

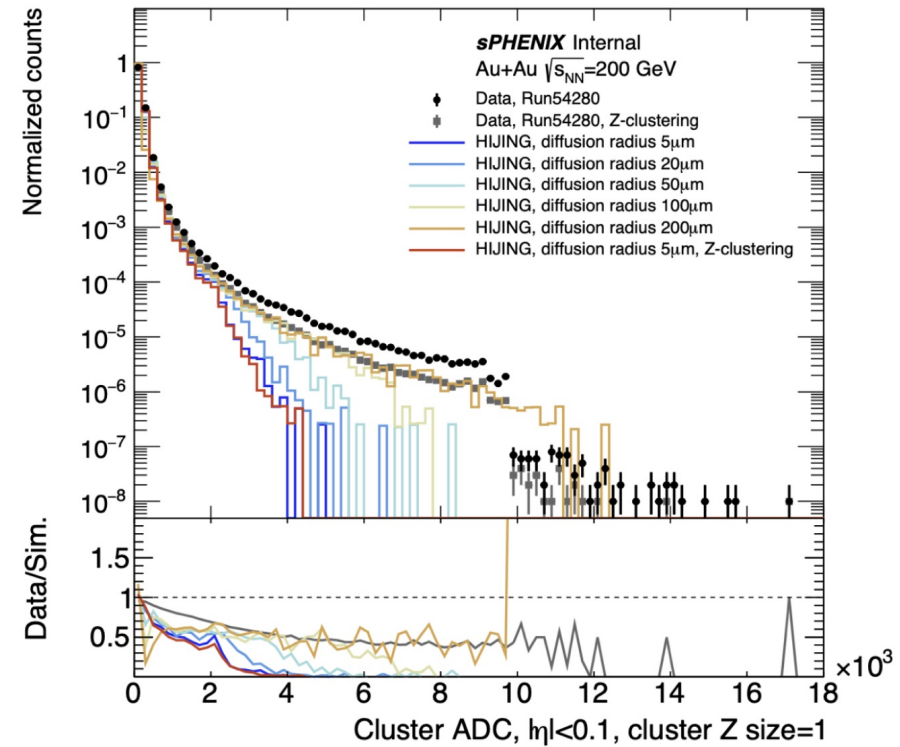
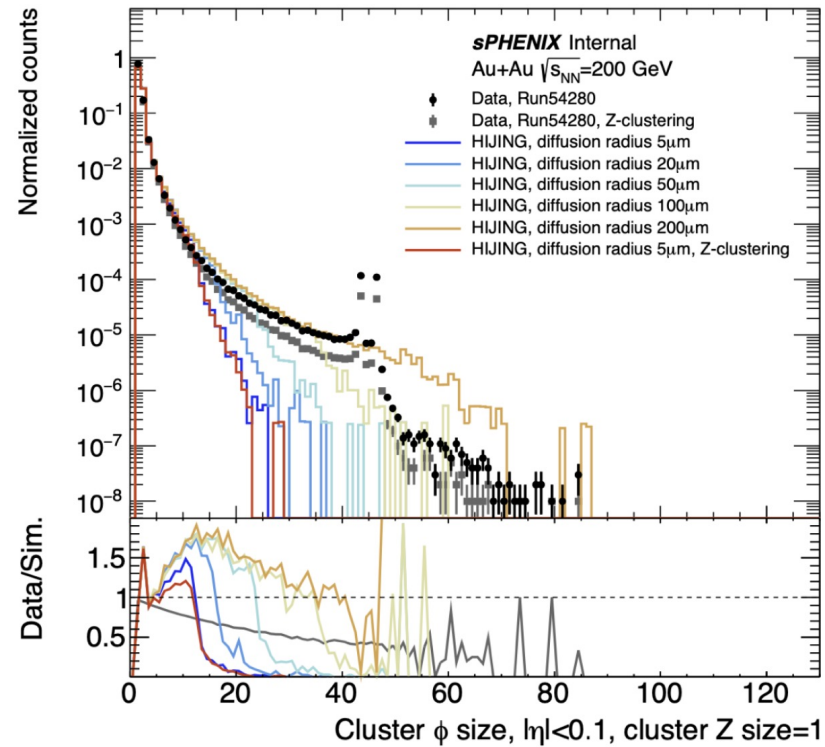
INTT Simulation Optimization

