be identified as approximating the Wasserstein distances between various distributions and allows $p(y)$ to be any sampleable distribution, including the physical, $p(z)$. Additionally, the (S)WAE method allows the encoder and decoder to be implicit probability models, while avoiding an adversarial training strategy which can lead to problems like mode collapse[33].

Both d_W and d_{SW} are true distance metrics[20]. The KL-divergence and adversarial schemes lack this property resulting in divergences and meaningless loss values which lead to problems during training and make it difficult to include additional, physically-motivated constraints. The Wasserstein distance is the cost to transport probability mass from one probability distribution to another according to a cost metric, $c(\cdot, \cdot)$, following the optimal transportation map. However, it is difficult to calculate for multivariate probability distributions when pairs from the optimal transportation map are unknown. However, for univariate probability distributions, there is a closed-form solution involving the difference between the inverse Cumulative Distribution Functions (CDF^{-1}s) of the two probability distributions. The SW distance approximates the Wasserstein distance by averaging the one-dimensional Wasserstein distance over many randomly selected slices—one-dimensional projections of the full probability distribution[20] (see "Training").

The SWAE loss takes the general form of the WAE loss

$$\mathcal{L}_{\text{SWAE}}(p(x), p_D(x \mid z), p_E(z \mid x)) = \underset{2A}{\mathbb{E}_{x\sim p(x)}\mathbb{E}_{p_E(z|x)}\mathbb{E}_{\tilde{x}\sim p_D(x|z)}[c(x,\tilde{x})]} + \underset{2B}{\lambda\, d_{\text{SW}}(p_E(z), p(z))}. \tag{2}$$

Term A of Eq. (2) compares pairs $(x, \tilde{x})$, where $\tilde{x}$ is the output of the encoder–decoder mapping. In term B of Eq. (2), matched pairs are not available so we instead use the SW distance approximation. Both loss terms use the cost metric $c(u, v) = ||u - v||^2$[20].

The SWAE allows us to train a probabilistic autoencoder that transforms between $\mathcal{X}$ and $\mathcal{Z}$ with a physical prior $p(z)$. However, since we are in an unsupervised setting, the true $p(x \mid z)$ is unknown. It is therefore crucial to ensure that the learned transformation is plausible and represents a series of physical interactions. To encourage this, we can easily impose supplemental physically-meaningful constraints on the SWAE model. These constraints can be relations between $\mathcal{Z}$ and $\mathcal{X}$ spaces or constraints on the internal properties of these respective spaces. In this work, we use one constraint from each category.

From the first category, we add a term comparing the unit vector parallel to the momentum of an easily identifiable particle in the latent and experimental spaces. This can be thought of as analogous to choosing a consistent basis and can be helpful for problems containing simple inversion symmetries. An example of such an inversion symmetry exists in the $Z \to e^+e^-$ study below. In particle experiments, misidentification of lepton charge in the process of data reconstruction is known to be extremely rare. This means a learned mapping which frequently maps electron/positron ($e^\mp$) information in $\mathcal{Z}$ to positron/electron ($e^\pm$) information in $\mathcal{X}$, and vice versa, would be unphysical. For a generative mapping $G : \mathcal{U} \to \mathcal{V}$, this anchor term takes the general form

$$\mathcal{L}_A(p(u), p_G(v \mid u)) = \mathbb{E}_{u\sim p(u)}\mathbb{E}_{v\sim p_G(v|u)}[c_A(u, v)]. \tag{3}$$

We chose $c_A(u, v) = 1 - \hat{\mathbf{p}}_u \cdot \hat{\mathbf{p}}_v$, where $\hat{\mathbf{p}}$ is the unit vector of the electron's momentum. We add the anchor loss in $\mathcal{Z}$ space, $\mathcal{L}_A(p(x), p_E(z \mid x))$, and in $\mathcal{X}$ space, $\mathcal{L}_A(p(z), p_D(x \mid z))$, to the SWAE loss with hyperparameter weightings β_E and β_D respectively.

From the second category, we enforce the Minkowski metric constraint internally for $\mathcal{Z}$ and $\mathcal{X}$ spaces respectively. A particle's nature, excluding discrete properties such as charge and spin, is described by four quantities related by the Minkowski metric. Arranging these quantities into a 4-vector defined as $p^\mu = (\mathbf{p}, E)$ where E is a particle's energy and $\mathbf{p}$ is a vector of its momentum in the $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ direction respectively, the constraint becomes

$$p^\mu p_\mu = E^2 - \mathbf{p}^2 = m^2, \tag{4}$$

where m is the particle's mass. We directly enforce this relationship in the model for all particles. [We note that initial experiments lacked this constraint yet the networks automatically learned this relationship from the data. However, directly including this constraint in the model architecture improved performance overall.]

Adding more physically-motivated constraints would be straightforward, however, in this work we only assume this minimal set and recommend that more robust data structures be considered first, as such constraints may become unnecessary (see "Conclusion").

OTUS in practice. In this section we briefly outline how OTUS might eventually be applied to problems in particle physics such as searches for new particles. However, we emphasize that this work only demonstrates a proof-of-principle version of OTUS. Follow-up work will be necessary to overcome some technical hurdles before OTUS could be applied to such a problem (see "Conclusion").

A main goal of particle physics is to discover the complete set of fundamental units of matter: particles. Therefore, searches for exotic particles are common practice in this field. These searches typically proceed by looking for anomalies in data which are better described by simulations which assume the existence of a new particle. It is therefore phrased as a hypothesis test between two theoretical models, θ_{SM}, which assumes only the particles in the Standard Model (SM), and θ_{BSM}, which assumes the existence of one or more new particles that lie Beyond the Standard Model (BSM). These distinct models will generate distinct latent signatures, $\{z_{\text{SM}} \mid \theta\}$ and $\{z_{\text{BSM}} \mid \theta\}$, which lie in $\mathcal{Z}$. As particle physics experiments do not observe the latent $\{z\}$ directly, the hypothesis test is performed in the observed space $\mathcal{X}$, see "Introduction" and "Objective and related work" for more details.

The goal for OTUS is to learn a simulation mapping from $\mathcal{Z} \to \mathcal{X}$ which is independent of the underlying model, θ, and can be applied to any z. This is achieved by carefully selecting control regions, $\{z_i \mid \theta_i\}$, which span $\mathcal{Z}$ and for which observed data, $\{x\}$, is available for training. See Fig. 2 for a visual description. These control regions have known distributions of outcomes in $\mathcal{X}$, which allows us to properly match distributions in $\mathcal{Z}$ to distributions in $\mathcal{X}$ for training OTUS. Since these control regions are chosen to span $\mathcal{Z}$, OTUS will then be able to interpolate to unseen signal regions. Neural networks in general are known to perform well at interpolation tasks[34], and recent work has shown that autoencoders in particular are proficient at learning manifold interpolation[35]. Still more work has suggested there might be a deeper connection to the structure of this manifold and optimal transport[36]. Therefore, it is reasonable to expect that OTUS will be able to interpolate well in this space. However, these claims should be thoroughly investigated in future work.

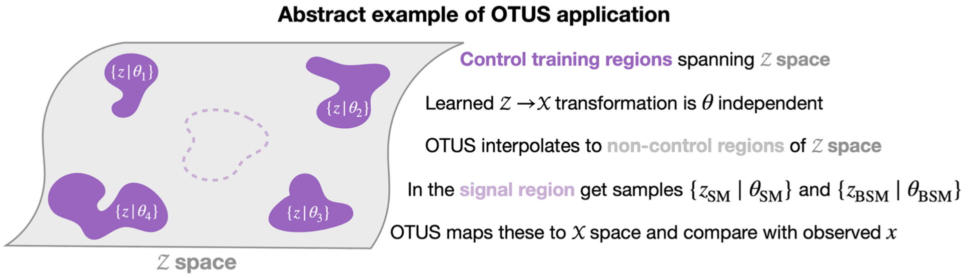
Abstract example of OTUS application
$\{z\,|\,\theta_1\}$
$\{z\,|\,\theta_2\}$
$\{z\,|\,\theta_4\}$
$\{z\,|\,\theta_3\}$
Control training regions spanning $\mathcal{Z}$ space
Learned $\mathcal{Z} \rightarrow \mathcal{X}$ transformation is θ independent
OTUS interpolates to non-control regions of $\mathcal{Z}$ space
In the signal region get samples $\{z_{\mathrm{SM}} \mid \theta_{\mathrm{SM}}\}$ and $\{z_{\mathrm{BSM}} \mid \theta_{\mathrm{BSM}}\}$
OTUS maps these to $\mathcal{X}$ space and compare with observed x
$\mathcal{Z}$ space

Figure 2. Schematic diagram of how OTUS can be used in an abstract analysis. The gray surface represents $\mathcal{Z}$. Different theoretical models, θ_i, will produce different signatures $\{z_i \mid \theta_i\}$ which lie in $\mathcal{Z}$. The goal of OTUS is to learn a general mapping from $\mathcal{Z} \rightarrow \mathcal{X}$ which is independent of the underlying theory, θ, and only depends on the information contained in $\{z \in \mathcal{Z}\}$. One trains OTUS using control region data which span $\mathcal{Z}$ and have known outcomes in $\mathcal{X}$. These allow us to pair distributions in $\mathcal{Z}$ with distributions in $\mathcal{X}$. From these examples, OTUS interpolates to the rest of $\mathcal{Z}$ and can then be used to generate $\{x_i\}$ from samples $\{z_i \mid \theta_i\}$ from regions not used during training, including the blinded signal region. This can then be used to search for new particles.

A signal region is a region in $\mathcal{Z}$ space where signatures of new particles might occur. SM predictions, $\{z_{\mathrm{SM}} \mid \theta_{\mathrm{SM}}\}$, and BSM predictions, $\{z_{\mathrm{BSM}} \mid \theta_{\mathrm{BSM}}\}$, would then be passed to OTUS to produce two simulated data samples $\{x_{\mathrm{SM}}\}$ and $\{x_{\mathrm{BSM}}\}$ which would be compared with observed data, $\{x\}$, via a hypothesis test to calculate the relative likelihood of the SM and BSM theories. This technique, simulation-based inference, is standard practice in particle physics and is applied to existing simulation methods.

