Optimal transport solutions for pileup mitigation at hadron colliders

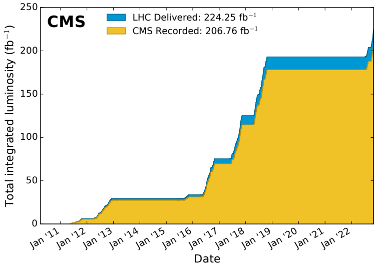
L. Gouskos [1] **F. Iemmi** [2] S. Liechti [4] B. Maier [1]
V. Mikuni [3] H. Qu [1]

[1]European Organization for Nuclear Research (CERN), Geneva
[2]Institute of High Energy Physics (IHEP), Beijing
[3]National Energy Research Scientific Computing Center (NERSC), Berkeley
[4]University of Zurich (UZH), Zurich

IHEP, Beijing, CN

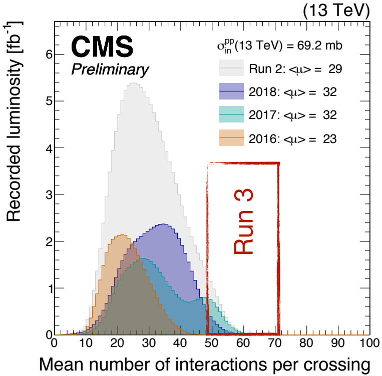
based on arXiv:2211.02029

PU mitigation at hadron colliders



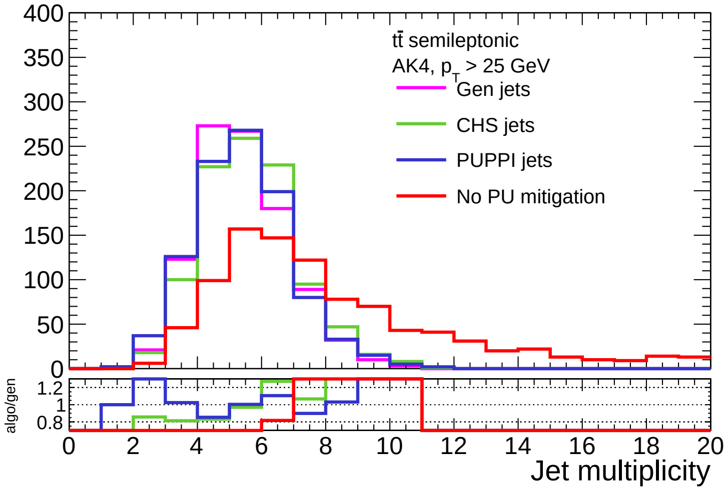
- At the LHC, usually interested in rare, **head-on pp collisions**
- Maximize probability of head-on collisions by **squeezing the proton bunches** as much as possible
- **Keep instantaneous luminosity high**
- High integrated luminosity $\Longrightarrow$ better physics outcome
- **Drawback**: non-negligible probability of having **more than one pp collision** per bunch crossing

PU mitigation at hadron colliders



- **Pileup**: additional pp collisions superimposing to main collision
- **PU** has **increased** in Run3 ($\langle nPU \rangle = 50$) and will increase in HL-LHC ($\langle nPU \rangle = 140$)
- Will severely **degrade quality of observables** (jet multiplicity, jet substructure, ...) if not properly treated
- **Easy task for charged** particles: use tracking information to disentangle particles
- **Very challenging for neutral** particles

How bad is pileup?



State-of-the-art at CMS: PUPPI [1407.6013]

- Starting from Run3, **default PU mitigation technique** in CMS **is PUPPI**
- **Rule-based** algorithm
- Calculates a weight $w \in [0, 1]$ for each particle in the event
  - Encodes the probability for a particle to be LV or not
  - Weight used to **reweight the particle 4-momentum before jet clustering**
- For charged: use tracking information and assign 0 or 1
- For neutrals: build α variable

$$\alpha_i = \log \sum_{j \neq i, \Delta R_{ij} < R_0} \left(\frac{p_{T,j}}{\Delta R_{ij}} \right)^2 \begin{cases} |\eta_i| < 2.5 & j \text{ are all charged particles from LV} \\ |\eta_i| > 2.5 & j \text{ are all kinds of particles} \end{cases}$$

- QCD is harder and more collimated than PU $\implies$ higher α than PU

State-of-the-art at CMS: PUPPI [1407.6013]