Milan Stojanovic¹, Cheng-Wei Shih²

¹Wayne State University

²National Central University & RIKEN

Description



Discrepancy of cluster phi size & ADC in data and simulation

- simple change of diffusion radius does not fix it

Diffusion algorithm

- ❑ Diffusion fixed within cylinder with constant radius
- ❑ Only do the charge diffusion if the number of strips involved for this G4step is smaller than 13

```
double diffusion_radius = diffusion_width;

if (Verbosity() > 5)
{
    std::cout << " segment " << i
        << " interval " << interval
        << " frac " << frac
        << " local_in.X " << hiter->second->get_local_x(0)
        << " local_in.Z " << hiter->second->get_local_z(0)
        << " local_in.Y " << hiter->second->get_local_y(0)
        << " pathvec.X " << gsl_vector_get(m_PathVec, 0)
        << " pathvec.Z " << gsl_vector_get(m_PathVec, 2)
        << " pathvec.Y " << gsl_vector_get(m_PathVec, 1)
        << " segvec.X " << gsl_vector_get(m_SegmentVec, 0)
        << " segvec.Z " << gsl_vector_get(m_SegmentVec, 2)
        << " segvec.Y " << gsl_vector_get(m_SegmentVec, 1) << std::endl
        << " diffusion_radius " << diffusion_radius
        << std::endl;
}

// Now find the area of overlap of the diffusion circle with each pixel and apportion the energy
for (int iz = minstrip_z; iz <= maxstrip_z; iz++)
{
    for (int iy = minstrip_y; iy <= maxstrip_y; iy++)
    {
        // Find the pixel corners for this pixel number
        double location[3] = {-1, -1, -1};
        layergeom->find_strip_center_localcoords(ladder_z_index, iy, iz, location);
        // note that (y1,z1) is the top left corner, (y2,z2) is the bottom right corner of the pixel - circle_rectangle_intersection expects this ordering
        int type = (ladder_z_index == 0 || ladder_z_index == 2) ? 0 : 1; // ladder ID 0 and 2 are type-A (1.6 cm), ladder ID 1 and 3 are type-B (2.0 cm)
        double y1 = location[1] - layergeom->get_strip_y_spacing() / 2.0;
        double y2 = location[1] + layergeom->get_strip_y_spacing() / 2.0;
        double z1 = location[2] + layergeom->get_strip_z_spacing(type) / 2.0;
        double z2 = location[2] - layergeom->get_strip_z_spacing(type) / 2.0;

        if (Verbosity() > 5)
        {
            std::cout << PHWHERE << " ladder_z_index " << ladder_z_index << " strip size in z (from CylinderGeomIntt) " << fabs(z1 - z2) << " strip size in y (from CylinderGeomIntt) " << fabs(y1 - y2) << std::endl;
        }

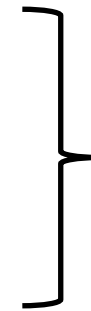
        // here m_SegmentVec.1 (Y) and m_SegmentVec.2 (Z) are the center of the circle, and diffusion_radius is the circle radius
        // circle_rectangle_intersection returns the overlap area of the circle and the pixel. It is very fast if there is no overlap.
        double striparea_frac = PHG4Utils::circle_rectangle_intersection(y1, z1, y2, z2, gsl_vector_get(m_SegmentVec, 1), gsl_vector_get(m_SegmentVec, 2), diffusion_radius) / (M_PI * (diffusion_radius * diffusion_radius));
```

All the G4Steps have the same charge diffusion radii, not nature

Plan overview

1. Understand diffusion logic

- ❑ Using particle gun MC
- ❑ Simplest environment where we can isolate effect



MC only

2. Use beam test data to tune MC

MC generation

Using module prepared by Cameron: [dNdeta_sPHENIX_simulations](#)

```
const string generator = "SIMPLE",  
const bool fullSim = true,  
const bool turnOnMagnet = false,  
const bool idealAlignment = true,  
const string particleType = "pi+"
```

```
set_eta_range(-0.005, 0.005)  
set_pt_range(10, 20.)
```