As a concrete example, let our BSM theory be the SM with the addition of a new particle, Z', with a mass of 0.030 [TeVc^{-2}], which decays into a pair of leptons, a flagship search for the Large Hadron Collider[37]. The latent space $\mathcal{Z}$ would include the two leptons produced by the decay of the Z', and the observed space $\mathcal{X}$ would include the leptons identified and measured by the detector. For OTUS to be able to predict the observed signatures from this latent space, it would need to interpolate between control regions which have similar relationships. Decays of existing particles to leptons, such as the 0.091 [TeVc^{-2}] Z and the 0.002 [TeVc^{-2}] J/ψ would allow OTUS to learn the mapping from latent leptons to observed leptons. Our theoretical Z' has a mass which lies between those of the particles in our control regions. OTUS would need to interpolate along this axis; control regions at various masses provided by the Z and J/ψ decays are therefore essential to describe and determine the nature of the interpolation. To verify the interpolation, one might compare the prediction of OTUS to observed data in the intermediate range between the Z' and the Z.

Alternatively, the Z' could have a heavier mass, e.g. 1 [TeVc^{-2}]. In this scenario, OTUS would be required to extrapolate along the mass axis. Naively, this sounds problematic as extrapolation is generally much less sound than interpolation, however this task is also required of current simulations for this scenario. Simulations succeed in such tasks when they have inductive biases which control their behavior even outside of training (tuning) regions. These inductive biases are based on physics principles and scale to the signal regions of interest. For neural networks, it has been shown that architectures with inductive bias constraints succeed at such extrapolation tasks[38]. Since a mature version of OTUS will manifestly include such inductive biases (see "Conclusion") it is reasonable to assume it can achieve this task as well as current simulation methods can.

Results

Demonstration in $Z \rightarrow e^+e^-$ decays. We first test OTUS on an important control region: leptonic decays of the Z-boson to electron-positron pairs, $Z \rightarrow e^+e^-$. The theoretical prior is well-known, and its parameters $\{\theta\}$, like the Z-boson's mass and its interaction strengths, are tightly constrained by precision experiments. We identify $\mathcal{Z}$ with the Z-boson's decay products: the electron, e^-, and positron, e^+, whose four-momenta span the space. We compose these into an eight-dimensional vector

$$z := \{z_{e^-}, z_{e^+}\} = \{\mathbf{p}^{e^-}, E^{e^-}, \mathbf{p}^{e^+}, E^{e^+}\}. \tag{5}$$

This simplistic vector description excludes categorical properties such as charge.

The model prior $p(z)$ can be simply expressed with quantum field theory and sampled. The subsequent step, where the electron and positron travel through the layers of detectors, depositing energy and causing particle showers, cannot be described analytically; a model will be learned by OTUS from data in control regions. Here we use simulated data samples, but specific (z, x) pairs are not used to mimic the information available when training from real data. The complex intermediate state with many low-energy particles and high-dimensional detector readouts is reduced and reconstructed yielding estimates of the electron and positron four-momenta. Therefore, $\mathcal{X}$ has the same structure and dimensionality as $\mathcal{Z}$, though the distribution $p(x)$ reflects the impact of the finite resolution of detector systems (see "Data generation").

Figure 3 shows distributions of testing data, unpaired samples from $\mathcal{X}$ and $\mathcal{Z}$ in several projections, and the results of applying the trained encoder and decoder to transform between the two spaces. Visual evaluation indicates qualitatively good performance, and quantitative metrics are provided. Measuring overall performance, the SW distances are as follows: $d_{\mathrm{SW}}(p(z), p_E(\tilde{z})) = 0.984$ [GeV^2], $d_{\mathrm{SW}}(p(x), p_D(\tilde{x})) = 1.33$ [GeV^2],

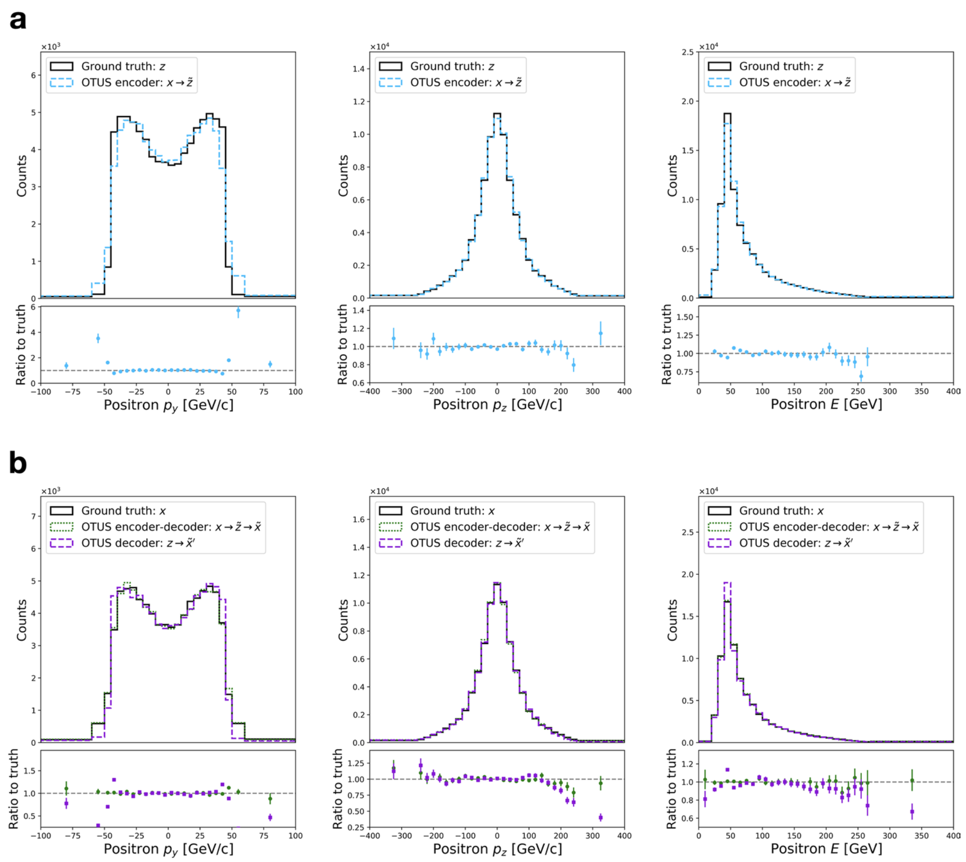

Figure 3. Performance of OTUS for $Z \to e^+e^-$ decays. (**a**) Matching of the positron's p_x, p_y, and E distributions in $\mathcal{Z}$. It shows distributions of samples from the theoretical prior, $\{z \sim p(z)\}$ (solid black), as well as the output of the encoder, $\{\tilde{z}\}$; the encoder transforms samples of testing data in experimental space, $\mathcal{X}$, to the latent space, $\mathcal{Z}$, and is shown as $x \to \tilde{z}$ (dashed cyan). (**b**) Matching of the positron's p_x, p_y, and E distributions in $\mathcal{X}$. It shows the testing sample $\{x \sim p(x)\}$ (solid black) in the experimental space, $\mathcal{X}$, as well as output from the decoder applied to samples drawn from $p(z)$, labeled as $z \to \tilde{x}'$ (dashed purple). Also shown are samples passed through both the decoder and encoder chain, $x \to \tilde{z} \to \tilde{x}$ (dotted green). Dotted green and solid black distributions are matched explicitly during training. Enhanced differences between dashed purple and solid black indicate the encoder's output needs improvement, as $p_E(z)$ does not fully match $p(z)$. If performance were ideal, the distributions in every plot would match up to statistical fluctuations. Residual plots show bin-by-bin ratios with statistical uncertainties propagated accordingly (see "Evaluation").

$d_{\mathrm{SW}}(p(x), p_D(\tilde{x}')) = 3.03$ [GeV^2]. Additionally, several common metrics are reported for each projection in Supplementary Tables 1 and 2. Details of the calculations are provided in Evaluation.

To ensure that the learned decoder reflects the physical processes being modeled, we inspect the transformation from $\mathcal{Z} \to \mathcal{X}$ in Fig. 4. The learned transfer function, $p_D(x \mid z)$, shows reasonable behavior, mapping samples from $\mathcal{Z}$ to nearby values of $\mathcal{X}$. This reflects the imperfect resolution of the detector while avoiding unphysical transformations such as mapping information on the far-end distribution tails in $\mathcal{Z}$ to the distribution peaks in $\mathcal{X}$.

Finally, we examine the distribution of a physically important derived quantity, the invariant mass of the Z-boson, see Fig. 5. This quantity was not used as an element of the loss function, and so provides an alternative measure of performance. The results indicate a high-quality description of the transformation from $\mathcal{Z}$ to $\mathcal{X}$. The performance of the transformation from $\mathcal{X}$ to $\mathcal{Z}$ is less well-described, likely because this relation is more strict in $\mathcal{Z}$ causing a sharper peak in the distribution. Such strict rules are difficult for networks to learn when not penalized directly or hard-coded as inductive biases, again signaling that a robust data representation will be crucial to improving performance (see "Conclusion").

Demonstration in semileptonic top-quark decays. The Z-boson control region is valuable for calibrating simulations of leptons such as electrons or muons, which tend to be stable and well-measured. We next test OTUS on the challenging task of modeling the decay and detection of top-quark pairs featuring more complex detector signatures. This control region has more observed particles and introduces additional complexities: unstable particles decaying in flight, significantly degraded resolution relative to leptons, undetected particles, and a stochastically variable number of observed particles.

