Foundations of a Fast, Data-Driven, Machine-Learned Simulator

Jessica N. Howard\textsuperscript{1,*}, Stephan Mandt\textsuperscript{2}, Daniel Whiteson\textsuperscript{1}, and Yibo Yang\textsuperscript{2}

\textsuperscript{1}Department of Physics & Astronomy, UC Irvine, Irvine, CA 92627
\textsuperscript{2}Department of Computer Science, UC Irvine, Irvine, CA 92627
\textsuperscript{*}jnhoward@uci.edu

Authors are listed alphabetically.

ABSTRACT

We introduce a novel strategy for machine-learning-based fast simulators, which is the first that can be trained in an unsupervised manner using observed data samples to learn a predictive model of detector response and other difficult-to-model transformations. Across the physical sciences, a barrier to interpreting observed data is the lack of knowledge of a detector’s imperfect resolution, which transforms and obscures the unobserved latent data. Modeling this detector response is an essential step for statistical inference, but closed-form models are often unavailable or intractable, requiring the use of computationally expensive, ad-hoc numerical simulations. Using particle physics detectors as an illustrative example, we describe a novel strategy for a fast, predictive simulator called Optimal Transport based Unfolding and Simulation (OTUS), which uses a probabilistic autoencoder to learn this transformation directly from observed data, rather than from existing simulations. Unusually, the probabilistic autoencoder’s latent space is physically meaningful, such that the decoder becomes a fast, predictive simulator for a new latent sample, and has the potential to replace Monte Carlo simulators. We provide proof-of-principle results for Z-boson and top-quark decays, but stress that our approach can be widely applied to other physical science fields.

1 Introduction

From measuring the masses of particles to deducing the likelihood of life elsewhere in the Universe, an essential step in the analysis of scientific data is statistical inference — drawing conclusions about the values of a model’s parameters given a sample of observed data. Nearly universally, however, the data which would be the most powerful for inference is inaccessible. Experimenters are thus limited to indirect data gathered by detectors with finite resolution. In statistical terms, these powerful but unobservable direct data belong to a latent space. Inference then requires understanding how events in the unobserved latent space are transformed into observed data. In many cases, this transformation is highly non-trivial, involving complex physical interactions that cannot be described analytically. A numerical simulation of the transformation can bridge this gap, but is often extremely computationally expensive [1].

Particle physics, which seeks to discover the nature of matter by studying fundamental particle interactions, relies heavily on numerical simulations for inference. Particle interactions produce secondary particles which are not directly observable and often transform in flight before passing through layers of detectors whose response can help reveal the particle’s identity and momentum. The net transformation from the latent space, particles produced in the initial interaction, to the observed detector response data is stochastic, governed by quantum mechanical randomness, and has no analytic description.

Currently, the field relies on Monte-Carlo based numerical simulations of the in-flight and detection processes to generate samples of possible observed data for a given latent interaction [2–4]. This approach is enormously computationally expensive [5, 6], as it requires the propagation and simulation of every individual particle, each of which creates subsequent showers of derivative particles, often numbering in the millions or billions. Additionally, these simulations depend on hundreds of parameters which must be hand-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, as well as many other fields in the physical sciences [7–10], the computational cost of numerical simulations has become a central bottleneck. Thus, a fast, interpretable, flexible, data-driven generative model which can transform between the latent space and the observed data would be a significant breakthrough for these fields. Recent advances in the flexibility and capability of machine learning models have allowed for their application as computationally inexpensive simulators. In particle physics, applications of these techniques have made impressive steps towards this goal, but fall short in crucial ways. For example, most approaches which leverage Generative Adversarial Networks (GANs) have the ability to mimic existing, observed datasets for fixed distributions in the latent space [11–13], but are unable to generate predictions for new values of latent variables, a crucial ability for a simulator. Other efforts are conditioned on latent variables [14], but require training with labeled pairs generated by the slow Monte-Carlo generators, incurring the computational cost they seek to avoid.

In this paper, we lay the foundations and provide a proof-of-principle demonstration for a novel approach: 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 observed space, \( \mathcal{X} \), relying on theoretical priors, \( p(z) \), where \( z \in \mathcal{Z} \) and observed data \( x \in \mathcal{X} \) but, crucially, no labeled pairs, \( \{z,x\} \). Our model is an application of a new type [15, 16] of probabilistic autoencoder, which learn two mappings: encoder (data \( \rightarrow \) latent, \( p_E(z|x) \)) and decoder (latent \( \rightarrow \) data, \( p_D(x|z) \)). A typical probabilistic autoencoder (i.e. a variational autoencoder (VAE) [13, 17]) uses a *simple, unphysical* latent space, \( \mathcal{Y} \), for computational tractability during learning. However, this choice leads 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 it compromises to requiring expensive simulated pairs e.g., a conditional VAE approach [18]. The innovation in the OTUS method is to align the VAE’s latent space, \( \mathcal{Y} \), with that of our inference task, \( \mathcal{Z} \). By making this change, our decoder becomes a computationally inexpensive, conditional simulator mapping \( \mathcal{Z} \rightarrow \mathcal{X} \) as well as a tractable transfer function, \( p_D(x|z) \).

For traditional VAEs, the task of identifying \( \mathcal{Y} \) with \( \mathcal{Z} \) is non-trivial 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 VAE: the Sliced Wasserstein Autoencoder (SWAE) [15, 16], 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 for the encoder and decoder network mappings to be inherently stochastic.

We suggest that an SWAE [15] can be used to achieve the broadest goals of simulators: learning the mapping from the physical latent space to observed detector response directly from samples of observed experimental data \( x \sim p(x) \) and theoretical priors \( p(z) \) in control regions. The resulting decoder \( \mathcal{Z} \rightarrow \mathcal{X} \) can be applied as a simulator, allowing for the generation of samples of simulated detector response from latent variables in a fraction of the time. In addition, the decoder’s numerically tractable detector response function, \( p_D(x|z) \), would be in great demand for other applications, such as direct calculation of likelihood ratios via integration [19]. This approach also results in several additional valuable elements. For instance, the encoder network’s \( \mathcal{X} \rightarrow \mathcal{Z} \) mapping can be used in unfolding studies [20, 21]. In addition, the mathematical attributes of the SW distance allows for the effortless inclusion of informed constraints on the mappings. Finally, the decoder’s mapping can be easily probed and visualized to ensure a physically meaningful response.

In the following, we present the foundations for the OTUS method, initial proof-of-principle demonstrations, and discuss the steps toward scaling this to a full simulation capable of replacing current Monte-Carlo methods.

## 2 Results

We first describe the state-of-the-art particle physics simulations and clearly define the goal of predictive simulators. We then provide the theoretical details of the SWAE [15] approach to simulation. Finally, we illustrate the performance of the method in examples of increasing complexity.
2.1 Sliced Wasserstein Autoencoders as Predictive Simulators

2.1.1 Particle Physics Simulations

The primary statistical task in particle physics, as in many areas of science, is to infer the values of model parameters, $\theta$, based on observed data, $x$. For example, physicists try to infer the mass of the Higgs boson from observations at the Large Hadron Collider. Inference about $\theta$ requires a statistical model, $p(x|\theta)$, which describes the probability to make various observations, $x$, given parameter values, $\theta$. Unfortunately, due to the indirect nature of the observations and the complexity of the detectors, closed-form expressions for such models do not exist. Previous solutions to this problem have relied on numerical Monte-Carlo based simulations [2–4].

