UNIVERSITY OF AMSTERDAM

Radboud Universiteit

Fast approximation of SUSY NLO cross-sections using Deep Learning

SUSY 2018, Barcelona

24.07.2018

Based on work by Sascha Caron, Jong-Soo Kim, Sydney Otten, Krzysztof Rolbiecki, Roberto Ruiz de Austri, Jamie Tattersall

This talk is …

- an attempt at a compromise between physics and machine learning
- tackling the thought that neural networks are magic black boxes you only have to throw data at for a perfect solution
- guiding through the process of constructing an AI for predicting an LHC observable
- demonstrating the fact that we can handle high-dimensional models and can go beyond low-dimensional simplified models
- showing that even for a regression task, classification, active learning and feature engineering (or injection of deeper expert knowledge) might help a lot.
- trying all of the above in less than 20 minutes.

What are we looking at and why?

There are reasons to believe that there is **physics beyond the standard model**:

- Hierarchy problem
- Dark Matter
- Gravity at the Planck scale
- Dark energy
- …

For the first two points, SUSY can provide a solution → Well motivated theory!

Many possible realizations of SUSY in nature exist. The manifold of parameter combinations is 19 dimensional (*pMSSM-19*) and continuous.

If a signal at the **LHC** appears that **cannot be explained by the SM**, we want to be able to tell if it can be explained by SUSY and **which Lagrangian parameters fit the signal** best.

→ To create such a mapping, we need to know LHC observables like σ.

→ For tools that help with this, it is crucial to know the NLO electroweakino (Neutralino and Chargino) cross section within ms.

Neutralinos and Charginos

Neutralinos

- four neutralinos exist: $\chi_i^0, \mathrm{i} = 1 \dots 4$
- They are mixtures of the neutral bino, wino and higgsinos:

$$\tilde{\chi}^0 = N\tilde{\psi}^0 \text{ with } N \text{ being a 4x4 mixing matrix and } \tilde{\psi}^0 = \begin{pmatrix} -i\tilde{b} \\ -i\tilde{w}^3 \\ \tilde{h}_1 \\ \tilde{h}_2 \end{pmatrix}$$

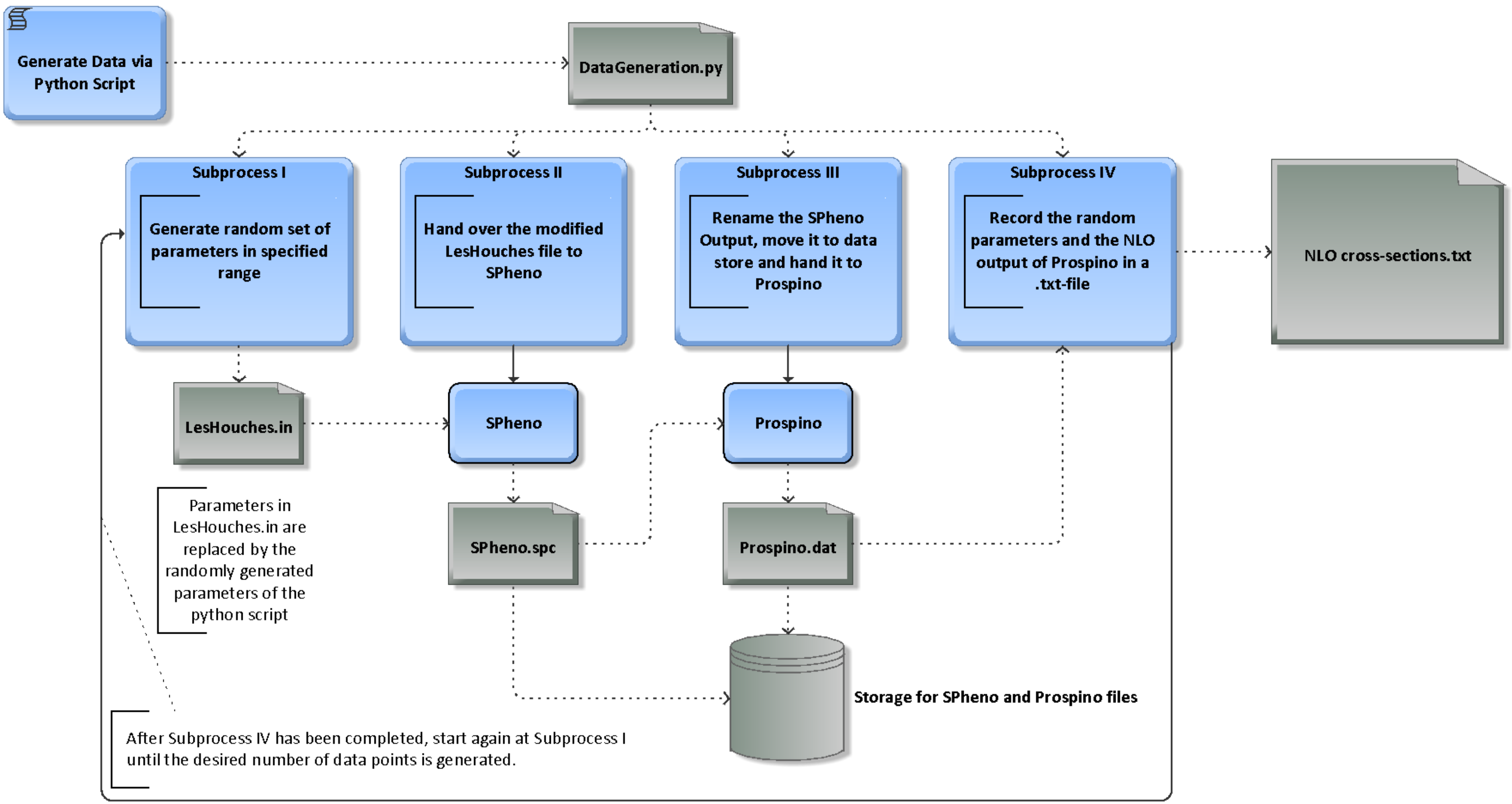
Charginos

- four charginos exist: $\chi_i^{\pm}, \mathrm{i} = 1, 2$
- They are mixtures of the charged wino and higgsinos:

$\tilde{\chi}^{-T} = \tilde{\psi}^{-T} U^T$ and $\tilde{\chi}^{+} = V\tilde{\psi}^{+}$ with U and V being 2x2 mixing matrices and

$\tilde{\psi}^{-} = \begin{pmatrix} -i\tilde{w}^{-} \\ \tilde{h}_1^{-} \end{pmatrix}$ and $\tilde{\psi}^{+} = \begin{pmatrix} -i\tilde{w}^{+} \\ \tilde{h}_2^{+} \end{pmatrix}$.

