A detailed wireframe model of a particle accelerator ring is the central background of the slide. The ring is shown in a perspective view, with various sections and components visible. The text is overlaid on the central part of the ring.

# Application of Machine Learning for the IPM-based beam profile reconstruction

M. Sapinski, R. Singh, D. Vilsmeier/GSI

GSI Accelerator Seminar

August 16, 2018

- Ionization Profile Monitors
- Profile distortion, previous approaches
- Virtual-IPM program
- High space charge regime: LHC beams
- Understanding profile distortion
- Correction using electron sieve
- Machine learning and Artificial Neural Networks
- Corrections based on Machine Learning
- IPM for micron-size beams using space charge
- Other examples of ML applications in instrumentation
- Conclusions

# Introduction to IPMs

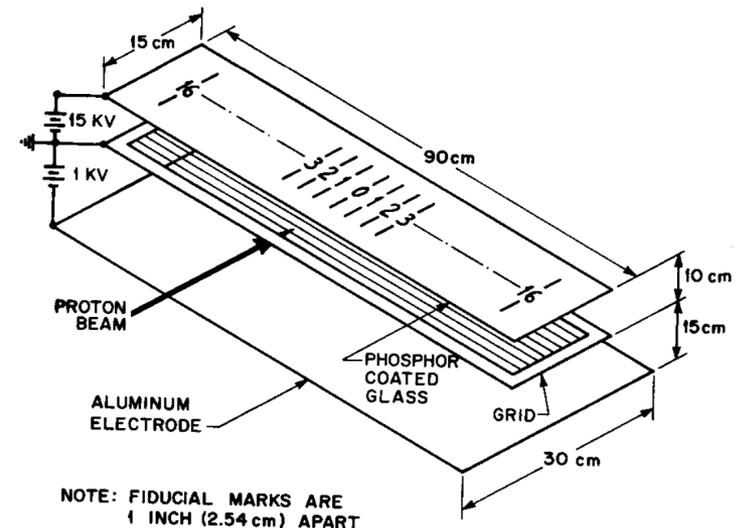
- First constructed in Argonne National Laboratory to measure beam profile on Zero Gradient Synchrotron (ZGS) in 1967 (around the same time in Budker Institute)
- Measures transverse profile of a particle beam.
- Rest gas (pressure  $10^{-8}$  mbar) is ionized by the beam.
- Electric field is used to transport electrons/ions to a detector.
- If electrons are used – additional magnetic field is usually applied to confine their movement.

374

## NONDESTRUCTIVE BEAM PROFILE DETECTION SYSTEMS FOR THE ZERO GRADIENT SYNCHROTRON\*

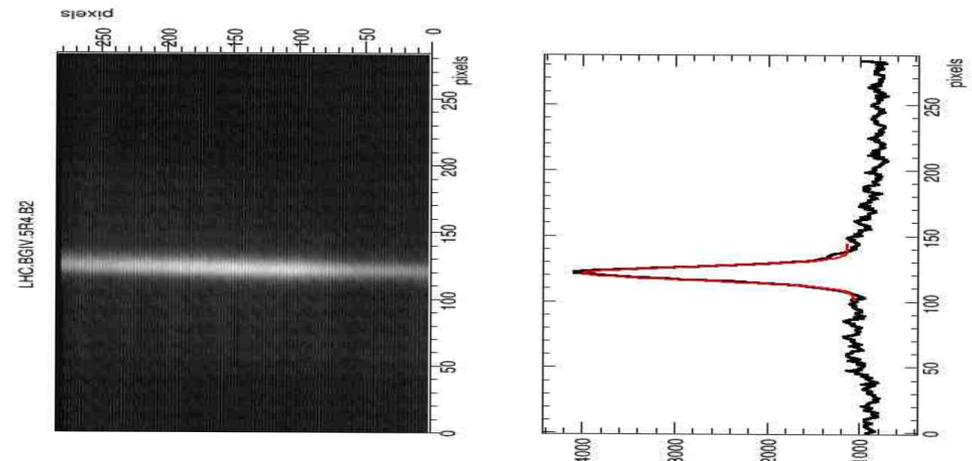
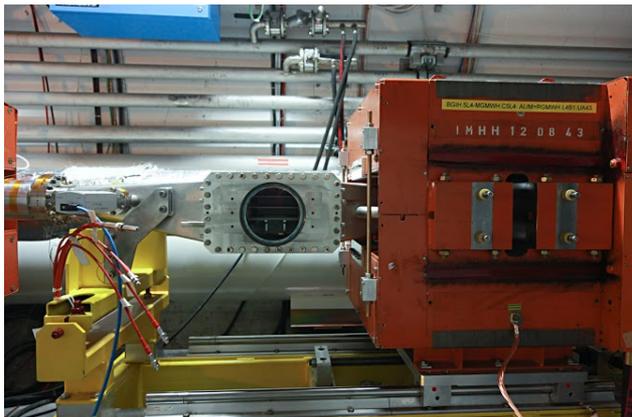
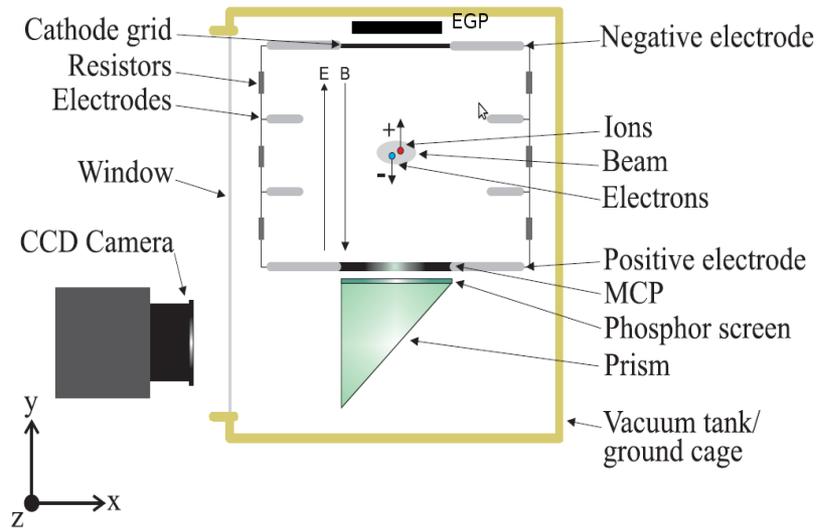
Fred Hornstra, Jr. and William H. DeLuca

Argonne National Laboratory  
Argonne, Illinois



## Ionization Profile Monitor (IPM):

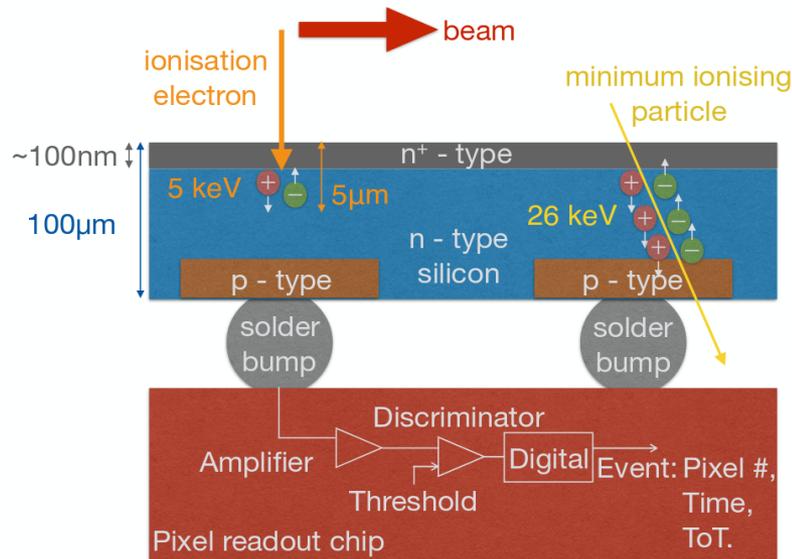
- Measures transverse profile of particle beam.
- Rest gas (pressure  $10^{-8}$  mbar) is ionized by the beam.
- Electric field is used to transport electrons/ions to a detector.
- If electrons are used – additional magnetic field is usually applied to confine their movement.



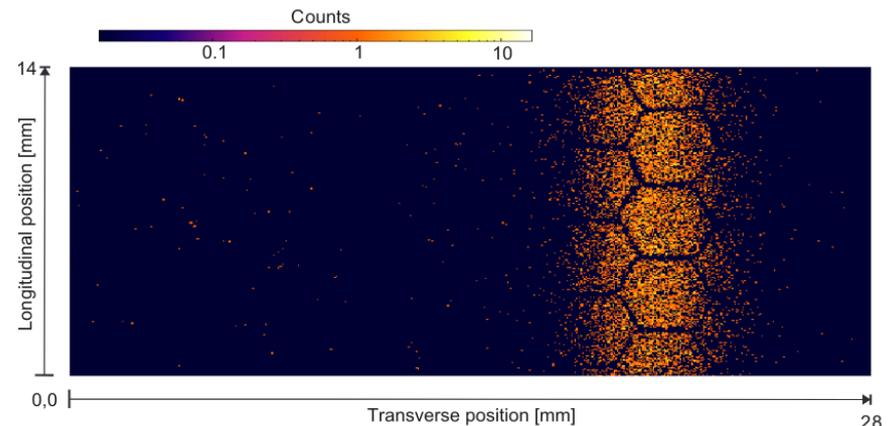
# Remark: New IPM detector technology

- Hybrid silicon pixel detector  
(in this case Timepix3)
- Relatively inexpensive
- Pixels  $55 \times 55 \mu\text{m}^2$
- Single chip  $256 \times 256$  pixels

- Electron arrival time resolution: **1.56 ns**
- Continuous measurement
- No capricious MCP
- Prototype working well on CERN PS

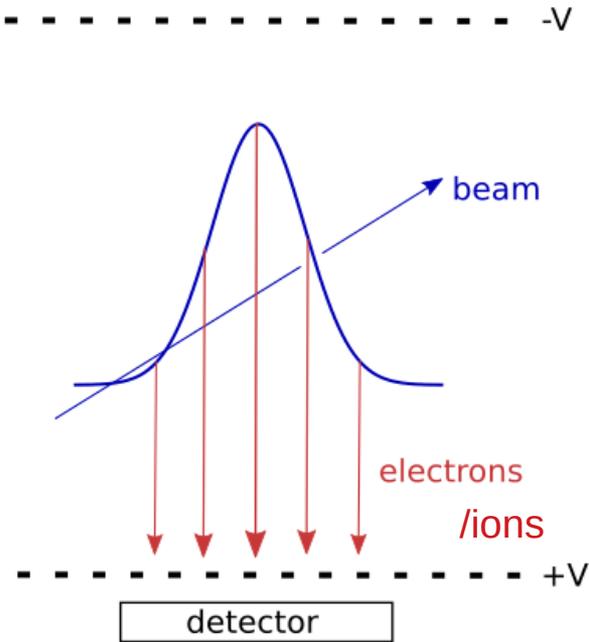


J. Storey et al., Proc. IBIC 2017(WEPC07)  
S. Levasseur et al., Proc of IPAC 2018(WEPAL075)



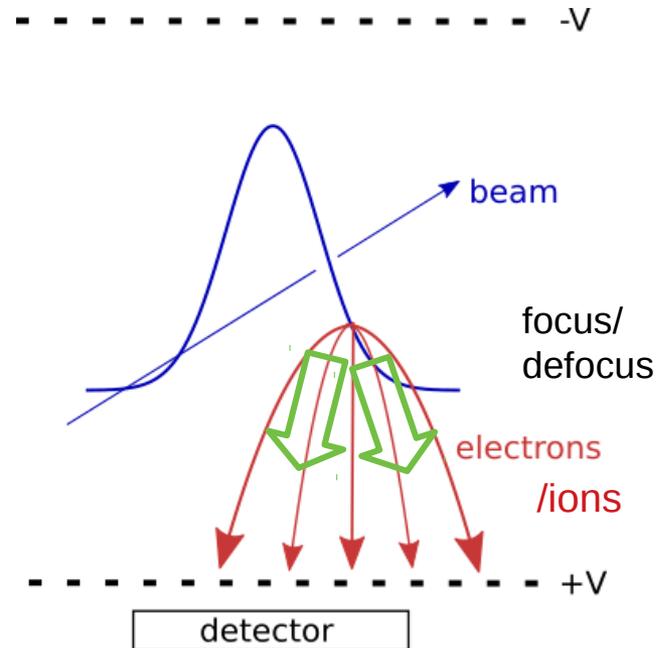
## Ideal case

- Particles are moving on straight lines towards the detector



## Real case

- Particle trajectories are influenced by initial momenta and by the interaction with the beam field

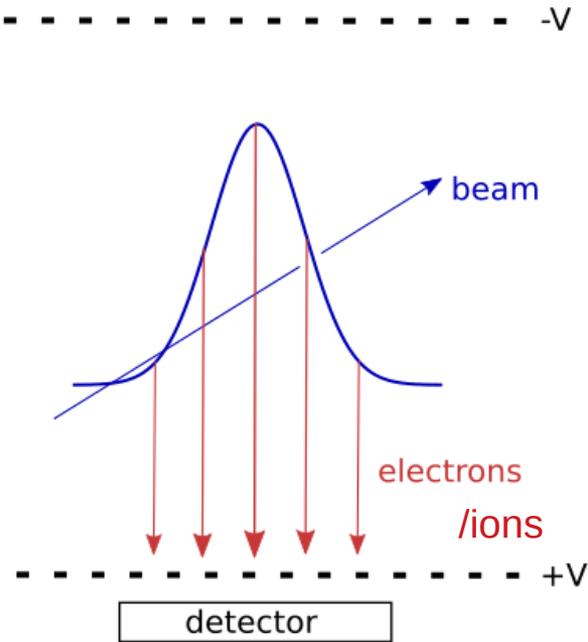


... instrumental effects such as camera tilt, optical point-spread-functions, point-spread functions due to optical system and multi-channel plate granularity etc, etc... come on top!

# Profile distortion in IPM

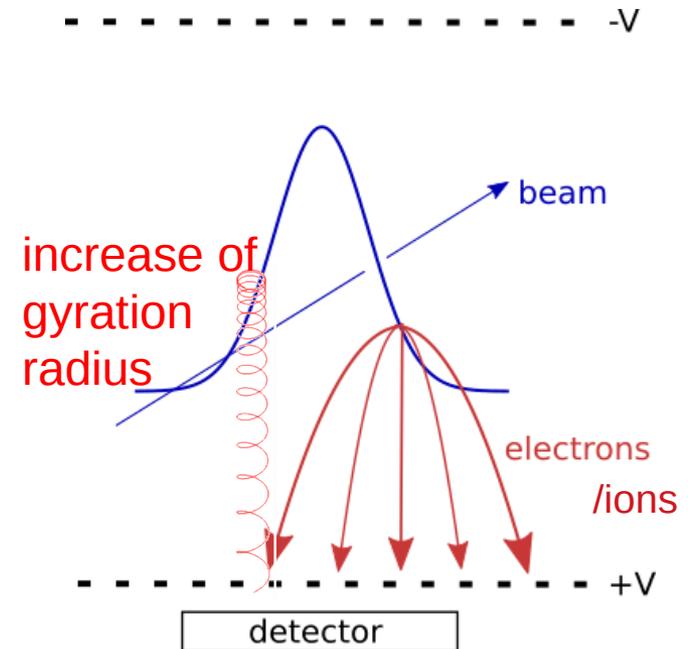
## Ideal case

- Particles are moving on straight lines towards the detector



## Real case

- Particle trajectories are influenced by initial momenta and by the interaction with the beam field



... instrumental effects such as camera tilt, optical point-spread-functions, point-spread functions due to optical system and multi-channel plate granularity etc, etc... come on top!