The initial creation of top-quark pairs, their leading-order decay $t\,\bar{t} \to W^+b\;W^-\bar{b}$, and the subsequent W-boson decays are well-described using quantum field theory, so $p(z \mid \theta)$ can be sampled. We select the modes

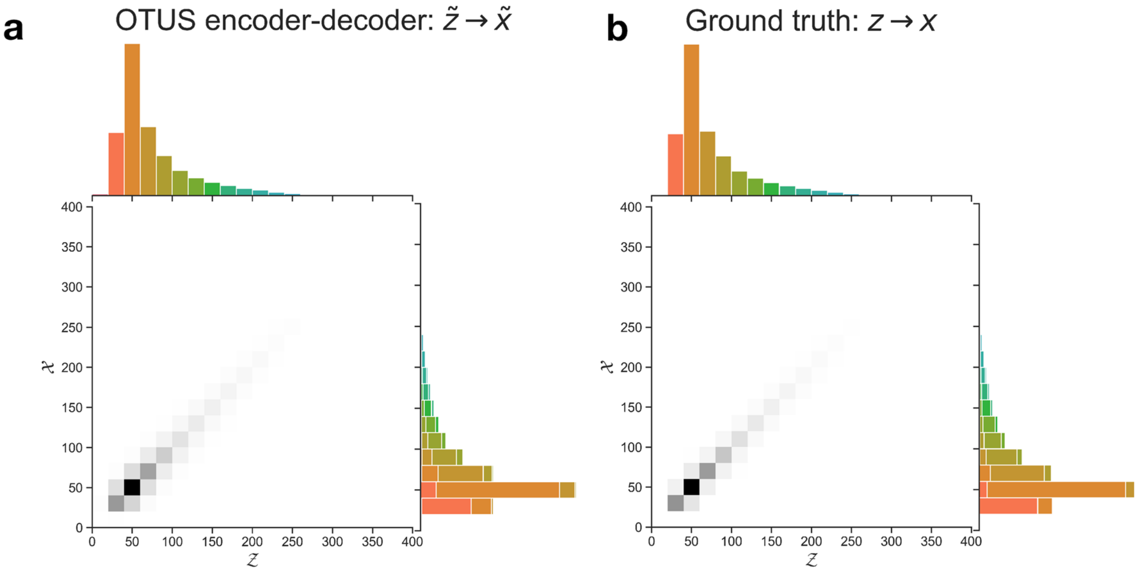

Figure 4. Visualization of the transformation from $\mathcal{Z} \rightarrow \mathcal{X}$ in the $Z \rightarrow e^+e^-$ study for positron energy. (**a**) The learned transformation of the decoder, $p_D(x \mid z)$. (**b**) The true transformation from the simulated sample, for comparison, though the true (z, x) pairs are not typically available and were not used in training. Colors in the $\mathcal{X}$ projection indicate the source bin in $\mathcal{Z}$ for a given sample.

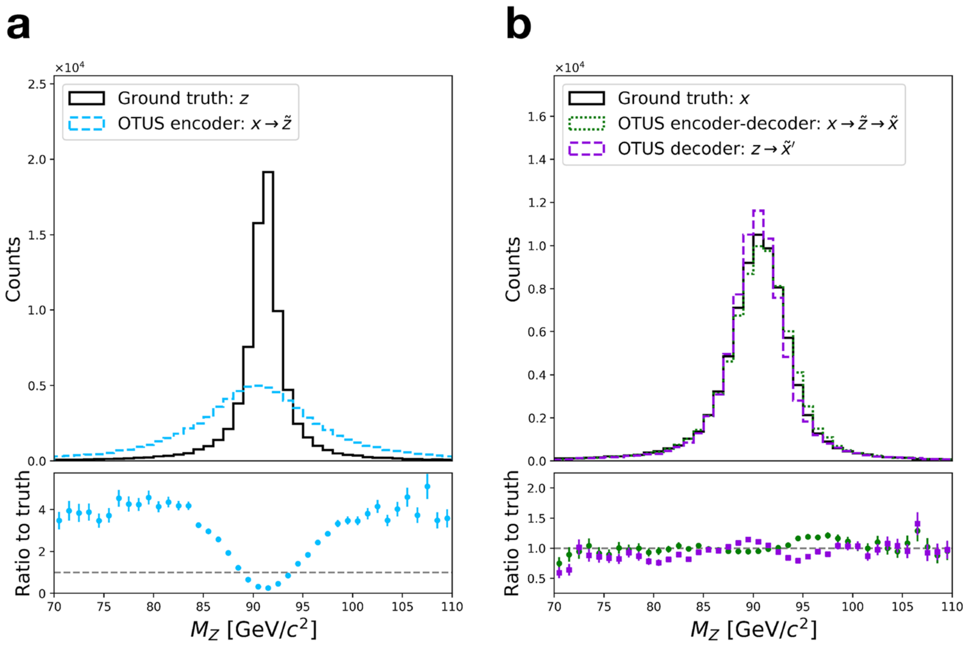

Figure 5. Performance of OTUS for $Z \rightarrow e^+e^-$ decays in a physically important derived quantity, the invariant mass of the electron-positron pair, M_Z. (**a**) Matching of the M_Z distribution in $\mathcal{Z}$. It shows distributions of samples from the theoretical prior, $\{z \sim p(z)\}$ (solid black), as well as the output of the encoder, $\{\tilde{z}\}$; the encoder transforms samples of testing data in experimental space, $\mathcal{X}$, to the latent space, $\mathcal{Z}$, and is shown as $x \rightarrow \tilde{z}$ (dashed cyan). (**b**) Matching of the M_Z distribution in $\mathcal{X}$. It shows the testing sample $\{x \sim p(x)\}$ (solid black) in the experimental space, $\mathcal{X}$, as well as output from the decoder applied to samples drawn from $p(z)$, labeled as $z \rightarrow \tilde{x}'$ (dashed purple). Also shown are samples passed through both the decoder and encoder chain, $x \rightarrow \tilde{z} \rightarrow \tilde{x}$ (dotted green). Dotted green and solid black distributions are matched explicitly during training. Enhanced differences between dashed purple and solid black indicate the encoder's output needs improvement, as $p_E(z)$ does not fully match $p(z)$. If performance were ideal, the distributions in every plot would match up to statistical fluctuations. Note that this projection was not explicitly used during training, but was inferred by the networks. Residual plots show bin-by-bin ratios with statistical uncertainties propagated accordingly (see "Evaluation").

$W^- \to e^- \, \bar{\nu}_e$ and $W^+ \to u \, \bar{d}$ as examples and assign our latent space to describe the four-momenta of these six of particles:

$$z := \{z_{e^-}, \, z_{\bar{\nu}_e}, \, z_b, \, z_{\bar{b}}, \, z_u, \, z_{\bar{d}}\} = \{\mathbf{p}^{e^-}, E^{e^-}, \mathbf{p}^{\bar{\nu}_e}, E^{\bar{\nu}_e}, \mathbf{p}^b, E^b, \mathbf{p}^{\bar{b}}, E^{\bar{b}}, \mathbf{p}^u, E^u, \mathbf{p}^{\bar{d}}, E^{\bar{d}}\} \tag{6}$$

with a total of twenty-four dimensions.

Unlike in the $Z \to e^+e^-$ study, the $\mathcal{X}$ space's structure is considerably different from that of the $\mathcal{Z}$ space. While the electron e^- is stable and readily identifiable, the other particles are more challenging. The neutrino, $\bar{\nu}_e$, is stable, yet invisible to our detectors, providing no estimate of its direction or momentum; instead its presence is inferred using momentum conservation $\mathbf{p}^{\nu} = -\sum \mathbf{p}^{\text{observed}}$. Unfortunately, soft initial state radiation and detector inefficiencies also contribute to missing momentum. The aggregate quantity is labeled $\mathbf{p}^{\text{miss}}$. The four quarks $\bar{b}$, u, $\bar{d}$ and b are strongly-interacting particles each producing complex showers of particles that are clustered together into jets to estimate the original quark momenta and directions. Unfortunately, despite significant recent progress[39–41], we cannot assume a perfect identification of the source particle in $\mathcal{Z}$ for a given jet observed in $\mathcal{X}$, causing significant ambiguity.

Additionally, a complete description of the $\mathcal{Z} \to \mathcal{X}$ transformation should include the possibilities for the number of jets in $\mathcal{X}$ to exceed the number of quarks, due to radiation and splitting, or to fail to match the number of quarks, due to jet overlap or detector inefficiency. We leave this complexity for future work and restrict our $\mathcal{X}$ space to contain exactly four jets.

The final complexity introduced in this study is the presence of a sharp lower threshold in transverse momentum, p_{T}. Experimental limitations require that jets with $p_{\text{T}} < 20$ [GeVc^{-1}] be discarded and therefore are not represented in the training dataset, as they would be unavailable in control region data. Mimicking this experimental effect, we directly impose this threshold on the decoder's output instead of the network learning it. Paralleling reality, such events are discarded before computing losses. This strategy requires modifications to both the model and training strategy (see "Methodology").

Our experimental data is the vector

$$x := \{x_{e^-}, x_{\text{miss}}, \, x_{\text{jet1}}, \, x_{\text{jet2}}, \, x_{\text{jet3}}, x_{\text{jet4}}\} = \{\mathbf{p}^{e^-}, E^{e^-}, \mathbf{p}^{\text{miss}}, E^{\text{miss}}, \mathbf{p}^{\text{jet1}}, E^{\text{jet1}}, \mathbf{p}^{\text{jet2}}, E^{\text{jet2}}, \mathbf{p}^{\text{jet3}}, E^{\text{jet3}}, \mathbf{p}^{\text{jet4}}, E^{\text{jet4}}\}, \tag{7}$$

with a total of twenty-four dimensions. If quark-jet assignment were possible, it would be natural to align the order of the observed jets with the order of their originating quarks in $\mathcal{Z}$ space. Lacking this information, it is typical to order jets by descending $|\mathbf{p}_{\text{T}}| = \sqrt{p_x^2 + p_y^2}$, where jet 1 has the largest $|\mathbf{p}_{\text{T}}|$.

Figure 6 shows distributions of testing data, unpaired samples from $\mathcal{X}$ and $\mathcal{Z}$ in several projections, and the results of applying the trained encoder and decoder to transform between the two spaces. Visual evaluation indicates qualitatively good performance, and quantitative metrics are also provided. Measuring overall performance the SW distances are as follows: $d_{\text{SW}}(p(z), p_E(\tilde{z})) = 22.3$ [GeV^2], $d_{\text{SW}}(p(x), p_D(\tilde{x})) = 232$ [GeV^2], $d_{\text{SW}}(p(x), p_D(\tilde{x}')) = 120$ [GeV^2]. Additionally, several common metrics are reported for each projection in Supplementary Tables 3 and 4. Details of the calculations are provided in "Evaluation".

To probe the $\mathcal{Z} \to \mathcal{X}$ transformation, we inspect the learned transfer function, $p_D(x \mid z)$ in Fig. 7. While the overall performance is worse in this more complex case, it still shows reasonable behavior, mapping samples from $\mathcal{Z}$ to nearby values of $\mathcal{X}$ and avoiding unphysical transformations such as mapping information on the far-end distribution tails in $\mathcal{Z}$ to the distribution peaks in $\mathcal{X}$. Additionally, cross-referencing with the true simulation's mapping shows the similar nature of the mappings.