MC generation

Changes to the core software: [PHG4InttHitReco.cc](#)

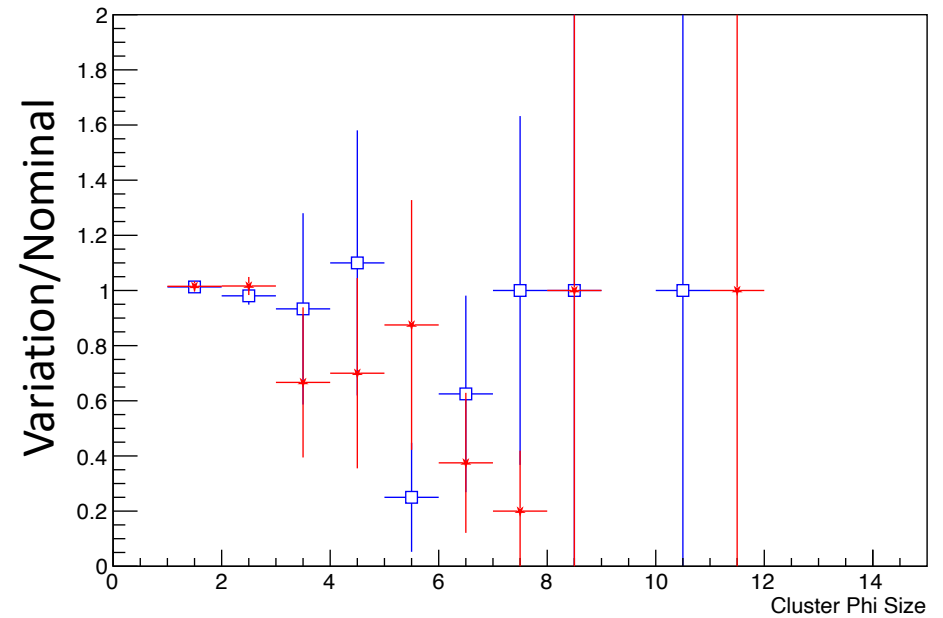
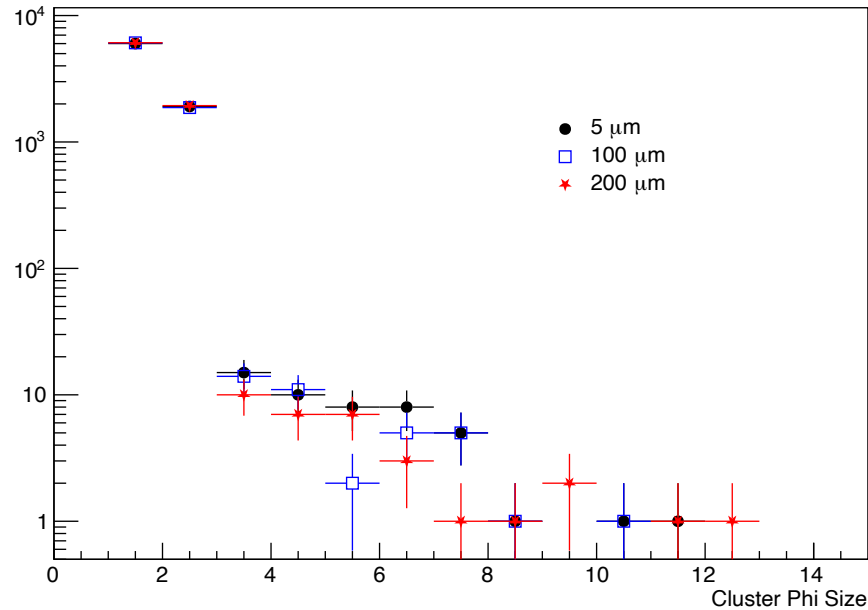
```
diffusion_width = 5.0e-04 #nominal  
→  
diffusion_width = 200.0e-04 #variation 1  
diffusion_width = 200.0e-04 #variation 2
```