How do we generate data?



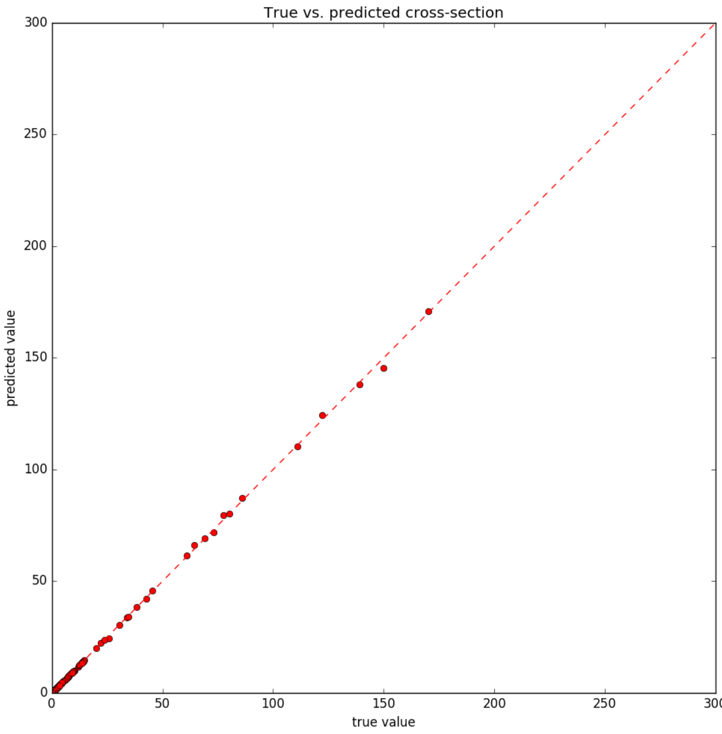
What is our mission?

Provide a computational tool

- that gives the NLO cross section for electroweakinos
- in milliseconds
- with a mean absolute percentage error (mape) below 1 %
- with a maximum percentage error below 10 %

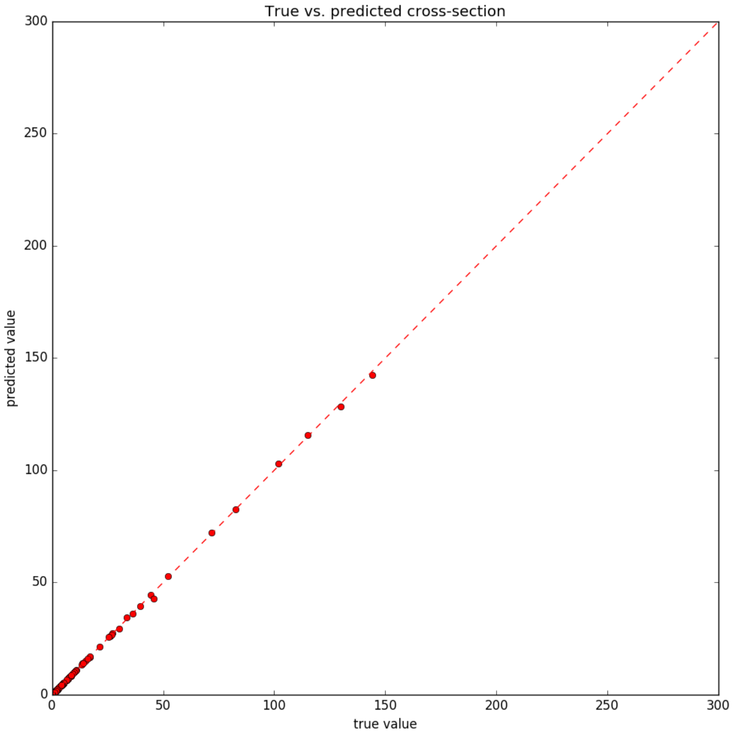
First results in 4D and 5D

$\chi_1^+\chi_1^-$ NLO cross section for 4D



Mape: 0.3 %, max. error < 5 %, #points: 18‘000

$\chi_1^+\chi_1^-$ NLO cross section for 5D



Mape: 0.3 %, max. error < 5 %, #points: 54‘000

Interlude: Decomposing the NLO

Generating NLO data takes very long! $t_{\sigma_{NLO}} \approx \frac{180s}{point}$

But luckily, $\sigma_{NLO} = K \cdot \sigma_{LO}$ and $t_{\sigma_{LO}} \approx \frac{1.5s}{point}$.

We think that:

- most difficulties already show for the LO cross section
- the K factors are easy because $K \in [1,2]$

→ We generate data for these two quantities separately.

→ Instead of 480 $\frac{NLO\ points}{day \cdot core}$, we can now generate 57600 $\frac{LO\ points}{day \cdot core}$.