Finally, we examine the distribution of physically important derived quantities, the invariant masses of the top-quarks and W-bosons estimated by combining information from pairs and triplets of objects, see Fig. 8. No exact assignments are possible due to the ambiguity of the jet assignment and the lack of transverse information for the neutrino, but a comparison can be made between the experimental sample in $\mathcal{X}$ and the mapped samples $\mathcal{Z} \to \mathcal{X}$. As in the $Z \to e^+e^-$ case, we see imperfect but reasonable matching on such derived quantities which the network was not explicitly instructed to learn.

Methodology

This section provides details on the methods used to produce the results in the previous section. We first describe the data generation process. We then describe the machine learning models used and strategies for how they were trained. Finally, we give details on the qualitative and quantitative evaluation methods used in the visualizations of the results.

Data generation. The data for this work was generated with the programs Madgraph5 v.2.6.3.2[42], Pythia v.8.240[3], and Delphes v.3.4.1[4]. ROOT v.6.08/00[43] was used to interface with the resulting Delphes output files. We used the default run cards for Pythia, Delphes, and Madgraph. Where relevant, jets were clustered using the anti-kt algoritm[44] with a jet radius of 0.5. The card files can be found with the code for this analysis (see "Code availability").

Samples of the physical latent space, $\mathcal{Z}$, were extracted from the Madgraph LHE files to form the 4-momenta of the particles. Samples of the data space, $\mathcal{X}$, were extracted from Delphes' output ROOT files. We selected for the appropriate final state: e^+, e^- in the $Z \to e^+e^-$ study and e^-, missing 4-momentum (i.e. MET $= (\mathbf{p}^{\text{miss}}, E^{\text{miss}})$), and 4 jets in the semileptonic $t\bar{t}$ study. If an event failed this selection, the corresponding $\mathcal{Z}$ event was also removed. Reconstructed data in $\mathcal{X}$ was extracted by default as $(p_{\text{T}}, \eta, \phi)$ of the object and converted into $(\mathbf{p}, E)$ via the following relations

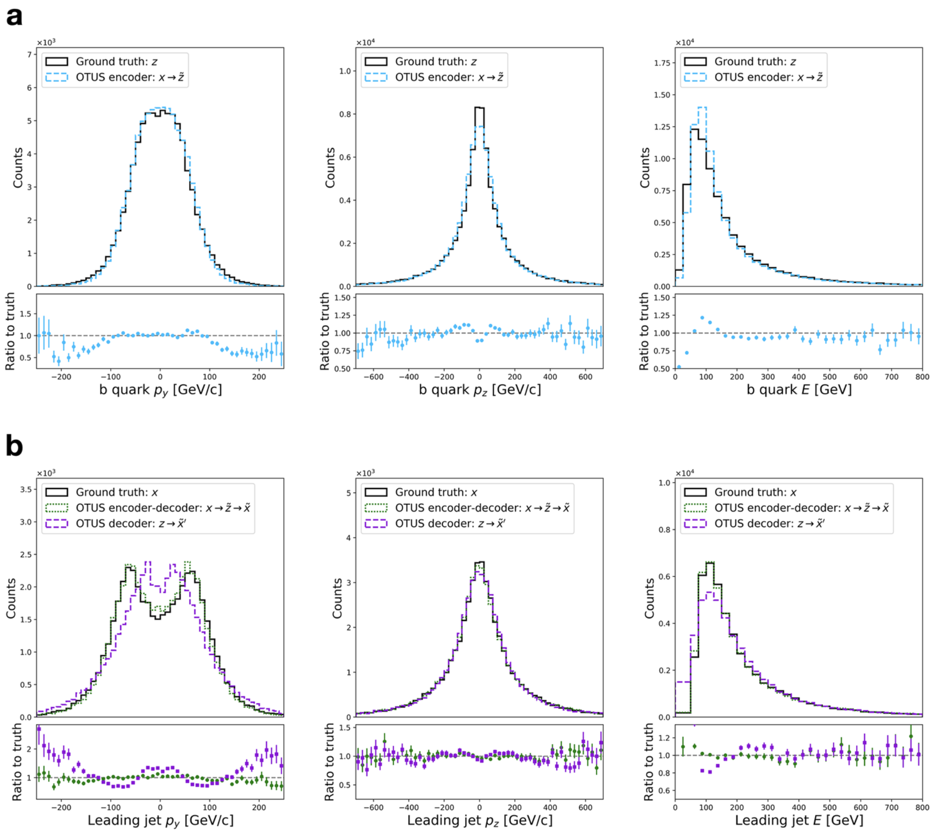

Figure 6. Performance of OTUS for semileptonic $t\bar{t}$ decays. (**a**) Matching of the b quark's p_x, p_y, and E distributions in $\mathcal{Z}$. It shows distributions of samples from the theoretical prior, $\{z \sim p(z)\}$ (solid black), as well as the output of the encoder, $\{\tilde{z}\}$; the encoder transforms samples of the testing data in experimental space, $\mathcal{X}$, to the latent space, $\mathcal{Z}$, and is shown as $x \to \tilde{z}$ (dashed cyan). (**b**) Matching of the leading jet's p_x, p_y, and E distributions in $\mathcal{X}$. It shows the testing sample $\{x \sim p(x)\}$ (solid black) in the experimental space, $\mathcal{X}$, as well as output from the decoder applied to samples drawn from the prior $p(z)$, labeled as $z \to \tilde{x}'$ (dashed purple). Also shown are samples passed through both the decoder and encoder chain, $x \to \tilde{z} \to \tilde{x}$ (dotted green). Dotted green and solid black distributions are matched explicitly during training. Enhanced differences between dashed purple and solid black indicate the encoder's output needs improvement, as $p_E(z)$ does not fully match $p(z)$. If performance were ideal, the distributions in every plot would match up to statistical fluctuations. Residual plots show bin-by-bin ratios with statistical uncertainties propagated accordingly (see "Evaluation").

$$\mathbf{p} := (p_x, p_y, p_z) = (p_{\mathrm{T}}\cos(\phi),\ p_{\mathrm{T}}\sin(\phi),\ p_{\mathrm{T}}\sinh(\eta)) \tag{8}$$

$$E = \sqrt{(p_{\mathrm{T}}\cosh(\eta))^2 + m^2}, \tag{9}$$

where m is the particle's definite mass and is zero for massless particles. Note that we are assuming natural units where the speed of light, c, is equal to unity. This equates the units of energy, E, momentum, $\mathbf{p}$, and mass, m. In our case, $m_{e^+} = m_{e^-} = 0$ [GeVc^{-2}] is a standard assumption given that the true value is very small compared to the considered energy scales. We additionally set $m = 0$ [GeVc^{-2}] for the 4 jets and MET since these objects have atypical definitions of mass.

In total, we generated 491,699 events for $Z \to e^+e^-$ and 422,761 events for semileptonic $t\bar{t}$. The last 160,000 events in each case were reserved solely for statistical tests after training and validation of OTUS.

Model. *Model choice.* In this section, we briefly survey the literature of machine learning methods which might be considered for this task. We discuss their features and whether they are compatible choices for this application.

We will primarily focus on OT-based probabilistic autoencoder methods (i.e. WAE[21] and its derivatives) but first we briefly address a derivative of VAEs, β-VAE. This method appears similar to WAE in the form of loss function that is used. Both have a data-space loss and a latent-space loss with a relative hyperparameter weighting β (or λ for the WAE). However, the β-VAE method is not principled in OT and thus is distinct from the WAE method and its derivatives. Most importantly for our application, the β-VAE (like its predecessor VAE)

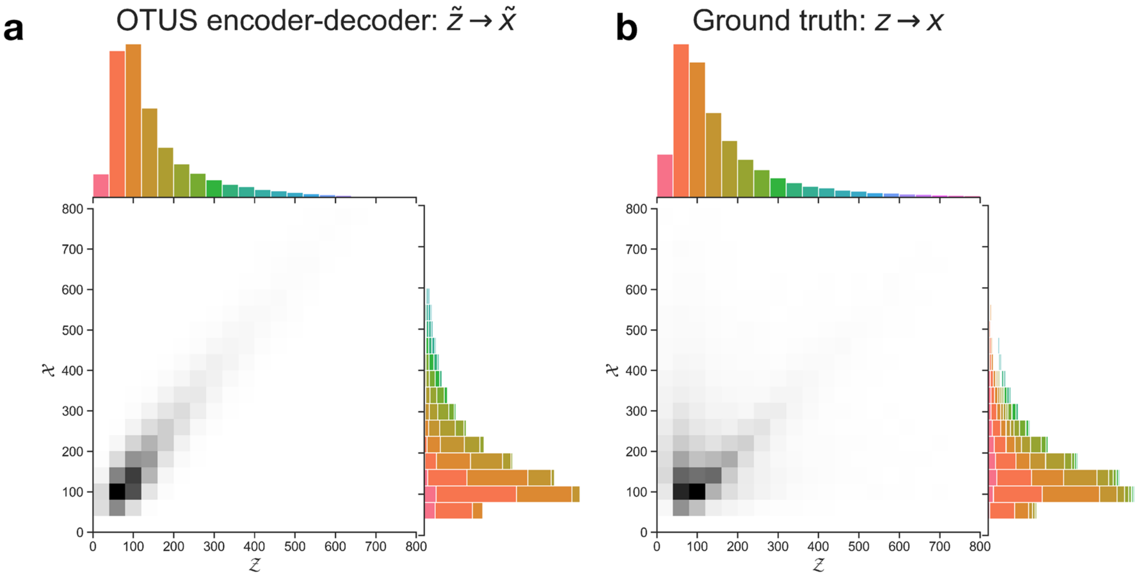

Figure 7. Visualization of the transformation from $\mathcal{Z} \to \mathcal{X}$ in the $t\bar{t}$ study for the energy of the b quark in $\mathcal{Z}$ to energy of the leading jet in $\mathcal{X}$. (**a**) The learned transformation of the decoder, $p_D(x \mid z)$. (**b**) The true transformation from the simulated sample, for comparison, though the true (z, x) pairs are not typically available and were not used in training. Note that the b quark will not always correspond to the leading jet, see the text for details. Colors in the $\mathcal{X}$ projection indicate the source bin in $\mathcal{Z}$ for a given sample.

is likelihood-based which precludes it from applications where the latent prior is not analytically known. The interested reader can find more information on these distinctions in the following reference[45].

The WAE method[21] provides a general framework for an autoencoder whose training is based on ideas from OT theory, namely the Wasserstein distance. This work defined a large umbrella under which a rich amount of subsequent literature falls (e.g. SWAE[20], Sinkhorn Autoencoders[46], CWAE[47]). The key difference between these methods and the original WAE method is the fact that each chooses a different $d_z(\cdot,\cdot)$ cost function. Therefore, the choice of method largely comes down to finding a suitable d_z for the given problem.

