Tracking in STAR

Yuri Fisyak

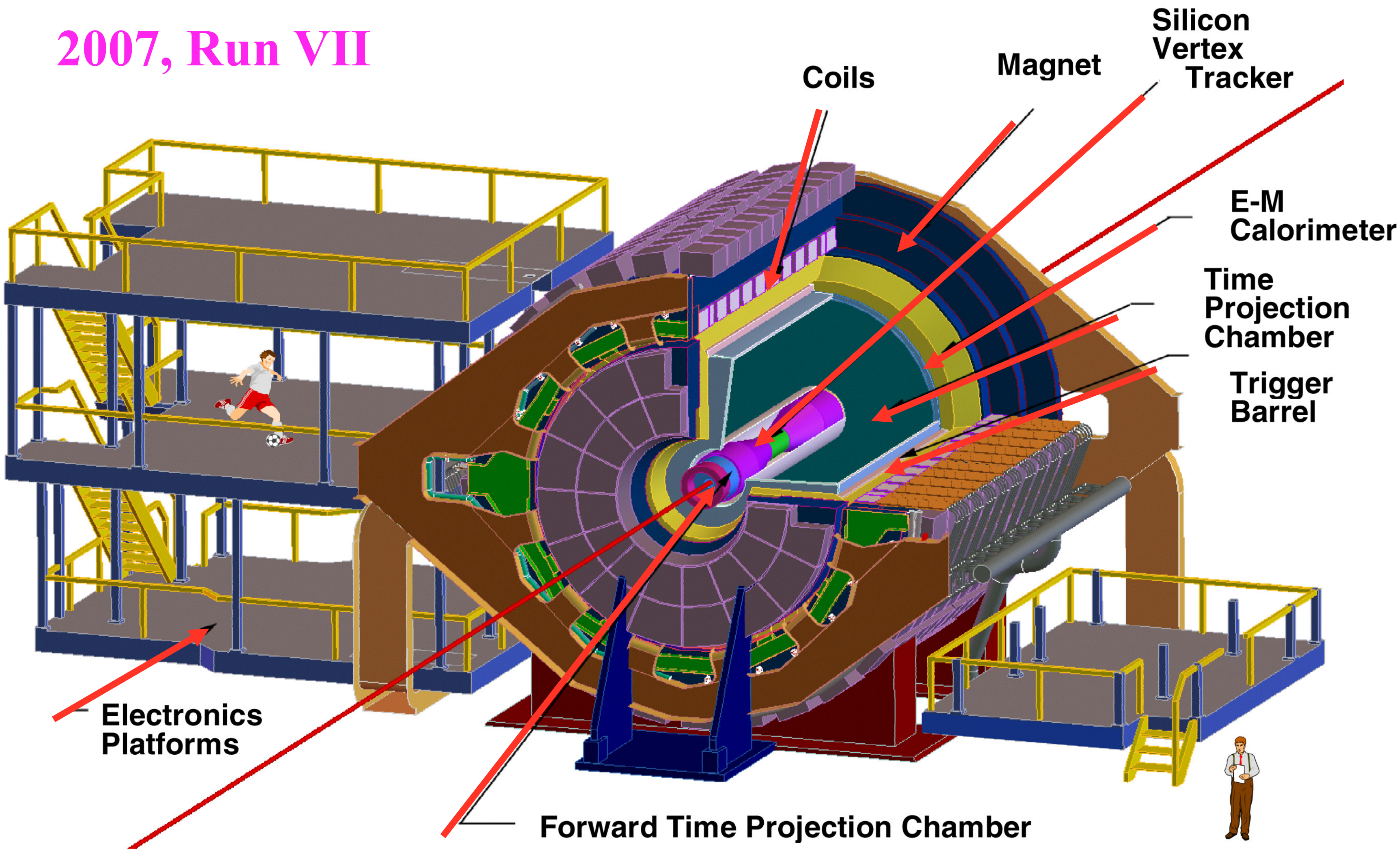
Outlook

- Introduction to STAR and STAR tracking history.
- STAR tracking review (November 14-19, 2011, LBNL)
- Plans, problems, discussions.

