**The Landscape of Unfolding with Machine Learning Paper Review**

Paper Review by Tyler Kim

**Motivation**

* Predictions from quantum field theory are at partons level which require precise and detailed simulations
* Original approach was to use *forward inferencing* which fold predictions using QCD effects, hadronization, and detector response
  + There are couple of problems with *forward inferencing*
    - Requires access to data
    - Requires access to detector simulation
* *Unfolding* is an alternative approach has advantages and problems
  + *Unfolding*: data are adjusted to provide and estimate of their pre-detector distributions
  + Advantages
    - Data analysis more possible for broader community
    - Enabling efficient combination of data
  + Problems
    - Widely used methods work on small dimensionality

**Proposal**

* Propose to use unfolding as an alternative for predicting parton level data

**ML-Unfolding Methods**

* Simulated samples come in pairs
* Two approaches
  + Reweight simulated samples
  + Generate unfolded samples from conditional probabilities

A diagram of a person's body

Description automatically generated with medium confidence

* Reweighting
* **Omnifold (Reweighting)**
  + Re-weighting based on Neyman-Pearson lemma
    - *Neyman-Pearson lemma*: optimally trained, calibrated classifier C, will learn the likelihood ratio of two underlying phase space distribution
  + Computes classifier weights at reco-level and uses paired simulated data to pull weights from reco-level to particle-level

A diagram of weights and weights

Description automatically generated

* + Bayesian network allows for learning distributions
* Mapping Distributions
  + Assumes that and describe same features at reco level
  + Train network to map event distributions form  to based on paired or unpaired simulated events and apply mapping to to generate

A diagram of a diagram

Description automatically generated

* **Schrodinger Bridge**
  + Uses forward-time stochastic differential equations (SDE) as a time-dependent process to define the transformation between to
    - * : deterministic part
      * : noise schedule
      * : noise infinitesimal
    - Need to find and
    - as the score
    - There is a lot of physics that makes solving this SDE easier
* **Direction Diffusion**
  + Describes time evolution between particle and reconstruction levels
  + Paired
    - Learn the velocity field () that transforms the density such that using the equation: for paired
    - Loss function:
  + Unpaired
    - Needed equation is:
    - Loss function:
  + Can add Bayesian layers, gaussian distributions, and KL-term for more distribution approach
* Generative Unfolding
  + Uses conditional generative networks to learn conditional probability describing the inverse simulation

A black text on a white background

Description automatically generated

* + Building forward surrogate network uses same data and has close to same setup going backwards due to Baye’s Theorem
* **Conditional INN**
  + Conditional invertible neural network
  + Creates a bijective mapping between the latent and phase space as an invertible function conditioned at the reco-level event
    - Learned density:
    - Loss function:
  + Transformer extension translate sequences of reco-level momenta into a sequence of particle-level momenta
* **Conditional Flow Matching**
  + Same as direction diffusion except CFM samples from a Gaussian latent distribution, conditional on a reco-level event
  + Advantage to Direct Diffusion is that this approach allows them to unfold the same reco-level event repeatedly with different noise from as a starting point
* **Transformer Conditional Flow Matching**
  + Steps
    - Reco-level and particle-level dimensions are individually mapped into a higher-dimensional embedding space
    - Reco-level embeddings are then fed to the transformer encoder
    - Updated embeddings are fed into a cross-attention block resolves the combinatorics between reco-level and particle-level objects
    - Outputs a final condition
  + For unfolding, sample from latent distribution and solve:
* **Latent Variational Diffusion** 
  + The goal is o reduce disparity between parameterizations of the set of observables to enable a more robust network
  + Map observables from particle/parton phase space to a latent space
  + Fixed length reco-level objects encoding mapping is learned by a deep feed-forward neural network
  + Variable-length inputs are used by a transformer encoder

**Key Findings**

Detector Unfolding: Z+jets

* **Datasets**
  + Omnifold Dataset: <https://zenodo.org/records/10668638>
  + Pythia Dataset: Pythia 8.244 with Tune 26
* **Preprocessing**
  + Use bigger version of public dataset from reference [14] which is found in reference [59]
    - Dataset is a bigger version of the Omnifold dataset: <https://zenodo.org/records/10668638>
    - 24M simulated events (20M for training, 4M for test)
  + Focus on six observables describing the leading jet:
    - : mass
    - : width
    - : multiplicity
    - : soft-drop mass
    - : momentum fraction
    - : N-subjettiness ratio
  + They take the distributions for each observable and apply a dedicated preprocessing to the jet multiplicity and the groomed momentum fraction distribtuions
* **Reweighting**