# Profile distortion without magnetic field (I)

- Effect already investigated in [W. DeLuca, IEEE 1969] (also observation of focusing when collecting electrons!)
- R.E.Thern, "Space-Charge Distortion in the Brookhaven Ionization Profile Monitor ", PAC 1987

$$\sigma_m = \sigma + 0.302 \frac{N^{1.065}}{\sigma^{2.065}} (1 + 3.6 R^{1.54})^{-0.435}$$

$N$  = beam current in  $10^{12}$  protons  
 $\sigma$  = root-mean-square beam size in mm.  
 $R$  = aspect ratio, (other plane)/(measured plane)

- simulations versus measurements
- quite good agreement for nominal extraction voltages
- doubts about accuracy of the correction due to disagreement for low extraction voltages

- W. Graves, "Measurement of Transverse Emittance in the Fermilab Booster", PhD 1994,
  - simulations with TOSCA2D
  - proposed correction:

$$\sigma_{beam} = C_1 + C_2 \sigma_{measured} + C_3 N.$$

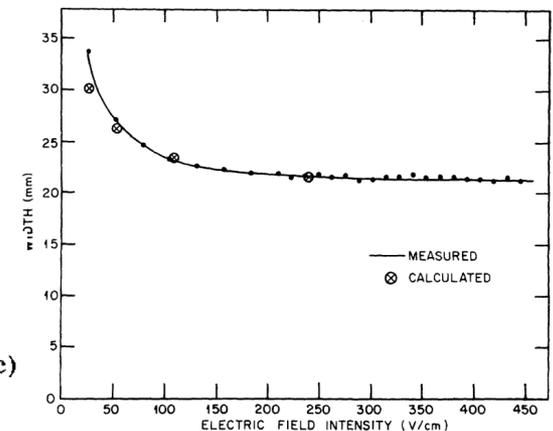


Fig. 4 Comparison of calculated and measured values of ion scatter as a function of electrode field strength.

# Profile distortion without magnetic field (II)

- J. Amundson, J. Lackey, P. Spentzouris, G. Jungman, L. Spentzouris  
"Calibration of the Fermilab Booster ionization profile monitor", PRSTAB 2003

- theoretical investigations (1<sup>st</sup> order):  $\langle |y_{\text{out}}| \rangle = \langle |y_{\text{in}}| \rangle + KN\sigma_{\text{real}}^{-1/2}$ ,

- 2D simulations using OCTAVE

- 1<sup>st</sup> and 2<sup>nd</sup> order corrections investigated:  $\sigma_{\text{measured}} = \sigma_{\text{real}} + C_1 N \sigma_{\text{real}}^{p_1} + C_2 N^2 \sigma_{\text{real}}^{p_2}$ .

- What if the beam is non-gaussian?

J. Egberts, "IFMIF-LIPAc Beam Diagnostics: Profiling and Loss Monitoring Systems", PhD 2012

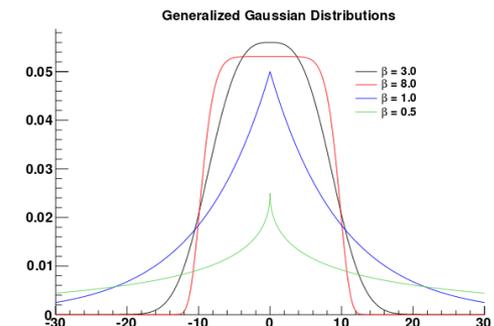
- generalized gaussian distribution

- correction via matrix multiplication:

$$P_{\text{corrected}, l} = \sum_j (A_{ij} P_{\text{measured}, j}), \text{ where } A=A(N, \sigma_{\text{measured}}, \kappa\text{-kurtosis})$$

- if solution not found immediately – iterative, convergent process.

$$P(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-\left(\frac{|x-\mu|}{\alpha}\right)^\beta}$$



# IPM simulation programs

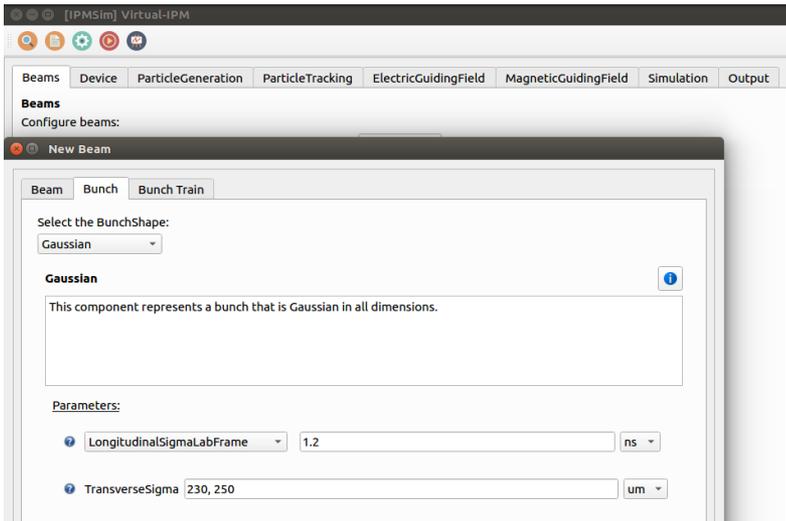
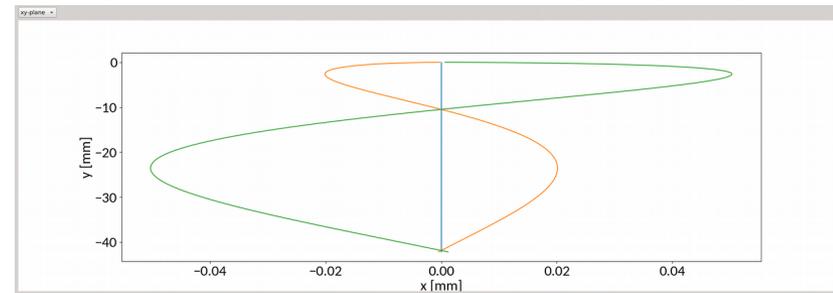
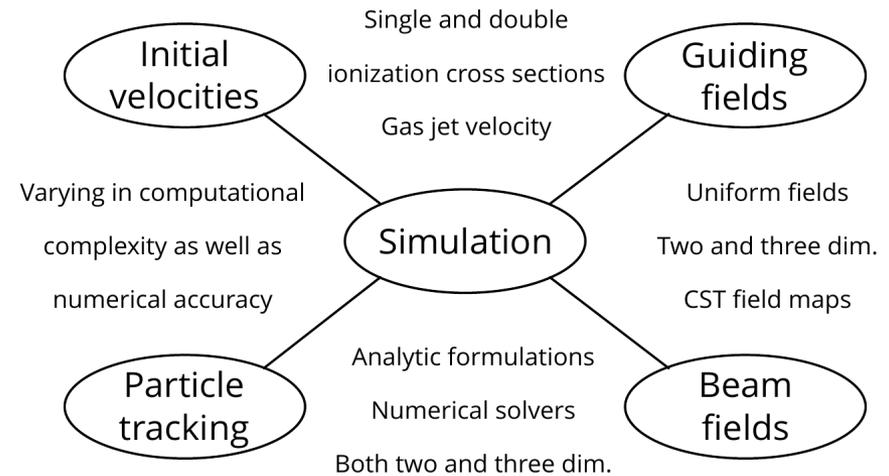
- Since ~2012 – looking for proper IPM simulation codes
- Available programs (CST, Geant4) – missing features
- Many ‘private’ codes exists.
- Workshops on IPM simulations: CERN 2016, GSI 2017, J-PARC 2018 (Sep)

Name/Lab	Language	Ionization	Guiding field	shape	Beam field	Tracking
GSI code	C++	simple DDCS	uniform E,B	parabolic 3D	3D analytic relativ.	numeric R-K 4 <sup>th</sup> order
PyECLOUD-BGI /CERN	python	realistic DDCS	uniform E,B	Gauss 3D	2D analytic relativ. only	analytic
FNAL	MATLAB	simple SDCS	3D map E,B	arbitrary	3D numeric relativ. (E and B)	num. MATLAB rel. eq. of motion
ISIS	C++	at rest	CST map E only	arbitrary (CST)	2D numeric (CST) non-relativ.	numeric Euler 2 <sup>nd</sup> order
IFMIF	C++	at rest	Lorenz-3E map E only	General. Gauss	numeric (Lorenz-3E) non-relativ.	
ESS	MATLAB	at rest	uniform E,B	Gauss 3D	3D numeric (MATLAB) relativ.	numeric MATLAB R-K
IPMSim3D /J-PARC	python	realistic DDCS	2D/3Dmap E, B	Gauss 3D	2D numeric (SOR) relativ. only	numeric R-K 4 <sup>th</sup> order

M. Sapinski et al, *Ionization Profile Monitor Simulations - Status and Future Plans*, Proc. of IBIC 2016, (TUPG71)

# Virtual-IPM program

- After looking for a proper program we decided to write our own called Virtual-IPM
- Written in Python with modern, modular architecture
- GUI in Qt
- Covers: IPM, BIF, gas jets
- Publicly available as python module:  
<https://pypi.org/project/virtual-ipm>
- pip install virtual-ipm
- Code on gitlab:  
<https://gitlab.com/IPMsim/Virtual-IPM>



D. Vilsmeier, et al., *A Modular Application for IPM Simulations*, Proc. of IBIC17 (WEPC07) and presentation at 1<sup>st</sup> ARIES Annual Meeting, Riga, May 2018

# High space-charge regime: LHC beams

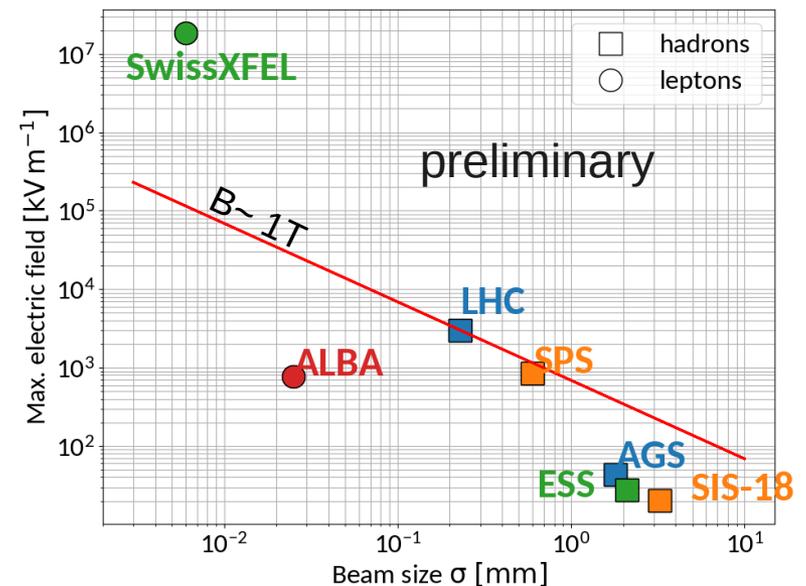
- Let's fix beam parameters the impact of various phenomena
  - for this beam maximum E-field is  $\sim$ MV/m
  - we do simulate beam B-field, but its impact is small because electrons/ions move slowly

- IPM corresponding to LHC/SPS IPMs but with Timepix3 detector resolution:

- distance between electrodes: 84 mm
- U=4 kV (E=48 kV/m)
- B=0.2 T
- Spatial resolution (binning): 55  $\mu$ m

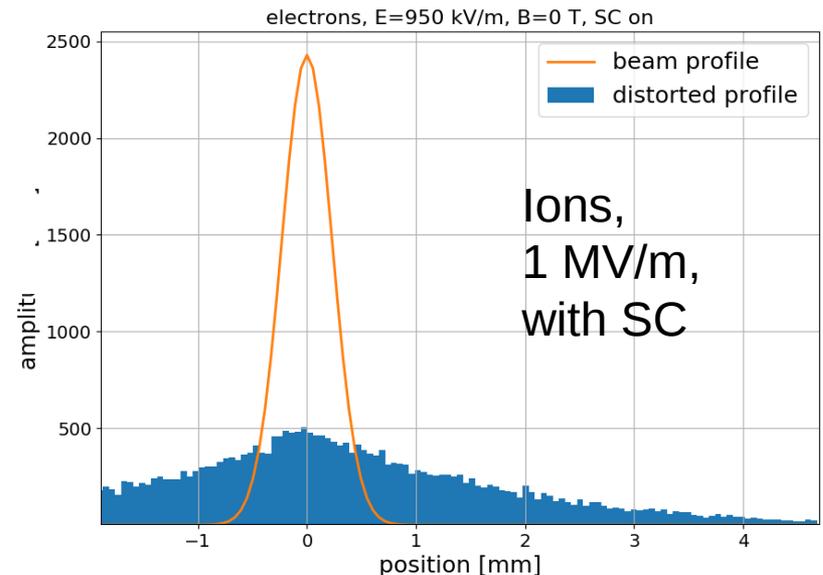
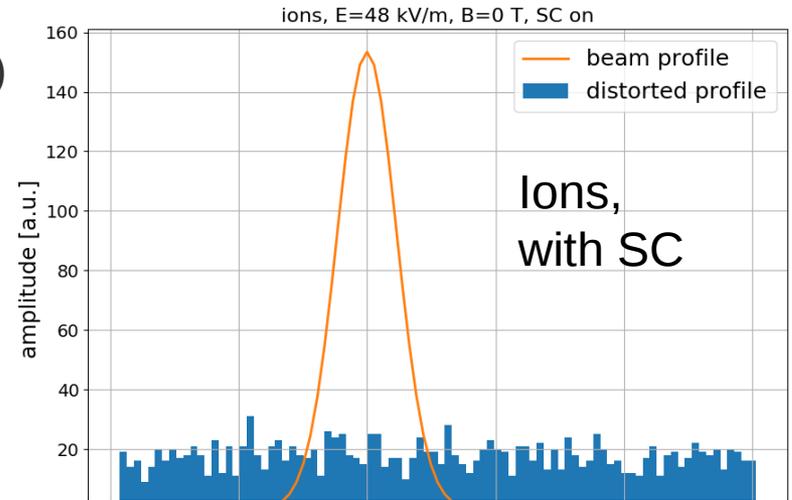
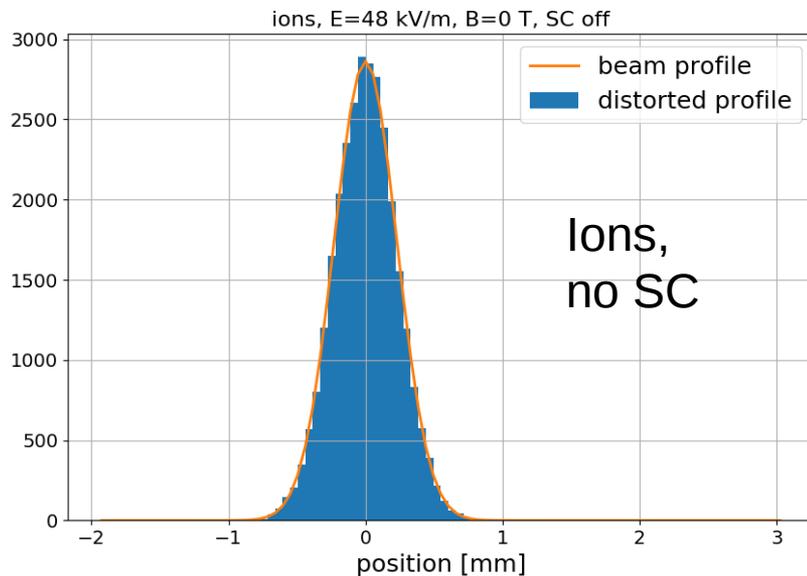
Here “space charge” refers to bunch field impact on electrons – it scales with relativistic gamma.