Commented out:

```
if (maxstrip_y - minstrip_y > 12 || maxstrip_z - minstrip_z > 12)
```

5k events generated for diffusion width
~10k clusters

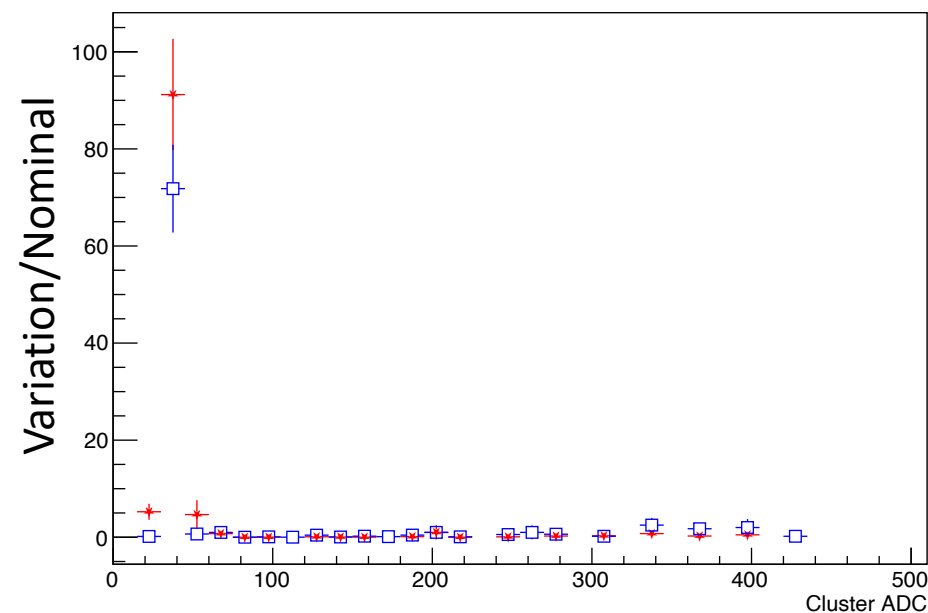
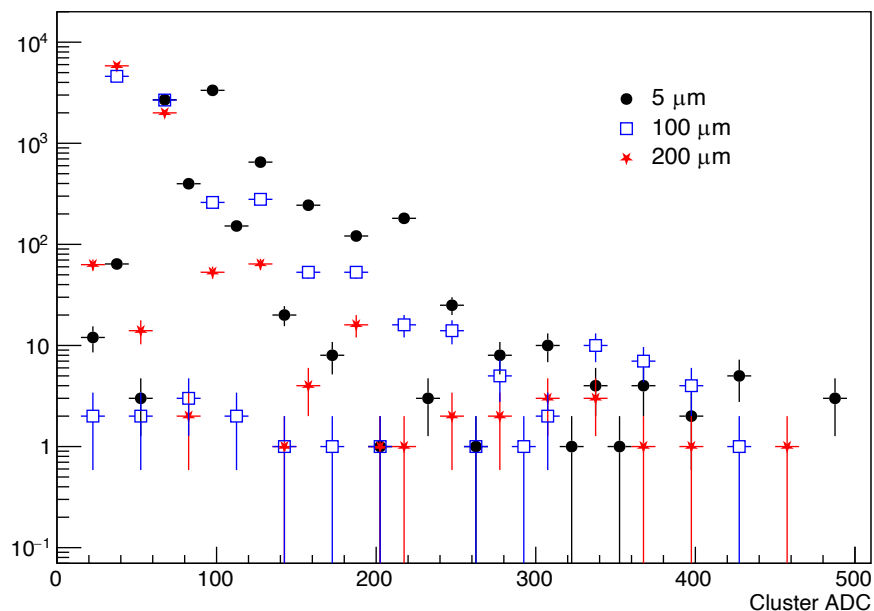
MC Results – Cluster Phi Size



Diffusion radius modification

- Cluster Size independent on the diffusion radius (at low Cluster Phi Size)

MC Results – Cluster ADC



Diffusion radius modification

➤ Cluster ADC decreases with diffusion width

Different behavior of Phi Size and ADC
Threshold effect?