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

The transfer function, $p(x|z)$, represents the multi-staged transformation from the unobserved latent space, $Z$, to the observed data space, $X$. As the latent space particles travel they may decay or interact to produce a subsequent shower of hundreds of secondary particles. These surviving particles then pass through the detector, which contains many layers of millions of sensors resulting in a high-dimensional response, $O(10^8)$. Finally, the full detector response is used to reconstruct an estimate of the identity and momentum of the original unobserved particles in the latent space. This final, observed data space, $X$, has a very similar dimensionality to that of $Z$, usually $O(10^4)$. 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 $p(x|z)$. Instead, particle physicists use simulations as a proxy for true transfer function.

To arrive at $p(x|\theta)$, samples of $z \sim p(z|\theta)$ are transformed via simulations into effective samples of $x \sim p(x|\theta)$, approximating the integral above. The 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 $p(z|\theta)$ is well-established from previous experiments, freeing $p(x|\theta)$ of surprises. Examples of control regions include the decays of heavy bosons (e.g. $Z$) or the top quark ($t$).

The computational cost of current simulations has been identified as the dominant source of systematic uncertainties and the largest bottleneck in the ability to test new models of particle physics [22]. A computationally inexpensive, flexible predictive simulator which can reliably map from $Z$ to $X$ such that it effectively approximates $p(x|z)$ would be a tremendous breakthrough. It would revolutionize statistical analysis in particle physics and have wide applications in other fields.

2.1.2 Statement of Objective

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

The simulator should learn a stochastic transformation $Z \rightarrow X$ such that samples $z$ drawn from $p(z|\theta)_{\text{control}}$ can be transformed into samples $x$ whose distribution matches that of the observed data $x_{\text{control}}$. Additionally, these control regions should be robust so that the simulator can reasonably approximate $p(x|\theta)$ for different, but related, values of $\theta$. This is the same task which faces traditional Monte-Carlo simulators such as GEANT4 [2].

The rapidly improving power and flexibility of machine learning (ML) models at learning difficult functions across a wide array of contexts suggests that these tools could be applied to the development of a fast simulator.

\footnote{Note that the dimensionality is not necessarily equal due to the imperfect nature of the detection process. For example, $Z$ may represent 4 quarks but $X$ may only contain 3 jets.}
Figure 1. Schematic diagram outlining the problem. 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) based solutions can quickly reproduce the requisite distributions in $\mathcal{X}$ but are not conditioned on the information in $\mathcal{Z}$ which limits their scope. We introduce a new class of methods which provides the best of both worlds. Optimal Transport based Unfolding and Simulation (OTUS) provides an avenue for a fast, data-driven simulation mapping from $\mathcal{Z} \rightarrow \mathcal{X}$ as well as an unfolding mapping from $\mathcal{X} \rightarrow \mathcal{Z}$.

The objectives described above translate to four constraints on the class of ML model and methods of learning. To generate samples of $x \in \mathcal{X}$ requires a (1) generative ML method. For $z \in \mathcal{Z}$, the simulator maps $z \rightarrow x$, such that the output $x$ depends on the input $z$, meaning that our map is (2) conditional. The inherent and unknown randomness of our problem 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, any method which meets these conditions should ideally produce a simulation mapping ($\mathcal{Z} \rightarrow \mathcal{X}$) which is inspectable and physically interpretable.

Generative ML models have demonstrated the ability to produce realistic samples of data in many settings, including natural images. Generative Adversarial Networks (GANs), which transform noise into artificial data samples, have been adapted to the task of particle physics simulation for both high-level and raw detector data, which can resemble images [11–14]. Unfortunately, this means that while these GANs have shown promise for mimicking existing datasets $x$ for a fixed set of $z$, they have not learned the general transformation $z \rightarrow x$ prescribed by $p(x|z)$, and so cannot generate fresh samples $x'$ for a new set of $z'$, thus failing condition (2). Other GAN-based
would like we have little knowledge about the true underlying stochastic transforms, assuming any particular parametric density which measures the distance between the distributions. At the same time, the output of the full autoencoder, $\tilde{x}$, is harder to match this latent space is seen as an extra, unnecessary step in training [12]. Therefore, VAEs have been largely ignored in favor of GANs in the pursuit of a fast particle physics simulator. Some studies have investigated VAEs for particle physics datasets such as natural images. However, in the particle physics community, optimizing the encoding mapping to match the physical latent space, $\tilde{x}$, is governed by quantum field theory, is ill-described by standard Gaussian distributions. In particular, the prior $p(z)$ is similarly of interest in unfolding applications [20, 21].

The choice of latent space prior, $p(y)$, is usually independent of the nature of the data, and is often chosen to be a multi-dimensional Gaussian for simplicity. This choice provides sufficient expressive power even for complex datasets such as natural images. However, in the particle physics community, optimizing the encoding mapping to match this latent space is seen as an extra, unnecessary step in training [12]. Therefore, VAEs have been largely ignored in favor of GANs in the pursuit of a fast particle physics simulator. Some studies have investigated VAEs in this context, but retain the unphysical form of the latent space prior [13, 24, 25], preventing them from being conditional generators, failing requirement (2).

2.1.3 Our Approach: OTUS

Our central innovation is to construct a probabilistic autoencoder in which the latent space prior, $p(y)$, is identical to the physical latent space, $p(y) \equiv p(z) = p(z|\theta)$, for the choice of a particular model, $\theta$. In this construction, the decoder learns $p_D(x|z)$ so that it can provide precisely the desired conditional transformation, $z \rightarrow x$. Additionally, the decoder’s $p_D(x|z)$ can be used as a tractable transfer function in approaches which estimate $p(x|\theta)$ via direct integration [26]. The encoder’s learned $p_E(\hat{y}|x)$ is similarly of interest in unfolding applications [20, 21].

The primary obstacle to using a VAE in this manner, and likely the cause for its lack of earlier investigation, is the requirement that the densities $p(y), p_E(\hat{y}|x),$ and $p_D(x|y)$ can be computed explicitly in order to optimize its variational objective. In most existing applications, the probability models $p(y), p_E(\hat{y}|x),$ and $p_D(x|y)$ are multivariate Gaussian distributions. In particular, the prior $p(y)$ 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 ill-described by standard Gaussian distributions and computing its density explicitly typically requires an expensive numerical procedure. Similarly, as we have little knowledge about the true underlying stochastic transforms, assuming any particular parametric density model for $p_E(\hat{y}|x)$ or $p_D(x|y)$, such as a multivariate Gaussian, would be inappropriate and overly restrictive. These concerns led us to use inherently stochastic, or implicit, models for $p(z), p_E(\hat{y}|x),$ and $p_D(x|z)$ that are fully sample driven.

Additionally, the VAE objective introduces technical disadvantages stemming from its formulation using KL-divergences. The KL-divergence, $D_{KL}(\cdot || \cdot)$, is not a true distance metric, and will diverge for non-overlapping distributions often leading to unusable gradients during training [15, 27]. Moreover, the specific use of $D_{KL}(p_E(\hat{y}|x)||p(z))$ within the VAE loss forces $p_E(\hat{y}|x)$ to match $p(z)$ for every value of $x \sim p(x)$ [16]. This means careful tuning of the weight of this term (e.g., with a $\beta$-VAE approach [13, 28]) is often needed to avoid the undesirable effect of the encoder mapping different parts of $\mathbb{X}$ to the same overlapping region in $\mathbb{Y}$. This is particularly problematic if we would like $\mathbb{Y}$ to represent a physically meaningful latent space.