$\sigma_x$	230 $\mu$ m
$\sigma_y$	270 $\mu$ m
$N_{\text{prot}}$	$1.4 \cdot 10^{11}$
$4\sigma_z$	1.1 ns
$E_{\text{beam}}$	6.5 TeV



# Understanding distortion: ions

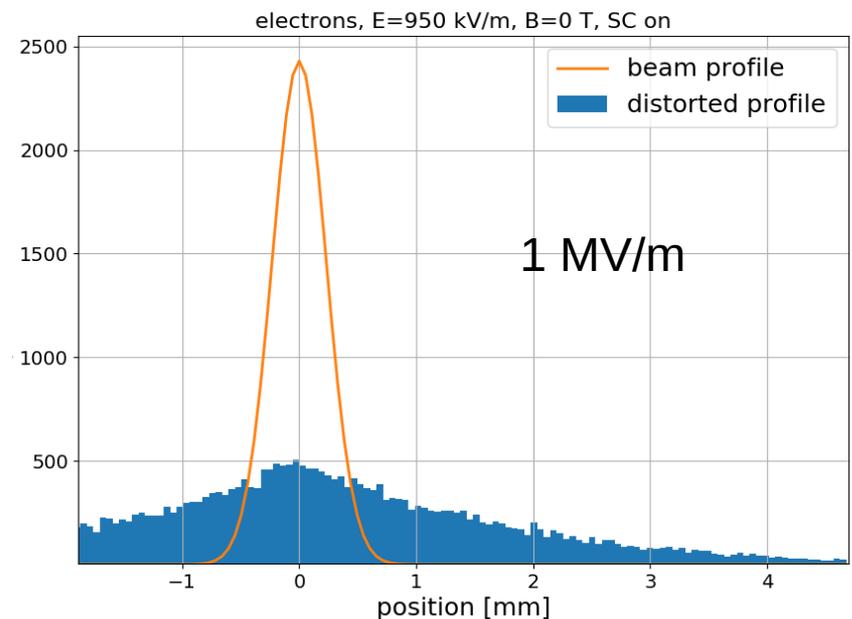
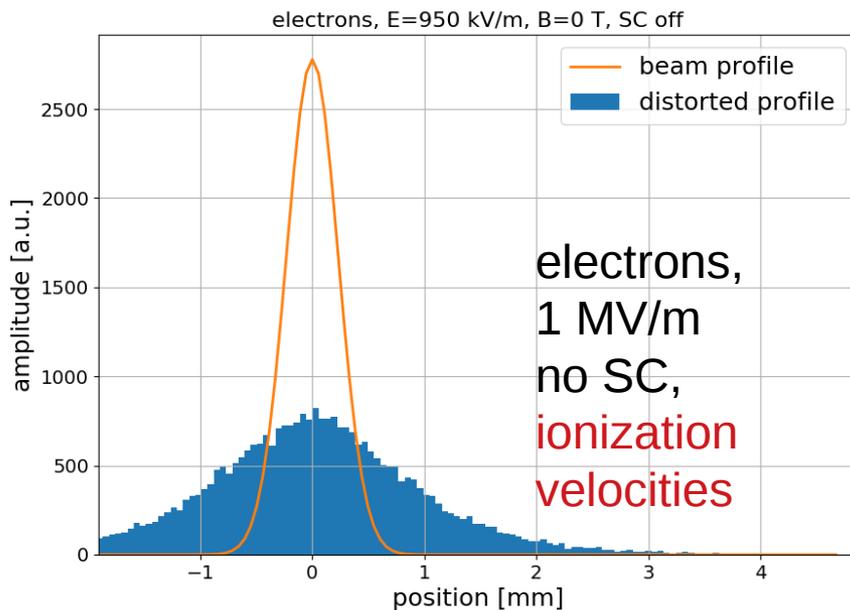
- The simplest device (no magnetic field)
- Tracked:  $H_2^+$  ions



Conclusion: ion-based device  
too sensitive to beam space  
charge → use electrons!

# Electron tracing – no B-field

- Immediately try higher extraction fields ( $\sim 1$  MV/m)

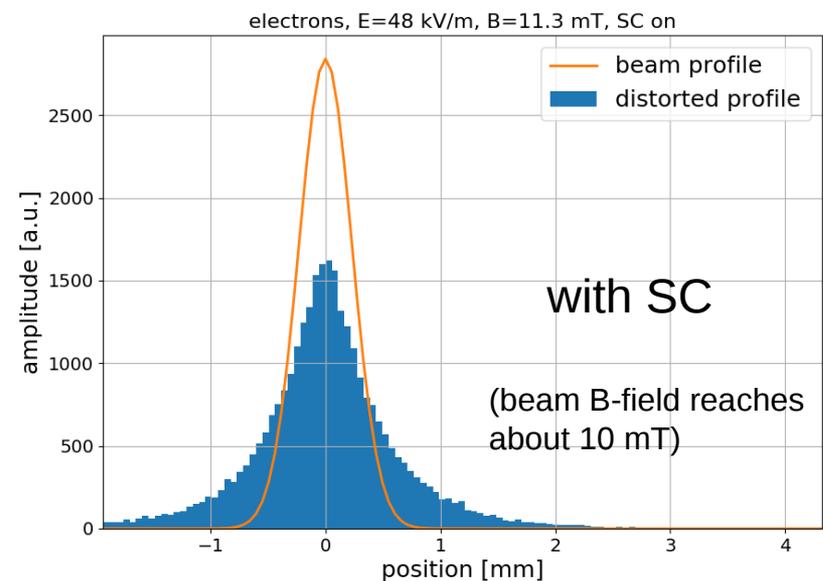
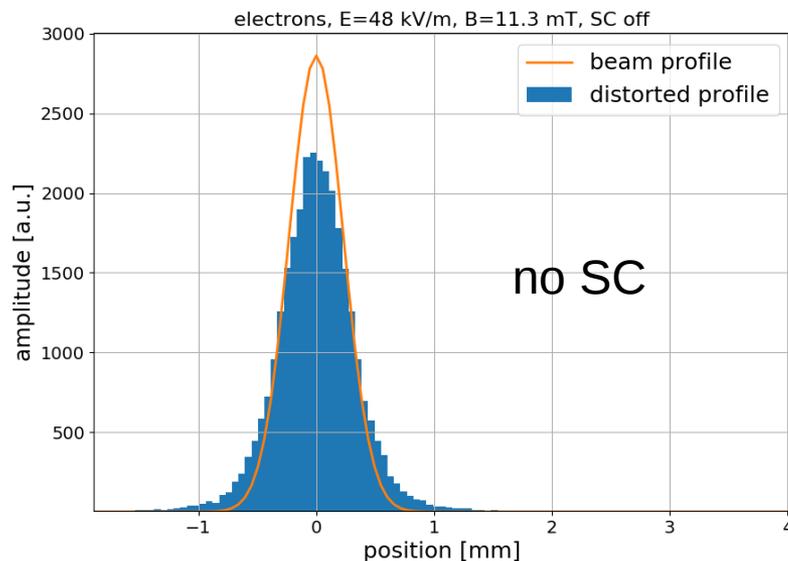


Space charge smears the profile, but even without space charge, the initial electron velocities from ionization broaden the peak enormously.

**Magnetic field is needed when using electrons with sum-mm beams.**

# Electron tracing – *single turn B-field*

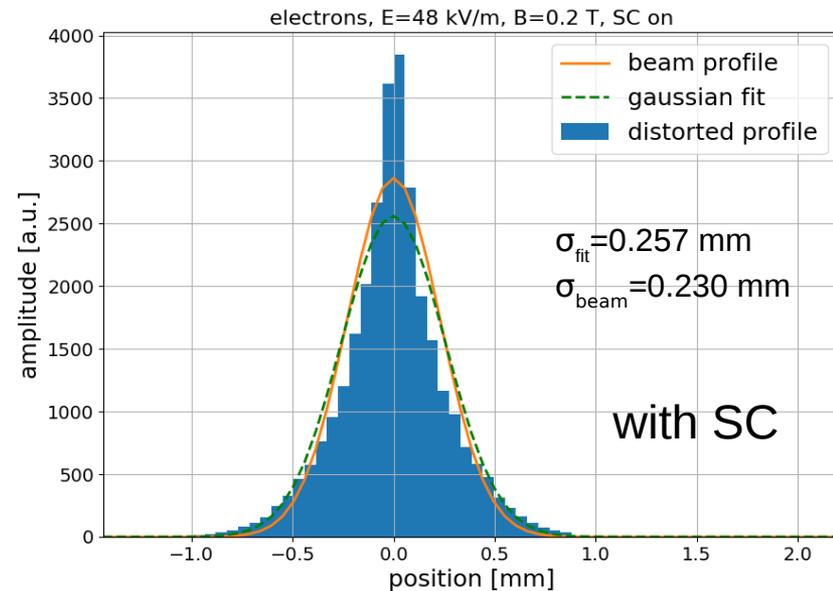
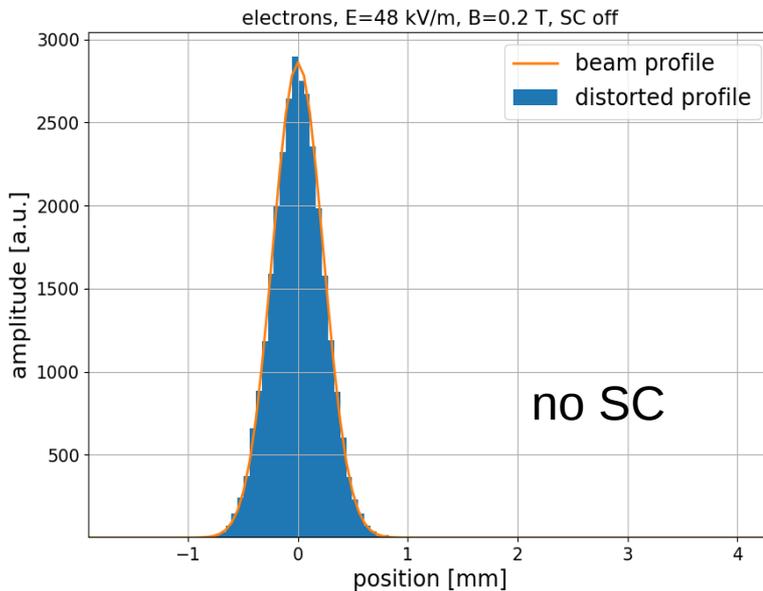
- Magnetic field tuned to single turn:  $B = \pi \sqrt{2 * m_e * E / (e * d)} = 11.3 \text{ mT}$
- Original idea in [F. Hornsta, M. Trump, PLACC 1970]



## Disadvantages of tuning magnetic field to single turn:

- Sensitivity to vertical beam position
- Sensitivity to fringe fields of neighboring magnets (as the field required is weak)
- Spread due to vertical component of electron velocities (and vertical beam size)
- Little help for the space-charge effects – electron trapping/additional kick in vertical direction

# Electron tracing – strong B-field

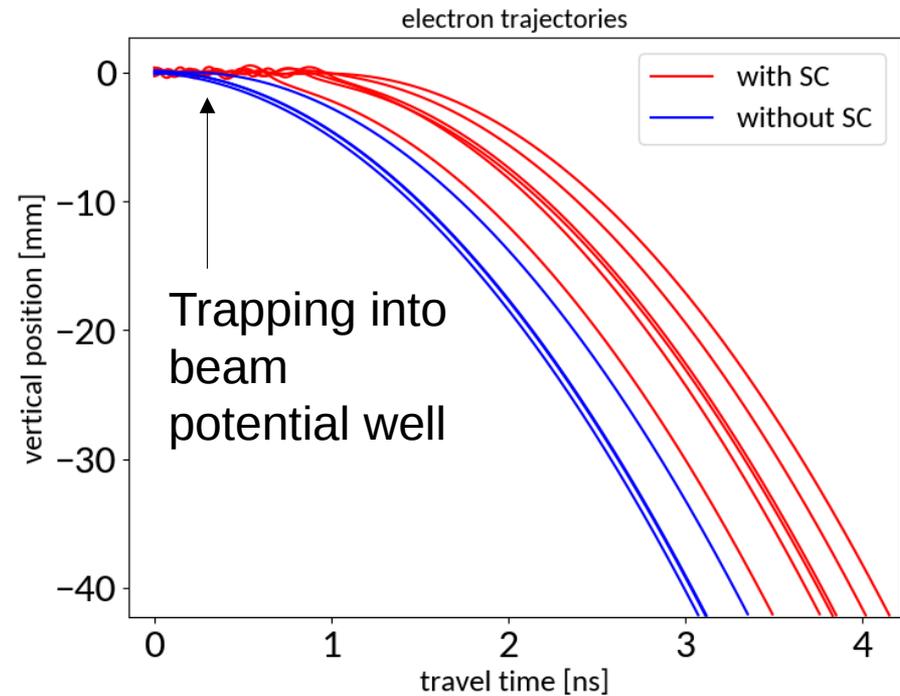
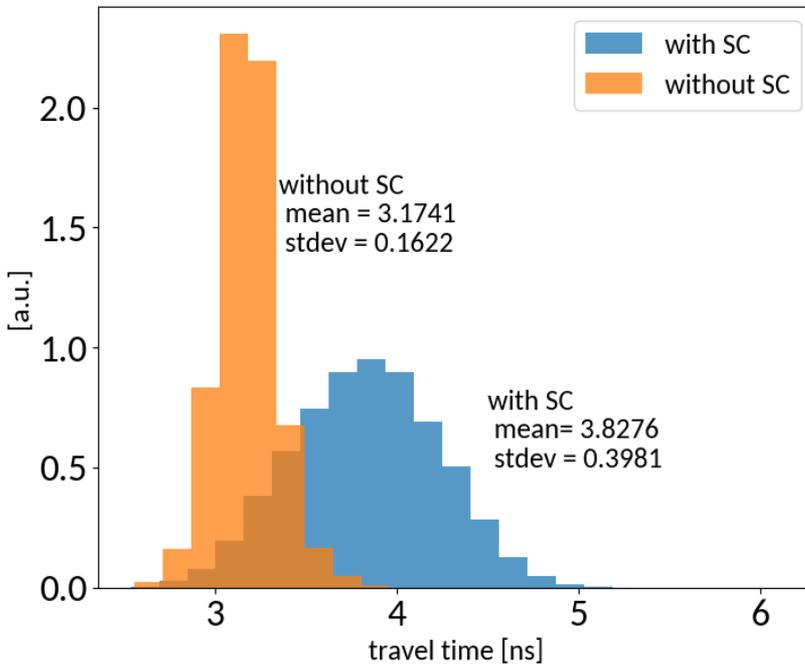


## Conclusions:

- Better use strong B-field rather than “one turn” field
- 0.2 T is not enough to counteract space charge effect for LHC beam.

# Electron tracing – strong B-field

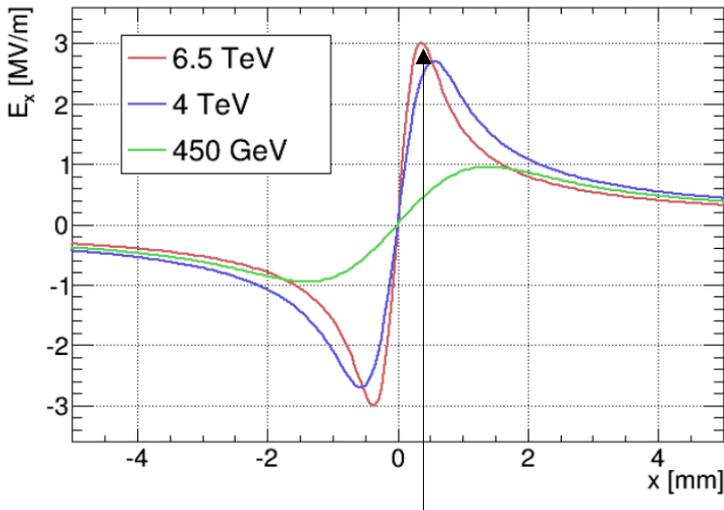
What happens to electrons?