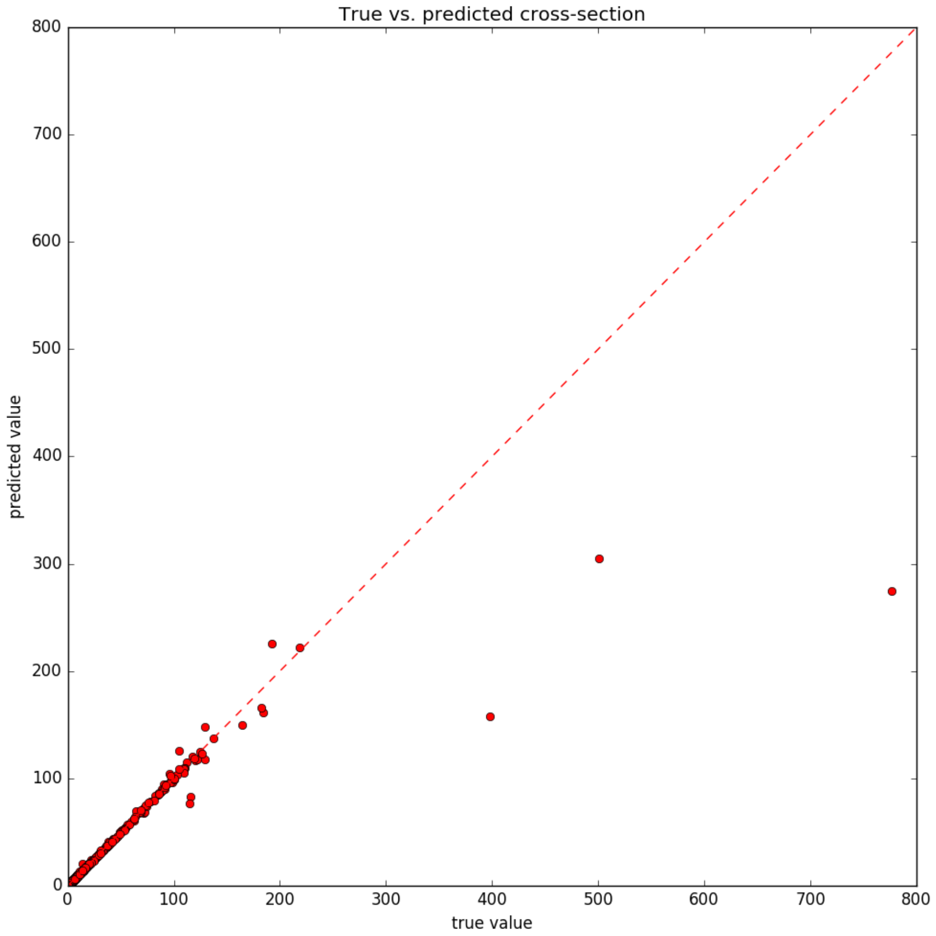
Results in 12D

Mape: 0.5 %

Max. error: 63 %

#points: 525‘000

$\chi_1^+\chi_1^-$ LO cross section for 12D



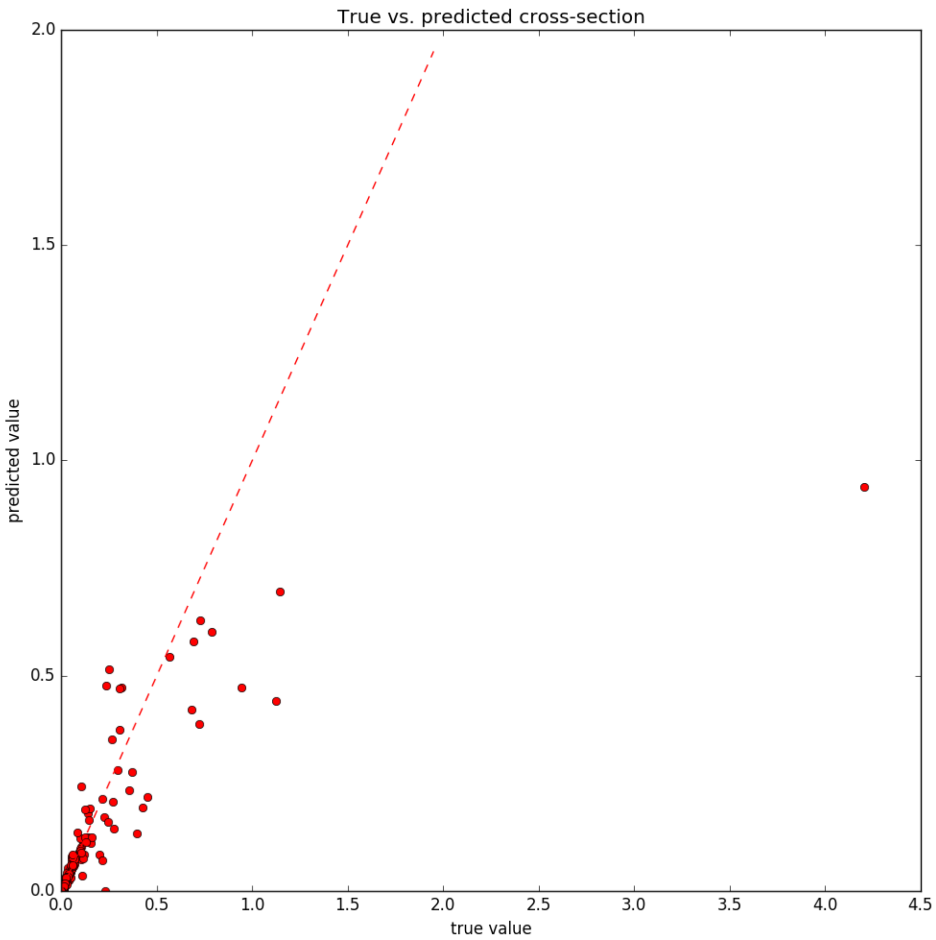
Neutralino-Neutralino pair production

$\chi_1^0 \chi_1^0$ NLO cross section for 18D

Mape: 3 %

Max. error > 100 %

#points: 870‘000



Interlude II: Cut off

The data generation so far was only restricted by the ranges of the MSSM parameters

→ cross sections as low as $10^{-10}\,pb$ were part of the results!

But since the integrated luminosity $L_{int} = 3000\,fb^{-1}$ and

$$L_{int} \cdot \sigma = \#events,$$

we can determine a threshold for σ by fixing a minimum number of events in the lifetime of the LHC.

→ Training threshold (#events=1): $\sigma_{thr,1} = \frac{fb}{3000} = \frac{1}{3 \cdot 10^6} pb \approx 3.3 \cdot 10^{-7} pb$

→ Validation threshold (#events=10): $\sigma_{thr,2} \approx 3.3 \cdot 10^{-6} pb$

Interlude III: Classification

→ Training threshold (#events=1): $\sigma_{thr,1} = \frac{fb}{3000} = \frac{1}{3 \cdot 10^6} pb \approx 3.3 \cdot 10^{-7} pb$

Extremely conservative cut off!

→ every point with a lower cross section is **not interesting**

Thus, we have **two classes**: 1) Interesting (label 1) 2) Not interesting (label 0).

This is an **easy classification problem**!