We resolve the above issues by applying an emerging class of probabilistic autoencoders, based on a fundamentally different metric, the Wasserstein distance, which is a well-behaved distance metric between arbitrary probability distributions rooted in concepts from Optimal Transport theory [15, 16]. The original Wasserstein Autoencoder
(WAE) [16] uses this metric to reimagine the VAE objective. The loss function is

\[
\mathcal{L}_{\text{WAE}}(p(x), p_d(x|z), p_e(z|x)) = \mathbb{E}_{x \sim p(x)} \mathbb{E}_{p_e(z|x)} \mathbb{E}_{x \sim p_d(x|z)} [c(x, \tilde{x})] + \lambda \; d_c(p_e(z), p(z))
\]

(1)

where \(\mathbb{E}\) denotes the expectation operator and \(c(\cdot, \cdot)\) is a cost metric. For the optimal \(p_e(z|x)\), \(\mathcal{L}_{\text{WAE}}\) becomes an upper bound on the Wasserstein distance between the true data distribution, \(p(x)\), and the decoder’s learned distribution, \(p_d(x)\); the bound is tight for deterministic decoders.

The data-space term (Eq. 1) constrains the decoder mapping assuming a given encoder mapping, while the latent-space term (Eq. 2) constrains the encoder mapping. The factor \(\lambda\) is a hyperparameter providing a relative weighting between the two terms. The difference between the marginal encoding distribution, \(p_e(z) = \int dx p_e(z|x) p(x)\), and the latent prior, \(p(z)\), is measured by \(d_c(\cdot, \cdot)\). 2 Unfortunately, both of the originally proposed options for \(d_c(\cdot, \cdot)\) [16] had undesirable features which made them ill-suited to solve this particle physics problem.

More recent work proposed the Sliced Wasserstein Autoencoder (SWAE) [15], which uses the Sliced Wasserstein (SW) distance as the \(d(\cdot, \cdot)\) metric. The SW distance, \(d_{SW}(\cdot, \cdot)\), is a mathematically rigorous approximation to the Wasserstein distance, \(d_W(\cdot, \cdot)\). The SWAE method completely grounds all parts of the loss function in Optimal Transport theory as the latent-space, data-space, and combined losses can all be identified as approximations to Wasserstein distances between various distributions. The SWAE method allows \(p(y)\) to be any sampleable distribution, including the prior over the parton-level latent space, \(p(z)\). Additionally, the (S)WAE method gives us the freedom to allow the encoder and decoder to be implicit probability models, while avoiding an adversarial training strategy which can lead to undesired effects such as mode collapse [29].

Both \(d_W\) and \(d_{SW}\) are true distance metrics [15]. In contrast, the KL-divergence, as well as adversarial schemes, do not have this property and can result in divergences and meaningless loss values. This introduces undesirable hurdles during training and makes it difficult to include additional, physically-motivated constraints. The Wasserstein distance measures the transportation cost of probability mass from one probability distribution to another according to a cost metric, \(c(\cdot, \cdot)\). The Wasserstein distance itself is difficult to calculate exactly for multivariate probability distributions when pairs from the optimal transportation map are unknown. 3 However, for univariate probability distributions, there is a closed-form solution related to the difference between the inverse Cumulative Distribution Functions (CDF\(^{-1}\)) 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 [15]. See Training for more details.

The SWAE loss takes the general form of the WAE loss,

\[
\mathcal{L}_{\text{SWAE}}(p(x), p_d(x|z), p_e(z|x)) = \mathbb{E}_{x \sim p(x)} \mathbb{E}_{p_e(z|x)} \mathbb{E}_{x \sim p_d(x|z)} [c(x, \tilde{x})] + \lambda \; d_{SW}(p_e(z), p(z))
\]

(3)

The data-space term (Eq. 3) compares pairs \(\{x, \tilde{x}\}\), where \(\tilde{x}\) is the output of the encoder-decoder mapping. For the latent-space term (Eq. 4) matched pairs are not available for computing the full Wasserstein distance, so we instead use the SW distance approximation. In this work, both loss terms use the cost metric \(c(u, v) = ||u - v||^2\) [15].

The SWAE allows us to build a probabilistic autoencoder that transforms between \(\mathcal{X}\) and \(\mathcal{Z}\) with a physical latent prior \(p(z)\). We would also like the learned transformations to be plausible and represent 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. We propose one constraint from each category for our particle physics examples.

\[\footnote{The fact that the latent-space loss compares \(p_E(z)\) and \(p(z)\) rather than \(p_E(z|x)\) and \(p(z)\) is the crucial innovation which allows for different parts of \(\mathcal{Z}\) to remain disjoint.}

\[\footnote{Calculating the Wasserstein distance involves optimizing over all possible transportation paths to find the one with minimal cost. If explicit pairs (i.e. from the joint distribution \(p(z, x)\)) are known the path has effectively been chosen, providing an upper bound on the Wasserstein distance.}\]
Falling in the first category, we add a term which compares the unit vector parallel to the momentum of an easily identifiable particle in the latent and observed spaces. This can be thought of as analogous to choosing a consistent basis; this can be helpful for problems which contain simple inversion symmetries, such as $Z \rightarrow e^+e^-$ below. We therefore call it an anchor term. This takes the general form

$$
L_A(p(a), p_G(b|a)) = \mathbb{E}_{a \sim p(a)}\mathbb{E}_{b \sim p_G(b|a)}[c_A(a, b)]
$$

(5)

In our studies we chose $c_A(a, b) = 1 - \hat{p}_a \cdot \hat{p}_b$, where $\hat{p}$ is the unit vector of the coordinates corresponding to the momentum of an easily identifiable particle, such as electron. We add the anchor loss in $\mathcal{Z}$ space, $L_A(p(x), p_E(z|x))$, and in $\mathcal{X}$ space, $L_A(p(z), p_D(x|z))$, to the SWAE loss with hyperparameter weightings $\beta_1$ and $\beta_2$ respectively. The well-behaved SWAE loss (compared to, e.g., an adversarial loss) allows us to conveniently minimize the combined loss without it diverging.

Falling in the second category, we enforce the Minkowski space 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 $p^\mu = (p, E)$ where $E$ is a particle’s energy and $\hat{p}$ is a vector of its momentum in the $x, y, z$ direction respectively, the constraint becomes

$$
 p^\mu p_\mu = E^2 - \hat{p}^2 = m^2
$$

(6)

where $m$ is the particle’s mass.

We directly enforce this relationship in the model for all particles to help improve performance. Initial experiments which did not include this constraint demonstrated the networks were able to automatically learn this relationship from the data. However, including this constraint directly in the model architecture improved performance overall.

It would be straightforward to add more physically motivated constraints, however, for the studies presented here, we only assume this minimal set of constraints. We recommend that a more robust structuring of the data and latent spaces be considered first, as it may make such constraints unnecessary (see Future Work).

### 2.2 Demonstration in $Z \rightarrow e^+e^-$ decays

As a first test of the performance of OTUS as a predictive generator, we consider an important control region, leptonic decays of the $Z$ boson, such as to electron-positron pairs, $Z \rightarrow e^+e^-$. The theoretical prior is well known, and its parameters $\theta$, such as the mass of the $Z$ boson and the strengths of its interactions, are tightly constrained by precision experiments. We identify the latent space, $\mathcal{Z}$, with the decay products of the $Z$ boson: the electron, $e^-$, and positron, $e^+$, whose four-momenta span the space. We compose these into an eight-dimensional vector:

$$
z := \{z_{e^-}, z_{e^+}\} = \{p_{e^-}, E_{e^-}, p_{e^+}, E_{e^+}\}
$$

(7)

where we have chosen a simplistic vector description of the data which excludes categorical properties such as charge.