- To translate into a weight, compare each particle's α with the mean and RMS of PU particles

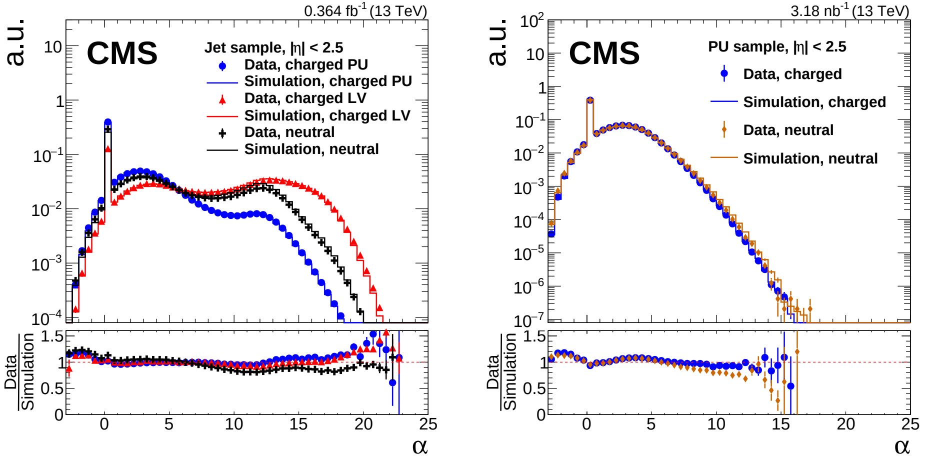
$$\text{signed}\chi_i^2 = \frac{(\alpha_i - \bar{\alpha}_{\text{PU}})|\alpha_i - \bar{\alpha}_{\text{PU}}|}{(\alpha_{\text{PU}}^{\text{RMS}})^2}$$

- Use **charged particles** for $\bar{\alpha}_{\text{PU}}$ and $(\alpha_{\text{PU}}^{\text{RMS}})^2$ computation
- Finally, assume signedχ^2 follows a χ^2 distribution and assign weight based on CDF

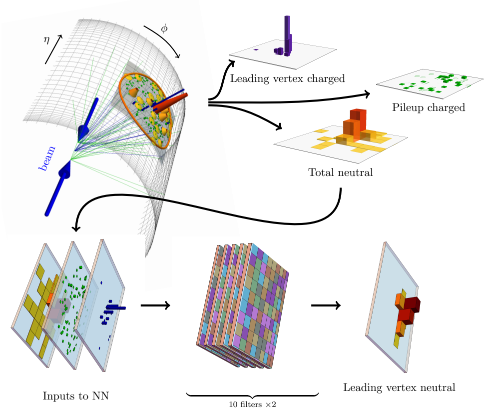
$$w_i = F_{\chi^2,\text{NDF}=1}(\text{signed}\chi^2)$$

- **LV particle** $\implies$ large signedχ^2 $\implies$ large CDF $\implies$ **large weight**
- **PU particle** $\implies$ small signedχ^2 $\implies$ small CDF $\implies$ **small weight**

State-of-the-art at CMS: PUPPI [1407.6013]

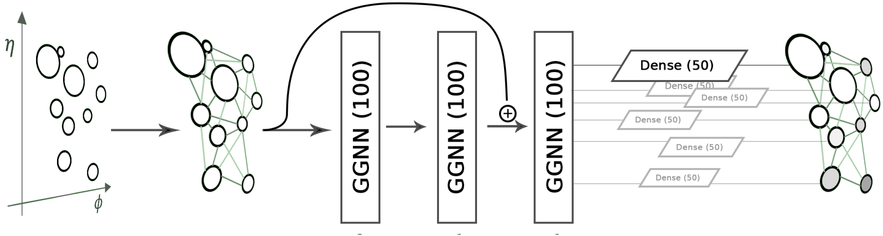


PileUp Mitigation with Machine Learning: PUMML [1707.08600]