The original WAE work proposes two specific options for the d_z, defining two versions of WAE: GAN-WAE and MMD-WAE. The first is an adversarial approach in which d_z is the Jensen–Shannon divergence estimated using a discriminator network. The second chooses d_z to be the Maximum Mean Discrepancy (MMD)[21].

The GAN-WAE strategy suffers from the same practical issues as other adversarial methods such as GANs (i.e. mode collapse). This possibility of training instability makes it an undesirable choice. The MMD-WAE does not have this training instability issue but requires an a priori choice of a kernel for the form of latent space prior, $p(z)$. This implies that we analytically know the desired prior form ahead of time, which is not the case for particle physics in general. Therefore, this option will not work for the applications explored in this work.

We now explore WAE derivatives which choose other choices for d_z that might be more amenable to our application. CWAE[47] chooses the Cramer-Wold distance as the d_z cost function. For a Gaussian latent space prior, this provides a computationally efficiency boost due to the existence of a closed-form solution. However, this assumption makes it unsuitable for our current application because our latent prior, $p(z)$, is non-Gaussian and often does not have a form which is known analytically a priori.

Two other derivatives allow for a flexible prior form which would be suitable for the task at hand. SWAE[20] chooses the d_z cost function to be the SW distance and Sinkhorn Autoencoder (SAE)[46] chooses it to be the Sinkhorn divergence which is estimated via the Sinkhorn algorithm. Both have comparable performance with trade-offs in performance and computational efficiency. SAE claims superior performance to SWAEs for Gaussian priors, while it is slightly more computationally intensive ($\mathcal{O}(M^2)$ as opposed to SWAEs best case $\mathcal{O}(M)$ or worst case $\mathcal{O}(M\log M)$). However, both methods are valid choices for this application. Therefore, we suggest that SAE performance on this task be explored in future work.

We also note the existence of other WAE-derivative methods which generalize the underlying OT framework. In our application, the d_z metric always compares distributions in the same ambient space $\mathcal{Z}$. Additionally, the overall loss function also approximates the Wasserstein distance between two distributions in the same ambient space $\mathcal{X}$, namely $W_c(p(x), p_D(x))$. However, recent work using the Gromov–Wasserstein distance[48] extends the underlying Optimal Transport (OT) framework to situations where the two probability measures μ and ν are not defined on the same ambient space (e.g. $\mathbb{R}^n$ and $\mathbb{R}^m$ with different dimensions n and m). For this application, this is an over-powered tool since by construction $p(z)$ and $p_E(z)$ ($p(x)$ and $p_D(x)$) always lie in the same ambient space. However, if one were attempting to study the optimal transportation between different spaces, this would be ideal. This would be an interesting direction to follow-up recent related work which connects OT and particle physics[36,49].

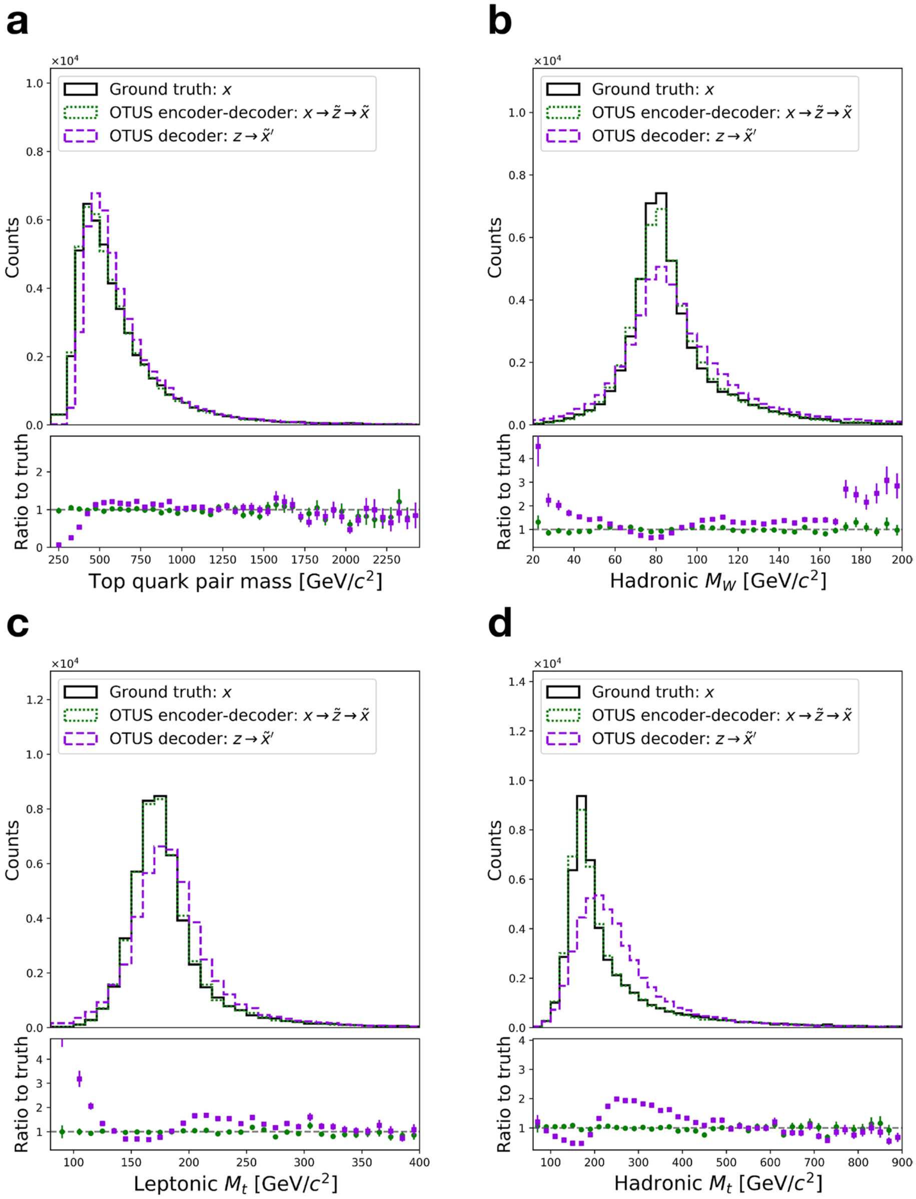

Figure 8. Performance of OTUS for semileptonic $t\bar{t}$ decays in physically important derived quantities in $\mathcal{X}$. (**a**) Matching of the invariant mass of the combined $t\bar{t}$ pair. (**b**) Matching of the invariant mass of the hadronically decaying W-boson, M_W. (**c**) Matching of the invariant mass of the top-quark, M_t, reconstructed using information from the leptonically decaying W-boson. (**d**) Matching of the invariant mass of the top-quark, M_t, reconstructed using information from the hadronically decaying W-boson. These show the testing sample $\{x \sim p(x)\}$ (solid black) in the experimental space, $\mathcal{X}$, as well as output from the decoder applied to samples drawn from $p(z)$, labeled as $z \to \tilde{x}'$ (dashed purple). Also shown are samples passed through both the decoder and encoder chain, $x \to \tilde{z} \to \tilde{x}$ (dotted green). Dotted green and solid black distributions are matched explicitly during training. Enhanced differences between dashed purple and solid black indicate the encoder's output needs improvement, as $p_E(z)$ does not fully match $p(z)$. Residual plots show bin-by-bin ratios with statistical uncertainties propagated accordingly (see "Evaluation").

Base model. Both the encoder and decoder models of OTUS are implicit conditional generative models, and operate by concatenating the input with random noise and passing the resulting vector through feedforward neural networks.

For a model, G, mapping from a space, $\mathcal{U}$, to a space, $\mathcal{V}$, the steps are as follows. (1) A sample of raw input data, $u \in \mathcal{U}$, is standardized by subtracting the mean and dividing by the standard deviation resulting in the standardized data vector, $\bar{u}$. (2) A noise neural network computes a conditional noise distribution $p_N(\epsilon \mid u)$, where the noise vector $\epsilon \sim p_N(\epsilon \mid u)$ has the same dimensionality as the core network prediction $\bar{w}$ (defined in the next step). (3) The standardized data vector, $\bar{u}$, and noise vector, ϵ, are then concatenated and fed into a core neural network. This network outputs the 3-momentum, $\mathbf{p}$, information of each particle in the standardized space, collected into a vector $\bar{w}$. (4) The vector $\bar{w}$ is then unstandardized by inverting the relationship in step 1, creating a vector w. (5) The Minkowski relation (see "Our approach: OTUS") is then enforced explicitly to reinsert the energy information of each particle, transforming w into the final $v \in \mathcal{V}$ which is distributed according to $p_G(v \mid u)$.

Both the encoding and decoding model's noise networks produce Gaussian-distributed noise vectors with mean and diagonal covariances $[\mu(x), \sigma^2(x))]$ and $[\mu(z), \sigma^2(z))]$ respectively. For the $Z \to e^+e^-$ study, the core and noise networks for both the encoder and decoder each used a simple feed-forward neural network architecture with a single hidden layer, with 128 hidden units and ReLU activation.

Model for semileptonic top-quark decay study. To better model the complexities in the semileptonic $t\bar{t}$ data, we introduced a restriction to the decoder model and modified the training procedure accordingly (see "Training"). With these modifications, the base model encountered difficulty during training, so we introduced the following three changes to the architecture for more effective training.

First, the conditionality of the noise network is removed and the noise is instead drawn from a fixed standard normal distribution, $p_N(\epsilon \mid u) = p_N(\epsilon) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Second, the model now has a residual connection such that the core network now predicts the change from the input u. The 3-momentum sub-vector of u is added to w before proceeding to imposing the Minkowski relation in step 5. This input-to-output residual connection provides an architectural bias towards identity mapping, when the model is initialized with small random weights.

Lastly, the core network itself is augmented with residual connections[50] and batch normalization[51]. An input vector to the core network is processed as follows: (A) A linear transform layer with K units maps the input to a vector $r \in \mathbb{R}^K$. (B) Two series of [BatchNorm, ReLU, Linear] layers are applied sequentially to r, without changing the dimensionality, resulting in $s \in \mathbb{R}^K$. (C) A residual connection from r is introduced, so that $s \to s + r$. (D) The resulting s is then transformed by a final linear layer with J units to obtain the output vector $t \in \mathbb{R}^J$. For the $t\bar{t}$ study, the input vector $[\bar{u}, \epsilon]$ is $24 + 18 = 42$ dimensional, the output dimension $J = 18$, and we set $K = 64$ for the core network, in both the encoder and decoder models.