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

Figure 2 shows the distribution of testing data, unpaired samples from $\mathcal{X}$ and $\mathcal{Z}$ in several projections, as well as the results of applying the trained encoder and decoder to transform between the two spaces. Visual evaluation indicates qualitatively good performance, but quantitative metrics are provided. To measure overall performance the Sliced Wasserstein distances are as follows: $d_{SW}(p(z), p_E(z)) = 1.09$ [GeV²], $d_{SW}(p(x), p_D(x)) = 1.69$ [GeV²],
\[ d_{SW}(p(x), p_D(x')) = 2.94 \text{ [GeV}^2]. \] Additionally, several common metrics are reported for each projection in Supplementary Table 1 and 2. Details of the calculations are provided in Evaluation.

To ensure that the learned decoder reasonably reflects the physical processes being modeled, we inspect the transformation from $Z \rightarrow \mathcal{X}$ in Figure 3. The learned transfer function, $p_D(x|z)$, shows reasonable behavior, mapping samples from $Z$ to nearby values of $\mathcal{X}$, reflecting the finite resolution of the detector while minimizing the overall transport and avoiding unphysical transformations with inversions or discontinuities. Additionally, cross-referencing with the mapping of the true simulation shows the nature of the mappings are very similar.

Finally, we examine the distribution of a physically important derived quantity, the invariant mass of the electron-positron pair, see Figure 4. Comparisons of this quantity were not used explicitly as an element of the loss function, and so provide an alternative measure of performance. The overall metrics indicate a high-quality description of the transformation from $Z$ to $\mathcal{X}$ in this context. The performance of the transformation from $\mathcal{X}$ to $Z$ is less well-described, likely because this relation is more strict in $Z$ causing a sharper peak in the distribution. Such strict rules are difficult for neural networks to describe when not penalized directly or hard-coded as inductive biases. This again signals that a more robust data representation will be crucial to improving performance, see Future Work for more details.

2.3 Demonstration in semileptonic top quark decays

The $Z$ boson control region is valuable for calibrating leptons such as electrons or muons, which tend to be stable and well-measured. We next apply OTUS to the challenging task of modeling the decay and detection of top quark pairs which features more complex detector signatures. Events in this control region have more observed particles, and also introduce several additional complexities: unstable particles decaying in flight, significantly degraded resolution relative to leptons, undetected particles, and a stochastically variable number of observed particles.

The creation of top quark pairs in collisions, their decay $t \bar{t} \rightarrow W^+ b \ W^- \bar{b}$, and the subsequent $W$ boson decays can be well-described using quantum field theory, and the prior $p(z|\theta)$ efficiently sampled. We select the modes $W^- \rightarrow e^- \bar{\nu}_e$ and $W^+ \rightarrow u \bar{d}$ as examples and assign our latent space to describe the four-momenta of these six of particles:

\[
\mathbf{z} := \{z_{e^-}, z_{\bar{\nu}_e}, z_b, z_u, z_d, z_{\bar{b}}\} = \{\mathbf{p}^{e^-}, E^{e^-}, \mathbf{p}^{\bar{\nu}_e}, E^{\bar{\nu}_e}, \mathbf{p}^b, E^b, \mathbf{p}^u, E^u, \mathbf{p}^d, E^d, \mathbf{p}^{\bar{b}}, E^{\bar{b}}\}
\]

with a total of twenty-four dimensions.

Unlike in the $Z \rightarrow e^+ e^-$ study, the structure of the observed space $\mathcal{X}$ is considerably different from that of the latent space. While the electron $e^-$ is stable and readily identifiable, the other particles are more challenging. The neutrino, $\nu_e$, is stable, yet invisible to our detectors, which provide no estimate of its direction or momentum; its presence can only be deduced by applying constraints of the conservation of momentum $\mathbf{p}^\nu = -\sum \mathbf{p}^{\text{obs}}$. Unfortunately, this is not the only source of missing momentum, soft initial state radiation and detector inefficiencies also contribute. The aggregate quantity is labeled $\mathbf{p}^\text{miss}$. The four quarks $\bar{b}, u, \bar{d}$ and $b$ are strongly-interacting particles, which produce complex showers of particles that can be clustered together into jets to provide an estimate of the original quark momentum and direction. Unfortunately, despite significant recent progress [30], we cannot assume a perfect identification of the source particle in $Z$ for a given jet observed in $\mathcal{X}$, leaving a significant ambiguity.

In addition, a full description of the $Z \rightarrow \mathcal{X}$ process should include 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. In this study, we set this complexity aside for future work and restrict our data space to samples with exactly four jets.

A final complexity introduced in this dataset is the presence of a sharp lower threshold in transverse momentum, $p_T$. Experimental limitations require a minimum threshold, so that jets which have $p_T < 20$ GeV are discarded and therefore are not represented in the training dataset, as they would not be available in control region data. To mimic this experimental effect, we directly impose this threshold on the decoder’s output instead of learning it directly. Paralleling reality, such events are discarded before computing losses. This strategy requires modifications to both the model and training strategy see Methods for details.
Figure 2. A study of the performance of OTUS for $Z \rightarrow e^+e^-$ decays. Top pane shows distributions of samples from the theoretical prior $p(z)$, as well as the output of the encoder, which transforms samples of testing data in observed space, $X$, to the latent space, $Z$, shown as $x \rightarrow \tilde{z}$. Bottom pane shows the testing sample in the observed space, $X$, as well as output from the decoder applied to samples drawn from the prior $p(z)$, labeled as $z \rightarrow \tilde{x}'$. Also shown are samples passed through both the decoder and encoder chain, $x \rightarrow \tilde{z} \rightarrow \tilde{x}$. Residual plots show bin-by-bin ratios with statistical uncertainties propagated accordingly, see Evaluation for more details.
Figure 3. Visualization of the transformation from $\mathcal{Z} \to \mathcal{X}$ in the $Z \to e^+e^-$ study for positron energy. Colors in the $\mathcal{X}$ projection indicate the source bin in $\mathcal{Z}$ for a given sample. Left is the learned transformation of the decoder, $p_D(x|z)$. Right is 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.

Our space of observed data is the vector:

$$x := \{x_{e^-}, x_{\text{miss}}, x_{\text{jet}1}, x_{\text{jet}2}, x_{\text{jet}3}, x_{\text{jet}4}\}$$

$$= \{p_{e^-}, E_{e^-}, p_{\text{miss}}, E_{\text{miss}}, p_{\text{jet}1}, E_{\text{jet}1}, p_{\text{jet}2}, E_{\text{jet}2}, p_{\text{jet}3}, E_{\text{jet}3}, p_{\text{jet}4}, E_{\text{jet}4}\}$$

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 that information, it is typical to order jets by descending $|p_T| = \sqrt{p_T^2 + p_Y^2}$, such that jet 1 will have the largest $|p_T|$.

Figure 5 shows the distribution of testing data, unpaired samples from $\mathcal{X}$ and $\mathcal{Z}$ in several projections, as well as the results of applying the trained encoder and decoder to transform between the two spaces. Visual evaluation indicates qualitatively good performance, but quantitative metrics are also provided. To measure overall performance the Sliced Wasserstein distances are as follows:

$$d_{SW}(p(z), p_E(\tilde{z})) = 13.0 \text{ [GeV}^2\text{]},$$

$$d_{SW}(p(x), p_D(\tilde{x})) = 231 \text{ [GeV}^2\text{]},$$

$$d_{SW}(p(x), p_D(\tilde{x}')) = 186 \text{ [GeV}^2\text{]}.$$ 

Additionally, several common metrics are reported for each projection in Supplementary Table 3 and 4. Details of the calculations are provided in Evaluation.