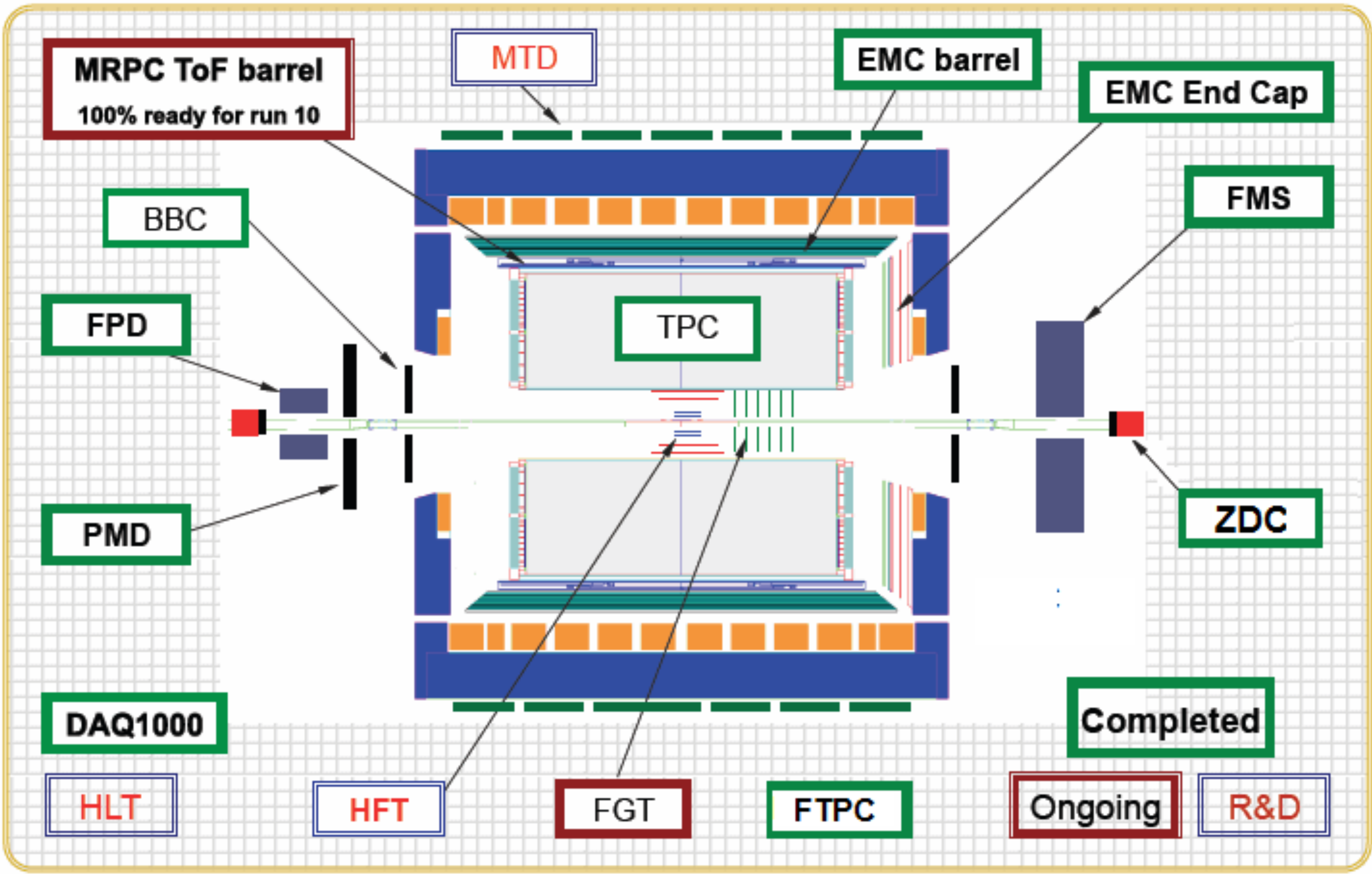
Introduction to STAR

- STAR is running detector for last 12 years.
- Right now we are taking data in Run XII (pp 200 GeV right now, pp 500 500 GeV next, heavy ion part is yet defined: AuAu, UU, or CuAu).
- STAR tracking software now is in its third development phase. Each phase corresponds to new challenges due to detector and accelerator upgrades.

STAR - Solenoidal Tracker At RHIC



The STAR Detector upgrades since 2008



STAR Upgrades time line

- 2008: SVT has been removed
- 2009: DAQ 1000, completed (1000 stands for ~1kHz data taking rate. I remind you that STAR has started as DAQ ~1Hz detector).
- 2010: ToF completed
- 2011:
  - FTPCs have been removes,
  - 60% of Forward Gem Tracker (FGT) installed,
  - Run XII is engineering run for FGT
- 2012:
  - FGT will be completed
  - Heavy Flavor Tracker (HFT) prototype
  - Muon Telescope Detector (MTD)
- 2014:
  - MTD completed
  - HFT completed

The first version of STAR Tracking (tpt)

- This version has been developed in 1995 - 2003.
- TPC pattern recognition based on ALEPH reconstruction + GEANE propagator & Kalman (re)fitter to account energy loss and multiple scattering.
- FTPC reconstruction, based on track follower with primary vertex reconstructed with TPC tracks.
- EST: Silicon hits to TPC track matcher.
- TPC reconstruction was ready by day 1 and used till 2005. No major problems has been observed. The development was mainly in calibration, alignment, and space charge effects.
- FTPC reconstruction has been developed and used since 2001.
- The main **problem** in this version was track **reconstruction** with **Silicon** hits. The reconstruction had lack of tracking precision, calibration and alignment tools.

STAR integrated tracking (Sti)

- The development of this version was started in 2001.
- The main ideas were :
  - To use Kalman filter in full scale
  - To use c++ with hope to improve maintainability and speed up reconstruction.
- The project has been completed in 2005.
- In 2006 Silicon detector calibration and alignment tools based on Sti have been developed. It was achieved designed Silicon hits precision ~50 μm.
- Run V (CuCu 200 GeV/n) and Run VII (AuAu 200 GeV/n) have been processed with Silicon detectors in reconstruction. Right now the analysis with attempt to extract charm signal from these data samples is **still** ongoing.

Stv, STAR Integrated tracker based on ROOT VMC Geometry

- The limitation of the present Sti:
  - Sti geometry is too simplified with respect to the real one, especially for dead materials.
  - Sti does not treat disk type tracking detectors like FGT.
- The limitations becomes essential due coming upgrades:
  - FGT, Forward GEM disks (coming 2012),
  - MTD, Muon Telescope Detector (2010-2014), and
  - HFT, Heavy Flavor Tracker: 2 layers of pixels, 1 layer of single side Silicon strips (IST) and SSD. (2014)
- The proposed solution is to use ROOT TGeo classes and approach a la GEANE. The first version of **Stv** has been developed and presented during STAR tracking review (Nov. 2011).

Cellular Automata (CA) as track seed finder

- One of the results of the First Tracking workshop was an idea to use Alice High Level Trigger (HLT) tracker based on CA and Kalman Filter (KF), developed by GSI group, as STAR TPC track seed finder in offline reconstruction (StiCA or/and StvCA).
- The goal is to port the CA codes after evaluation to STAR HLT.
- StiCA and StvCA were a part of STAR tracking review last November.
- I would like to give credits to
  - Igor & Maksym who integrated CA in STAR framework, and
  - STAR framework (root4star) which allowed to run simultaneously in the same library release 4 different versions of reconstruction: Sti, Stv, StiCA and StvCA.

STAR Tracking review

Focused on the 3 components to be reviewed

- AgML: Abstract geometry model
- CA: Cellular Automaton seed finder:
  - performance StiCA versus Sti.
- Stv: VMC tracker, performance:
  - Stv versus Sti
  - StvCA versus Stv, Sti and StiCA