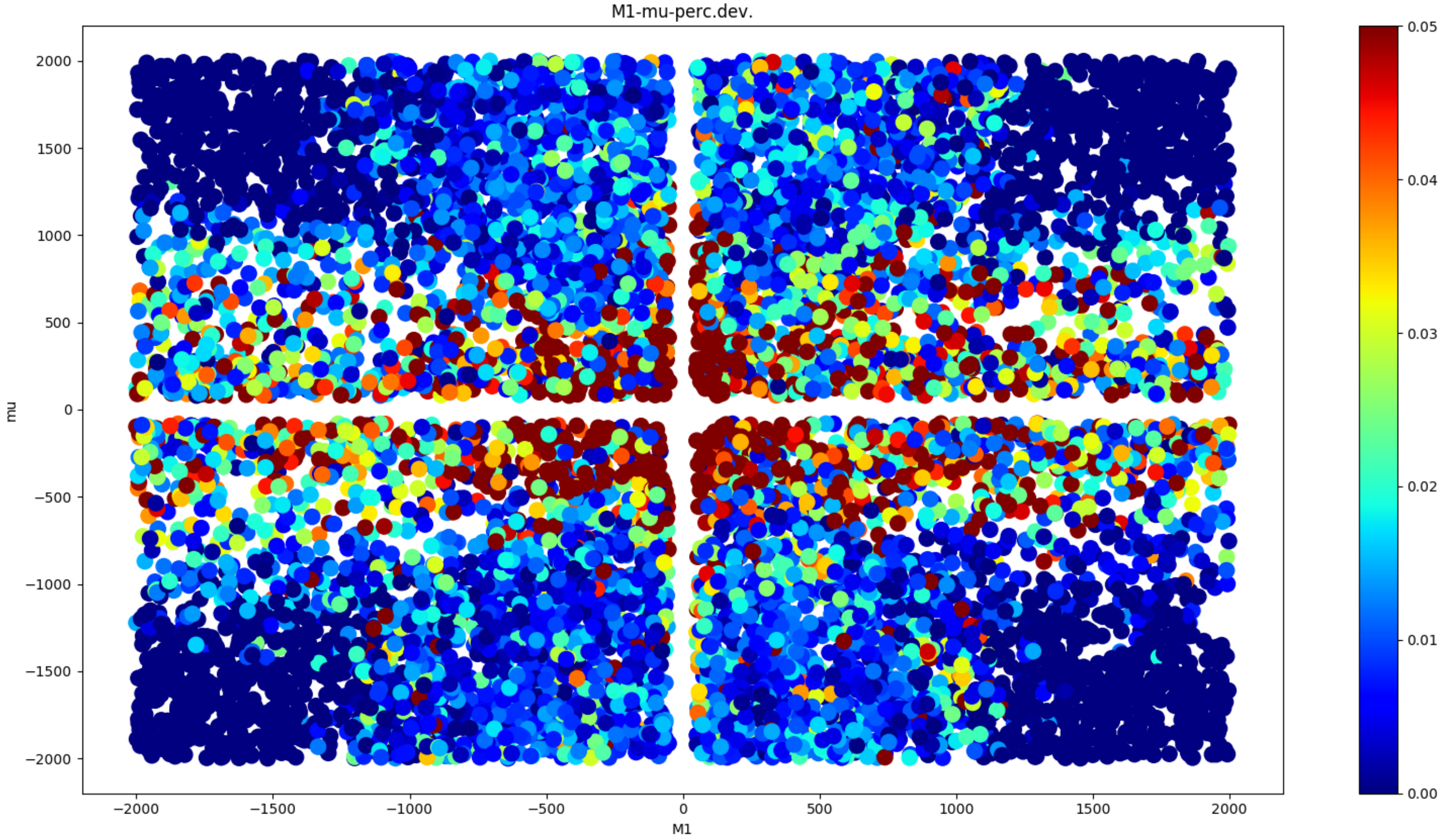
The NN needs to predict either a 0 or a 1 for a given set of MSSM parameters.

→ Accuracy is close to 100% in all tested 19D cases

→ The data generation script now is **more intelligent**:
By providing the **trained weights**, it incorporates the NN predictions and only generates **interesting data**.