To probe the nature of the $\mathcal{Z} \to \mathcal{X}$ transformation, we inspect the learned transfer function, $p_D(x|z)$; see Figure 6. While the overall quality of the modeling is not as high in this more complex case, it still shows reasonable behavior, mapping samples from $\mathcal{Z}$ to nearby values of $\mathcal{X}$ avoiding unphysical transformations with inversions or discontinuities. Additionally, cross-referencing with the mapping of the true simulation shows the nature of the mappings are very similar.

Finally, we examine the distribution of physically important derived quantities, the invariant mass of the top quarks and $W$ bosons, which are estimated by combining information from pairs and triplets of objects, see Figure 7. 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 observed 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.
Figure 4. Performance of OTUS for $Z \rightarrow e^+e^-$ decays in a physically important derived quantity, the invariant mass of the electron-positron pair, $M_Z$. 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 for more details.
Figure 5. A study of the performance of OTUS for $t\bar{t}$ decays. Top pane shows distributions of samples from the theoretical prior $p(z)$, as well as the output of the encoder, which transforms samples of the testing data in observed space, $X$, to the latent space, $Z$, shown as $x \rightarrow \tilde{z}$. Bottom pane shows the testing sample in the observed space, $X$, as well as output from the decoder applied to samples drawn from the prior $p(z)$, labeled as $z \rightarrow \tilde{x}'$. Also shown are samples passed through both the decoder and encoder chain, $x \rightarrow \tilde{z} \rightarrow \tilde{x}$. Residual plots show bin-by-bin ratios with statistical uncertainties propagated accordingly, see Evaluation for more details.
Figure 6. Visualization of the transformation from $\mathcal{Z} \rightarrow \mathcal{X}$ in the $t\bar{t}$ study for the energy of the $b$ quark in $\mathcal{Z}$ to energy of the leading jet in $\mathcal{X}$. 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. Left is the learned transformation of the decoder, $p_D(x|z)$. Right is 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.

Figure 7. Performance of OTUS for $t\bar{t}$ decays in physically important derived quantities, the invariant mass of the $W$ bosons, $M_W$, and top quarks, $M_t$, as reconstructed from particle pairs and triplets. Also shown is the invariant mass of the combined $t\bar{t}$ pair and the $p_T$ of the leading jet. Residual plots show bin-by-bin ratios with statistical uncertainties propagated accordingly, see Evaluation for more details.
2.4 Discussion

2.4.1 Implications for Physics and Beyond

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

In general, this approach can be applied to any process in the physical sciences where unobserved latent phenomena \( \mathcal{Z} \) can be described in the form of a prior model \( p(\mathcal{Z}) \) and are translated to an empirical set of observed data \( \mathcal{X} \) via an unknown transformation. For example, in molecular simulations in chemistry, observations could be measurements of real-world molecular dynamics, \( p(\mathcal{Z}) \) would represent the model description of the system, and \( p(\mathcal{X}|\mathcal{Z}) \) would model the effects of real-world complications [7]. In cosmology, \( \mathcal{X} \) could be the distribution of mass in the observed universe, \( p(\mathcal{Z}) \) could describe its distribution in the early universe, and \( p(\mathcal{X}|\mathcal{Z}) \) would model the universe’s unknown expansion dynamics, e.g., due to inflation [9, 31]. In climate simulations, \( p(\mathcal{Z}) \) could correspond to the climate due to a physical model, while \( p(\mathcal{X}|\mathcal{Z}) \) takes unknown geography-specific effects into account [8]. Additionally, an immediate and promising application of our approach is in medical imaging, which uses particle physics simulations to model how the imaging particles (for example, x-rays) interact with human tissue — an approach that suffers from the great computational cost of these simulations [10].

Additionally, features of this method can be easily adapted to suit the needs of the particular problem. For example, in this work we were interested in low-dimensional data, however the method could also handle high-dimensional datasets. Moreover, the encoding and decoding mappings can be stochastic, as in this work, or deterministic. Lastly, while a goal of this work was to be completely unsupervised, and thus data-driven, our method 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 \( \{\mathcal{Z}, \mathcal{X}\} \) (e.g., from simulation runs). These pairs effectively sample the joint distribution, \( p(\mathcal{Z}, \mathcal{X}) \). This can be interpreted as specifying a transportation plan between the marginals \( p(\mathcal{Z}) \) and \( p(\mathcal{X}) \), which ultimately makes the estimation of the Wasserstein distances \( d_W(p(\mathcal{X}), p_D(\mathcal{X})) \) and \( d_W(p(\mathcal{Z}), p_E(\mathcal{Z})) \) tractable in this high-dimensional setting. Since this is not necessarily the optimal transportation plan, these are only upper-bounds on the true Wasserstein distances calculated with the optimal transportation plan, leading to the alternative objectives

\[
\mathcal{L}_{\text{paired}}(p_D(\mathcal{X}|\mathcal{Z}), p(\mathcal{Z}, \mathcal{X})) = \mathbb{E}_{(\mathcal{Z}, \mathcal{X}) \sim p(\mathcal{Z}, \mathcal{X})} [c(\mathcal{Z}, \mathcal{X})] \\
\mathcal{L}_{\text{paired}}(p_E(\mathcal{X}|\mathcal{Z}), p(\mathcal{Z}, \mathcal{X})) = \mathbb{E}_{(\mathcal{Z}, \mathcal{X}) \sim p(\mathcal{Z}, \mathcal{X})} [c(\mathcal{Z}, \mathcal{X})]
\]

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

2.4.2 Future Work

This work has 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 space.

First, the structure of the latent and data space can significantly affect the performance and reliability of the resulting simulations. Particle physics data has rich structures often governed by group symmetries and conservation laws. Our current description of the data is simplistic, a vector format which 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 might explore network architectures and losses that can better capture the full nature of these data structures [32, 33].

The next technical hurdle will be the ability to handle variable input and output states. The same prior \( p(\mathcal{Z}) \) 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 \( Z \)
The vector $\bar{w}$ is then unstandardized by inverting the relationship in step 1, creating a vector $\bar{w}$.

Both the encoding and decoding model’s noise networks produced Gaussian-distributed noise vectors with mean $\phi$ and diagonal covariances $\mu$. (2) A noise neural network computes a conditional noise distribution $p_N(\varepsilon|u)$, where the noise vector $\varepsilon \sim p_N(\varepsilon|u)$ has the same dimensionality as the core network prediction $\bar{w}$ (defined in the next step). (3) The standardized data, $\bar{u}$, and noise vector, $\varepsilon$, are then concatenated and fed into a core neural network. This network outputs the 3-momentum, $\bar{p}$, information of each particle in the standardized space, collected into a vector $\bar{w}$.

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, $U$, to a space, $V$, the steps are as follows. (1) A sample of raw input data, $u \in U$, is standardized by subtracting the mean and dividing by the standard deviation resulting in the standardized data, $\bar{u}$. (2) A noise neural network computes a conditional noise distribution $p_N(\varepsilon|u)$, where the noise vector $\varepsilon \sim p_N(\varepsilon|u)$ has the same dimensionality as the core network prediction $\bar{w}$ (defined in the next step). (3) The standardized data, $\bar{u}$, and noise vector, $\varepsilon$, are then concatenated and fed into a core neural network. This network outputs the 3-momentum, $\bar{p}$, 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 discussed in the Results section is then enforced explicitly to reinsert the energy information of each particle, transforming $w$ into the final $v \in V$ which is distributed according to $p_G(v|u)$.