Performance studies

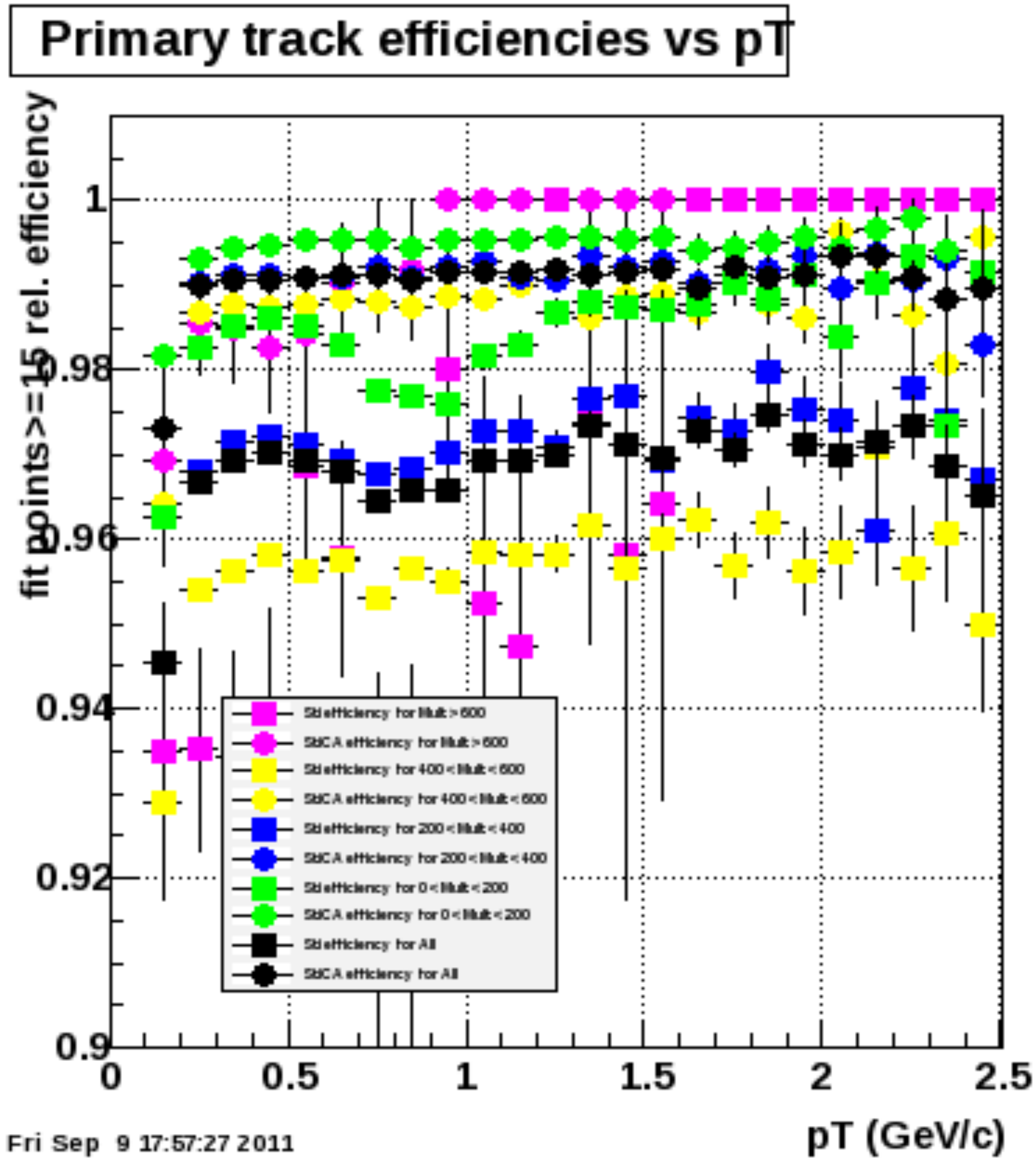
For performance studies included 3 main points:

- Simulation (MC): comparison reconstructed tracks with MC ones. Efficiencies and track parameters QA. http://www4.rcf.bnl.gov/~fisyak/star/RECO/Eval/MC.
- Track by track comparison (TbyT): QA based on comparison of reconstructed real data by different trackers (Sti, StiCA, Stv and StvCA). The track by track matching from different trackers are based on tracks reconstructed from the same hits. This studies was done for the most critical data samples (2004, 2009, 2010). http://www4.rcf.bnl.gov/~fisyak/star/RECO/Eval/TbyT/
- Physics working group analysis results based on samples obtained with different trackers.

2010 AuAu 200, StiCA (•) versus Sti(■)

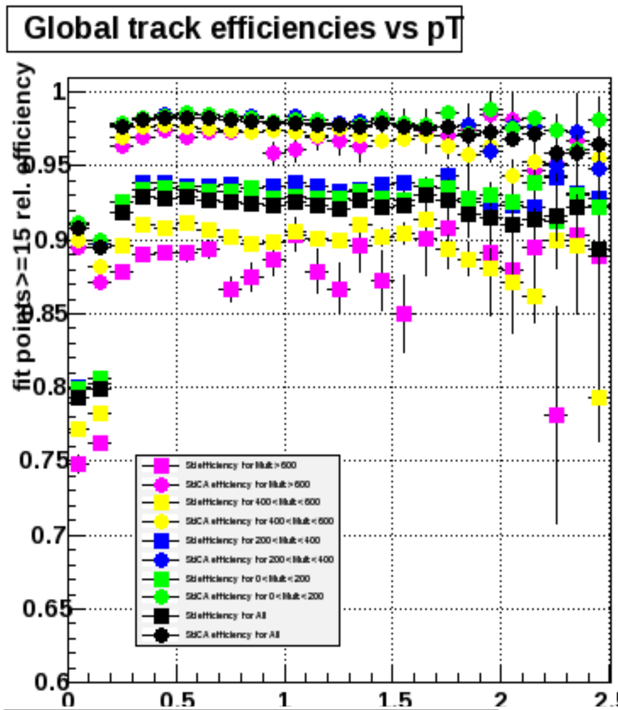
The same statement as before:

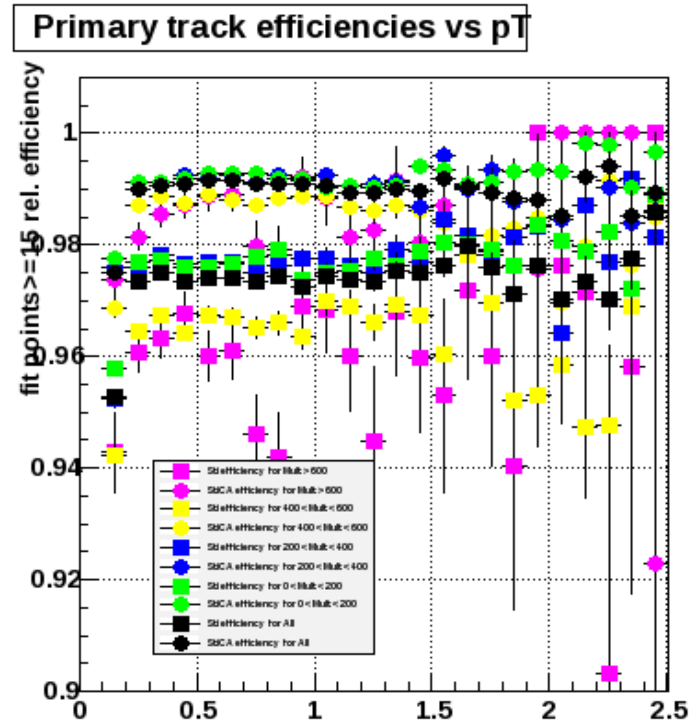
- StiCA efficiency is higher than Sti by ~2%.
- StiCA efficiency has less multiplicity dependence.