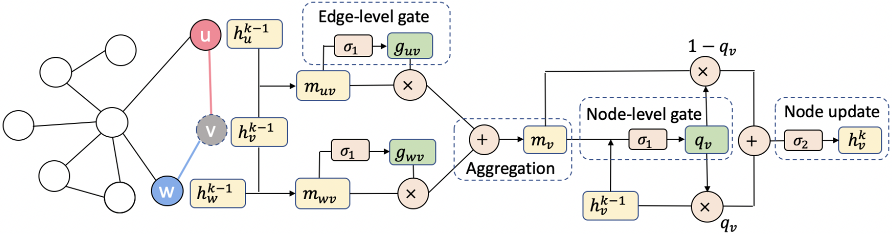
- **Treat jets as squared images**, use CNNs
- Input is a three-channel image
  - Charged radiation from LV
  - Charged radiation from PU
  - Total neutral radiation
- Output is regressed neutral radiation from LV
- Image-based approach **overlooks complex detector geometry**

PUPPI with Machine Learning: PUPPIML [1810.07988]



- PUPPI is based on **local information: use GNN** to collect it in **more expressive** ways
- Developed at Delphes particle-level (before interaction with the detector)
- **Fully supervised: use truth-labels** coming from Delphes simulation
- These are **not available in full Geant4**-based simulations!

Semi-supervised PUPPI [2203.15823]



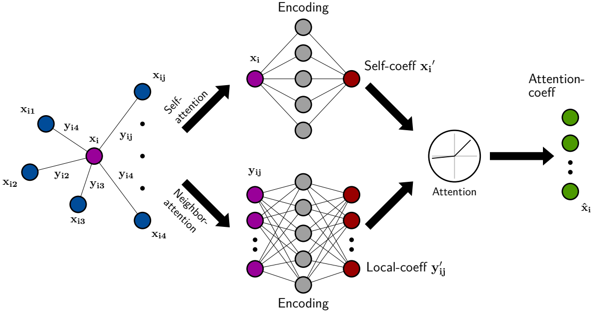
- PUPPI is based on **local information: use GNN** to collect it in **more expressive** ways
- Developed at Delphes particle-level (before interaction with the detector)
- **Semi-supervised:** train on charged (labels exist in Geant4 as well), apply on neutrals
- Can train on data, but **requires extrapolations** (charged $\rightarrow$ neutrals, central $\rightarrow$ forward)

Overview of PU mitigation techniques

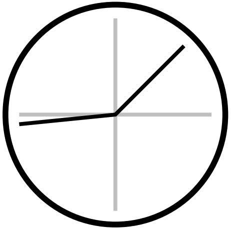
- Currently in use (e.g., CMS): **PUPPI** [1407.6013]
  - **Rule-based** algorithm
  - For each neutral particle, consider the energy of neighboring particles
  - Extract a probability for the particle to be LV or PU
  - Relies on properties of charged particles and **extrapolates to neutrals**
- Nature and complexity of task inspired **machine-learning-based approaches**
  - PUMML: treat jets as images, reconstruct LV neutral radiation [1707.08600]
  - PUPPIML: use GNN, rely on Delphes truth labels [1810.07988]
  - Semi-supervised PUPPI: train on charged, apply on neutrals [2203.15823]
- **Recurring problem**: lack of truth **labels for neutrals** in full simulation
- We developed a **new ML-based approach** to overcome this bottleneck
  - Use **Attention-Based Cloud Network** (ABCNet, [2001.05311]) **combined with optimal transport**
  - **TOTAL**: **Training Optimal Transport with Attention Learning**
  - Train model on a Delphes-based simulation of the CMS Phase2 detector

Attention-Based Cloud network

- ABCNet is an **graph neural network** enhanced **with attention mechanisms**
  - Treat particle collision data as a set of permutation-invariant objects
  - Attention mechanisms filter out the particles that are not relevant for the learning process
- Implemented inside custom **graph attention pooling layers** (GAPLayers)



Attention mechanism



Attention

- Add together self- (x_i') and local- (y_{ij}') coefficients and apply non-linearity

$$c_{ij} = \text{LeakyRelu}(x_i' + y_{ij}')$$

- Align coefficients c_{ij} by applying SoftMax

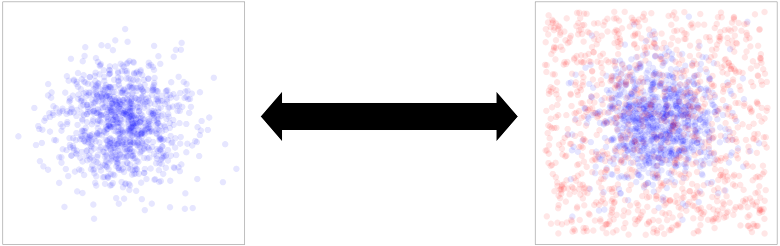
$$c_{ij}' = \frac{\exp(c_{ij})}{\sum_k \exp(c_{ik})}$$