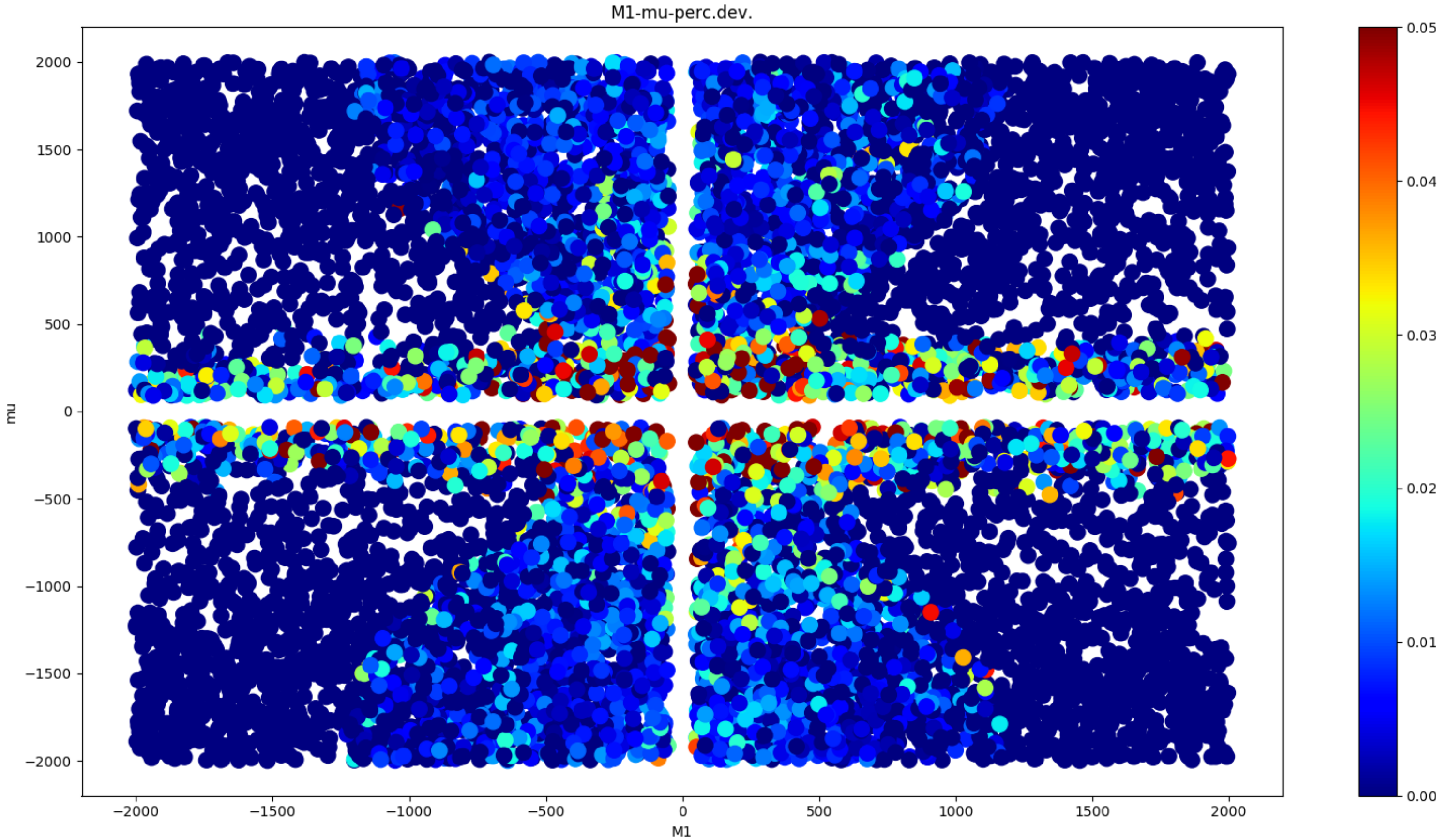
...back to 3D

Mape: 2.3 %, max. error: 307 %, 31'075 points



3D – oversampling problematic regions

Mape: 0.8 %, max. error: 37 %, 237‘000 points



→ Active Learning

Modifying the mapping

Idea: **Train on SPheno output** instead of MSSM parameters:

- Back to 19 dimensional parameter space but collecting the relevant SPheno output:
  - Neutralino/Chargino mass(es)
  - Three squark masses
  - 4 or 8 mixing matrix entries
- No disadvantage: most people interested in the cross-section run SPheno anyway.

→ Number of dimensions reduced from 19 to 6 - 11

→ NN save themselves the conversion step SPheno performs

→ **Better results!**

Intermediate Results

Classification accuracy			
Pair	Training validation	1 event validation	10 event validation
$\chi_1^0\chi_1^0$	99.877 %	99.667 %	99.996 %
$\chi_1^0\chi_1^+$	99.993 %	99.51 %	100 %
$\chi_1^0\chi_2^0$	99.66 %	98.7%	99.97 %

Regression mapes and #points			
Pair	1 event validation	10 event validation	#points
$\chi_1^0\chi_1^+$	1.57 %	1.41 %	196‘000
$\chi_1^0\chi_2^0$	2.48 %	1.6 %	169‘000
$\chi_1^0\chi_1^0$	1.48 %	1.38 %	310‘000
$\chi_1^0\chi_1^0$	1.55 %	1.28 %	447‘000
$\chi_1^0\chi_1^0$	1.27 %	1.1 %	607‘000

What happened next…

- For the first 1-2 months, the **19-dimensional $\chi_1^0\chi_1^0$-case** with SPheno output as neural network input was investigated and then the time-line splits.

At the end, there were **three different approaches** to solving the LO problem:

~~1. A **direct approach** mapping the pMSSM-19 parameters onto the cross-section~~

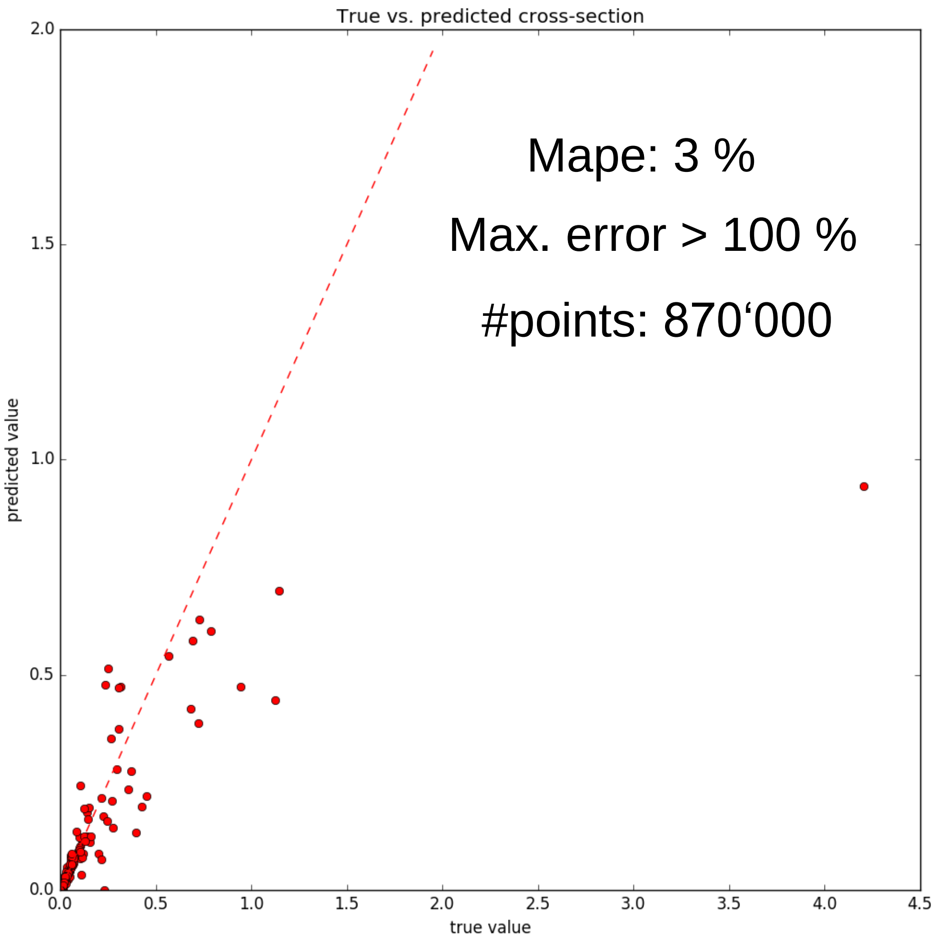
2. A **direct approach** mapping the SPheno output onto the cross-section