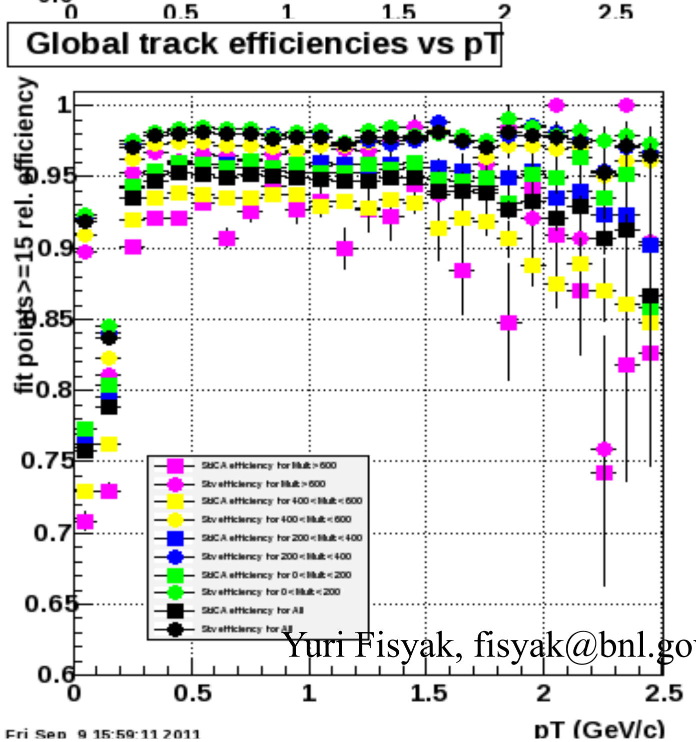
2004
AuAu200

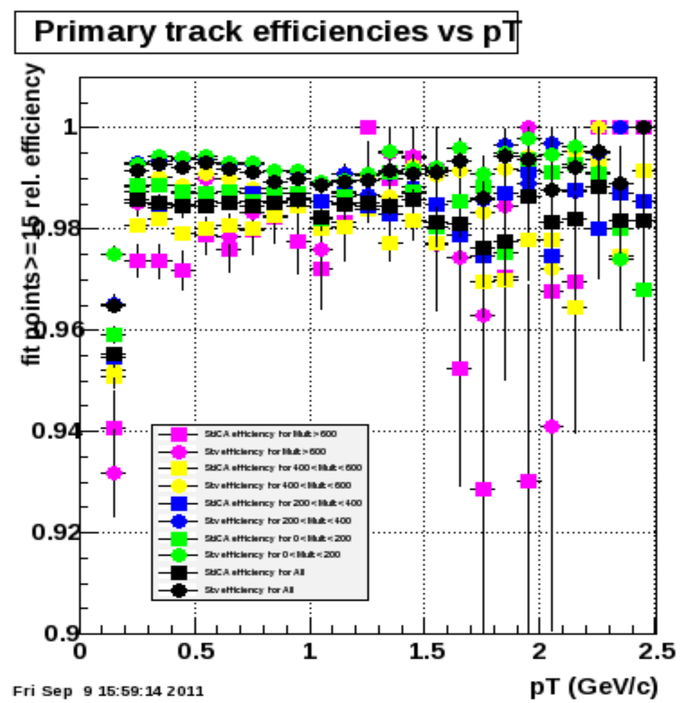
StiCA (•) versus Sti (■)





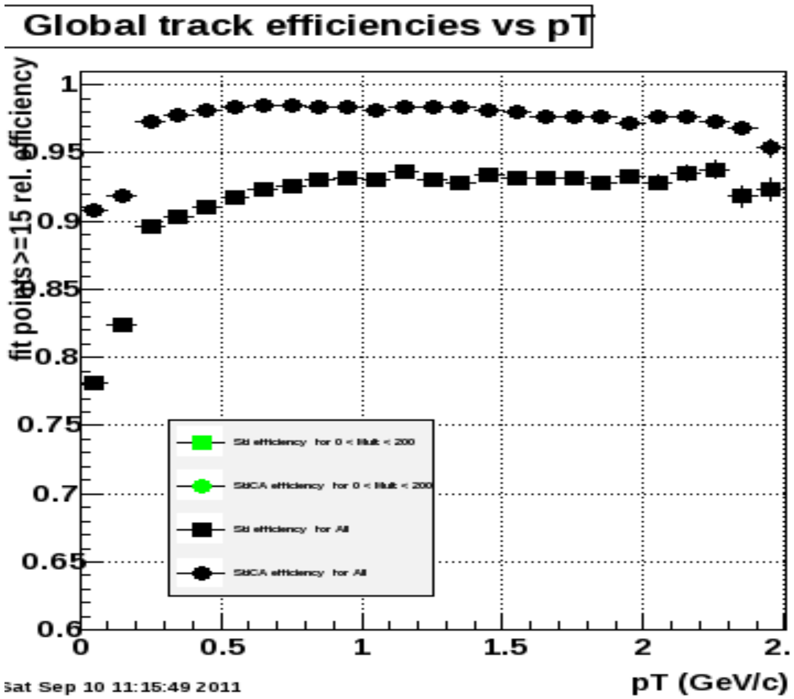
Stv(•) versus StiCA (■)