- Get attention coefficients by multiplying y_{ij}' by c_{ij}'

$$\hat{x}_i = \text{Relu}\left(\sum_j c_{ij}' y_{ij}'\right)$$

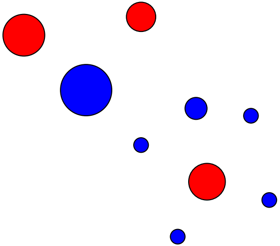
A novel approach to PU mitigation

- Definition of truth labels is highly non trivial in simulations at hadron colliders
- **Our approach**: simulate **identical** proton-proton **collisions in two scenarios**
  1. Only the hard interaction is simulated: **no-PU sample**
  2. Pileup is superimposed to the hard interaction: **PU sample**
- **Do not assign per-particle labels**: rather just assign a “global“ label to samples
- Train network to **learn differences between the two samples**



How to learn: OT concepts for a loss function

- We build a custom **loss inspired by optimal transport** ideas (OT)



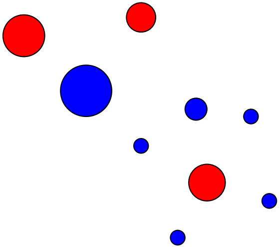
- OT example: the **Earth Mover's Distance** is the minimum work to move **earth** to fill some **holes**

$$EMD(\vec{x}, \vec{y}) = \min_{f} W(f, \vec{x}, \vec{y})$$

- With OT you can **match distributions** (e.g., earth-holes)
- We want to match the distributions for the no-PU particles and **PU particles weighted by an ABCNet weight** ($\vec{\omega}$)

Limitation of the EMD loss function

- Using the **EMD loss comes with some limitations**



- We consider 9000 particles per event
- **High computational cost**: feasible flows f are 9000x9000 matrices
- Can only match **3D distributions**
- We worked on a modified loss to
  - **Match higher-dimensionality distributions**
  - Solve OT efficiently
- **Sliced Wasserstein Distance** (SWD)

Efficient OT: sliced Wasserstein distance (SWD)

- The optimal transport problem has a **closed form for 1D problems**:

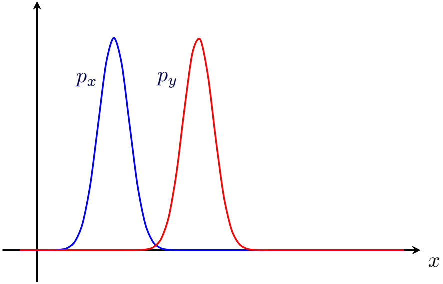
$$W_c(p_X, p_Y) = \int_0^1 c\left(P_X^{-1}(\tau), P_Y^{-1}(\tau)\right) \mathrm{d}\tau$$

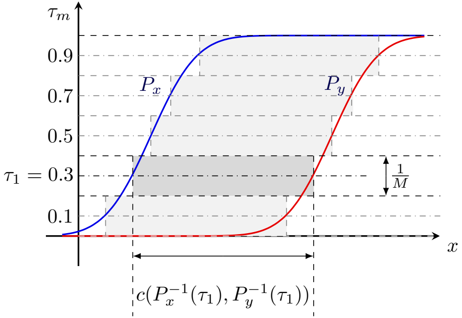
where p_X, p_Y are 1D PDFs, $P_X^{-1}(\tau), P_Y^{-1}(\tau)$ are the respective CDFs and $c(\cdot,\cdot)$ is the transportation cost function

- No guarantee that the integral is solvable (it depends on the form of $c(\cdot,\cdot)$)
- The **integral can always be approximated** by the finite sum

$$\frac{1}{M}\sum_{m=1}^{M} c\left(P_X^{-1}(\tau_m), P_Y^{-1}(\tau_m)\right), \qquad \tau_m = \frac{2m-1}{2M}$$

Example: M = 5





- $m \in \{1, 2, 3, 4, 5\} \implies \tau_m = \frac{2m-1}{2M} \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$