Data vs MC Comparison

❑ Data

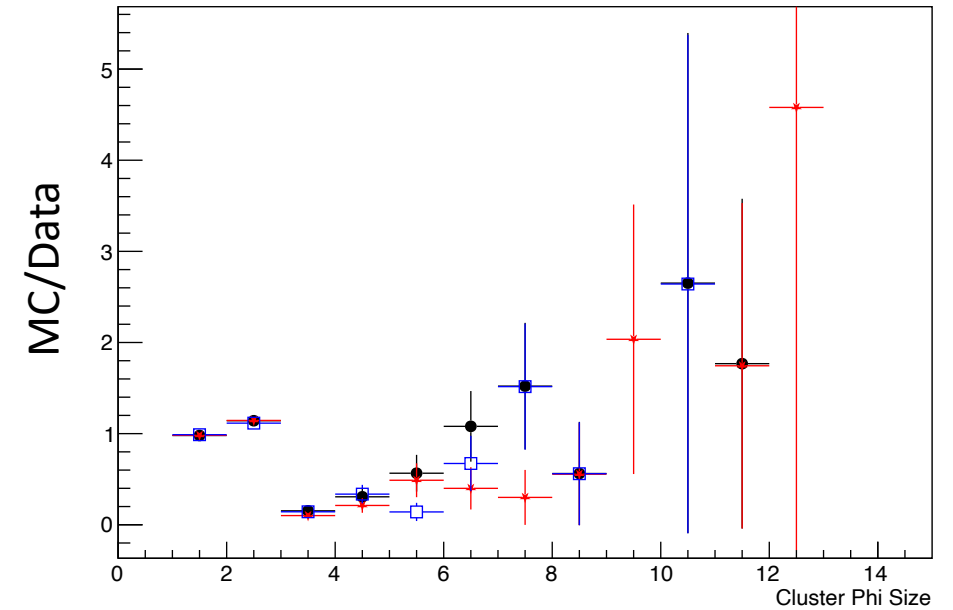
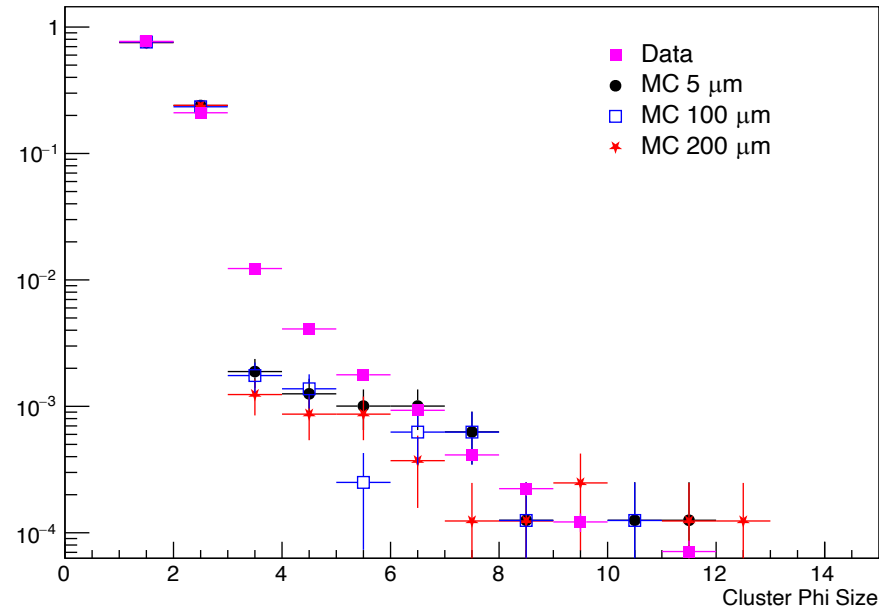
- Positron beam
- $p_T = 1 \text{ GeV}$
- Bias = 50 V