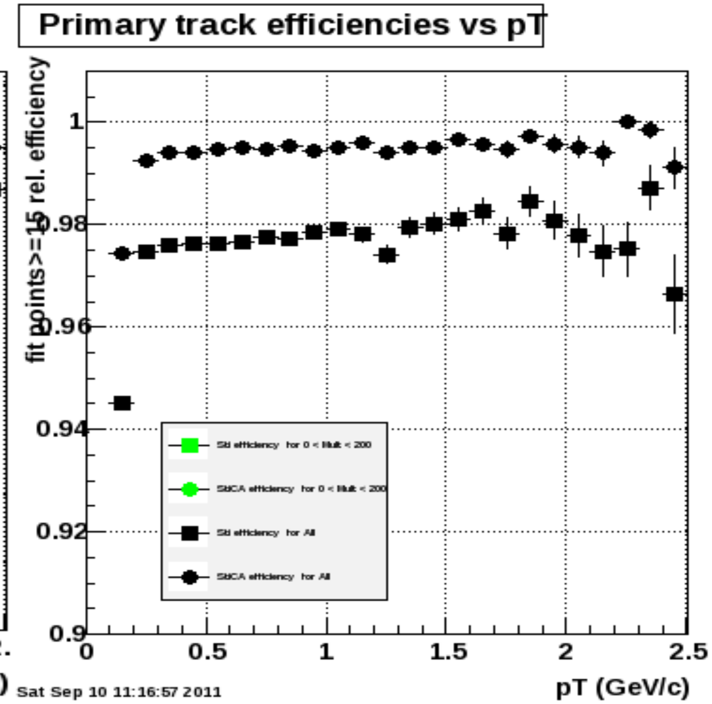




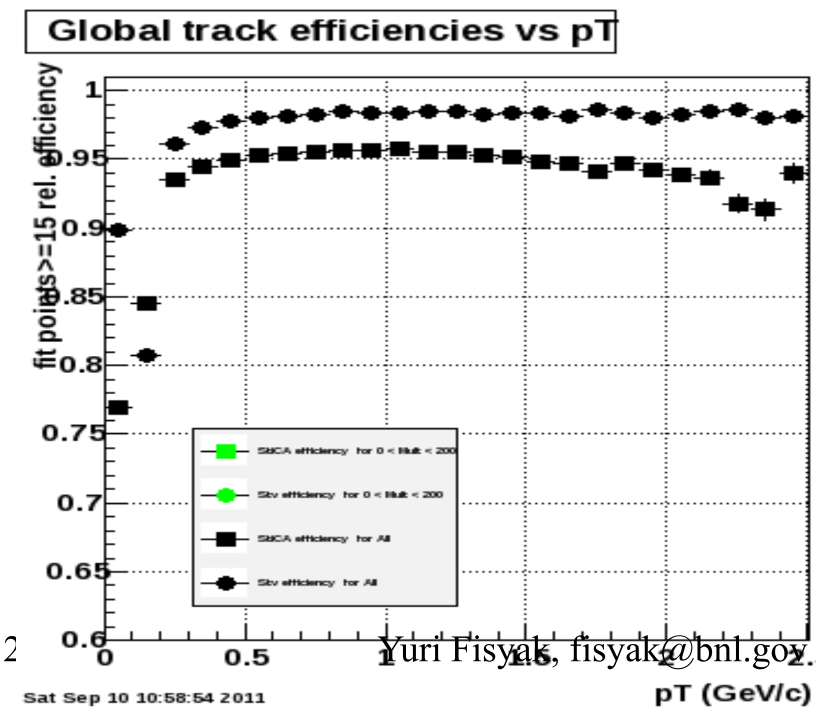
2009, pp500

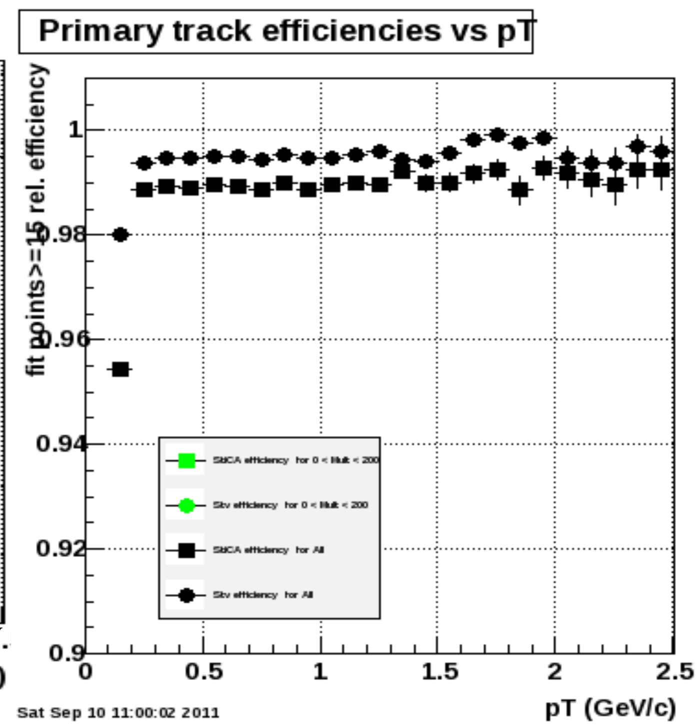
StiCA(•) versus Sti (■)





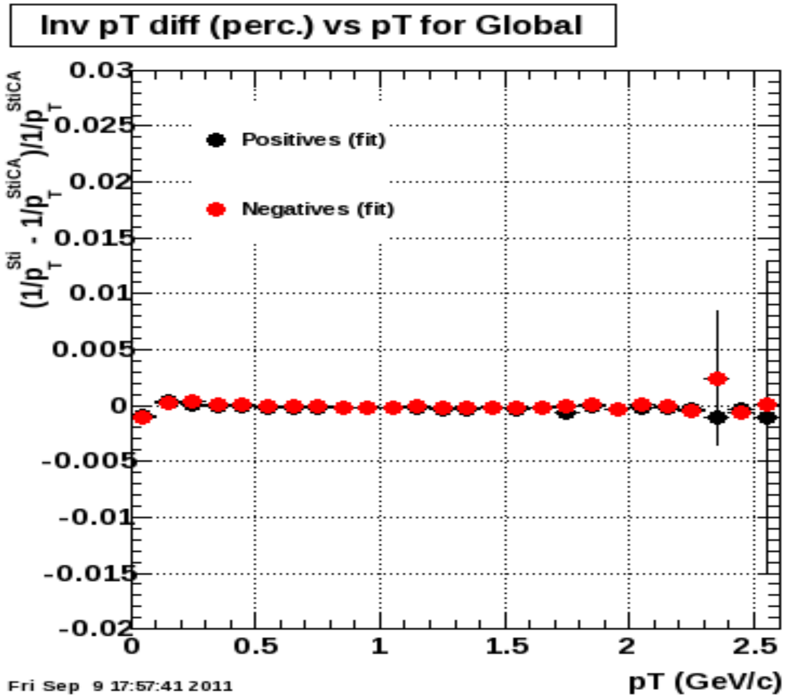
Stv(•) versus StiCA(■)