Efficient OT: sliced Wasserstein distance (SWD)

- In the **special case of discrete distributions** (discrete in nature, or resulting from a sampling), PDFs are sums of Dirac's deltas

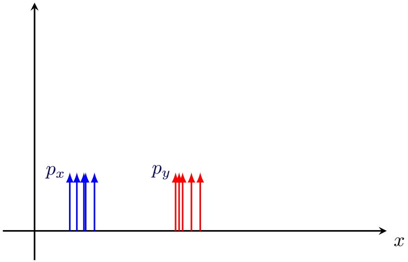
$$p_x = \frac{1}{M}\sum_{m=1}^{M}\delta(x - x_m); \qquad p_y = \frac{1}{M}\sum_{m=1}^{M}\delta(y - y_m);$$

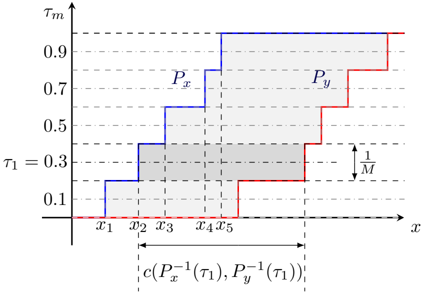
- The integral of a Dirac's delta is the Heaviside's step function $\Theta \implies$ $\implies$ CDFs are Heaviside functions

$$P_x(t) = \int_{-\infty}^{t} p_x(z)\mathrm{d}z = \frac{1}{M}\int_{-\infty}^{t}\sum_{m=1}^{M}\delta(z - x_m)\mathrm{d}z = \frac{1}{M}\sum_{m=1}^{M}\Theta(t - x_m)$$

- **If we sort the samples by feature**, the CDFs become a **sum of steps**

Example: M = 5





- $m \in \{1, 2, 3, 4, 5\} \implies \tau_m = \frac{2m-1}{2M} \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$
- Note that

$$P_x^{-1}(\tau_m) = x_m; \qquad P_y^{-1}(\tau_m) = y_m$$

Efficient OT: sliced Wasserstein distance (SWD)

- Note that

$$P_x^{-1}(\tau_m) = x_m; \qquad P_y^{-1}(\tau_m) = y_m$$

- Therefore

$$W_c(p_X, p_Y) = \frac{1}{M}\sum_{m=1}^{M} c\left(P_X^{-1}(\tau_m), P_Y^{-1}(\tau_m)\right) = \frac{1}{M}\sum_{m=1}^{M} c\left(x_m, y_m\right)$$

- The **1D OT problem is reduced to a sorting** of the 1D feature
  - **Fast and easy to solve**

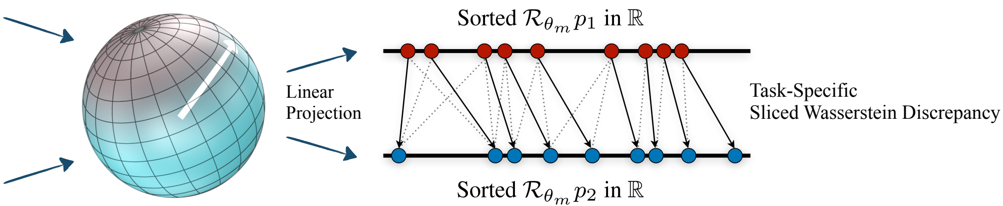
Efficient OT: sliced Wasserstein distance (SWD)

CHECKPOINT

1. Optimal transport problem has a closed form in 1D
2. For sampled distributions, the problem is reduced to a sorting of the 1D feature
3. Particles have multi-dimensional distributions though. How to apply this?

Efficient OT: sliced Wasserstein distance (SWD)

- Each particle is a sample from a *n*-D feature space
- **SWD**: take *n*-D feature space and **project (slice)** it **to 1D**
- Project on a vector belonging to S^{n-1}
- For robustness, take **multiple random slices**
- Now can **solve the 1D OT problem for each slice**
- **Sort particles by slice**
- The **average on all slices and particles** becomes the **loss function**



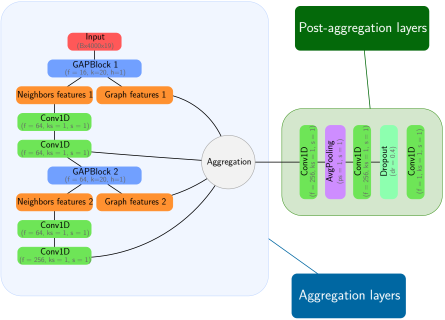
Energy conservation in OT: MET constraint