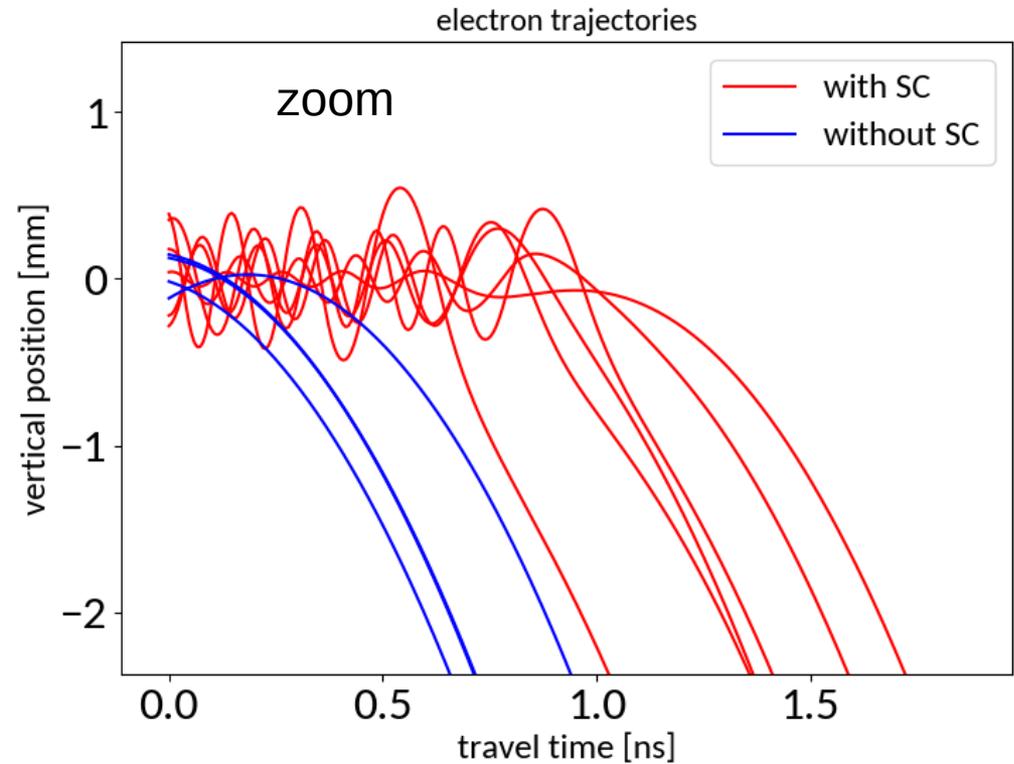
Electrons are trapped in bunch field for the time when bunch passes. They make several oscillations around bunch center. Complex movement!

# Electron tracing – strong B-field

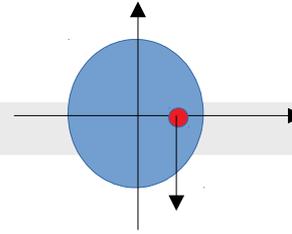
More investigations – what happens to electrons?



Field maximum at  $\frac{1}{2}\pi\sigma_{\text{beam}}$



# Electron tracing – strong B-field



## Why profile deformation is complex?

Electron produced at:

$$x = \frac{1}{2} \pi \sigma_{\text{beam}}, y = 0$$

1.5 ns before the bunch center

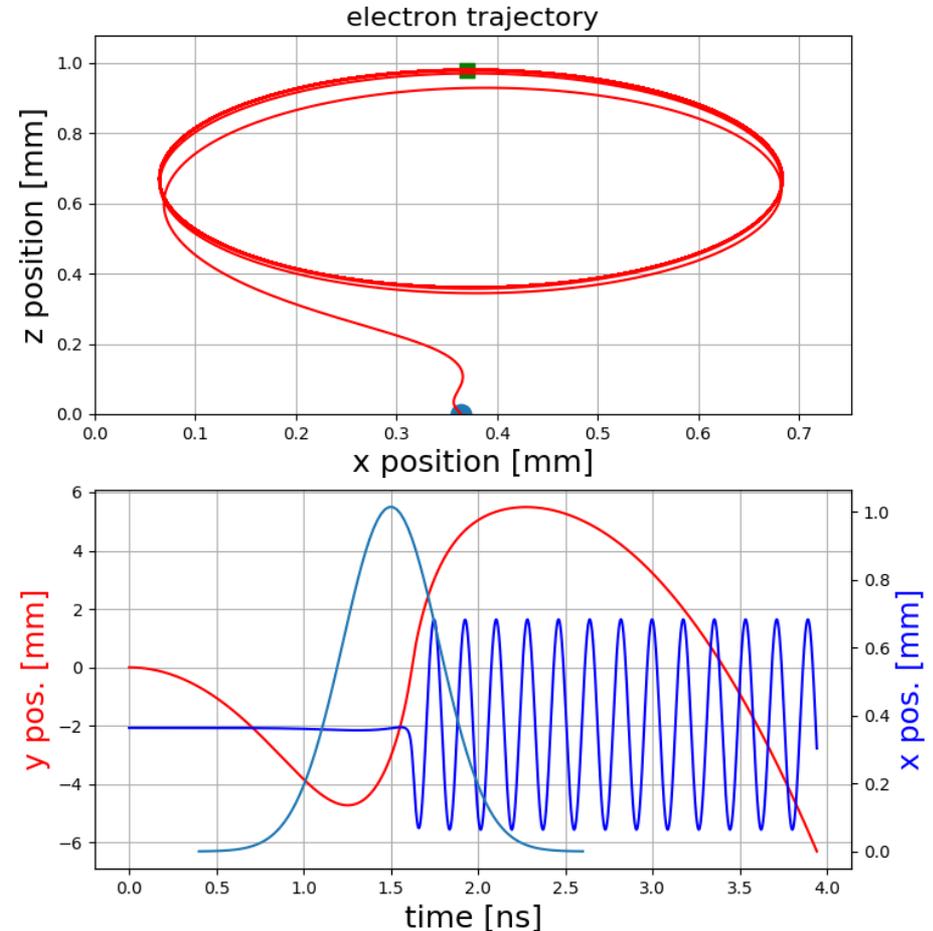
$$v_{\text{ionization}} = 0$$

Electron is moving towards the detector when arriving bunch sucks it back so strongly that it is launched well above the bunch.

Large momentum transfer in transverse direction.

ExB drift pushes it ~1 mm along the beam.

Final gyroradius: 310  $\mu\text{m}$ .



# Electron tracing – strong B-field

## Why profile deformation is complex?

Electron produced at:

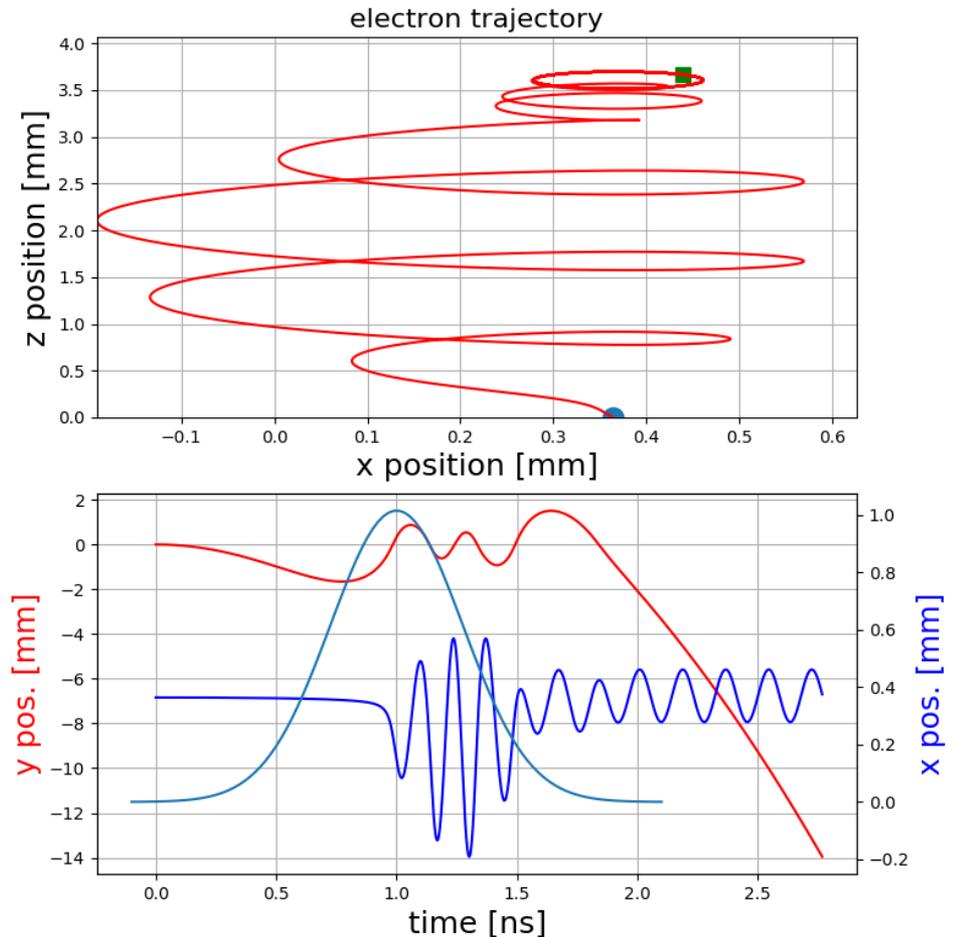
$$x = \frac{1}{2} \pi \sigma_{\text{beam}}, \quad y = 0$$

1.25 ns before the bunch center.

$$v_{\text{ionization}} = 0$$

Electron does not get as much initial kick as in the previous example and is trapped within the bunch (in both: horizontal and vertical directions). Inside the bunch the  $E \times B$  drift and gyrations mix. Maximum transverse momentum is larger than in previous case but final  $p_{\perp}$  is smaller.

Final gyroradius: 92  $\mu\text{m}$ .



# Electron tracing – strong B-field

Why profile deformation is complex?

Electron produced at:

$$x = \frac{1}{2} \pi \sigma_{\text{beam}}, \quad y = 0$$

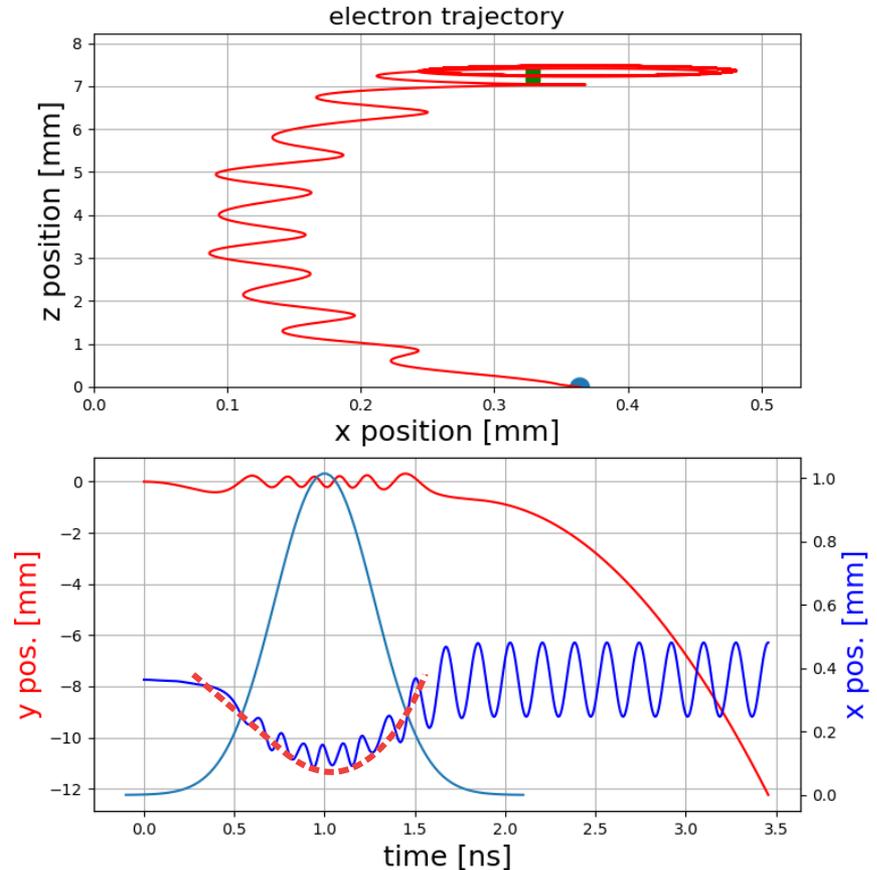
1.0 ns before the bunch center.

$$v_{\text{ionization}} = 0$$

On top of gyration and ExB drift electron experiences (probably) so called polarization drift:

$$v_{\text{polarization}} = \frac{m}{qB^2} \frac{d\mathbf{E}_{\perp}}{dt}$$

Final gyroradius: 120  $\mu\text{m}$ .



# Electron tracing – strong B-field

Why profile deformation is complex?

Electron produced at:

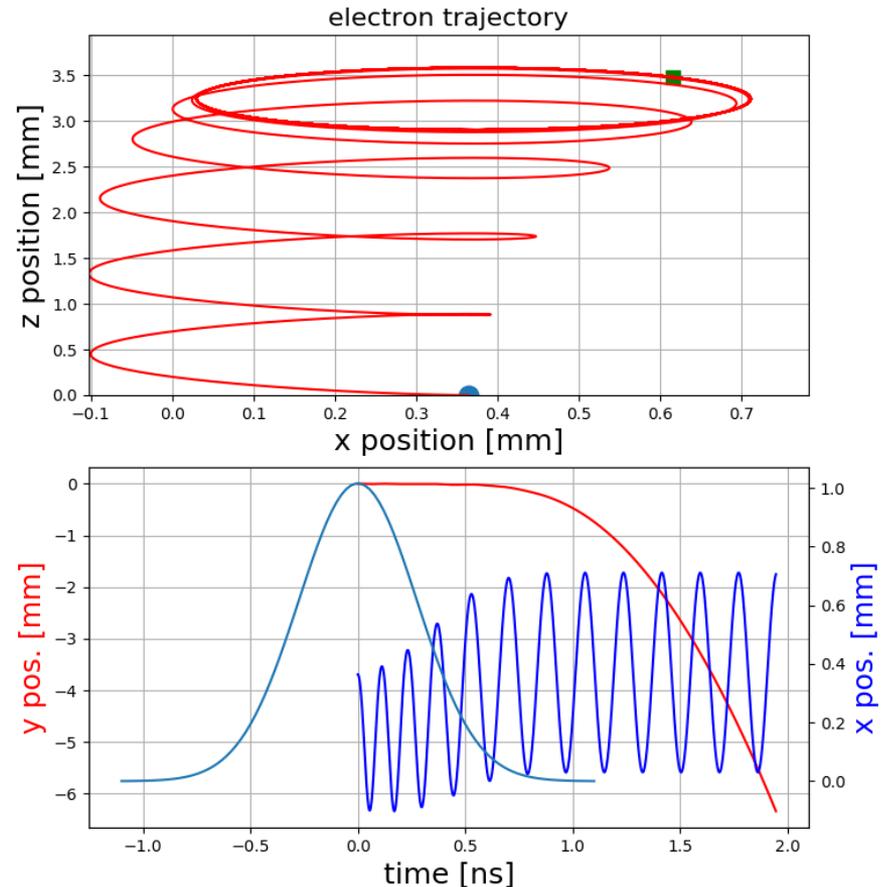
$$x = \frac{1}{2} \pi \sigma_{\text{beam}}, \quad y = 0$$

at the bunch center.