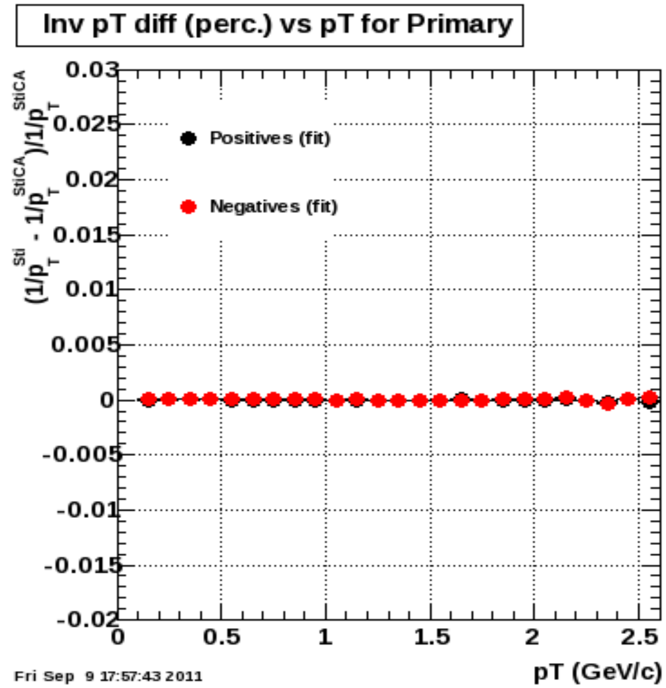




Momentum difference, 2010 AuAu200

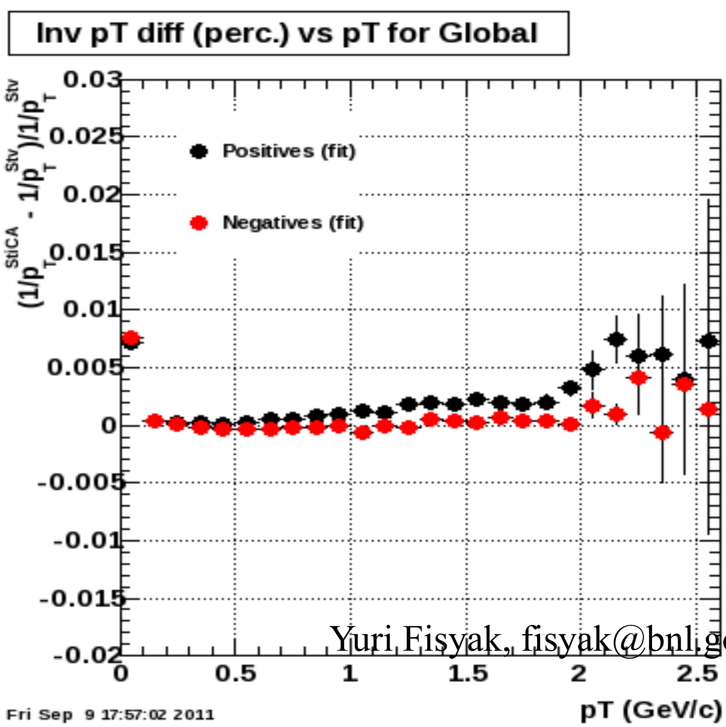
Sti versus StiCA: No difference

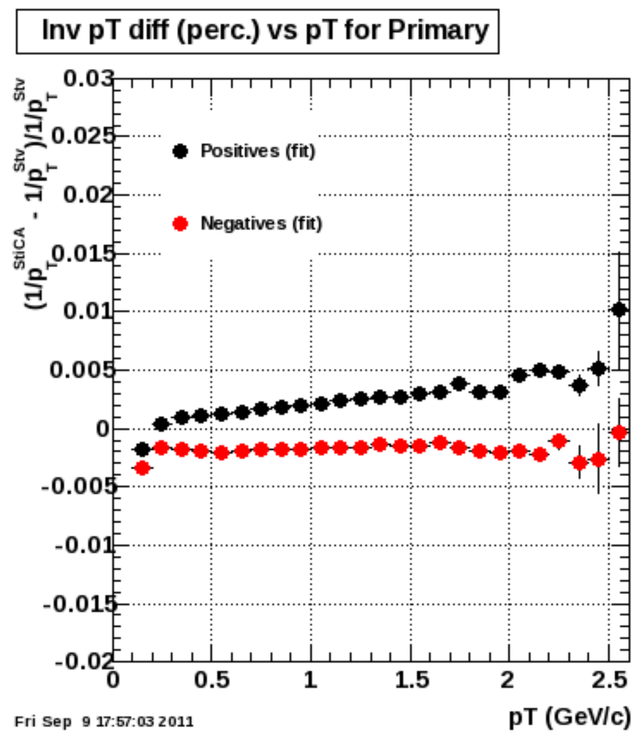




Stv versus StiCA:

- Small difference for global tracks
- For primary tracks sign splitting on the level a few per mills.