- SWD focuses on the optimal matching between individual particles in no-PU and PU samples
  - No guarantee that energy is conserved between the two
- Add an **event-level MET constraint** term to the loss
  - Enforce energies in no-PU and PU events to be similar
- Final loss function:

$$\mathcal{OT} = \mathrm{SWD}(\vec{x}_p \cdot \vec{\omega}, \vec{x}_{np}) + \mathrm{MSE}\left(\mathrm{MET}(\vec{x}_p \cdot \vec{\omega}), \mathrm{MET}(\vec{x}_{np})\right)$$

where $\vec{x}_p$ = PU sample; $\vec{x}_{np}$ = no-PU sample; MSE = mean squared error

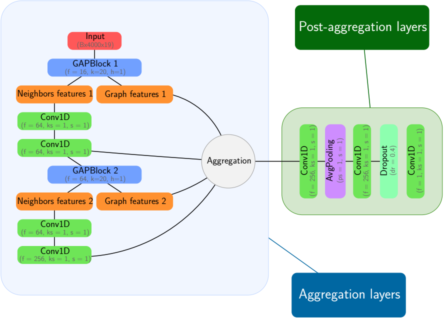
The model



- **9 input features**:
  - (p_T, η, ϕ, E)
  - Charge
  - PDG ID
  - dXY & dZ impact parameters
  - PUPPI weight
- **Loss**: $\text{SWD}(\vec{x}_p \cdot \vec{\omega}, \vec{x}_{np}) + \text{MET}$ constraint
- **Cost function**: squared distance
- **Sliced features**: (p_T, η, ϕ, E)
- **Output**: per-particle weight $\vec{\omega}$

- Train on **300k events**, equally split between QCD multijet, $t\bar{t}$ dileptonic and VBF Higgs(4ν) processes
- Consider **9000 particles per event** (zero-padding included)
- Gather the **20 *k*-nearest neighbors** for each particle when building graph

The model



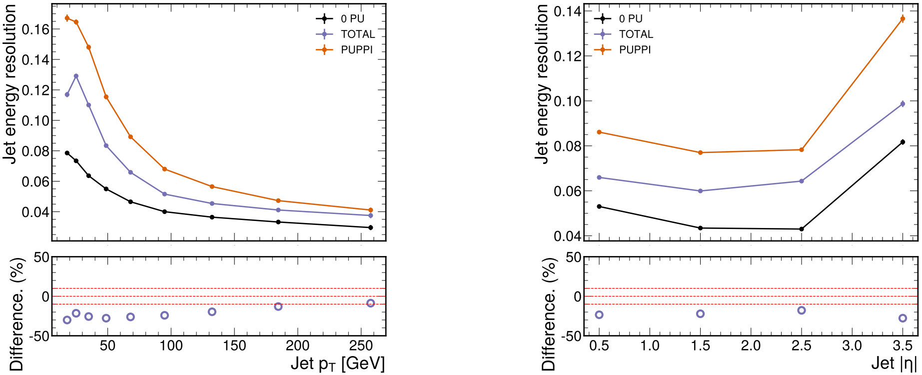
- **Compare** TOTAL **with PUPPI and no-PU** scenario
- **Reweight** each particle's 4-momentum by the network weight
- **Cluster** TOTAL jets and TOTAL MET

- We define the resolution as:

$$\delta = \frac{q_{75\%} - q_{25\%}}{2}$$

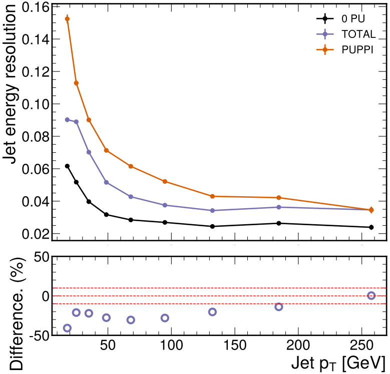
where $q_{X\%}$ is the X-th quantile of the considered response distribution