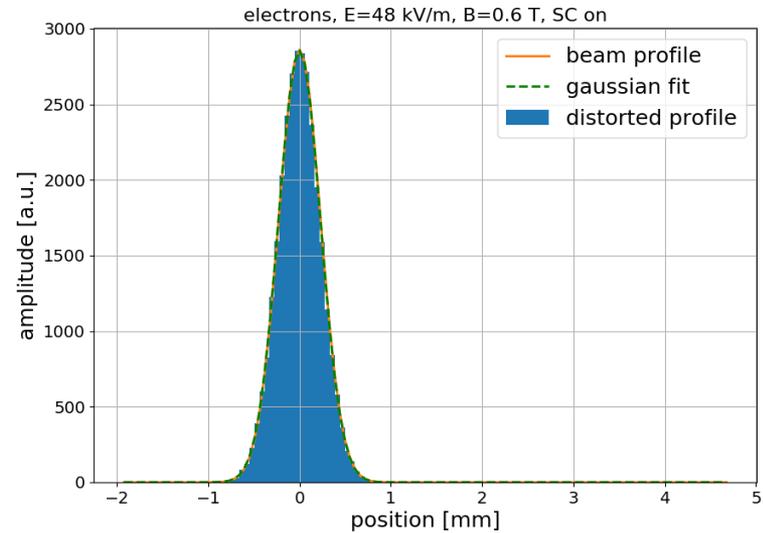
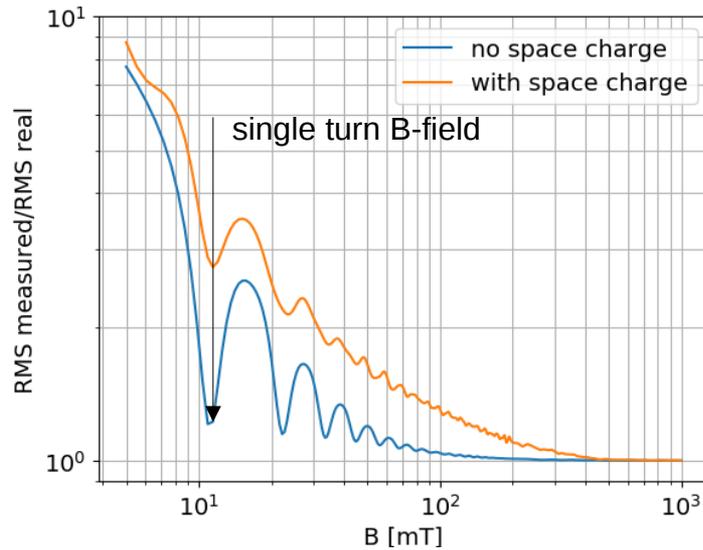
$$v_{\text{ionization}} = 0$$

Gyration and ExB drift. Transverse momentum transfer is very large.

Final gyroradius: 340  $\mu\text{m}$ .



# Electron tracing – very strong B-field



- unpractical solution: big and expensive magnet!

# Correction methods: sampling gyroradius

- No simple analytic procedure/formula (as for electric field case) found

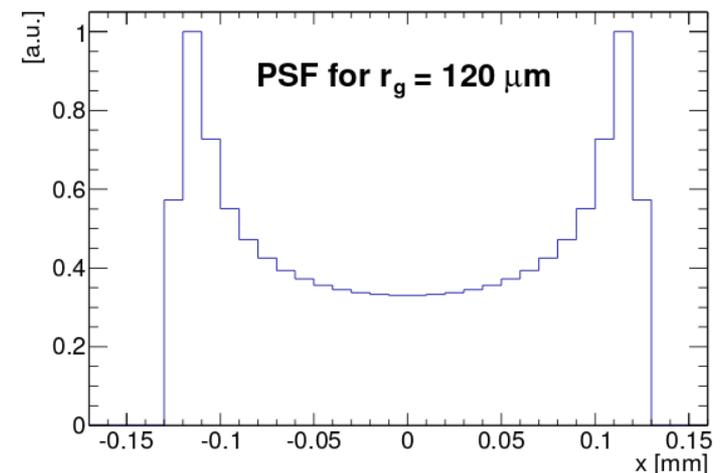
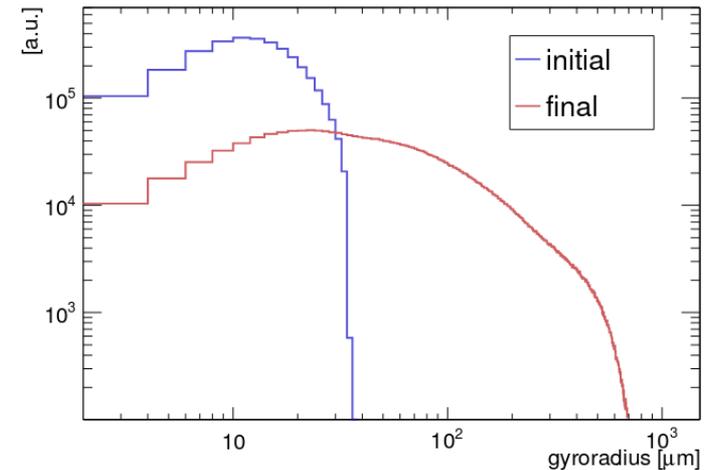
- **Electron sieve**

- MS. et al., *Investigation of the effect of beam space-charge on electrons in ionization profile monitors*, Proc. of HB2014 (MOPAB42)
- D. Vilsmeier, CERN-THESIS-2015-035

- technically difficult

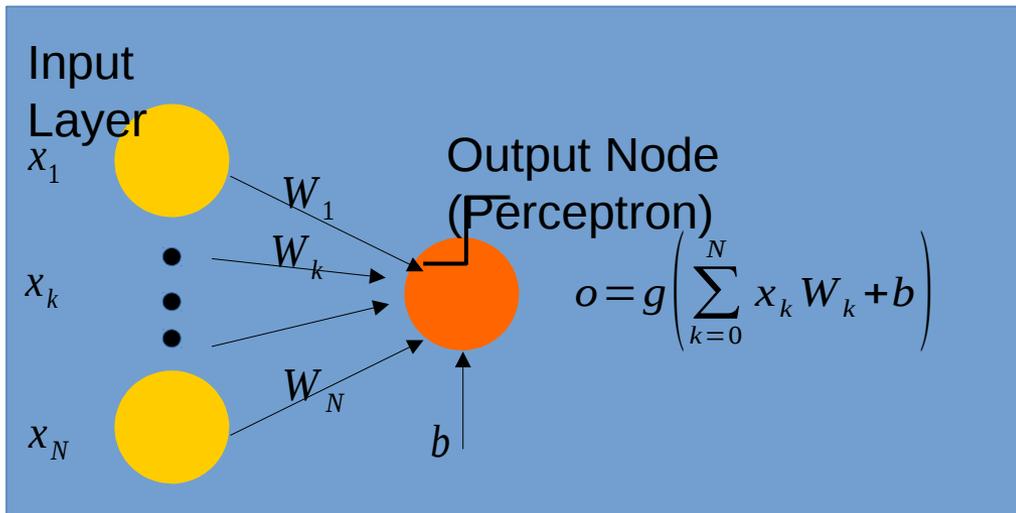
- Methods based on Machine Learning algorithms: finding arbitrary mapping between distorted profile and original one.

- R. Singh, M. S., D. Vilsmeier, *Simulation supported profile reconstruction with machine learning*, Proc. of IBIC17 (WEPC06)
- D. Vilsmeier et al., *Reconstructing Space-Charge Distorted IPM Profiles with Machine Learning Algorithms*, Proc. of IPAC 2018 (WEPAC008)



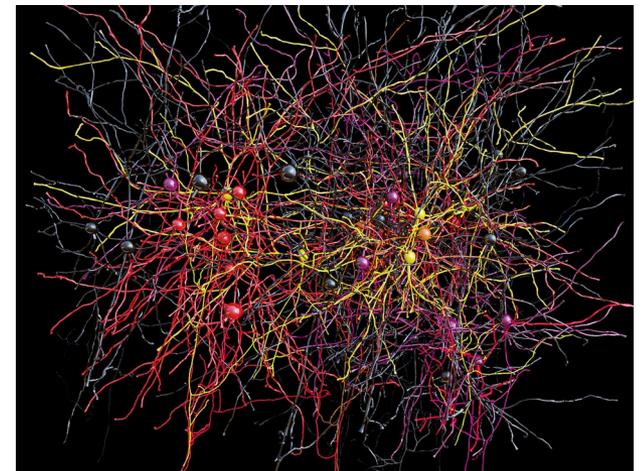
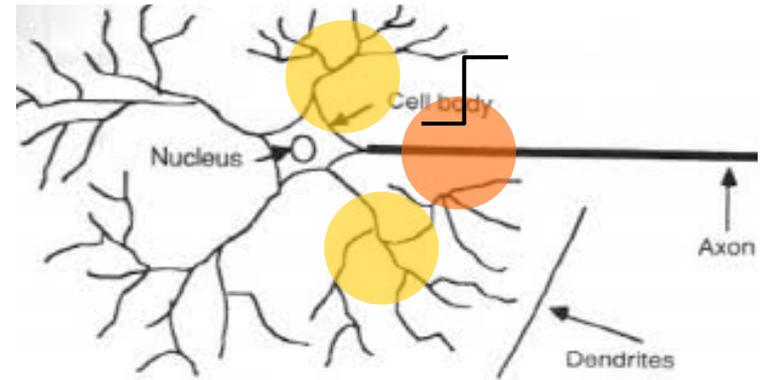
# Artificial Neural Network

- Machine Learning - algorithms which can learn and make predictions on data, **without explicit programming**
- Biologically inspired → Brain cells → neurons, computation via connections and thus Networks
- The basic node of ANNs is “Perceptron”



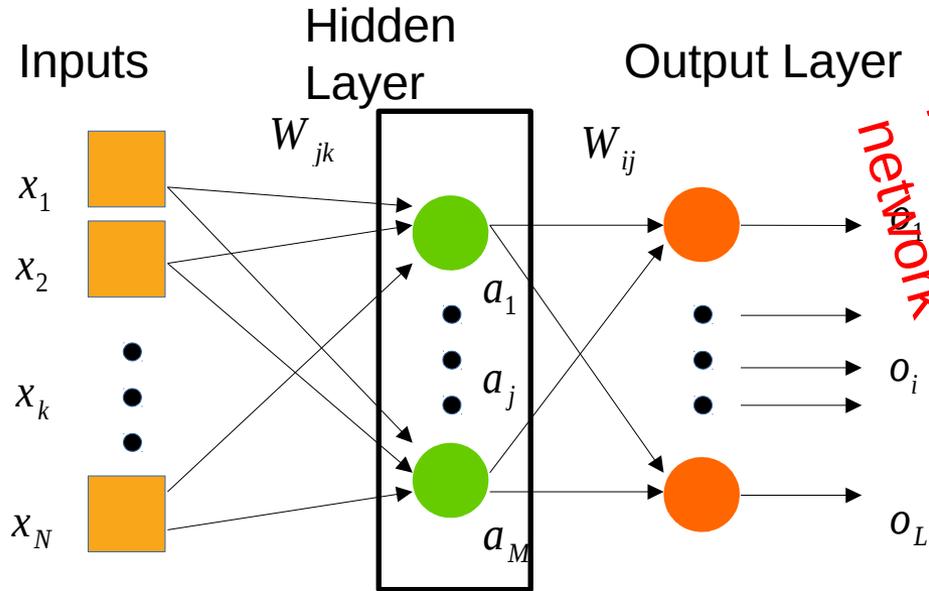
## Perceptron parameters:

- Weights from the inputs ( $X$ ) and bias ( $b$ )
- $g$  is the activation function, a step-like function with a threshold



[<https://www.wired.com/2016/03/took-neuroscientists-ten-years-map-tiny-slice-brain>]

# Hidden layers



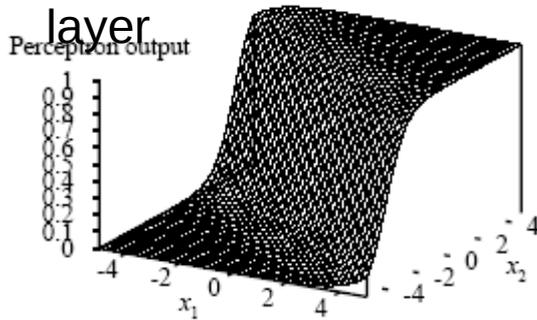
- Each hidden layer and output layer node is a perceptron

$$o_i = g \left( \sum_{j=0}^M W_{ij} \left( g \left( \sum_{k=0}^N x_k W_{jk} + b_j \right) \right) + b_i \right)$$

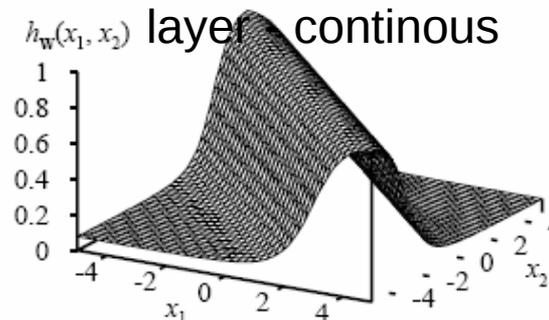
Adding “hidden” layer(s) allow non-linear target functions to be represented

# Multi-layer perceptron (MLP)

Perceptron: No hidden layer



One hidden layer continuous



(In practice we use additional hidden layers even to continuous problems)

• Carla P Gomes, Lecture Notes CS 4700: Foundations of Artificial Intelligence

- **Universal approximation theorem:**

Every bounded continuous “target” function can be approximated with arbitrarily small error, by network with single hidden layer [Cybenko 1989; Hornik et al. 1989]

If we have any unknown function,  $y = f(x, y, z \dots)$  it can be approximated by:

$$o_i = g \left( \sum_{j=0}^M W_{ij} \left( g \left( \sum_{k=0}^N x_k W_{jk} + b_j \right) \right) + b_i \right) \quad \text{universal approximator}$$

# MLP Network training

## ➤ How it works:

- Activation function **g** must be **differentiable**, eg. sigmoid or tanh.
- Initial weights chosen randomly.

$$o_i = g \left( \sum_{j=0}^M W_{ij} \left( g \left( \sum_{k=0}^N x_k W_{jk} + b_j \right) \right) + b_i \right)$$

$inp_i$

- For training record (or a batch of records) a **cost function** (or loss or error) is calculated, for instance mean squared error:
- (y-desired output, o-actual output)

$$E = \sum_{i=0}^L (y_i - o_i)^2$$

- The **cost function gradient** is calculated for each layer:

$$\frac{\delta E}{\delta W_{ij}^1} = a_j Err_i g'(inp_i)$$

$$\frac{\delta E}{\delta W_{jk}^2} = x_k g'(inp_j) \sum_{i=0}^M W_{ij} Err_i g'(inp_i)$$

- **New weights** are calculated:
- Repeat for new record  
(but you can use the same record later again)

$$W(t+1) = W(t) + \alpha \frac{\delta E}{\delta W}$$

$\alpha$ -learning rate

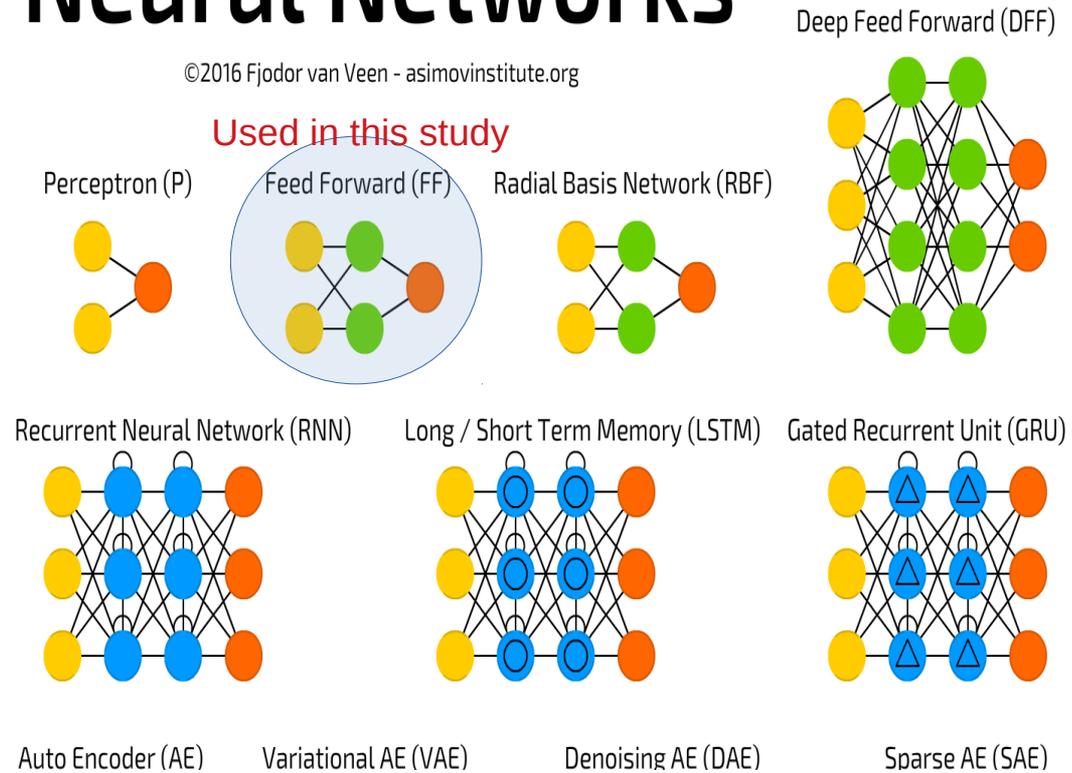
A mostly complete chart of

# Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

- Backfed Input Cell
- Input Cell
- Noisy Input Cell
- Hidden Cell
- Probablistic Hidden Cell
- Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell

Used in this study



Source: Fjodor Van Veen, Asimov Institute, Utrecht

... 20 more

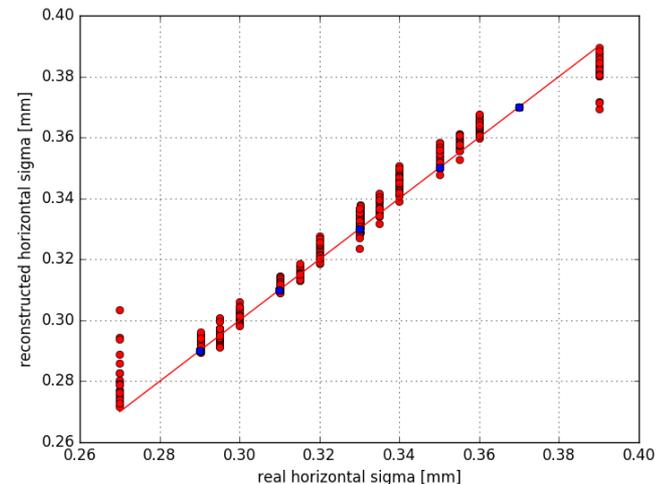
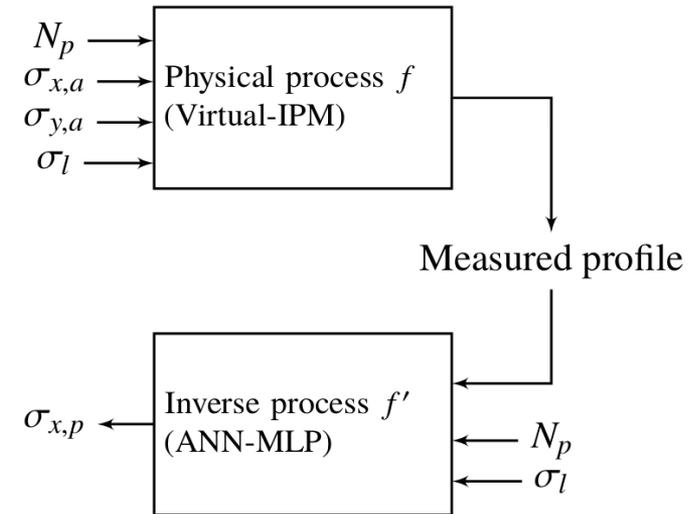
# Profile correction using ANN (I)

## First approach:

- Training sample on grid (375 points):

$\sigma_x$ [mm]	0.29, 0.31, 0.33, 0.35, 0.37
$\sigma_y$ [mm]	0.4, 0.45, 0.5, 0.55, 0.6
$N_p$ [ $10^{11}$ ]	1.1, 1.25, 1.40, 1.55, 1.7
$4 \cdot \sigma_z$ [ns]	0.9, 1.05, 1.2

- 2-layer network, 4 validation samples:
  - 1%, 25%, 50% off
  - 100% off – outside grid
- Convolute with Point Spread Function
- Use *tensorflow* and *Matlab NN toolbox*
- Value of  $\sigma_x$  restored with 1% accuracy!
- Good performance with noise.
- R. Singh, M. Sapinski, D. Vilsmeier, *Simulation supported profile reconstruction with machine learning*, Proc. of IBIC17 (WEPC06)



# Profile correction using ANN (II)

## Second approach:

- Training: sample of 13500 random points
- 4-layer network
- Validation set – also random
- Verify also other Machine Learning algorithms:
  - linear regression,
  - kernel Ridge regression,
  - support vector machine

D. Vilsmeier et al., *Reconstructing Space-Charge Distorted IPM Profiles with Machine Learning Algorithms*, Proc. of IPAC 2018, paper WEPAK008

Surprisingly: **even linear regression gives very good results.**

But we have not studied noise here

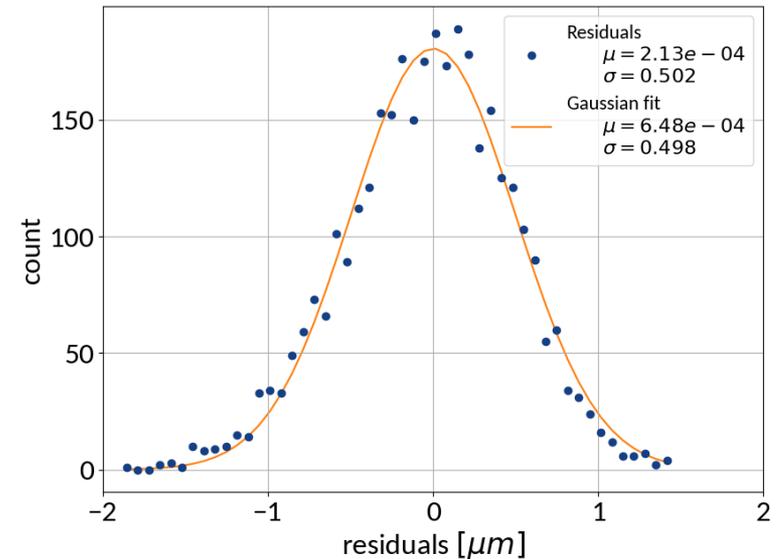


Table 2: Resulting Scores for the Different Models. Values are given in units of  $1 \mu\text{m}$ ,  $1 \mu\text{m}^2$  respectively.

	$\mu(\text{res})$	$\sigma(\text{res})$	<b>R2</b>	<b>EV</b>	<b>MSE</b>
<b>LR</b>	0.012	0.449	0.99976	0.99976	0.201
<b>KRR</b>	0.005	0.340	0.99986	0.99986	0.115
<b>SVR</b>	0.006	0.349	0.99985	0.99985	0.121
<b>MLP</b>	0.232	0.370	0.99977	0.99984	0.190

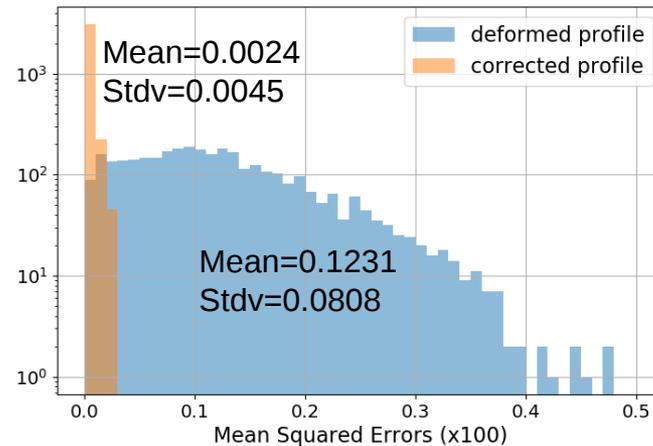
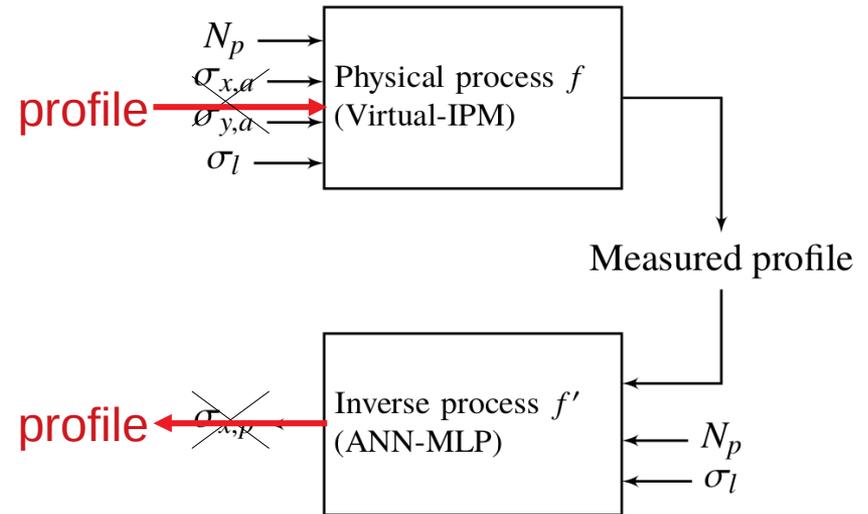
# Profile reconstruction using ANN

## Third approach:

- Training: sample of 13500 random points (only gaussian profiles)
- 2-layer network
- Validation set – also random, gaussian and non-gaussian
- Profile: 98 bins, 55  $\mu\text{m}/\text{bin}$
- Quality of profile reconstruction assessed using Mean Squared Deviations between real beam profile and corrected profile:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$

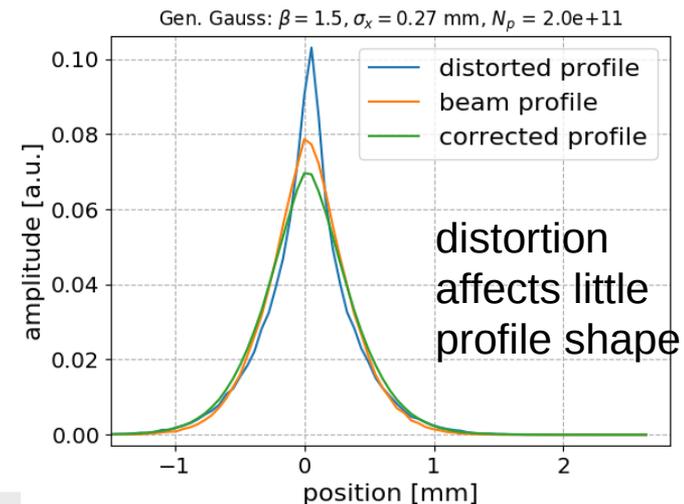
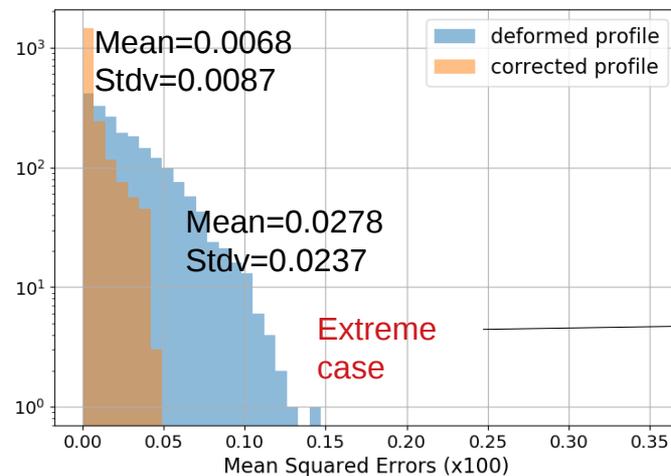
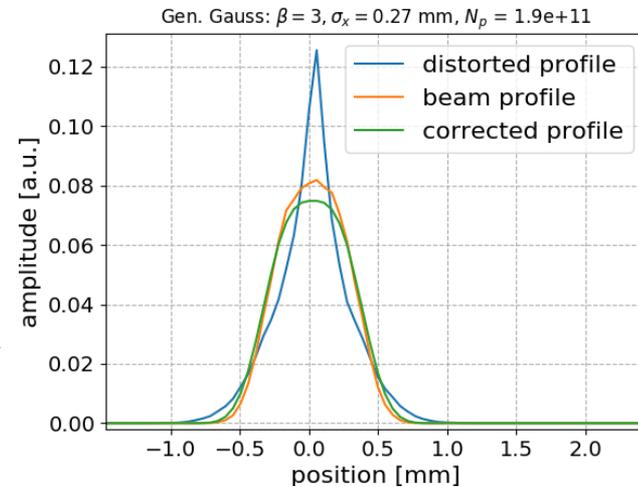
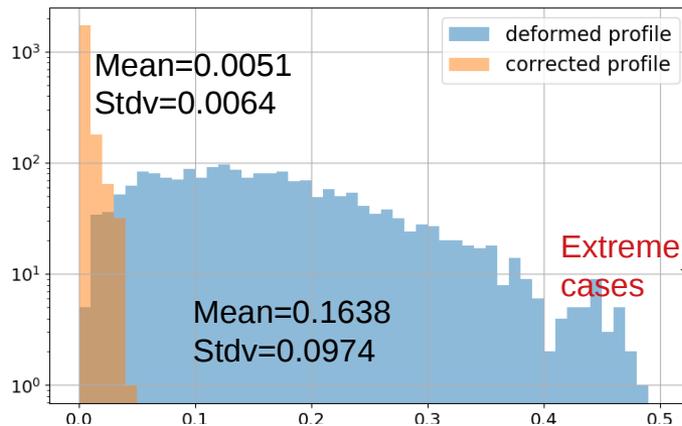
- Results for gaussian profiles:  
Very good profile shape reconstruction



# Reconstructing the profile: generalized gaussian

$$\frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x-\mu|/\alpha)^\beta}$$

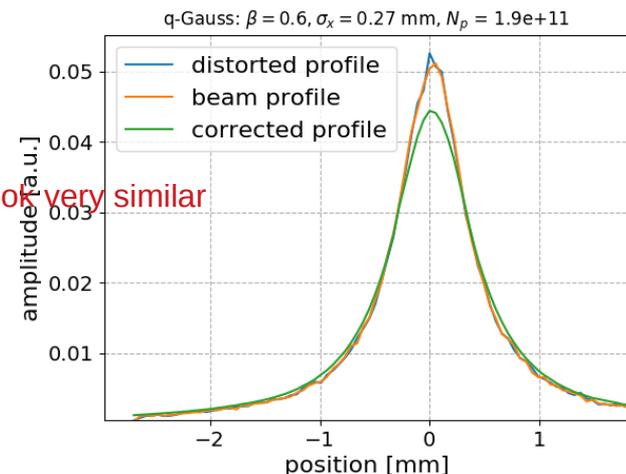
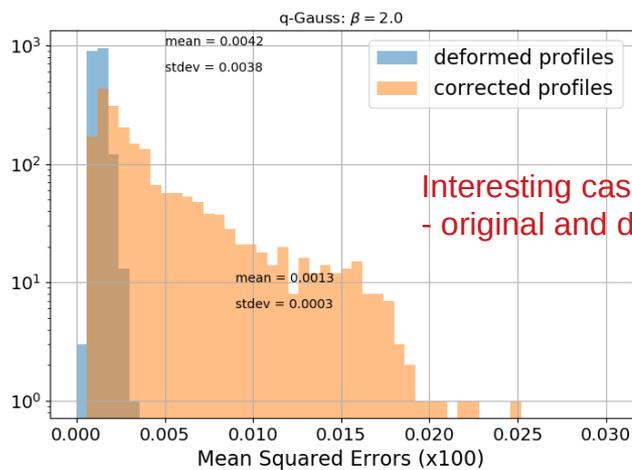
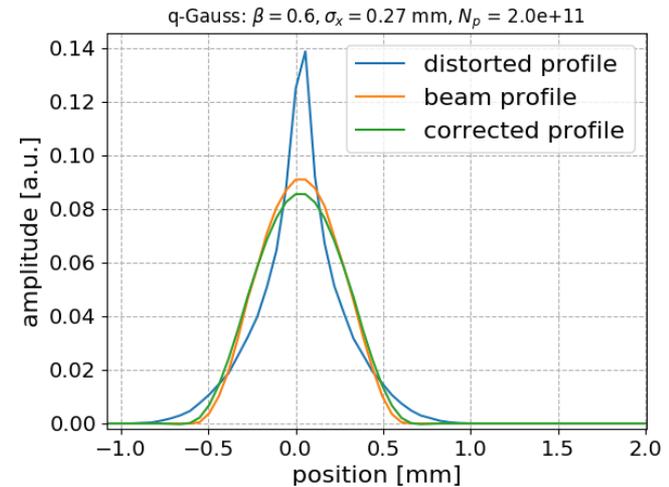
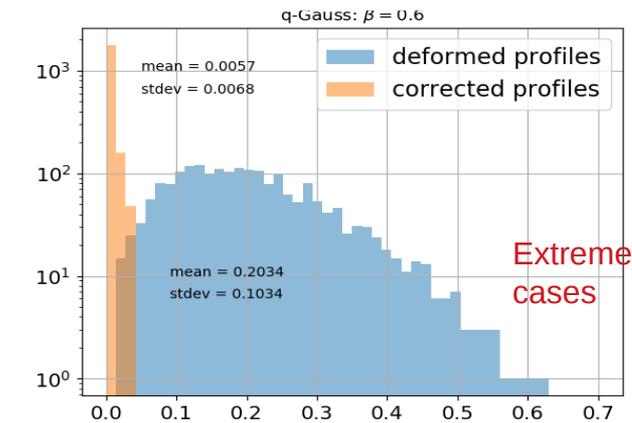
- Generalized-gaussian profiles with beta=3 and 1.5
- Neural network trained only on gaussian profiles!