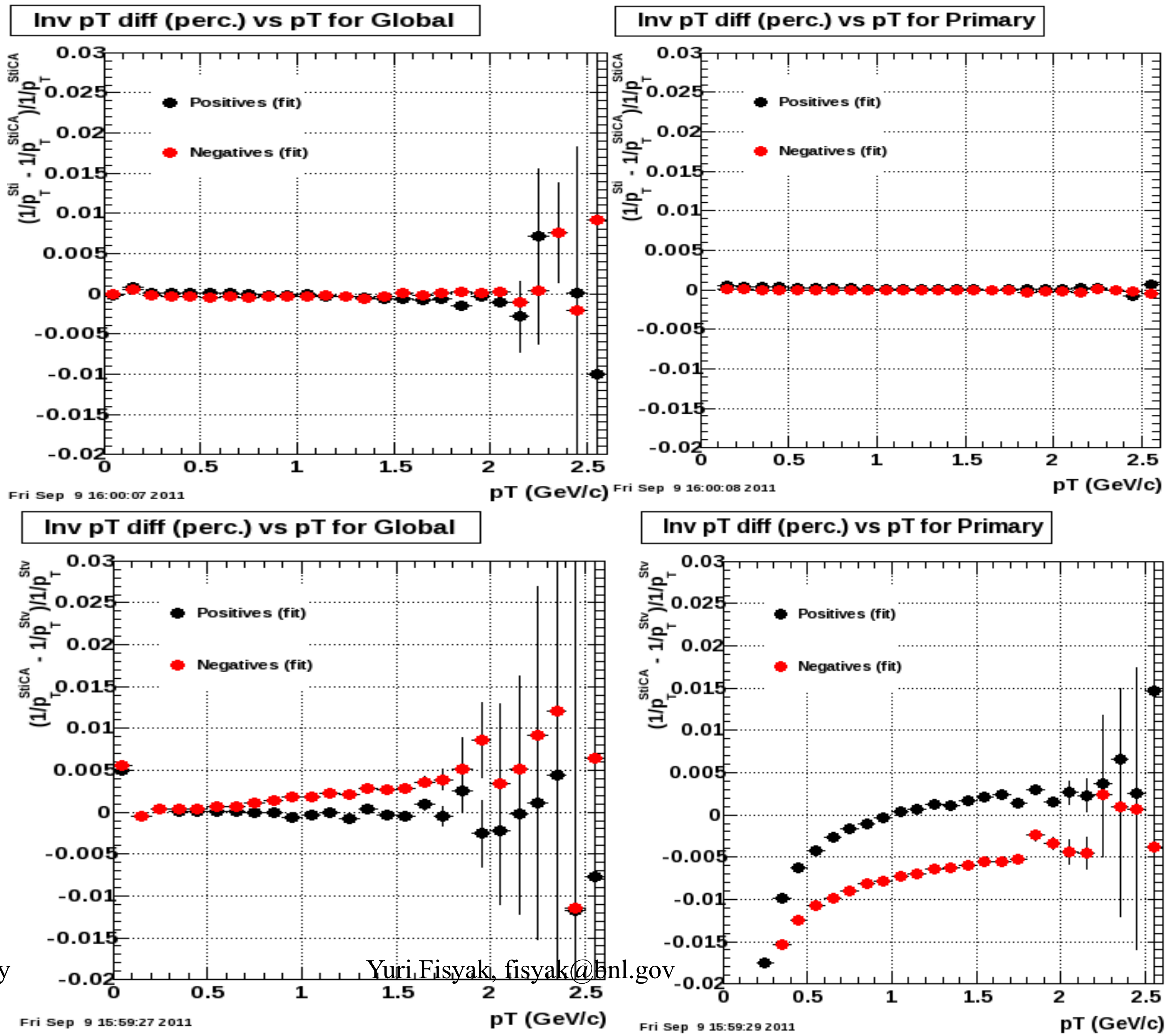




2004, AuAu200

StiCA versus Sti

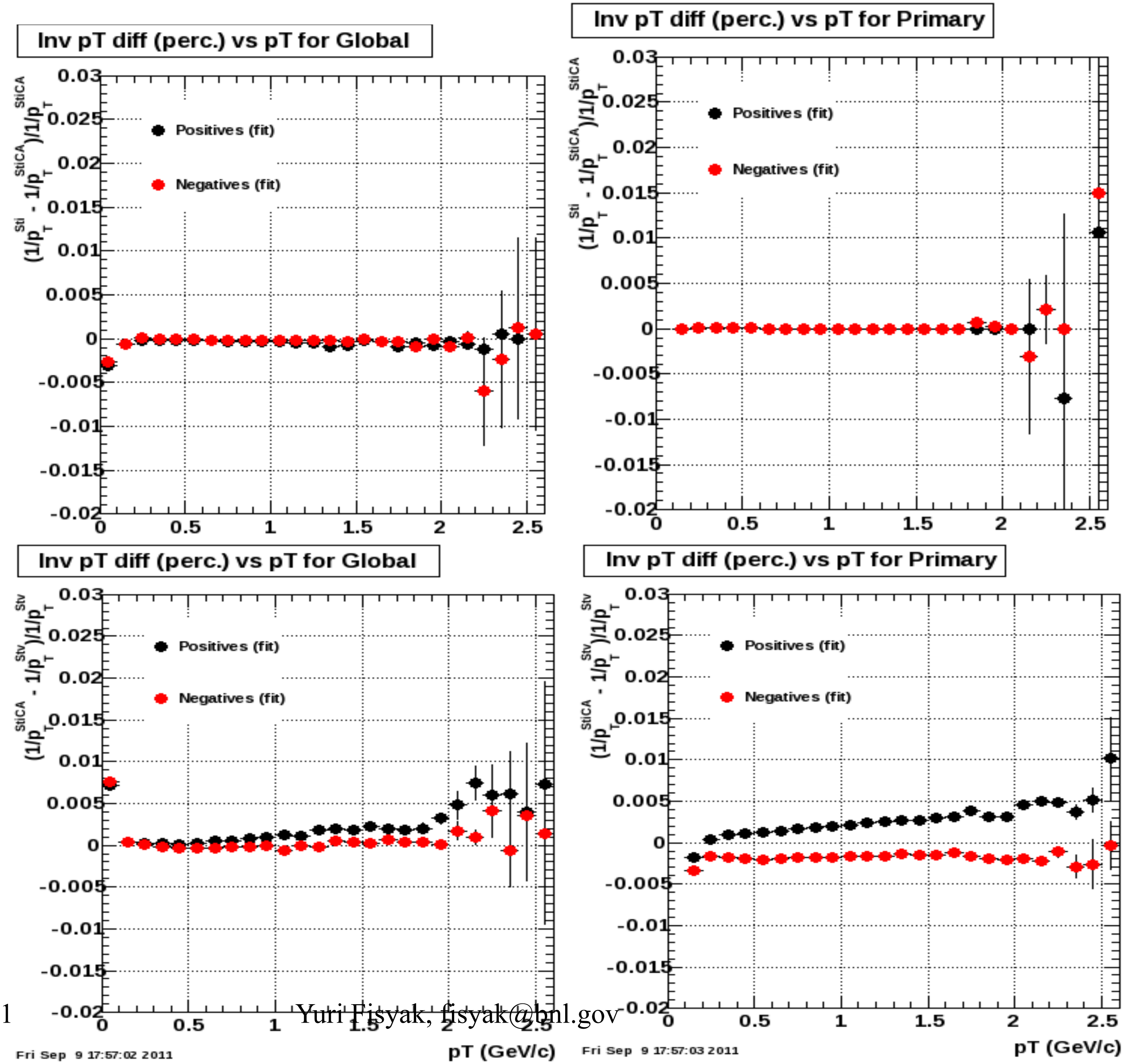
Stv versus StiCA: Sign splitting and shift for primaries on the level ~1%



2010, pp500

StiCA versus Sti:
No differences

Stv versus StiCA:
The same as for 2010 AuAu200



Committee Recommendations: CA track seed finder

"…The Cellular Automaton seed finder used with the Sti-tracker has demonstrated significant improvements in the tracking efficiency (particularly for high multiplicity events) and is found to have a positive effect on the CPU-usage for event reconstruction. For the tracking part of the reconstruction chain, CPU performance has been shown to improve by 10-30%. The average improvement in the track reconstruction efficiency was found to be at the level of 7% and 2% for global and primary tracks, respectively.

These improvements warrant endorsement of CA as the official new seed finder for STAR. Final track-quality cross-checks for specific data selection (detailed below) still need to be completed. Remaining integration and support issues associated with the use of external packages (mentioned later) need to be addressed as well. At the same time, the committee feels that no additional review is required for the cross-checks requested, and should be handled by the S&C Leadership and PAC directly. **The desirable time frame for deployment of the software for data production is early January of 2012 (for any next official data production in the year 2012).**

Together with the STAR version of the CA seed finder, results from recent work on CA optimization by GSI colleagues were presented. **The CA tracking algorithms look extremely promising for other STAR applications including the High Level Trigger and online reconstruction applications**. It appears that it is possible to keep CPU utilization to less than 50 ms per STAR minimum bias 200 GeV Au+Au event. **These achievements are quite remarkable and further developments within the STAR HLT team are strongly encouraged.** However, we advise against importing the newer CA codes into the STAR offline framework and we recommend that future STAR CA seed finder optimization efforts (for use in offline) should be stopped because potential gains from adopting/testing of the newer versions are undermined by the limited manpower available for the offline production…."

Vertex reconstruction (what do we now)

STAR has long standing issue with vertex reconstruction:

- The main reason is simply psychological.
  - STAR is mainly heavy ion experiment and
  - It is quite easy to find primary vertex for central AuAu events where you have hundreds tracks.
  - You take “good” tracks and fit primary vertex.
  - After that you look for secondary vertices which could be strange particle decays.
  - With this approach not whole track information is used.
  - In pp-interactions this approach fails. We lose low multiplicity events and this requires a special treatment for this case.

Vertex reconstruction (why we need upgrade)

- With luminosity for pp500 in Run XI increased by a factor of 2 we have up to 80 pile-up collisions (during TPC sensitivity time, ±40 μsec × 1 MHz) and this means that we have to handle up to 160 primary vertices.
- We would like to have the same treatment for primary and secondary vertices, including decays and conversions, and secondary interactions with detector material.
- To make beam line constrain natural part of the vertex finder i.e. to treat a beam line as an additional track which can be used in the primary vertex finding.
- We would like to improve vertex ranking as belonging to the triggered interaction.