Both the encoding and decoding model’s noise networks produced Gaussian-distributed noise vectors with mean and diagonal covariances $[\mu(x), \sigma^2(x)]$ and $[\mu(z), \sigma^2(z)]$ respectively. For the $Z \rightarrow e^+e^-$ 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.
3.2.2 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, $p_N(e|u) = p_N(e) = N(0, 1)$. 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 [36] and batch normalization [37]. 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 $s \rightarrow s + r$. (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 $[\vec{u}, \epsilon]$ 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.

3.3 Training

3.3.1 Base Training Strategy

The model is trained by minimizing the SWAE loss function augmented with anchor terms (see the Results section),

$$\mathcal{L}_{SWAE}(p(x), p_D(x|z), p_E(z|x)) = \mathbb{E}_{x \sim p(x)} \mathbb{E}_{z \sim p_E(z|x)} \mathbb{E}_{e \sim p_D(z|\theta)} [c(x, \tilde{x})] + \lambda d_{SW}(p_E(z), p(z)) + \beta_1 \mathcal{L}_A(p(x), p_E(z|x)) + \beta_2 \mathcal{L}_A(p(z), p_D(x|z)),$$

with respect to parameters of the encoder $p_E(z|x)$ and decoder $p_D(x|z)$ 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:

$$\hat{\mathcal{L}}_{SWAE} = \frac{1}{M} \sum_{m=1}^{M} c(x_m, \tilde{x}_m) + \lambda \frac{1}{L \times M} \sum_{l=1}^{L} \sum_{m=1}^{M} c((\theta_l \cdot z_m)_{\text{sorted}}, (\theta_l \cdot \tilde{z}_m)_{\text{sorted}}) + \beta_1 \frac{1}{M} \sum_{m=1}^{M} c_A(x_m, \tilde{z}_m) + \beta_2 \frac{1}{M} \sum_{m=1}^{M} c_A(z_m, \tilde{x}_m)$$

where $\{x_m\}_{m=1}^{M}$ and $\{z_m\}_{m=1}^{M}$ are $M$ instances of data ($X$) and latent ($Z$) samples, $\{\tilde{x}_m \sim p_D(\cdot|z_m)\}_{m=1}^{M}$ are drawn from the decoder, and $\{\tilde{z}_m \sim p_D(\cdot|z_m)\}_{m=1}^{M}$ are drawn from the auto-encoding chain $x \rightarrow \tilde{z} \rightarrow \tilde{x}$. 4 The estimation of the SW term $d_{SW}(p(z), p_E(z))$ additionally 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 computing the inverse of two univariate CDFs, by simply 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 $\theta_l$; we refer interested readers to [15] for more technical details of the Sliced Wasserstein distance. We use the squared norm as the cost metric $c(a, b) = ||a - b||^2$ in the SWAE loss [15]. The anchor cost, $c_A$, between two observation vectors $a, b$ (which can reside in either $X$ or $Z$ space) is defined as

$$c_A(a, b) := 1 - \hat{p}_a \cdot \hat{p}_b,$$

where $\hat{p}_a$ is the unit vector of the coordinates of $a$ corresponding to the momentum of a pre-specified particle, and $\hat{p}_b$ is defined analogously with respect to the same particle; this is chosen as the electron in our experiments. For example, $c_A(x, \bar{z})$ would be computed as

$$c_A(x, \bar{z}) = 1 - \hat{p}_x \cdot \hat{p}_{\bar{z}} = 1 - \frac{p_{\bar{z}}^e}{||p_{\bar{z}}^e||} \cdot \frac{p_x^e}{||p_x^e||}$$

---

4This is equivalent to drawing a sample $(x, \tilde{z}, \tilde{x})$ from the joint distribution $p(x)p_E(\tilde{z}|x)p_D(\tilde{x}|\tilde{z})$. 

---

16/23
Figure 8. Diagrams depicting the network structure of the model. The upper diagram shows the full OTUS model where gray indicates information used in the calculation of losses only. The middle diagram shows the internal structure present in both the encoder and decoder models. The lower model shows the setup used for a post-processing decoder network loss. See the description in the text for more details.

At a higher level, the computation of $\hat{\mathcal{L}}_{SWAE}$ based on a mini-batch proceeds as follows. Following the path through the full model, a batch of samples $X \sim p(x)$ from $\mathfrak{X}$ space is passed to the encoder model, $E$, producing $\tilde{Z} \in \tilde{Z}$ distributed according to $p_E(z|x)$. The encoding anchor loss term $L_{AE}(X, \tilde{Z}) \equiv L_A(p(x), p_E(z|x))$ is then computed along with the Sliced Wasserstein distance latent loss, $\hat{d}_{SW}(Z, \tilde{Z}) \equiv \hat{d}_{SW}(p(z), p_E(z))$. 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 $L_{AD}(Z, \tilde{X}') \equiv L_A(p(z), p_D(x|z))$ is then computed. Finally, the data space loss, chosen to be $\text{MSE}(X, \tilde{X})$, is computed. See Figure 8 for a visual representation. We can then minimize the tractable Monte-Carlo estimate of the objective, $\hat{\mathcal{L}}_{SWAE}$, 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, we also consider an auxiliary strategy of directly minimizing the more computationally convenient $\hat{d}_{SW}(p(z), p_E(z))$. When minimized over all $p_E(z|x)$ that satisfies the constraint $p_E(z) = p(z)$, the data term of the WAE loss, Eq. 1, becomes an upper bound on $d_W(p(x), p_D(x))$; the bound is tight for deterministic decoders [16]. The overall WAE loss $\mathcal{L}_{WAE}$ is a relaxation of the exact...
where $X_t$ as was previously explained, experimental limitations in the semileptonic 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,

\[
\tilde{p}_D(x) = \frac{p_D(x) \mathbf{1}_S(x)}{p_D(S)} \tag{15}
\]

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 $d_{SW}(p(x), \tilde{p}_D(x))$. 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 Eq. 1, so it becomes

\[
\mathbb{E}_{x \sim p(x)} \mathbb{E}_{\tilde{p}_D(x)} \mathbb{E}_{\tilde{y} \sim p_D(y|x)} [c(x, \tilde{x})] \rightarrow \mathbb{E}_{x \sim p(x)} \mathbb{E}_{\tilde{p}_D(x)} \mathbb{E}_{\tilde{y} \sim p_D(x)} [\mathbf{1}_S(\tilde{y}) \frac{\tilde{p}_D(x)}{p_D(S)} c(x, \tilde{x})] \tag{16}
\]

Letting $\theta$ 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 case (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

\[
p_D(S) \approx \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}_S(\tilde{x}_m) \tag{18}
\]

variational bound, and recovers the latter as $\lambda \to \infty$. 

\[
18/23
\]
3.3.3 Parameter Optimization

For the $Z \rightarrow e^+ e^-$ study, we used the base training strategy. We optimized $\hat{L}_{SWAE}$ for 80 epochs with anchor penalties $\beta_1 = \beta_2 = 50$, 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 $\hat{L}_{SWAE}$ by $\tilde{p}_D(x)$ (e.g., using the modified data loss term Eq. 16). In the spirit of projected gradient descent, we interleaved the minimization of the joint loss $\hat{L}_{SWAE}$ with “projection” steps that minimized the auxiliary encoder loss $\hat{d}_SW(p(z), p_E(z))$; we ran 400 epochs in each step, for a total of 24 alternating steps. Finally, we fine-tuned the decoder by minimizing $\hat{d}_SW(p(x), \tilde{p}_D(x))$ for 100 epochs. The input-to-output residual connection (see Sec. 3.2) 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 $\beta_1 = \beta_2 = 0$.