A group of graphs showing the number of jets

Description automatically generated with medium confidence

* + - Jet multiplicity distribution
      * Add uniform noise, to smooth the distribution for the jet multiplicity distribution
    - Groomed momentum fraction
      * Move the peak of the groomed momentum fraction features to and add uniform noise
      * Take the logarithm to make the distribution more uniform
      * Shift and scale the distribution to stretch from -1 to +1
      * Take the inverse error function to transform its shape to an approximate normal distribution
    - All observables
      * Standardized by subtracting from the means the dividing by the standard deviation
* **Experimentation**
* **Reweighting**
  + Test OmniFold reweighting and Bayesian version Omnifold reweighting and compare their metrics
  + Evaluated performance on the same dataset by splitting it into two halves with one half having noise added to it
    - Noisy half
      * Trained OmniFold and the Bayesian counterpart together (bOmniFold is just Omnifold with Bayesian twist) for 30 epochs
      * Unfolded and true particle-level events agreement is better with the exception of sparsely populated tails
      * OmniFold and bOmniFold have small differences

A graph of a number of graphs

Description automatically generated with medium confidence

A table with numbers and letters

Description automatically generated

* + Then task classifiers to learn the likelihood ratio between Pythia and Herwig on the Pythia dataset
    - This ratio is used to reweight Herwig onto Pythia
      * OmniFold trained for 50 epochs on 2M events and tested on 664k events
      * OmniFold rapidly overtrains even after 20 epochs
      * bOmniFold does not overtrain due to the inherent dropouts and regularizations
        + larger epoch-to-epoch fluctuations and has worse minimum validation loss than OmniFold

A graph of a number of data

Description automatically generated

A graph with red and green lines

Description automatically generated

* **Mapping Distributions**
  + The models were trained Adam optimizer and generated data generated by sampling with the MAP prediction.
  + Uncertainties derived by sampling 50 times
  + Schrodinger’s Bridge
    - Precise agreement between unfolded and truth-level observables
    - Largest deviation was in low-statistics edges but the bulks of the distributions are well described
    - Bayesian uncertainty covers the deviations from the truth
  + Direct Diffusion
    - Velocity field encoded with standard Bayesian MLP
    - Both paired and unpaired approaches are precise in terms of agreements of ground truth and unfolded
    - The paired direction diffusion is more stable

A group of graphs showing different colored lines

Description automatically generated

* **Generative Unfolding**
  + Relies on paired training data
  + cINN, CFM, and VLD are reproduced at the per-cent level or better
  + cINN and CFM have similar performance yet VLD approach shows slightly larger deviations from target distributions

**A group of graphs with numbers and lines

Description automatically generated with medium confidence**

* **Learned Mapping between Reconstruction and the Truth**
  + One question is the learned distributions have failure modes that cannot be seen from the marginal distribution
  + Used trained classifier to find mismatches resulting in an ROC-AUC values in range of .5-.55

**A screenshot of a computer generated image

Description automatically generated**

Unfolding to parton level: top pairs

* **Data**
  + Apply ML-methods to top quark pair production, unfolding from reco-level to parton level
  + The task is to map reco-level 4-momenta to parton-level 4-momenta
  + Simulation with Pythia 8.306 and detector effects with Delphes 3.5.0
  + The anti-k algorithm was used to construct jets
* **Generative Unfolding**
  + Test with cINN and CFM models and their transformer variants of which the cINN has the linear layers in the last block replaced with Bayesian layers
  + The lepton and neutrino kinematics are learned slightly better than the quark kinematics
  + Transformer-enhanced networks perform better than the CFM which in turn beats the cINN

A screenshot of a graph

Description automatically generated

* **Generative Unfolding using Physics**
  + Problem with the intermediate on-shell propagators was solved using mass parametrization proposed in reference [32]
    - Directly predicts the top and W-kinematics and makes simpler decay kinematics accessible via correlations
  + Use the Breit-Wigner mapping to convert sharp mass peaks into a Gaussian-like shape
  + Just used the CFM and transformer enhanced CFM for testing and found that the difference between the two models was smaller
  + Trained a classifier to distinguish the generated events from the training data truth
    - The level of agreement significantly improved, going from VLD to the CFM, and then adding the transformer feature of the transformer enhanced CFM to encode combinatorics

A screenshot of a graph

Description automatically generated

A graph of a graph with different colored lines

Description automatically generated with medium confidence