Training.

Base training strategy. As described in "Our approach: OTUS", the model is trained by minimizing the SWAE loss function augmented with anchor terms

$$\begin{aligned}\mathcal{L}_{\text{SWAE}}(p(x), p_D(x \mid z), p_E(z \mid x)) = {} & \mathbb{E}_{x \sim p(x)} \mathbb{E}_{z \sim p_E(z|x)} \mathbb{E}_{\tilde{x} \sim p_D(x|z)} [c(x, \tilde{x})] + \lambda d_{\text{SW}}(p_E(z), p(z)) \\ & + \beta_E \mathcal{L}_A(p(x), p_E(z \mid x)) + \beta_D \mathcal{L}_A(p(z), p_D(x \mid z)),\end{aligned} \tag{10}$$

with respect to parameters of the encoder $p_E(z \mid x)$ and decoder $p_D(x \mid z)$ distributions.

As each term in the loss function has the form of an expectation, we approximate each with samples and compute the following Monte-Carlo estimate of the loss:

$$\begin{aligned}\hat{\mathcal{L}}_{\text{SWAE}} = {} & \frac{1}{M} \sum_{m=1}^{M} c(x_m, \tilde{x}_m) + \lambda \frac{1}{L * M} \sum_{l=1}^{L} \sum_{m=1}^{M} c((\theta_l \cdot z_m)_{\text{sorted}}, (\theta_l \cdot \tilde{z}_m)_{\text{sorted}}) \\ & + \beta_E \frac{1}{M} \sum_{m=1}^{M} c_A(x_m, \tilde{z}_m) + \beta_D \frac{1}{M} \sum_{m=1}^{M} c_A(z_m, \tilde{x}'_m),\end{aligned} \tag{11}$$

where $\{x_m\}_{m=1}^M$ and $\{z_m\}_{m=1}^M$ are M instances of $\mathcal{X}$ and $\mathcal{Z}$ samples, $\{\tilde{z}_m \sim p_E(\cdot \mid x_m)\}_{m=1}^M$ are drawn from the encoder, $\{\tilde{x}'_m \sim p_D(\cdot \mid z_m)\}_{m=1}^M$ are drawn from the decoder, and $\{\tilde{x}_m \sim p_D(\cdot \mid \tilde{z}_m)\}_{m=1}^M$ are drawn from the auto-encoding chain $x \to \tilde{z} \to \tilde{x}$. [This is equivalent to drawing a sample $(x, \tilde{z}, \tilde{x})$ from the joint distribution $p(x) p_E(\tilde{z} \mid x) p_D(\tilde{x} \mid \tilde{z})$.] The estimation of $d_{\text{SW}}(p(z), p_E(z))$ uses L random slicing directions $\{\theta_l\}_{l=1}^L$ drawn uniformly from the unit sphere, along which the samples $z_m \sim p(z)$ and $\tilde{z}_m \sim p_E(z)$ are compared; this involves estimating each CDF^{-1} by sorting the two sets of projections in ascending order as $\{(\theta_l \cdot z_m)_{\text{sorted}}\}_{m=1}^M$ and $\{(\theta_l \cdot \tilde{z}_m)_{\text{sorted}}\}_{m=1}^M$, for each direction θ_l; we refer interested readers to[20] for more technical details of the Sliced Wasserstein distance. We use the squared norm as the cost metric $c(u, v) = ||u - v||^2$ in the SWAE loss[20]. The anchor cost, c_A, between two observation vectors u, v (which can reside in either $\mathcal{X}$ or $\mathcal{Z}$ space) is defined as $c_A(u, v) := 1 - \hat{\mathbf{p}}_u \cdot \hat{\mathbf{p}}_v$, where $\hat{\mathbf{p}}_u$ is the unit vector of the coordinates of u corresponding to the momentum of a pre-specified particle, and $\hat{\mathbf{p}}_v$ is defined analogously with respect to the same particle; this is chosen as the electron in our experiments. For example, $c_A(x, \tilde{z})$ would be computed as

$$c_A(x, \tilde{z}) = 1 - \hat{\mathbf{p}}_x^{e-} \cdot \hat{\mathbf{p}}_{\tilde{z}}^{e-} = 1 - \frac{\mathbf{p}_x^{e-}}{\|\mathbf{p}_x^{e-}\|} \cdot \frac{\mathbf{p}_{\tilde{z}}^{e-}}{\|\mathbf{p}_{\tilde{z}}^{e-}\|}. \tag{12}$$

At a higher level, the computation of $\hat{\mathcal{L}}_{\text{SWAE}}$ based on a mini-batch proceeds as follows. Following the path through the full model, a batch of samples $X \sim p(x)$ from $\mathcal{X}$ space is passed to the encoder model, E, producing $\tilde{Z} \in \mathcal{Z}$ distributed according to $p_E(z \mid x)$. The encoding anchor loss term $L_{A,E}(X, \tilde{Z}) \equiv \mathcal{L}_A(p(x), p_E(z \mid x))$ is then computed along with the SW distance latent loss, $\hat{d}_{\text{SW}}(Z, \tilde{Z}) \equiv \hat{d}_{\text{SW}}(p(z), p_E(z))$. The samples $\tilde{Z}$ and $Z \sim p(z)$ are then passed independently in parallel through the decoder model, D, producing $\tilde{X}$ and $\tilde{X}'$, respectively. The decoding anchor loss term $L_{A,D}(Z, \tilde{X}') \equiv \mathcal{L}_A(p(z), p_D(x \mid z))$ is then computed. Finally, the data space loss, chosen to be MSE$(X, \tilde{X})$, is computed. See Supplementary Fig. 1 for a visual representation. We can then minimize the tractable Monte-Carlo estimate of the objective, $\hat{\mathcal{L}}_{\text{SWAE}}$, by stochastic gradient descent with respect to parameters of the encoder and decoder networks.

Since the original (S)WAE aimed to ultimately minimize $d_W(p(x), p_D(x))$ via an approximate variational formulation, [When minimized over all $p_E(z \mid x)$ that satisfies the constraint $p_E(z) = p(z)$, term A of Eq. (1) becomes an upper bound on $d_W(p(x), p_D(x))$; the bound is tight for deterministic decoders[21]. The overall WAE loss $\mathcal{L}_{\text{WAE}}$ is a relaxation of the exact variational bound, and recovers the latter as $\lambda \to \infty$.] we also consider an auxiliary strategy of directly minimizing the more computationally convenient SW distance $d_{\text{SW}}(p(x), p_D(x))$ to train a decoder, or minimizing $d_{\text{SW}}(p(z), p_E(z))$ to train an encoder. This can be done by simply optimizing the Monte-Carlo estimates

$$\hat{d}_{\text{SW}}(p(x), p_D(x)) = \frac{1}{L * M} \sum_{l=1}^{L} \sum_{m=1}^{M} c((\theta_l \cdot x_m)_{\text{sorted}}, (\theta_l \cdot \tilde{x}'_m)_{\text{sorted}}) \tag{13}$$

$$\hat{d}_{\text{SW}}(p(z), p_E(z)) = \frac{1}{L * M} \sum_{l=1}^{L} \sum_{m=1}^{M} c((\theta_l \cdot z_m)_{\text{sorted}}, (\theta_l \cdot \tilde{z}_m)_{\text{sorted}}), \tag{14}$$

where the samples $\{x_m, z_m, \tilde{x}'_m, \tilde{z}_m\}_{m=1}^{M}$ are defined the same way as before. Auxiliary training of the encoder was found helpful for escaping local minima when optimizing the joint loss $\hat{\mathcal{L}}_{\text{SWAE}}$, and auxiliary fine-tuning of the decoder in post-processing also improved the decoder's fit to the data.

Note that the idea of training a decoder by itself is similar in spirit to GANs, but again with the major distinction and innovation that we use samples from a physically meaningful prior $p(z)$ instead of an uninformed generic one (e.g. Gaussian), as we are also interested in a physical conditional mapping $p_D(x \mid z)$ in addition to achieving good fit to the marginal $p(x)$.

Training an (S)WAE with a restricted decoder. As was previously explained, experimental limitations in the semileptonic $t\bar{t}$ study require a minimum threshold, so that jets which have $p_{\text{T}} < 20$ [GeVc^{-1}] are discarded and therefore are not represented in the training dataset, as they would not be available in control region data. Denoting the region of $\mathcal{X}$ space which passes this threshold by S, we are faced with the task of fitting a distribution $p_D(x)$ over $\mathcal{X}$ while only having access to data samples in the valid subset $S \subset \mathcal{X}$.

We propose a general method for fitting an (S)WAE such that its marginal data distribution $p_D(x)$, when restricted to the valid set S, matches that of the available data. We first define the restricted marginal data distribution,

$$\bar{p}_D(x) = \frac{p_D(x)\mathbf{1}_S(x)}{P_D(S)}, \tag{15}$$

where $\mathbf{1}_S(x)$ is the indicator function of S so that it equals 1 if $x \in S$, and 0 otherwise, and $P_D(S) := \int dt p_D(t)\mathbf{1}_S(t)$ normalizes this distribution. Note that $P_D(S)$ depends on the decoder parameters, and can be identified as the probability that the data model $p_D(x)$ yields a valid sample $x \in S$.