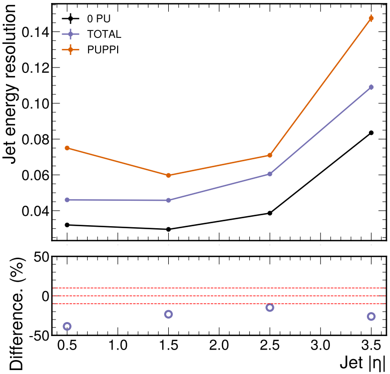
Results: QCD multijet



- Jet energy resolution as a function of jet p_T (left) and jet η (right)
- Improvement up to 30% in JER, up to 25% in η resolution

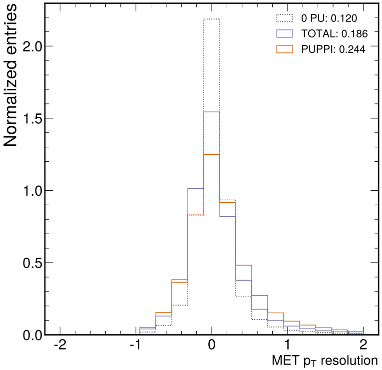
Results: dileptonic $t\bar{t}$

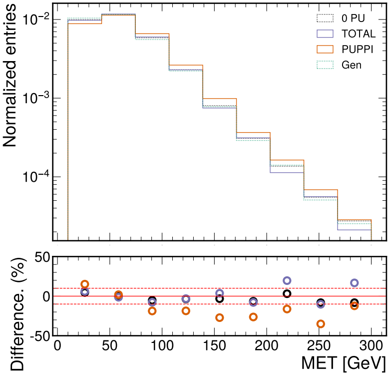




- Jet energy resolution as a function of jet p_T (left) and jet η (right)
- Improvement up to 40% in JER, up to 40% in η resolution

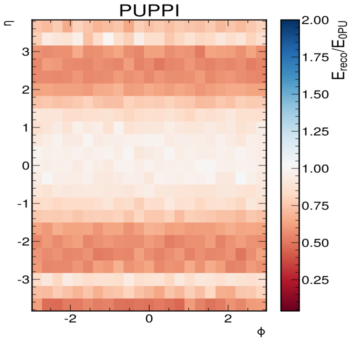
Results: dileptonic $t\bar{t}$

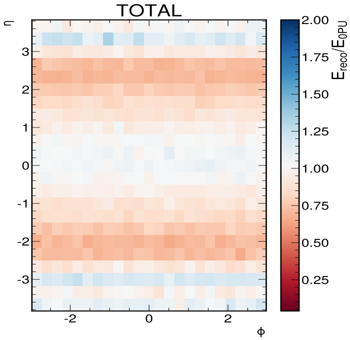




- MET p_T resolution (left); MET p_T distribution (right)
  - MET resolution is reduced by 24%

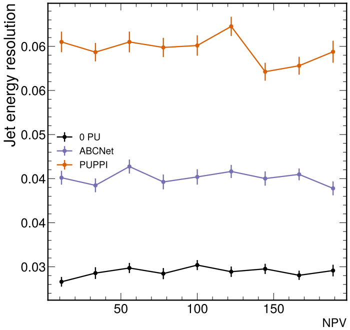
Inspecting particles weights





- Ratio $\frac{p_T \times \omega}{p_{T,\mathrm{noPU}}}(\eta, \phi)$ for PUPPI and TOTAL (right) in QCD multijet events
- Smoother behavior for TOTAL in central and forward region
- Still room for improvement in transition region

Robustness



- Evaluate resolution on **processes and PU scenarios unseen during training**
- Network is trained on QCD+$t\bar{t}$+VBF with $\langle \text{NPV} \rangle = 140$
- Evaluate on W+jets production, flat NPV between 0 and 200

Conclusions

- We presented **novel algorithm to reject PU particles** at high-intensity hadron colliders
  - Trained and tested on Delphes simulation of Phase2 CMS detector
- We are Training Optimal Transport with Attention Learning: **TOTAL**
- We **solved the longstanding problem of neutral labels** in PU mitigation
- **We do not rely on explicit, per-particle labeling**
- **Learning** happens **through OT in a self-supervised fashion**
- Such an algorithm will be **crucial at the High-Luminosity LHC**, where much harsher data-taking conditions are expected
- Our **approach can be generalized** to a wide range of denoising problems
  - Only needed input is a reliable simulation of signal and noise