# Reconstructing the profile: q-Gaussian

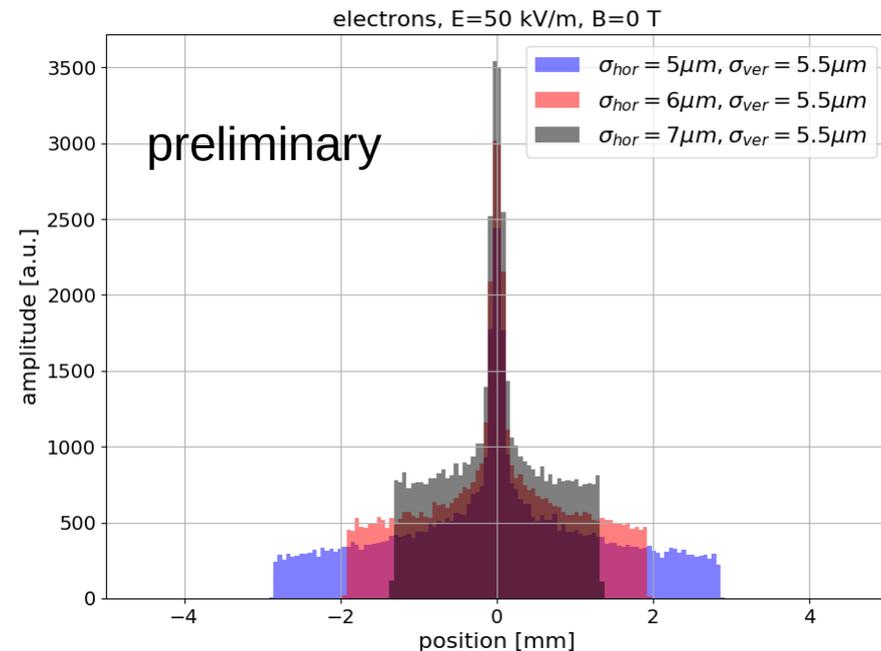
$$\frac{\sqrt{\beta}}{C_q} e_q(-\beta x^2)$$

- Used for beam halo parametrization, beta=0.6 and 2.0
- **Neural network trained only on gaussian profiles!**



# Remark: measuring micrometer-size beams

- If we understand the beam profile deformation, we could use it to measure high-brightness **beams smaller than the resolution of the detector.**
- Example SwissXFEL: 5.8 GeV electron beam, 230 pC bunch charge, 21 fs bunch length, 5-7  $\mu\text{m}$  transverse size.
- Even if bunch size is 1/10th of detector resolution,  
**the shape of the deformed profile strongly depends on the bunch size!**
- Alternative to  
*R. Tarkeshian et al.*  
*Phys. Rev. X 8, 021039*  
(beam width reconstruction based on ion energies)



- For detection of failing Beam Position Monitors (BPM), optics correction
- BPM linearization
- Beam Loss Monitors: patten detection
  
- Lot of studies in accelerator control, optimization, modelling and failure prediction.

9th International Particle Accelerator Conference  
ISBN: 978-3-95450-184-7

IPAC2018, Vancouver, BC, Canada

JACoW Publishing  
doi:10.18429/JACoW-IPAC2018-WEPAF062

## MACHINE LEARNING METHODS FOR OPTICS MEASUREMENTS AND CORRECTIONS AT LHC

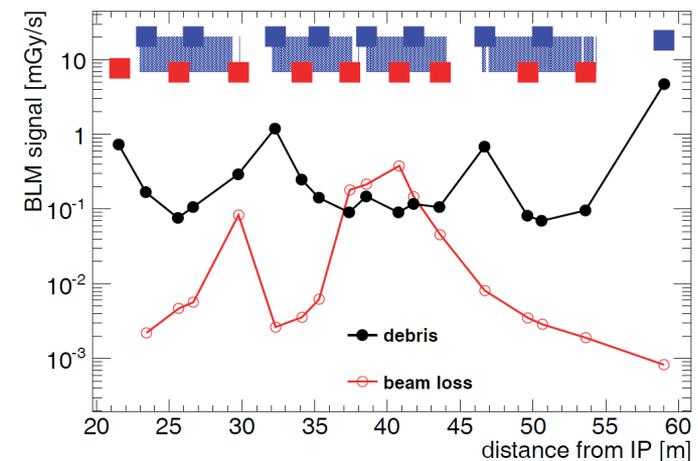
E. Fol\*, F. Carlier, J. Coello de Portugal, A. Garcia-Tabares Valdivieso, R. Tomás  
CERN, Geneva, Switzerland

Proceedings of BIW2012, Newport News, VA USA

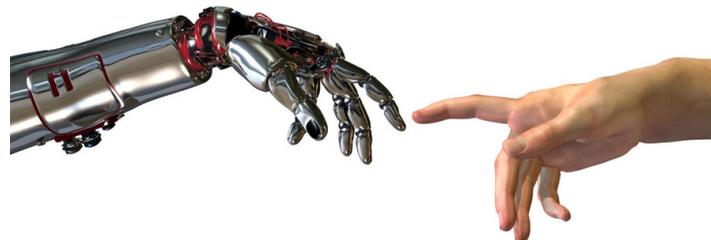
MOPG020

## APPLICATION OF NEURAL NETWORK ALGORITHMS FOR BPM LINEARIZATION\*

J. Musson#, C. Seaton, J. Yan, M. Spata, Thomas Jefferson National Accelerator Facility, Newport News, VA, 23606, U.S.A.



- Beam size measurement of bright beams is challenging. Even magnetic IPMs suffer from measured profile deformation which is difficult to correct.
- Machine Learning algorithms perform very efficient profile correction.
- ML techniques become a standard tool for physicists and engineers.
- Sometimes they can give hint about simpler solutions (linear regression).
- Can save lots of money (a set of 1T magnets for IPM ~ 5M€).
- Modern tools (eg. tensorflow+keras) are easy to use.
- In the field of Beam Instrumentation the use of these techniques is still in early phase.



# Further reading and playing

- “How could a Kangaroo climb Everest?” - about minimization algorithms:  
<ftp://ftp.sas.com/pub/neural/kangaroos>
- ANN recognizing drawings: <https://quickdraw.withgoogle.com>
- Music composed by AI: <http://www.flow-machines.com/ai-makes-pop-music/>
- Unreasonable effectiveness of ANN:  
<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- E. Musk concerned about AI: <https://www.youtube.com/watch?v=0NTb10Au-Ic>
- AI algorithms in social media – very interesting:  
[https://www.ted.com/talks/zeynep\\_tufekci\\_we\\_re\\_building\\_a\\_dystopia\\_just\\_to\\_make\\_people\\_click\\_on\\_ads](https://www.ted.com/talks/zeynep_tufekci_we_re_building_a_dystopia_just_to_make_people_click_on_ads)
- ANN playing with images:  
<https://nerdist.com/why-are-googles-neural-networks-making-these-brain-melting-images>
- ...

## Acknowledgments:

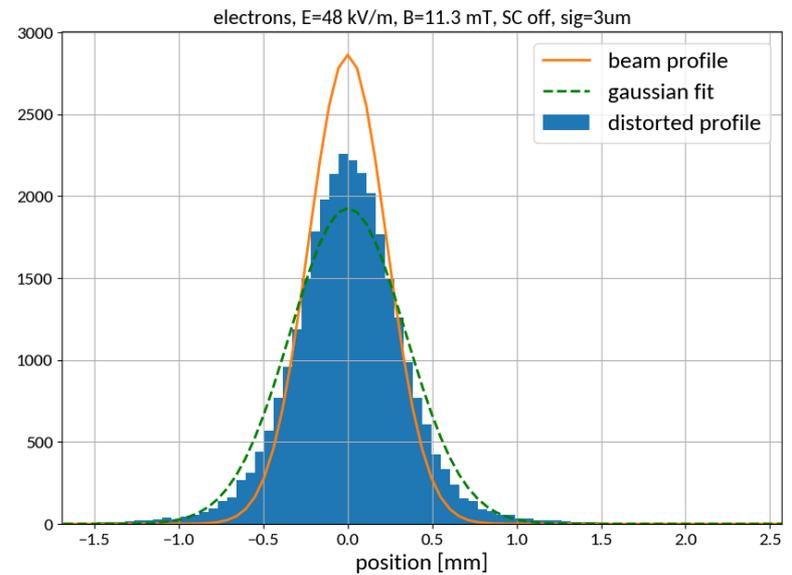
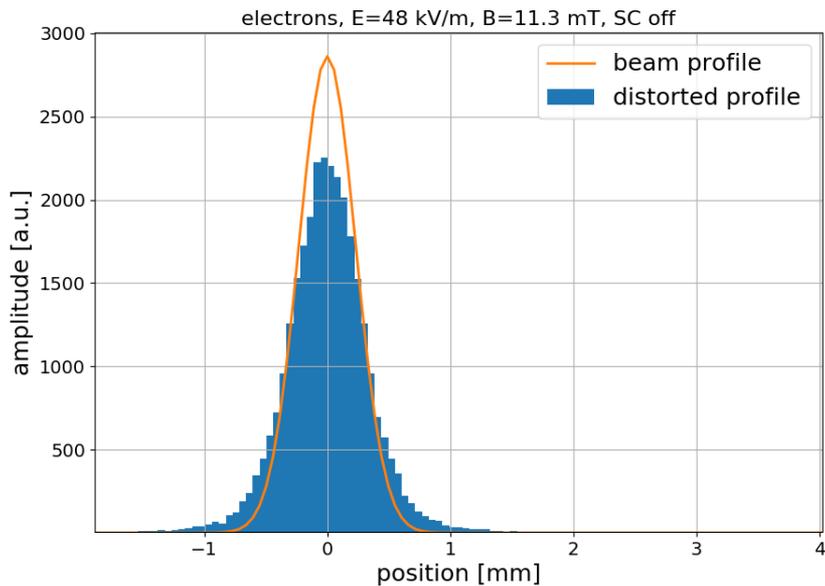
A. Reiter, P. Forck, K. Sato

# Additional slides

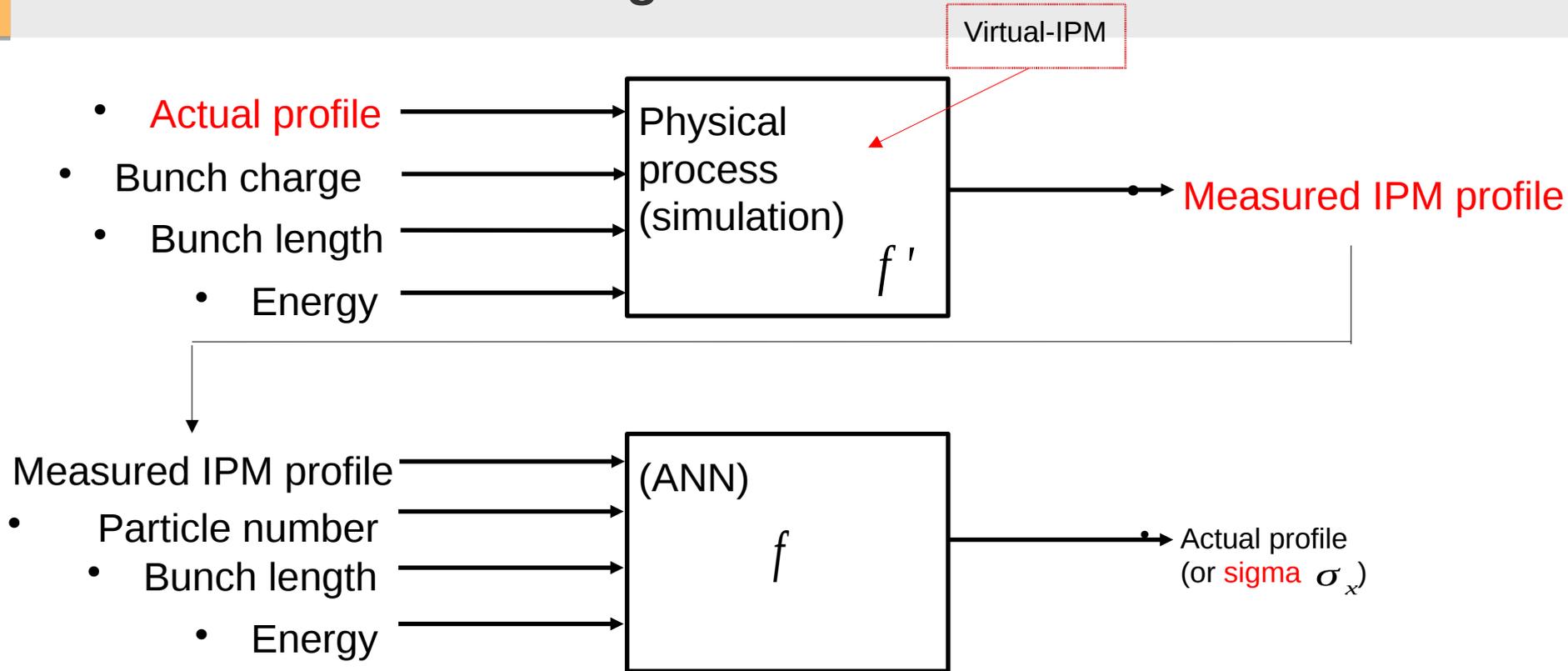
- Some algorithms known since 40's (Gauss Newton or Levenberg-Marquardt).
- **Backpropagation with Gradient Descent** developed in 70's
  - speeds up in ANN training – it triggered a wave of interest in ANN applications – still most popular.