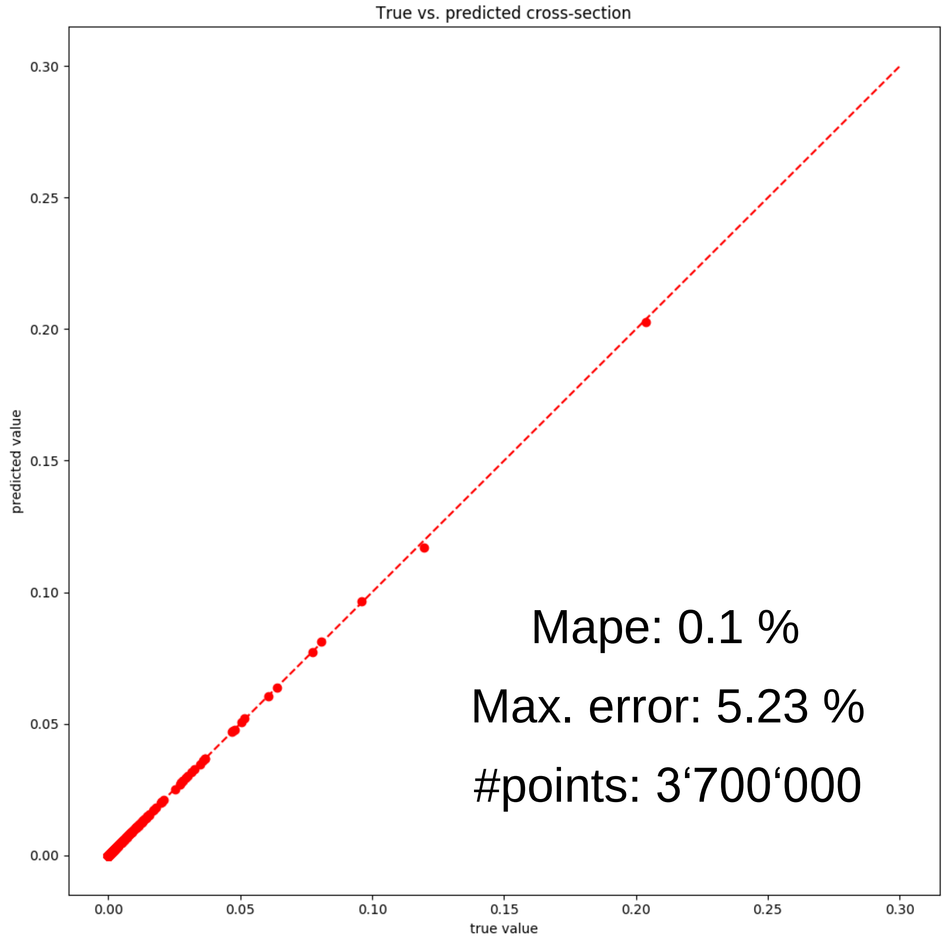
3. A **decomposing approach** that turns the cross-section into a sum of coefficients depending on the electroweakino and squark masses times coupling terms.

After seeing the second direct approach work much better than the first one, the first one was dismissed!

For the rest of the time, 2. and 3. were investigated in parallel.

Regression on $\chi_1^0\chi_1^0$ at LO then and now





K factor predictions

Pair	Test MAPE	Test Max. Errors	# training samples	# errors > 5%
$\chi_i^+ \chi_i^-$	0.078%	2.9%	$9 \cdot 10^4$	0
$\chi_i^+ \chi_j^-$	0.12%	3.1%	$9 \cdot 10^4$	0
$\chi_i^0 \chi_j^+$	0.12%	5.9%	$9 \cdot 10^4$	1
$\chi_i^0 \chi_i^0$	0.32%	9.6%	$9 \cdot 10^4$	12
$\chi_i^0 \chi_j^0$	0.18%	6.49%	$2.9 \cdot 10^5$	3

Architectural recommendations from experience

(for regression and classification problems in physics)

- Lots of data
- Adequate preprocessing (z-score normalization, log, min-max), …
- 3-4 or 8 (only with selu) hidden layers
- (Several) hundred neurons per layer
- Up-to-date optimizer (e.g. Adam/Nadam)
- Activation functions with appropriate initialization of weights
- Learning Rate Scheduling and Early Stopping
- Regularization if overfitting occurs (L1/L2 weight regularization, Dropout, …)

Decomposing LO I

It is possible to write the cross-sections as a sum:

$$\sigma = \sum_{i,j} A_{ij} C_i C_j$$

Very general approach: also suited for the NMSSM and categories of pairs!

The coupling terms C_i and C_j depend only on mixing matrix elements and constants.

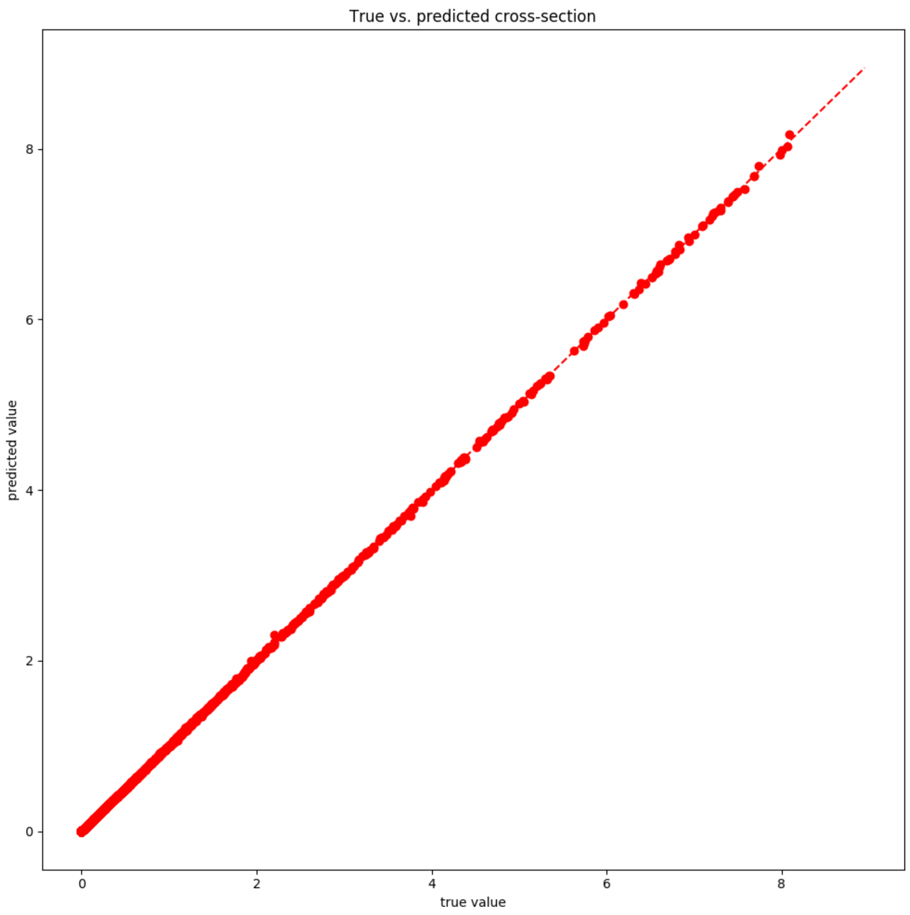
The coefficients A_{ij} depend on the electroweakino and squark masses.