Vertex reconstruction (KFParticle)

KFParticle class (developed by GSI team, see https://www.gsi.de/documents/DOC-2010-Jun-126-1.pdf) which provides:

- Simple construction of the particles from the tracks or other particles.
- Adding measurements to the reconstructed mother particle.
- Simple access to the parameters of the particle.
- Transport of the particles (even in the non uniform magnetic field).
- Impact parameter calculation to the vertex or other particle.
- Adding vertex constrain to track (fit global track with a vertex constrain ➔ primary track fit)
- Kalman techniques is used for vertex finding, fitting and smoothing.
- There is a possibility to add mass constrain to fit decay vertices.

Adaptive vertex fitter with annealing

The most recent reference: “Track and vertex reconstruction: From classical to adaptive methods", Are Strandlie and Rudolf Frühwirt, Rev. Mod. Phys. 82, 1419-1458 (2010).

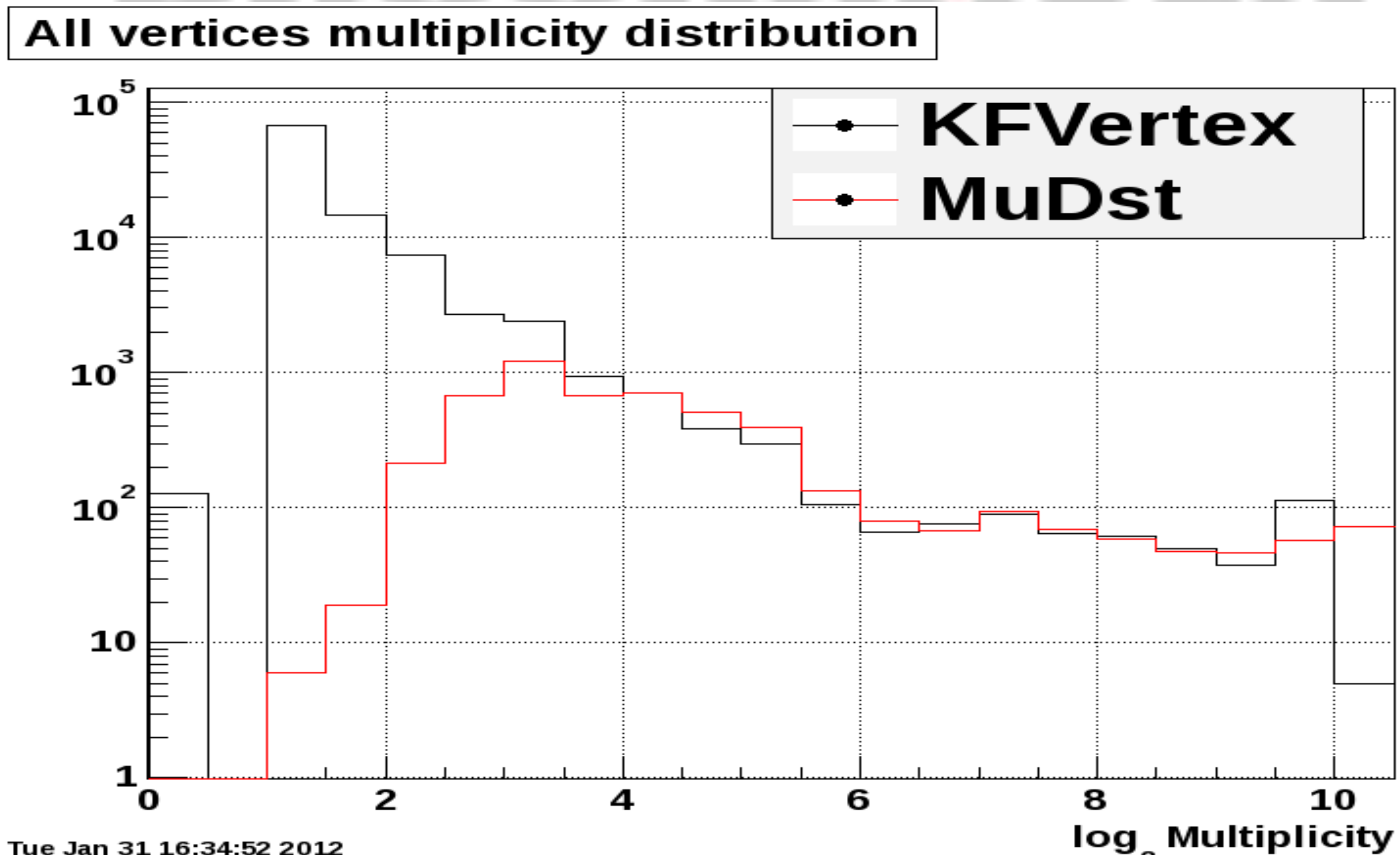
Annealing:

- The main idea is that we don’t really know errors of track parameters.
  - For STAR we fit tracks with π mass hypothesis.
  - For π we can more or less trust track parameters but the track can be generated by K, proton or electrons and for those particles the errors will be underestimated.
- Deterministic Annealing helps to reach the optimal solution.
  - We heat the system i.e. start with high temperature (at $T \gg 1$) by increasing track parameter errors, fit vertex and reiterate fit with decreasing temperature.
  - The final value is $T \rightarrow 1$.

Adaptive fit means that we can allow multiple tracks to be shared between multiple vertices with weight depending on

- $P_{ik} = \exp(-\chi^2_{jk}/2T) / (\exp(-\chi^2_{cut}/2T) + \exp(-\chi^2_{jk}/2T))$,
  - X^2_{jk} – measured distance of track j from vertex k
  - X^2_{cutt} –the cut-off parameters
  - T – the annealing factor (temperature)
- The fit means finding minimum of $X^2_{total} = \Sigma_k\{10^3 - T\bullet\Sigma_j \log(P_{ik})\}$. 10^3 is penalty factor for adding vertex.
- The final step would be unique association tracks to the vertices depending on above weights.

Prototype KFVertexFinder results: Multiplicity for all found vertices



Vertex finder

- Together with GSI group a prototype of Adaptive vertex fitter with annealing based on KFParticle class has been developed.
- The prototype has shown the expected performance.
- The next steps are:
  - To create a StKFVertexFinder as an option for the STAR vertex package
  - Revised and add ranking scheme for StKFVertexFinder.
  - Add kinematical fit for strange particle decay candidates.
  - Ask collaboration for Vertex review.

Problems

The first of all I would like to say that this workshop is very useful to share ideas.

The goal of the workshop is to share codes.

I gave examples of code sharing with GSI group.

But still there are problems with porting codes to the specific framework.

- Naming conventions
- Data formats
- Test suites for different experiments
- …

We did try to address these issues during the first workshop but we still do not see any movement.

Thank you