Our goal is then to minimize $d_W(p(x), \bar{p}_D(x))$. This can be done by minimizing the same variational upper bound as in a typical (S)WAE, but with an adjustment to the data loss function in term A of Eq. (1), so it becomes

$$\mathbb{E}_{x \sim p(x)} \mathbb{E}_{p_E(z|x)} \mathbb{E}_{\tilde{x} \sim \bar{p}_D(x|z)}[c(x, \tilde{x})] \to \mathbb{E}_{x \sim p(x)} \mathbb{E}_{p_E(z|x)} \mathbb{E}_{\tilde{x} \sim p_D(x|z)} \left[\frac{\mathbf{1}_S(\tilde{x})}{P_D(S)} c(x, \tilde{x}) \right]. \tag{16}$$

Letting θ denote the parameters of the model, it can be shown that the gradient of the modified cost function has the simple form

$$\nabla_\theta \frac{\mathbf{1}_S(\tilde{x})}{P_D(S)} c(x, \tilde{x}) = \frac{\mathbf{1}_S(\tilde{x})}{P_D(S)} \nabla_\theta c(x, \tilde{x}). \tag{17}$$

This means that training an (S)WAE with a restricted decoder by stochastic gradient descent proceeds as in the unrestricted base training strategy, except that only the valid samples in S contribute to the gradient of the data loss term, with the contribution scaled inversely by the factor $P_D(S)$, which can be estimated by drawing samples $\tilde{x}'_m \sim p_D(x)$ [This is equivalent to passing $z_m \sim p(z)$ through the decoder to produce $\tilde{x}'_m$.] and forming the Monte-Carlo estimate

$$P_D(S) \approx \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}_S(\tilde{x}'_m). \tag{18}$$

Parameter optimization. For the $Z \to e^+e^-$ study, we used the base training strategy. We optimized $\hat{\mathcal{L}}_{\text{SWAE}}$ for 80 epochs with anchor penalties $\beta_E = \beta_D = 50$, followed by another 800 epochs with the anchor penalties set to 0. For the semileptonic $t\bar{t}$ study, we modified the base training strategy to accommodate a restricted decoder, substituting all appearances of $p_D(x)$ in the loss $\hat{\mathcal{L}}_{\text{SWAE}}$ by $\bar{p}_D(x)$ (e.g. using the modified data loss term Eq. (16)). We optimized the resulting loss $\hat{\mathcal{L}}_{\text{SWAE}}$ till convergence, for about 1000 epochs. Then we froze the encoder and fine-tuned the decoder by minimizing $\hat{d}_{\text{SW}}(p(x), \bar{p}_D(x))$ for 10 epochs, with a reduced learning rate. The input-to-output residual connection (see "Model") in the $t\bar{t}$ model allowed for sufficiently high $P_D(S) \approx 0.6$ and reliable gradient estimates during training, and the architectural bias towards identity mapping made the anchor losses redundant, so we set $\beta_E = \beta_D = 0$.

In both studies, we found that a sufficiently large batch size significantly improved results. This is likely do to increasing the accuracy of gradient estimates for stochastic gradient descent and also the CDF^{-1} in the SWAE latent loss. In all of our experiments, we used the Adam optimizer[52] with $L = 1000$ number of slices, a batch size of $M = 20{,}000$, and learning rate of 0.001. We tuned the λ hyperparameter of the (S)WAE loss $\hat{\mathcal{L}}_{\mathrm{SWAE}}$ on the validation set; we set $\lambda = 1$ for the $Z \to e^+e^-$ model, and $\lambda = 20$ for the $t\bar{t}$ model.

Evaluation. This section provides details on the various qualitative and quantitative evaluation techniques used in this work.

As common in the literature[11,12,14], we visualize our results along informative one-dimensional projections using histograms (e.g. Figs. 3, 5). We choose the bin sizes such that the error on the counts can be approximated as Gaussian distributed. These histograms are accompanied by residual plots, showing the ratio between the histograms from generated samples and the histogram from true samples, with accompanying statistical errors[53][Specifically, for a bin with counts h_1 and h_2, respectively, the error on the ratio, $r = h_2/h_1$ is $\sigma_r = r\sqrt{\frac{1}{h_2} + \frac{1}{h_1}}$.]. We also visualize the generative mappings using transportation plots (e.g. Fig. 4) that allow us to confirm the physicality of the learned mappings.]

In addition to qualitative comparisons, we also evaluated the results using several quantitative metrics. To this end, we calculate the Monte-Carlo estimate of the SW distance, $\hat{d}_{\mathrm{SW}}(\cdot,\cdot)$, using $L = 1000$ slices according to the cost metric $c(u, v) = ||u - v||^2$. The results are reported for each study in the text. In addition, we apply several statistical tests on the considered one-dimensional projections, which we report in Supplementary Tables 1–4. First, we calculate the reduced χ^2, χ^2_R, for each comparison and report it along with the degrees-of-freedom (dof). Second, we calculate the unbinned two-sample, two-sided Kolmogorov–Smirnov distance. Lastly, we calculate the Monte-Carlo estimate of the Wasserstein distance, $\hat{d}_W(\cdot,\cdot)$, according to the cost metric $c(u, v) = ||u - v||^2$. All statistical tests were carried-out using two separate test sets not used during training or validation of the networks. The number of samples in each test set were 80,000 in the $Z \to e + e-$ study and 47,856 in the semileptonic $t\bar{t}$ study.[Note that the number of samples in the semileptonic $t\bar{t}$ study is lower due to the hard p_{T} cutoff constraint as described in "Demonstration in semileptonic top-quark decays". The events present are ones that passed this cutoff constraint.]

Conclusion

OTUS is a data-driven, machine-learned, predictive simulation strategy which suggests a possible new direction for alleviating the prohibitive computational costs of current Monte-Carlo approaches, while avoiding the inherent disadvantages of other machine-learned approaches. We anticipate that the same ideas can be applied broadly outside of the field of particle physics.

In general, OTUS can be applied to any process where unobserved latent phenomena $\mathcal{Z}$ can be described in the form of a prior model, $p(z)$, and are translated to an empirical set of experimental data, $\mathcal{X}$, via an unknown transformation. For example, in molecular simulations in chemistry observations could be measurements of real-world molecular dynamics, $p(z)$ would represent the model description of the system, and $p(x \mid z)$ would model the effects of real-world complications[7]. In cosmology, $\mathcal{X}$ could be the distribution of mass in the observed universe, $p(z)$ could describe its distribution in the early universe, and $p(x \mid z)$ would model the universe's unknown expansion dynamics (e.g. due to inflation)[9,54]. In climate simulations, $p(z)$ could correspond to the climate due to a physical model, while $p(x \mid z)$ takes unknown geography-specific effects into account[8]. Additionally, an immediate and promising application of OTUS is in medical imaging, which uses particle physics simulations to model how the imaging particles (e.g X-rays) interact with human tissue and suffers from the great computational cost of these simulations[10]. We note that our method assumes a high degree of mutual information between $\mathcal{Z}$ and $\mathcal{X}$ in the desired application. Therefore, in situations where such mutual information is low (e.g. chaotic turbulent flows) the transformations learned by this method would likely be less reliable.

Moreover, features of this method can be adapted to suit the particular problem's needs. For example, in this work we were interested in low-dimensional data, however the method could also be applied to high-dimensional datasets. Moreover, the encoding and decoding mappings can be stochastic, as in this work, or deterministic. Lastly, while this work aimed to be completely unsupervised, and thus data-driven, OTUS can be easily extended to a semi-supervised setting. In this case, the data would consist mostly of unpaired samples but would have a limited number of paired examples (z, x) (e.g. from simulation runs). These pairs sample the joint distribution, $p(z, x)$, which, combined with the decoder $p_D(\tilde{x} \mid z)$, yields a transportation map γ between $p(x)$ and $p_D(\tilde{x})$, $\gamma(p(x), p_D(\tilde{x})) := \int dz p(z, x) p_D(\tilde{x} \mid z)$. Since calculating the Wasserstein distance between $p(x)$ and $p_D(\tilde{x})$ involves finding the optimal transportation map, this particular choice yields an upper bound on the Wasserstein distance. We can similarly construct a transportation map between $p(z)$ and $p_E(\tilde{z})$ using $p(z, x)$ and $p_E(z \mid x)$. This makes directly optimizing the Wasserstein distances $d_W(p(x), p_D(x))$ and $d_W(p(z), p_E(z))$ tractable in this high-dimensional setting. Therefore, we get the alternative objectives

$$\mathcal{L}_{\mathrm{paired}}(p_D(x \mid z), p(z, x)) = \mathbb{E}_{(z,x)\sim p(z,x)} \mathbb{E}_{\tilde{x}\sim p_D(x|z)}[c(x, \tilde{x})] \tag{19}$$

$$\mathcal{L}_{\mathrm{paired}}(p_E(z \mid x), p(z, x)) = \mathbb{E}_{(z,x)\sim p(z,x)} \mathbb{E}_{\tilde{z}\sim p_E(z|x)}[c(z, \tilde{z})], \tag{20}$$

which are upper bounds on $d_W(p(x), p_D(x))$ and $d_W(p(z), p_E(z))$ respectively. These terms can be incorporated alongside the unsupervised SWAE loss, to leverage paired examples $\{(z, x) \sim p(z, x)\}$ in a semi-supervised setting.

We have demonstrated the ability of OTUS to learn a detector transformation in an unsupervised way. The results, while promising for this initial study, leave room for improvement. Several directions could lead to higher fidelity descriptions of the data and latent spaces.

First, the structure of the latent and data spaces can significantly affect the performance and physicality of the resulting simulations. Particle physics data has rich structures often governed by group symmetries and conservation laws. Our current vector format description of the data omits much of this complicated structure. For example, we omitted categorical characteristics of particles like charge and type. Knowledge of such properties and the associated rules likely would have excluded the necessity of terms like the anchor loss. Therefore, future work should explore network architectures and losses that can better capture the full nature of these data structures[38,55].

The next technical hurdle is the ability to handle variable input and output states. The same $p(z)$ can lead to different detected states as was described, but not explored, in the semileptonic $t\bar{t}$ study where the number of jets can vary. Additionally, it should be possible to handle mixtures of underlying priors in the latent space. This can cause the number and types of latent-space particles to vary from one sample to another. For example, the Z boson can decay into $Z \rightarrow \mu^+\mu^-$ in addition to $Z \rightarrow e^+e^-$; a simulator should be able to describe these two cases holistically.

Finally, an essential feature of a predictive simulator is that it learns a general transformation, allowing it to make predictions for points in the latent space which lie outside of the control regions. This would require structuring the latent and data spaces to accommodates data from several control regions, such that the network may learn to interpolate between them. Since networks excel at interpolation we expect that this will be a straightforward step.

Data availibility

The datasets generated and analysed during the current study are available in the DRYAD repository, doi: 10.7280/D1WQ3R.

Code availability

The code used during the current study are available in the Zenodo repository and are linked to the dataset, 10.5281/zenodo.4706055.