❑ MC

- π^+ particle gun
- $10 < p_T < 20 \text{ GeV}$
- Bias = 100 V

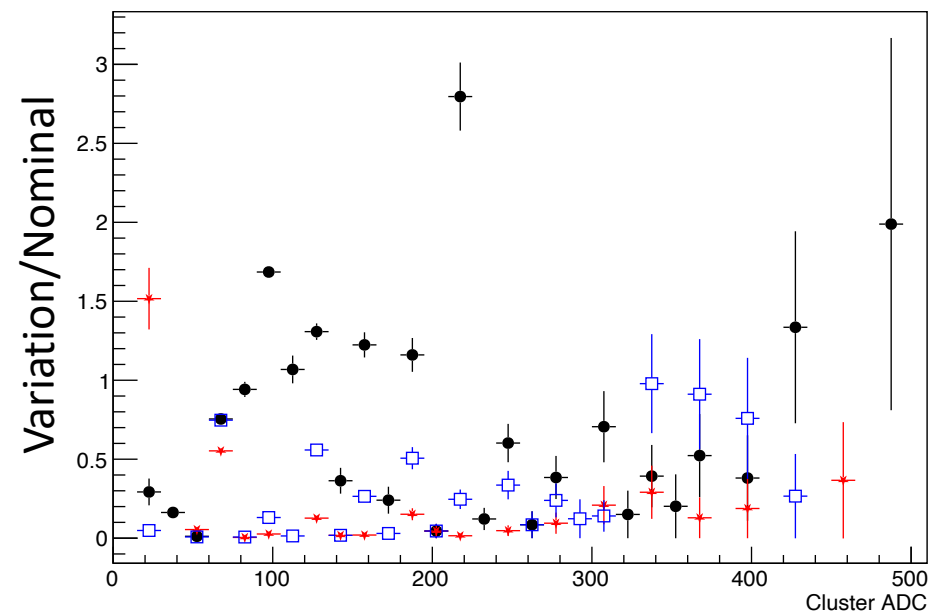
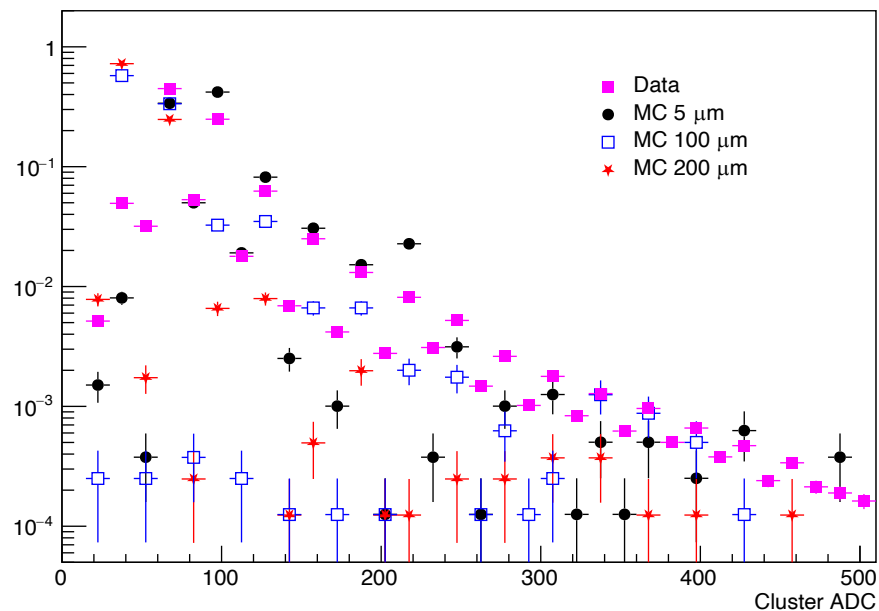
NOT apple to apple comparison

Data vs MC – Cluster Phi Size



Significant discrepancies start for Phi Size > 2

MC Results – Cluster ADC



Best agreement with nominal (5 μm) width?

Summary & Next steps

- ❑ Framework for studying diffusion implementation ready
 - Particle gun generator
- ❑ Change of diffusion width does not change cluster phi size at lower values
- ❑ Adjust selection for Data comparison