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 [38] with $L = 1,000$ number of slices, a batch size of $M = 20,000$, and learning rate of 0.001. When using an (S)WAE scheme, $\lambda = 1$ was used.

3.4 Evaluation

Evaluation of generative models typically rely on qualitative visualizations of the results [11, 12, 14]. In accordance with this standard, we depict visualizations of the results in important one-dimensional projections (e.g. Fig. 2 and Fig. 3). Residual plots which take the ratio between the histograms from generated samples to the histogram from true samples are also shown. Statistical uncertainties of the histogram bins are propagated accordingly and shown as error bars in the residual plots. We additionally visualize the generative mappings using the purely qualitative transportation plots (e.g. Fig. 3).

In addition to these visual comparisons, we also calculated quantitative metrics on our results. It is non-trivial to directly evaluate the similarity of two high-dimensional distributions in a statistically rigorous way. However, statistical evaluation of one-dimensional projections is much more well defined, but sacrifices knowledge about the high-dimensional behavior.

To give an impression of the high-dimensional matching of these distributions we calculate the MC estimate of the Sliced Wasserstein distance, $\hat{d}_SW(\cdot, \cdot)$, using $L = 1,000$ slices according to the cost metric $c(u, v) = ||u - v||^2$. The results are reported for each study in the text. For the one-dimensional projections we can evaluate performance in a more statistically rigorous fashion. We apply several familiar statistical tests which are reported in Supplementary Information. First, we calculate the reduced $\chi^2$, $\chi^2_R$, 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 MC estimate of the Wasserstein distance, $\hat{d}_W(\cdot, \cdot)$, according to the cost metric $c(u, v) = ||u - v||^2$.

For statistical tests it is crucial for the samples from each distribution to be completely independent. Given the generative nature of our problem we must take great care to ensure this independence. For example, if our true baseline was produced by Delphes mapping a set of samples $z_1 \rightarrow x_1$ we must use a separate set of samples, $z_2$, to pass through the decoding mapping $z_2 \rightarrow \tilde{x}_2$ in order to be able to safely compare $x_1$ and $\tilde{x}_2$. To ensure this independence, we constructed two separate testing sets which were not used during training or validation of the networks; one test set represents the true baseline and the other is run through the trained networks to produce the final results. The number of samples in these test sets were 80,000 in the $Z \rightarrow e^+ e^-$ study and 47,202 in the semileptonic $t\bar{t}$ study.  

4 Acknowledgements

JNH 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. YY acknowledges funding from the Hasso Plattner Foundation. SM acknowledges support by the National Science Foundation under Grants

---

6Note 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 section 2.3. The events present are ones that passed this cutoff constraint.
1928718, 2003237 and 2007719, Intel, Qualcomm, 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.

5 Author Contributions Statement

Using the CASRAI CRediT Contributor Roles Taxonomy: Conceptualization, JNH, SM, DW, YY; Data curation, JNH; Formal analysis, JNH, YY; Funding acquisition, JNH, SM, DW; Investigation, JNH, SM, DW, YY; Methodology, JNH, YY; Project administration, JNH, SM, DW; Software, JNH, YY; Supervision, SM, DW; Validation, YY; Visualization, JNH; Writing – original draft, JNH, DW; Writing – review & editing, JNH, SM, DW, YY.

References

[1] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. 2020. arXiv: 1911.01429 [stat.ML].

[2] S. Agostinelli et al. “Geant4 - a simulation toolkit”. In: Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 506.3 (2003), pp. 250–303. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(03)01368-8. URL: http://www.sciencedirect.com/science/article/pii/S0168900203013688.

[3] Torbjorn Sjostrand, Stephen Mrenna, and Peter Z. Skands. “PYTHIA 6.4 Physics and Manual”. In: JHEP 0605 (2006), p. 026. DOI: 10.1088/1126-6708/2006/05/026. arXiv: hep-ph/0603175 [hep-ph].

[4] J. de Favereau et al. “DELPHES 3, A modular framework for fast simulation of a generic collider experiment”. In: JHEP 02 (2014), p. 057. DOI: 10.1007/JHEP02(2014)057. arXiv: 1307.6346 [hep-ex].

[5] G. Aad et al. “The ATLAS Simulation Infrastructure”. In: The European Physical Journal C 70.3 (Sept. 2010), pp. 823–874. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-010-1429-9. URL: http://dx.doi.org/10.1140/epjc/s10052-010-1429-9.

[6] G L Bayatian et al. CMS Physics: Technical Design Report Volume 1: Detector Performance and Software. Technical Design Report CMS. There is an error on cover due to a technical problem for some items. Geneva: CERN, 2006. URL: https://cds.cern.ch/record/922757.

[7] Frank Noé et al. “Machine Learning for Molecular Simulation”. In: Annual Review of Physical Chemistry 71.1 (2020). PMID: 32092281, pp. 361–390. DOI: 10.1146/annurev-physchem-042018-052331. eprint: https://doi.org/10.1146/annurev-physchem-042018-052331. URL: https://doi.org/10.1146/annurev-physchem-042018-052331.

[8] David Rolnick et al. Tackling Climate Change with Machine Learning. 2019. arXiv: 1906 . 05433 [cs.CY].

[9] Arnaud Delaunoy et al. Lightning-Fast Gravitational Wave Parameter Inference through Neural Amortization. 2020. arXiv: 2010.12931 [astro-ph.IM].

[10] Jiajia Zhou et al. “Emerging role of machine learning in light-matter interaction”. In: Light: Science & Applications 8.1 (2019), p. 84. DOI: 10.1038/s41377-019-0192-4. URL: https://doi.org/10.1038/s41377-019-0192-4.

[11] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. “CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks”. In: Phys. Rev. D 97 (1 Jan. 2018), p. 014021. DOI: 10.1103/PhysRevD.97.014021. arXiv: 1712.10321 [hep-ex]. URL: https://link.aps.org/doi/10.1103/PhysRevD.97.014021.
[12] Anja Butter, Tilman Plehn, and Ramon Winterhalder. “How to GAN LHC Events”. In: SciPost Phys. 7 (6 2019), p. 75. DOI: 10.21468/SciPostPhys.7.6.075. arXiv: 1907.03764 [hep-ph]. URL: https://scipost.org/10.21468/SciPostPhys.7.6.075.

[13] Sydney Otten et al. Event Generation and Statistical Sampling for Physics with Deep Generative Models and a Density Information Buffer. 2019. arXiv: 1901.00875 [hep-ph].

[14] Luke de Oliveira, Michela Paganini, and Benjamin Nachman. “Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis”. In: Computing and Software for Big Science 1.1 (2017), p. 4. DOI: 10.1007/s41781-017-0004-6. arXiv: 1701.05927 [stat.ML]. URL: https://doi.org/10.1007/s41781-017-0004-6.

[15] Soheil Kolouri et al. Sliced-Wasserstein Autoencoder: An Embarrassingly Simple Generative Model. 2018. arXiv: 1804.01947 [cs.LG].

[16] Ilya Tolstikhin et al. Wasserstein Auto-Encoders. 2017. arXiv: 1711.01558 [stat.ML].

[17] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. 2013. arXiv: 1312.6114 [stat.ML].

[18] Artidoro Pagnoni, Kevin Liu, and Shangyan Li. Conditional Variational Autoencoder for Neural Machine Translation. 2018. arXiv: 1812.04405 [cs.CL].

[19] V. M. Abazov et al. “A precision measurement of the mass of the top quark”. In: Nature 429.6992 (2004), pp. 638–642. DOI: 10.1038/nature02589. URL: https://doi.org/10.1038/nature02589.