Received: 30 October 2021; Accepted: 13 April 2022
Published online: 09 May 2022

References

1. Cranmer, K., Brehmer, J. & Louppe, G. The frontier of simulation-based inference. *Proc. Natl. Acad. Sci.* **117**, 30055–30062 (2020).
2. Agostinelli, S. *et al.* GEANT4—a simulation toolkit. *Nucl. Instrum. Methods Phys. Res. Sect. A* **506**, 250–303 (2003).
3. Sjostrand, T., Mrenna, S. & Skands, P. Z. PYTHIA 6.4 physics and manual. *J. High Energy Phys.* **2006**(05), 026 (2006).
4. de Faverea, J. *et al.* DELPHES 3: a modular framework for fast simulation of a generic collider experiment. *J. High Energy Phys.* **2006**(02), 057 (2014).
5. Aad, G. *et al.* The ATLAS simulation infrastructure. *Eur. Phys. J. C* **70**, 823–874 (2010).
6. CMS Collaboration, Bayatian, G. L., et al. CMS physics: technical design report volume 1: detector performance and software (2006).
7. Noé, F., Tkatchenko, A., Müller, K. R. & Clementi, C. Machine learning for molecular simulation. *Annu. Rev. Phys. Chem.* **71**, 361–390 (2020).
8. Rolnick, D., et al. Tackling climate change with machine learning. Preprint at https://arxiv.org/abs/1906.05433 (2019)
9. Delaunoy, A., et al. Lightning-fast gravitational wave parameter inference through neural amortization. Preprint at https://arxiv.org/abs/2010.12931 (2020)
10. Zhou, J., Huang, B., Yan, Z. & Bünzli, J. C. G. Emerging role of machine learning in light-matter interaction. *Light Sci. Appl.* **8**, 84 (2019).
11. Paganini, M., de Oliveira, L. & Nachman, B. CaloGAN: simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial network. *Phys. Rev. D* **97**, 014021 (2018).
12. Butter, A., Plehn, T. & Winterhalde, R. How to GAN LHC events. *SciPost Phys.* **7**, 75 (2019).
13. Otten, S. *et al.* Event generation and statistical sampling for physics with deep generative models and a density information buffer. *Nat. Commun.* **12**, 2985 (2021).
14. de Oliveira, L., Paganini, M. & Nachman, B. Learning particle physics by example location-aware generative adversarial networks for physics synthesis. *Comput. Softw. Big Sci.* **1**, 4 (2017).
15. Buhmann, E. *et al.* Getting high: high fidelity simulation of high granularity calorimeters with high speed. *Comput. Softw. Big Sci.* **5**, 13 (2021).
16. Deja, K., Dubiński, J., Nowak, P., Wenzel, S. & Trzciński, T. End-to-end sinkhorn autoencoder with noise generator. *IEEE Access* **9**, 7211–7219 (2021).
17. Hashemi, B., Amin, N., Datta, K., Olivito, D. & Pierini, M. LHC analysis-specific datasets with generative adversarial networks. Preprint at https://arxiv.org/abs/1901.05282 (2019).
18. Lu, Y., Collado, J., Whiteson, D. & Baldi, P. Sparse autoregressive models for scalable generation of sparse images in particle physics. *Phys. Rev. D* **103**, 036012 (2021).
19. Andreassen, A., Feige, I., Frye, C. & Schwartz, M. JUNIPR: a framework for unsupervised machine learning in particle physics. *Eur. Phys. J. C* **79**, 2 (2019).
20. Kolouri, S., Pope, P. E., Martin, C. E. & Rohde G. K. Sliced-wasserstein autoencoder: an embarrassingly simple generative model. Preprint at https://arxiv.org/abs/1804.01947 (2018).
21. Tolstikhin, I., Bousquet, O., Gelly, S. & Schoelkopf, B. Wasserstein auto-encoders. Preprint at https://arxiv.org/abs/1711.01558 (2017).
22. Kingma, D. P. & Welling M. Auto-encoding variational Bayes. Preprint at https://arxiv.org/abs/1312.6114 (2013).
23. Pagnoni, A., Liu, K. & Li, S. Conditional variational autoencoder for neural machine translation. Preprint at https://arxiv.org/abs/1812.04405 (2018).
24. Abazov, V. *et al.* A precision measurement of the mass of the top quark. *Nature* **429**, 638–642 (2004).
25. Bellagente, M. *et al.* Invertible networks or partons to detector and back again. *SciPost Phys.* **9**, 74 (2020).
26. Andreassen, A., Komiske, P. T., Metodiev, E. M., Nachman, B. & Thaler, J. OmniFold: a method to simultaneously unfold all observables. *Phys. Rev. Lett.* **124**, 182001 (2020).

27. Aad, G. *et al.* Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys. Lett. B* **716**, 1–29 (2012).
28. Chatrchyan, S. *et al.* Observation of a New Boson at a mass of 125 GeV with the CMS experiment at the LHC. *Phys. Lett. B* **716**, 30–61 (2012).
29. Castro, A. (on behalf of the ATLAS and CMS Collaborations). Top Quark Mass Measurements in ATLAS and CMS. Preprint at https://arxiv.org/abs/1911.09437 (2019).
30. Zhang, C., Bütepage, J., Kjellström, H. & Mandt, S. Advances in variational inference. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**, 2008–2026 (2018).
31. Arjovsky, M., Chintala, S. & Bottou L. Wasserstein GAN. Preprint at https://arxiv.org/abs/1701.07875 (2017).
32. Burgess, C. P., et al. Understanding disentangling in β-VAE. Preprint at https://arxiv.org/abs/1804.03599 (2018).
33. Salimans, T., et al. Improved techniques for training GANs. Preprint at https://arxiv.org/abs/1606.03498 (2016).
34. Baldi, P., Cranmer, K., Faucett, T., Sadowski, P. & Whiteson, D. Parameterized neural networks for high-energy physics. *Eur. Phys. J. C* **76**, 235 (2016).
35. Batson, J., Haaf, C. G., Kahn, Y. & Roberts, D. A. Topological obstructions to autoencoding. *J. High Energy Phys.* **2021**, 280 (2021).
36. Komiske, P. T., Metodiev, E. M. & Thaler, J. The hidden geometry of particle collisions. *J. High Energy Phys.* **2020**, 6 (2020).
37. Aaij, R. *et al.* Searches for low-mass dimuon resonances. *J. High Energy Phys.* **10**, 156 (2020).
38. Battaglia, P. W., et al. Relational inductive biases, deep learning, and graph networks. Preprint at https://arxiv.org/abs/1806.01261 (2018).
39. Fenton, M. J., et al. Permutationless many-jet event reconstruction with symmetry preserving attention networks. Preprint at https://arxiv.org/abs/2010.09206 (2020).
40. Erdmann, J., Kallage, T., Kröninger, K. & Nackenhorst, O. From the bottom to the top-reconstruction of $t\,\bar{t}$ events with deep learning. *J. Instrum.* **14**, P11015 (2019).
41. Erdmann, J. *et al.* A likelihood-based reconstruction algorithm for top-quark pairs and the KLFitter framework. *Nucl. Instrum. Methods Phys. Res. Sect. A* **748**, 18–25 (2014).
42. Alwall, J., Herquet, M., Maltoni, F., Mattelaer, O. & Stelzer, T. MadGraph 5: going beyond. *J. High Energy Phys.* **2011**(06), 128 (2011).
43. Brun, R. & Rademakers, F. ROOT—an object oriented data analysis framework. *Nucl. Instrum. Methods Phys. Res. Sect. A* **389**, 81–86 (1997).
44. Cacciari, M., Salam, G. & Soyez, G. The anti-k_t jet clustering algorithm. *J. High Energy Phys.* **04**, 063 (2008).
45. Bousquet, O., Gelly, S., Tolstikhin, I., Simon-Gabriel, C. & Schoelkopf, B. From optimal transport to generative modeling: the VEGAN cookbook. Preprint at https://arxiv.org/abs/1705.07642 (2017).
46. Patrini, G., et al. *Sinkhorn AutoEncoders*. Preprint at https://arxiv.org/abs/1810.01118 (2018).
47. Knop, S., Tabor, J., Spurek, P., Podolak, I., Mazur, M. & Jastrzębski, S. *Cramer-Wold AutoEncoder* Preprint at https://arxiv.org/abs/1805.09235 (2018).
48. Vayer, T., Flamary, R., Tavenard, R., Chapel, L. & Courty, N. *Sliced Gromov–Wasserstein*. Preprint at https://arxiv.org/abs/1905.10124 (2019).
49. Cai, T., Cheng, J., Craig, K. & Craig, N. Linearized optimal transport for collider events. *Phys. Rev. D* **102**, 116019 (2020).
50. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. Preprint at https://arxiv.org/abs/1512.03385 (2015).
51. Ioffe, S. & Szeged, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. Preprint at https://arxiv.org/abs/1502.03167 (2015).
52. Kingma, D. P. & Ba, J. L. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2014).
53. Bevington, P. R. & Robinson, D. K. *Data Reduction and Error Analysis for the Physical Sciences* 3rd edn. (McGraw-Hill, 2003).
54. Seljak, U., Aslanyan, G., Feng, Y. & Modi, C. Towards optimal extraction of cosmological information from nonlinear data. *J. Cosmol. Astropart. Phys.* **12**, 009 (2017).
55. Bogatskiy, A., et al. Lorentz group equivariant neural network for particle physics. Preprint at https://arxiv.org/abs/2006.04780 (2020).

Acknowledgements

J.N.H. acknowledges support by the National Science Foundation under Grants DGE-1633631 and DGE-1839285, and U.S. Department of Energy, Office of Science under the Grant DE-SC0009920. Y.Y. acknowledges funding from the Hasso Plattner Foundation. S.M. acknowledges support by the National Science Foundation under Grants 2047418, 1928718, 2003237 and 2007719, Intel, Disney, Qualcomm, the U.S. Department of Energy, Office of Science under the grant SC0022331, and the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0021. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA) or the National Science Foundation.

Author contributions

Using the CASRAI CRediT Contributor Roles Taxonomy: Conceptualization, J.N.H., S.M., D.W., Y.Y.; Data curation, J.N.H.; Formal analysis, J.N.H., Y.Y.; Funding acquisition, J.N.H., S.M., D.W.; Investigation, J.N.H., S.M., D.W., Y.Y.; Methodology, J.N.H., Y.Y.; Project administration, J.N.H., S.M., D.W.; Software, J.N.H., Y.Y.; Supervision, S.M., D.W.; Validation, Y.Y.; Visualization, J.N.H.; Writing–original draft, J.N.H., D.W.; Writing-review and editing, J.N.H., S.M., D.W., Y.Y.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-10966-7.

Correspondence and requests for materials should be addressed to J.N.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2022