→ Keep the exact dependence on mixing matrix elements and perform the regression on A_{ij}.

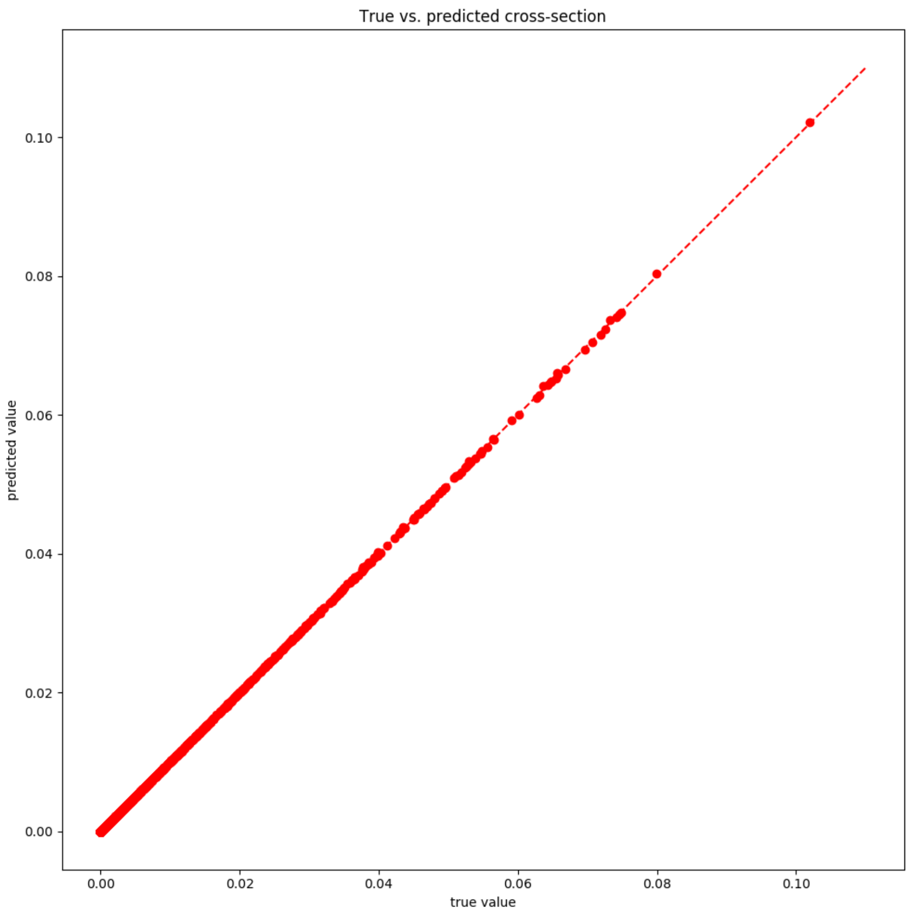
→ Generate data by choosing mixing matrix elements in Prospino input templates such that $C_i, C_j = \{-1,0,1\}$ and the equations are solved.

→ Train a neural network for each coefficient and then plug everything together for the full prediction.

