SFT Meeting

Status Update

Konrad Helms

4th October 2023

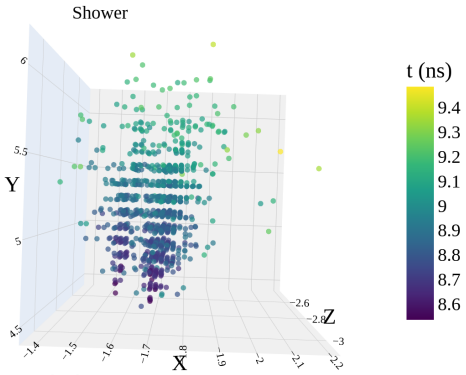
Investigation of MCtruthE/totalE Cut

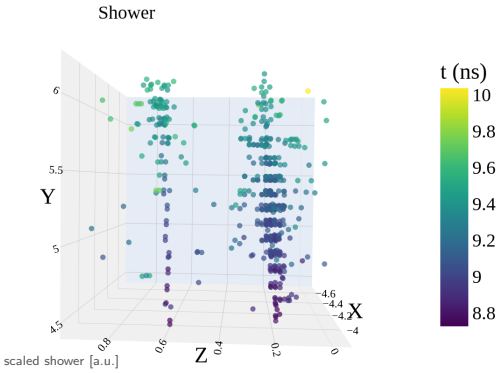
- removed cut:

$$\frac{\sum_i^{\#\text{ MC truth hits}} E_{\text{MC truth, hit } i}}{\sum_j^{\#\text{ all hits}} E_{\text{hit } j}} \geq 0.9$$

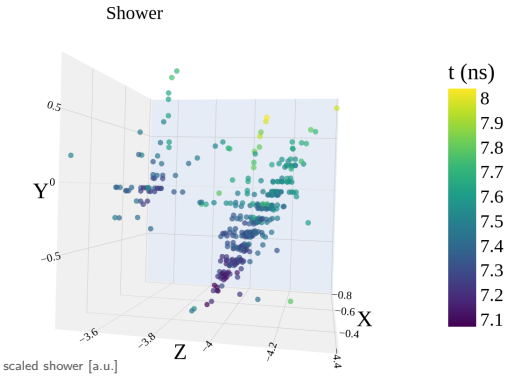
- investigated (some) showers:

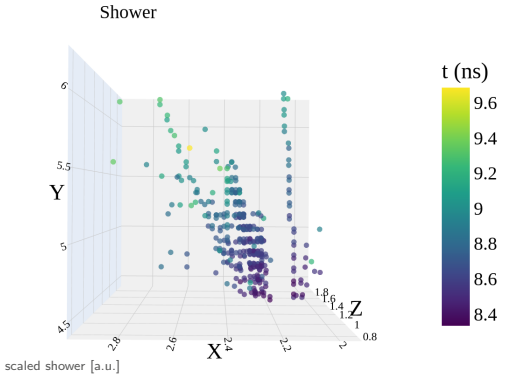
Showers without the cut #1





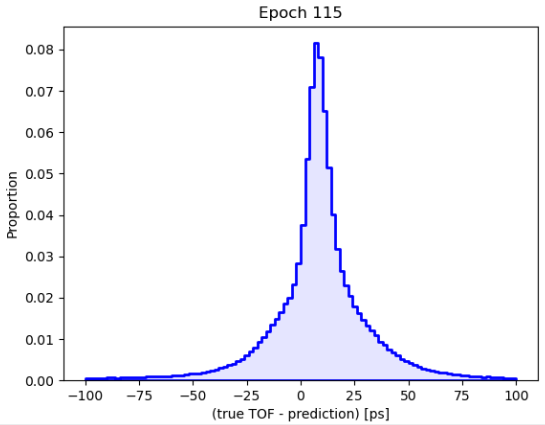
Showers without the cut #1





5 Layer CNN - On Data Without Cut

RMS90: 18.5 ps



Cut OLD

```
auto isCleanShower = [&navEcalHit](ReconstructedParticle* pfo, MCParticle* mcTrue) -> bool {
    /* check if the shower contains other hits with the highest energy contributions from another MC particles.
       NOTE: Contributions from other MC particles are still allowed if they are not dominating by the energy.
    */
    Cluster* cluster = pfo->getClusters()[0];
    for (auto hit : cluster->getCalorimeterHits()) {
        const std::vector<LCObject*>& simHits = navEcalHit.getRelatedToObjects(hit);
        const std::vector<float>& weights = navEcalHit.getRelatedToWeights(hit);
        // it should really be always 1, but just in case
        if (simHits.size() == 0) continue;
        SimCalorimeterHit* simHit = static_cast<SimCalorimeterHit*>(simHits[std::max_element(weights.begin(), weights.end()) - weights.begin()]);
        int nCont = simHit->getNMCContributions();
        MCParticle* particleCont = nullptr;
        double highestEnergyCont = -1.;
        for (int i = 0; i < nCont; i++) {
            if (simHit->getEnergyCont(i) > highestEnergyCont) {
                highestEnergyCont = simHit->getEnergyCont(i);
                particleCont = simHit->getParticleCont(i);
            }
        }
        double trueEnergyFraction = highestEnergyCont / simHit->getEnergy();
        if (particleCont != mcTrue || trueEnergyFraction < 0.5) return false;
    }
    return true;
};
```

Cut NEW

```
auto isCleanShower = [&navEcalHit](ReconstructedParticle* pfo, MCParticle* mcTrue) -> float {
    /* clean shower == true particle contributes >= 90% energies */
    Cluster* cluster = pfo->getClusters()[0];
    float totalEnergy = 0.;
    float showerEnergy = 0.;
    for (auto hit : cluster->getCalorimeterHits()) {
        const std::vector<LCObject*>& simHits = navEcalHit.getRelatedToObjects(hit);
        const std::vector<float>& weights = navEcalHit.getRelatedToWeights(hit);
        // it should really be always 1, but just in case
        if (simHits.size() == 0) continue;
        SimCalorimeterHit* simHit = static_cast<SimCalorimeterHit*>(simHits[std::max_element(weights.begin(), weights.end()) - weights.begin()]);
        int nCont = simHit->getNMCContributions();
        for (int i = 0; i < nCont; i++) {
            totalEnergy += simHit->getEnergyCont(i);
            if (simHit->getParticleCont(i) == mcTrue) showerEnergy += simHit->getEnergyCont(i);
        }
    }
    if (totalEnergy == 0.) {
        return 0.;
    } else {
        return (showerEnergy / totalEnergy);
    }
};
```

RANSAC TOF Prediction