[20] Marco Bellagente et al. Invertible Networks or Partons to Detector and Back Again. 2020. arXiv: 2006.06685 [hep-ph].

[21] Anders Andraeassen et al. “Omnifold: A Method to Simultaneously Unfold All Observables”. In: Physical Review Letters 124.18 (May 2020). ISSN: 1079-7114. DOI: 10.1103/physrevlett.124.182001. URL: http://dx.doi.org/10.1103/PhysRevLett.124.182001.

[22] Andrea Castro. Top Quark Mass Measurements in ATLAS and CMS. 2019. arXiv: 1911.09437 [hep-ex].

[23] Cheng Zhang et al. “Advances in variational inference”. In: IEEE transactions on pattern analysis and machine intelligence 41.8 (2018), pp. 2008–2026.

[24] Erik Buhmann et al. Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed. 2020. arXiv: 2005.05334 [physics.ins-det].

[25] Kamil Deja et al. End-to-end Sinkhorn Autoencoder with Noise Generator. 2020. arXiv: 2006.06704 [cs.LG].

[26] V. M. Abazov et al. “A precision measurement of the mass of the top quark”. In: Nature 429.6992 (2004), pp. 638–642. DOI: 10.1038/nature02589. URL: https://doi.org/10.1038/nature02589.

[27] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. 2017. arXiv: 1701.07875 [stat.ML].

[28] Christopher P. Burgess et al. Understanding disentangling in β-VAE. 2018. arXiv: 1804.03599 [stat.ML].

[29] Tim Salimans et al. Improved Techniques for Training GANs. 2016. arXiv: 1606.03498 [cs.LG].

[30] Michael James Fenton et al. “Permutationless Many-Jet Event Reconstruction with Symmetry Preserving Attention Networks”. In: (Oct. 2020). arXiv: 2010.09206 [hep-ex].

[31] Uroš Seljak et al. “Towards optimal extraction of cosmological information from nonlinear data”. In: Journal of Cosmology and Astroparticle Physics 2017.12 (Dec. 2017), pp. 009–009. DOI: 10.1088/1475-7516/2017/12/009. URL: https://doi.org/10.1088%5C%2F1475-7516%5C%2F2017%5C%2F12%5C%2F009.
6 Supplementary Information

Table 1. This table shows the $z$ space statistical test results for the $Z \rightarrow e^+e^-$ dataset. See Evaluation for detailed information about the calculations of these statistics.

|                  | $z$ vs $\bar{z}$ |          |          |
|------------------|-------------------|----------|----------|
|                  | $W$ [GeV$^2$]     | $(\chi^2_R, \text{dof})$ | KS       |
| Fig. 2a          | $1.32 \times 10^{+00}$ | (50.075, 23) | $1.56 \times 10^{-02}$ |
| Fig. 2b          | $1.77 \times 10^{+00}$ | (1.512, 26) | $7.51 \times 10^{-03}$ |
| Fig. 2c          | $1.17 \times 10^{+00}$ | (8.815, 26) | $1.49 \times 10^{-02}$ |
| Fig. 4a          | $2.81 \times 10^{+01}$ | (831.521, 39) | $2.59 \times 10^{-01}$ |

Table 2. This table shows the $\chi$ space statistical test results for the $Z \rightarrow e^+e^-$ dataset. See Evaluation for detailed information about the calculations of these statistics.

|                  | $x$ vs $\bar{x}$ |          |          | $x$ vs $\bar{x}'$ |          |          |
|------------------|-------------------|----------|----------|--------------------|----------|----------|
|                  | $W$ [GeV$^2$]     | $(\chi^2_R, \text{dof})$ | KS       | $W$ [GeV$^2$]     | $(\chi^2_R, \text{dof})$ | KS       |
| Fig. 2d          | $7.99 \times 10^{-01}$ | (3.000, 23) | $9.29 \times 10^{-03}$ | $1.21 \times 10^{+00}$ | (39.903, 23) | $1.40 \times 10^{-02}$ |
| Fig. 2e          | $2.67 \times 10^{+00}$ | (1.214, 26) | $7.84 \times 10^{-03}$ | $8.50 \times 10^{+00}$ | (4.326, 26) | $6.91 \times 10^{-03}$ |
| Fig. 2f          | $6.15 \times 10^{-01}$ | (0.568, 26) | $5.89 \times 10^{-03}$ | $3.04 \times 10^{+00}$ | (9.481, 26) | $1.91 \times 10^{-02}$ |
| Fig. 4b          | $6.21 \times 10^{-01}$ | (5.518, 39) | $1.57 \times 10^{-02}$ | $7.26 \times 10^{-01}$ | (20.714, 39) | $5.06 \times 10^{-02}$ |

Table 3. This table shows the $z$ space statistical test results for the semileptonic $t\bar{t}$ dataset. See Evaluation for detailed information about the calculations of these statistics.

|                  | $z$ vs $\bar{z}$ |          |          |
|------------------|-------------------|----------|----------|
|                  | $W$ [GeV$^2$]     | $(\chi^2_R, \text{dof})$ | KS       |
| Fig. 5a          | $4.35 \times 10^{+00}$ | (3.185, 49) | $1.59 \times 10^{-02}$ |
| Fig. 5b          | $2.53 \times 10^{+01}$ | (3.755, 55) | $1.23 \times 10^{-02}$ |
| Fig. 5c          | $2.10 \times 10^{+01}$ | (9.495, 31) | $2.76 \times 10^{-02}$ |
|        | $x$ vs. $\tilde{x}$ |        | $x$ vs. $\tilde{x}'$ |
|--------|---------------------|--------|---------------------|
|        | W [GeV$^2$]         | KS     | W [GeV$^2$]         | KS     |
| Fig. 5d| $8.09 \times 10^{-00}$ | $(1.102, 49)$ | $3.92 \times 10^{-03}$ | $(1.26 \times 10^{+02}$ | $(31.485, 49)$ | $3.78 \times 10^{-02}$ |
| Fig. 5e| $1.69 \times 10^{+02}$ | $(1.828, 55)$ | $1.44 \times 10^{-02}$ | $(7.86 \times 10^{+02}$ | $(6.689, 55)$ | $2.39 \times 10^{-02}$ |
| Fig. 5f| $1.00 \times 10^{+02}$ | $(1.593, 30)$ | $1.53 \times 10^{-02}$ | $(7.74 \times 10^{+02}$ | $(48.558, 30)$ | $5.75 \times 10^{-02}$ |
| Fig. 7a| $6.43 \times 10^{+02}$ | $(3.489, 43)$ | $2.24 \times 10^{-02}$ | $(3.62 \times 10^{+03}$ | $(92.257, 43)$ | $1.58 \times 10^{-01}$ |
| Fig. 7b| $6.22 \times 10^{+00}$ | $(14.813, 35)$ | $4.07 \times 10^{-02}$ | $(2.17 \times 10^{+02}$ | $(132.703, 35)$ | $1.17 \times 10^{-01}$ |
| Fig. 7c| $5.01 \times 10^{+00}$ | $(3.461, 30)$ | $1.08 \times 10^{-02}$ | $(1.31 \times 10^{+02}$ | $(100.467, 30)$ | $1.41 \times 10^{-01}$ |
| Fig. 7d| $1.04 \times 10^{+02}$ | $(15.263, 41)$ | $4.28 \times 10^{-02}$ | $(2.16 \times 10^{+03}$ | $(178.352, 41)$ | $2.49 \times 10^{-01}$ |

**Table 4.** This table shows the $\tilde{X}$ space statistical test results for the semileptonic $t\bar{t}$ dataset. See Evaluation for detailed information about the calculations of these statistics.