# Single-turn magnetic field

## ➤ Influence of vertical beam size



# Profile correction using ANN



Training "grid" (375 points):

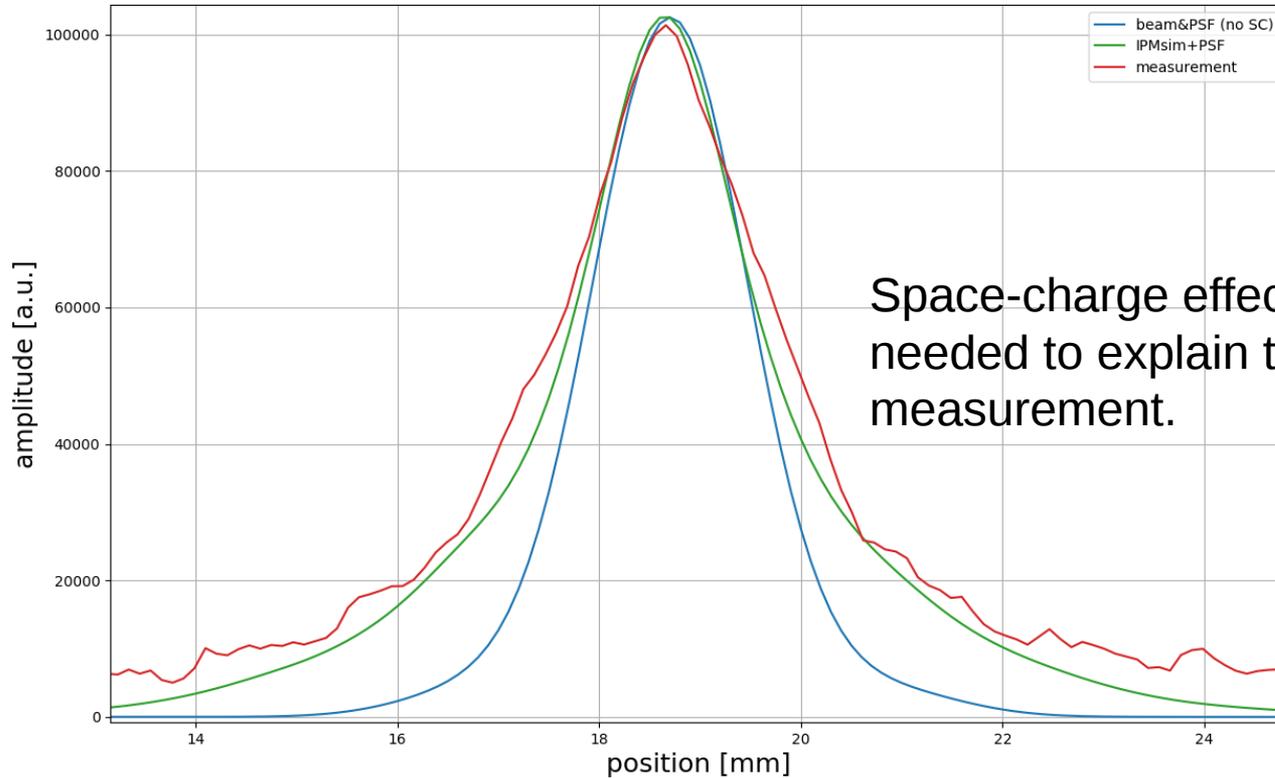
Using *tensorflow* and *Matlab NN toolbox*

$\sigma_x$	0.29, 0.31, 0.33, 0.35, 0.37
$\sigma_y$	(mm) 0.4, 0.45, 0.5, 0.55, 0.6
$N_p$	(mm) $1.1e11, 1.25e11, 1.40e11, 1.55e11, 1.7e11$
$\sigma_l$	0.9, 1.05, 1.2 (ns)

# What is Machine Learning?

- Algorithms which can learn and make predictions on data, **without explicit programming**.
- The term by Arthur Samuel (IBM) in 1959.
- Machine learning is closely related to **computational statistics** and to **mathematical optimization**.
- Data mining is a sub-field of Machine Learning known as unsupervised learning.
- Expert systems – are made of digitized/encoded expert knowledge. They are not Machine Learning algorithms. Still useful is there is little data available for training. Mixed systems are also available.

# Space-charge on SPS beam (16 mT)



Space-charge effect clearly needed to explain this measurement.

From: <https://www.solver.com/training-artificial-neural-network-intro>

- *There is no best answer to the layout of the network for any particular application. There are general rules:*
  - As the complexity between input and output increases, the number of the perceptrons in the hidden layer should also increase.
  - If the process being modeled is separable into multiple stages, then additional hidden layer(s) may be required. Otherwise additional layers may simply enable memorization of the training set, and not a general solution effective with other data.
  - The amount of training data sets an upper bound for the number of perceptrons in the hidden layer(s).  
If you use too many perceptrons the training set will be memorized.
  - ->generalization of the data will not occur, making the network useless on new data sets.

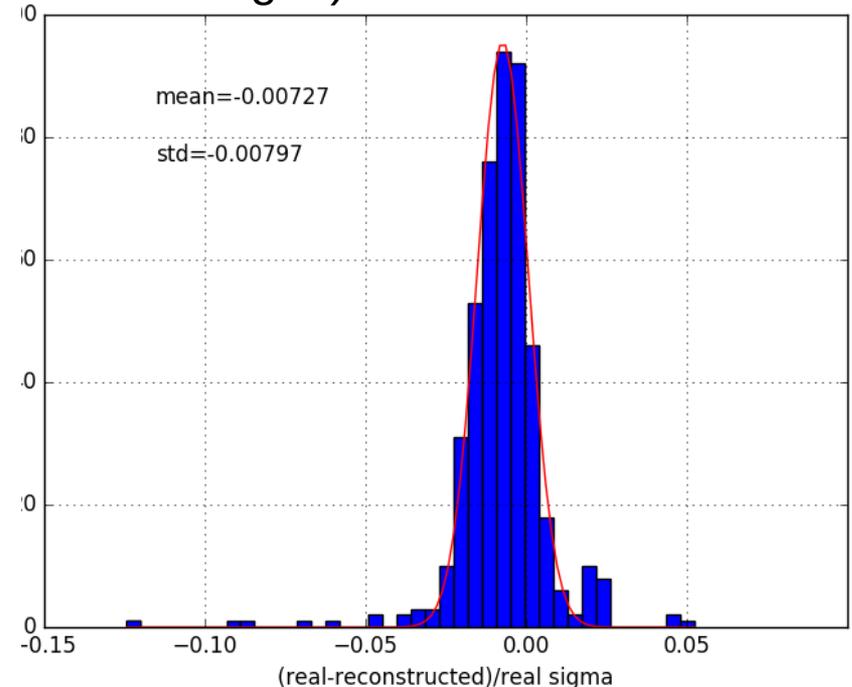
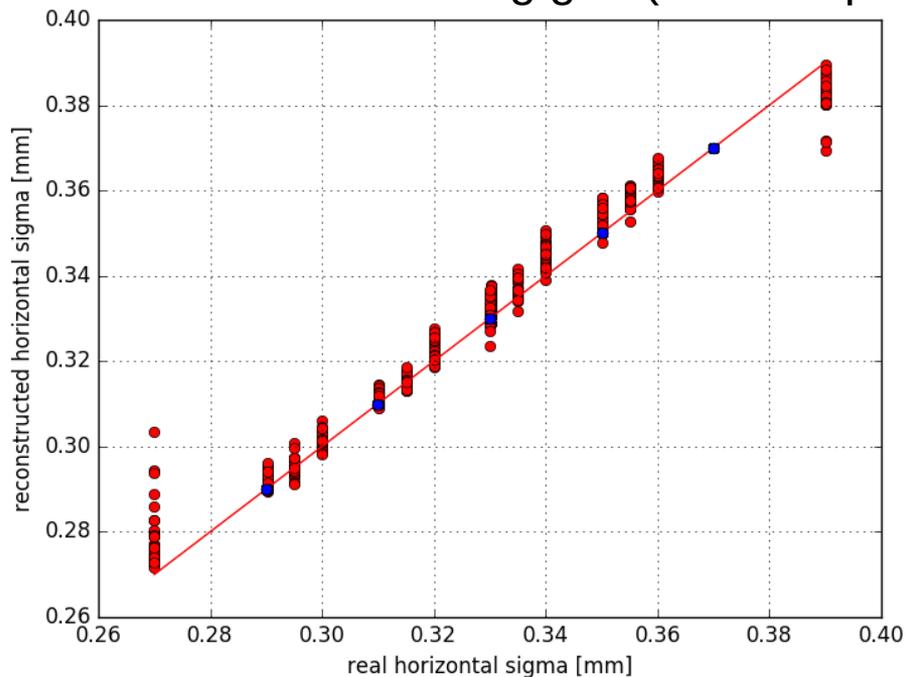
## Validation "grid" (128 points)

4 validation data sets (inputs and outputs) created:

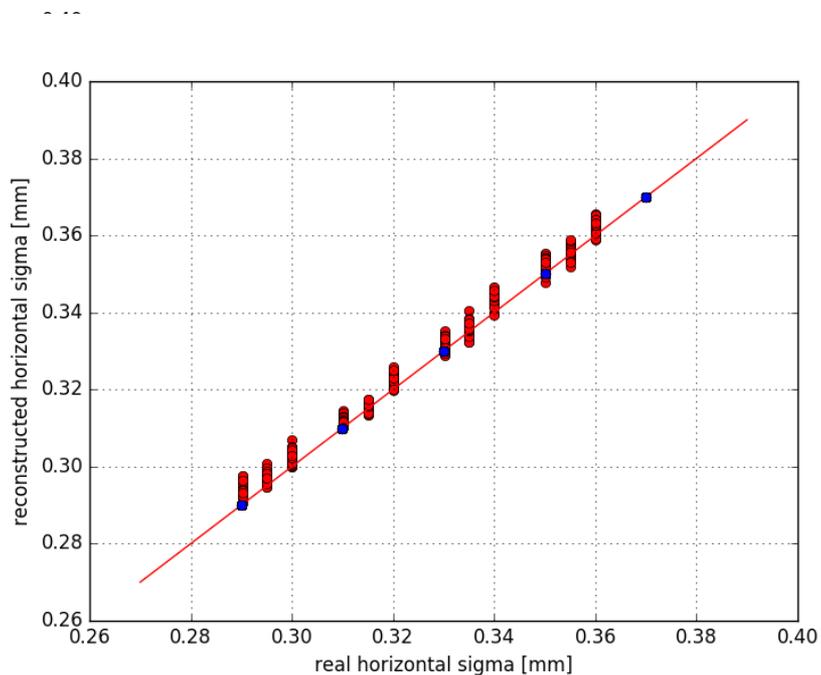
- 1% off the training grid in each dimension (within in grid)
- 25% off the training grid in each dimension
- 50% off the training grid in each dimension
- 100% off the training grid (the next point outside the grid)

For 12 runs:  
sigma systematically  
overestimated by  
0.4% with error 0.8%

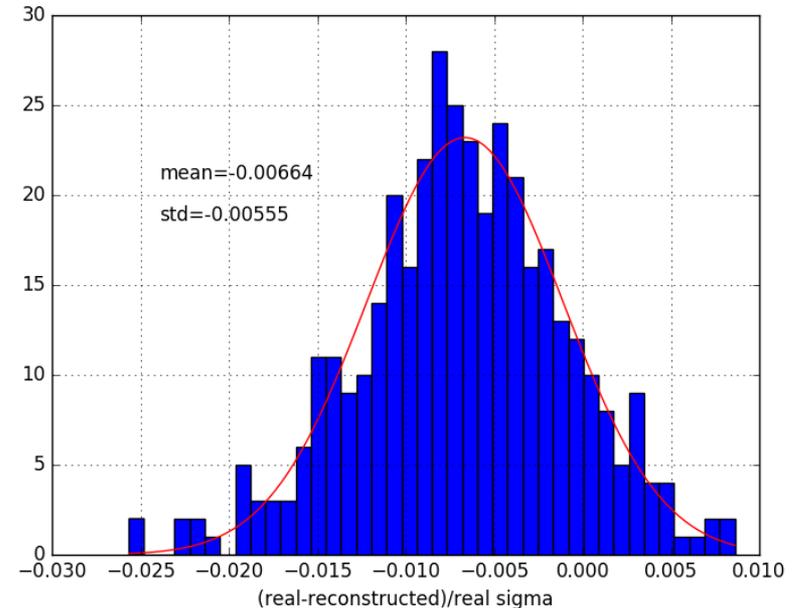
Much smaller than  
measurement errors!



Removing the validation sample  
outside of “training” area



For 12 runs:  
sigma systematically  
overestimated by 0.05% with  
error 0.7%



# Example 2: IPM profile corrections



- 
- Using higher electric and magnetic fields (expensive, sometimes impractical).
- **Electrons + electric and magnetic fields:** Sieve method (deconvolve with PSF of radius of Gyration) – difficult in practice.

*[Dominik Vilsmeier, Bachelor Thesis, CERN]*

- **Electric fields only (ions):** several calibration/correction attempts.

*[eg. R. E. Thern, PAC1987, J. Amundson et al., PRSTAB 6, 102801 (2003)]*

Latest work: Assumption on input beam distribution (Generalized Gaussian) and iterative procedure for input reconstruction from distorted profile using the data generated from simulation tool.

*[Jan Egberts, PhD Thesis, CEA Saclay]*

Approximate target function

$$o_i = g \left( \sum_{j=0}^M W_{ij} \left( g \left( \sum_{k=0}^N x_k W_{jk} + b_j \right) \right) + b_i \right)$$

Solve optimization problem with training data

STC

$$E = \sum_{i=0}^L (y_i - o_i)^2 + \lambda \sum_{j=0}^M \sum_{k=0}^N (W_{ij})^2$$

Calculate gradient, update weights

$$\frac{\delta E}{\delta W_{ij}} = a_j \text{Err}_i g'(inp_i)$$

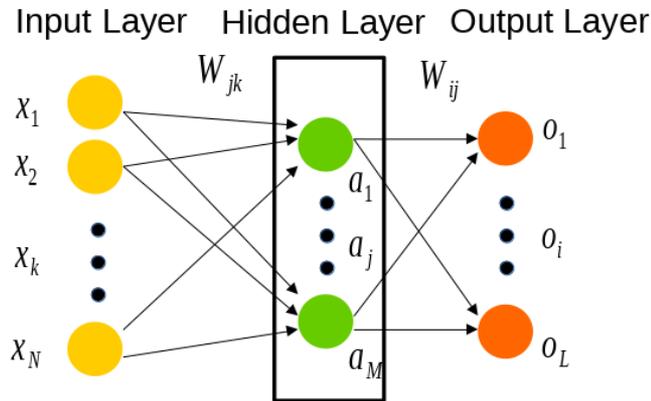
$$W_{ij}(t+1) = W_{ij}(t) + \alpha \frac{\delta E}{\delta W}$$

Validate with other data, "validation data" to check the generalization or "learning"

If not, change the number of units or architecture

- Supervised learning, unsupervised learning, reinforcement learning
- Batch learning, incremental learning
- Functions: Activation function, Target function, Objective or error function
- Optimization: Gradient descent, Levenberg-Marquardt, Epoches, Learning rate, Momentum
- For more: How could a Kangaroo climb Everest?  
<ftp://ftp.sas.com/pub/neural/kangaroos>
- Generalization: Cross validation, regularization, early stopping

# Calculating backpropagation



$$o_i = g \left( \sum_{j=0}^M W_{ij} \left( g \left( \sum_{k=0}^N x_k W_{jk} + b_j \right) \right) + b_i \right) = g(s)$$

$$E = \sum_{i=0}^L (y_i - o_i)^2 \quad - \quad E = (y - o)^2$$

>

$$\frac{dE}{dW_{11}} = \frac{dE}{do} \cdot \frac{do}{ds} \cdot \frac{ds}{dW_{11}}$$

$$-2(y-o)$$

$$g'(s)$$

$$a_1$$

$$\frac{\delta E}{\delta W_{ij}^1} = a_j \text{Err}_i g'(inp_i)$$

# Examples of ML-based projects (I)

## AlphaGo :

- Go is **difficult for algorithms** because of number of configurations ( $>2 \times 10^{170}$ , chess only  $\sim 5 \times 10^{52}$ ), atoms in the Universe  $\sim 10^{80}$ .
- The program uses Artificial Neural Network for learning and Monte Carlo Tree Search for decide about next move.
- 1 year learning time, 183 MWh energy, excessive data sample – not the way human learns, but:
  - AlphaGo won against the highest-qualified humans.
  - It has exhibited **creative skills** making moves seldom done by humans.