Decomposing LO II: Charginos

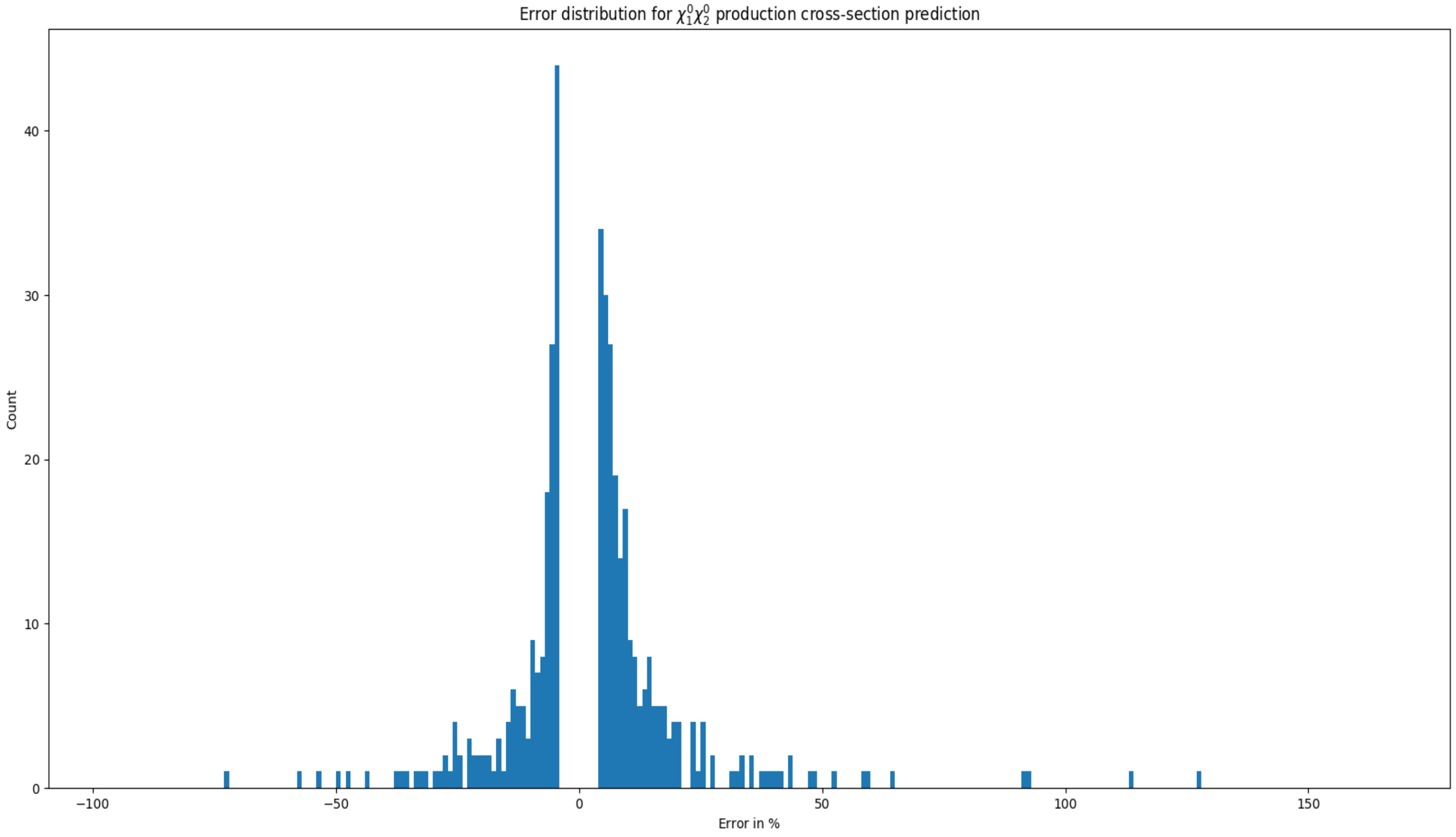


Identical Charginos

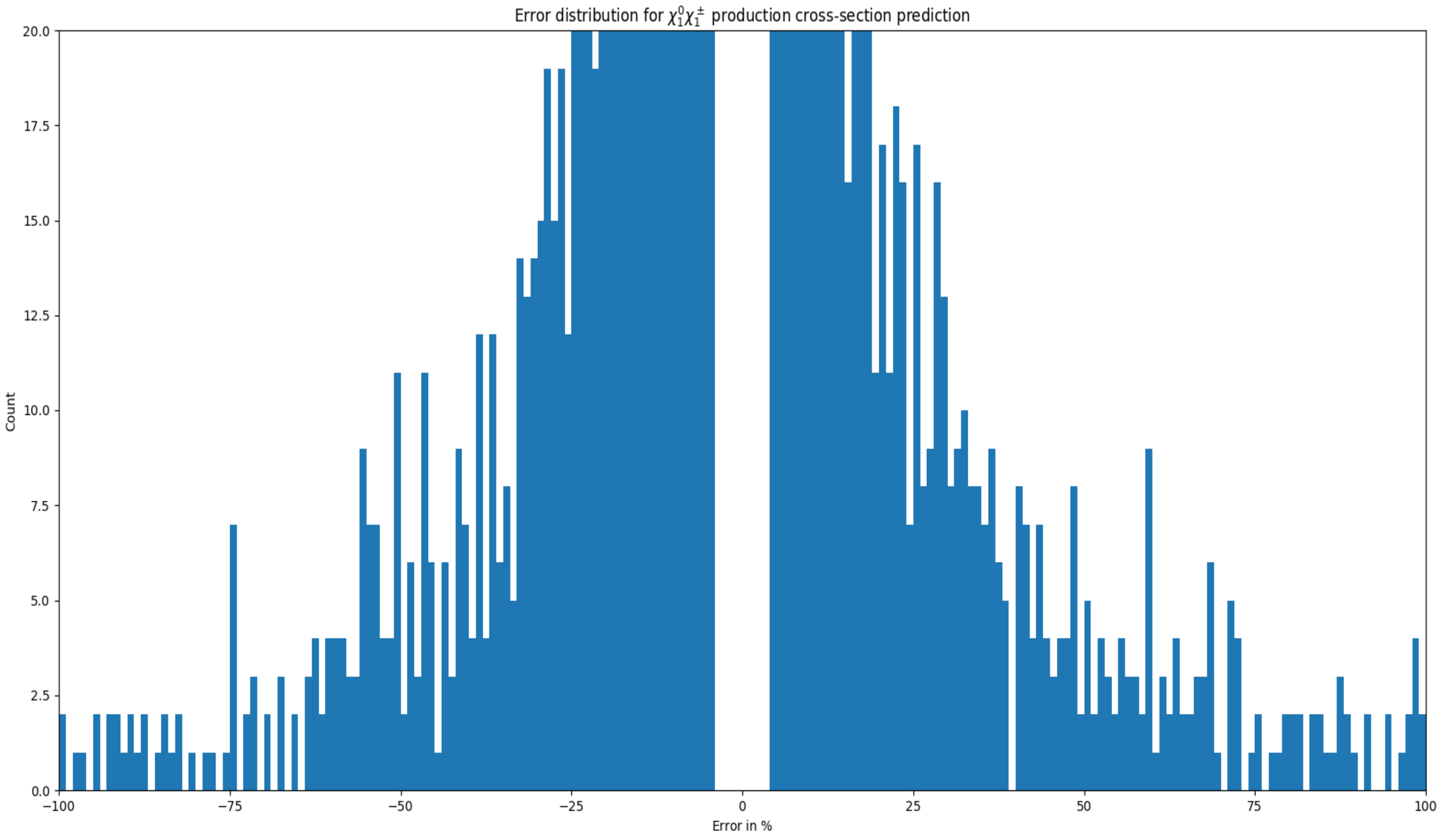


Different Charginos

Decomposing LO III: Neutralinos



Decomposing LO IV: Neutralino-Chargino



Decomposing LO V: Problem

Pair	Test MAPE	Test Max. Errors
$\chi_i^+\chi_i^-$	0.35%	4.7%
$\chi_i^+\chi_j^-$	0.1%	1.36%
$\chi_i^0\chi_j^+$	2.9%	8668%
$\chi_i^0\chi_j^0$	0.9%	166.4%

Question: How can such huge errors occur for a combination of extremely precise regressors?

Simplified example: Assume $\sigma = A \cdot C_1 + \text{B} \cdot C_2 + \text{C} \cdot C_3$.

Let $C_1 = -C_3 = 1$ and $C_2 = 10^{-5}$. We now predict the coefficients whose true values are $A = B = C = 1$ with deviations $\Delta A = 1\%, \Delta B = -4\%, \Delta C = 3\%$.

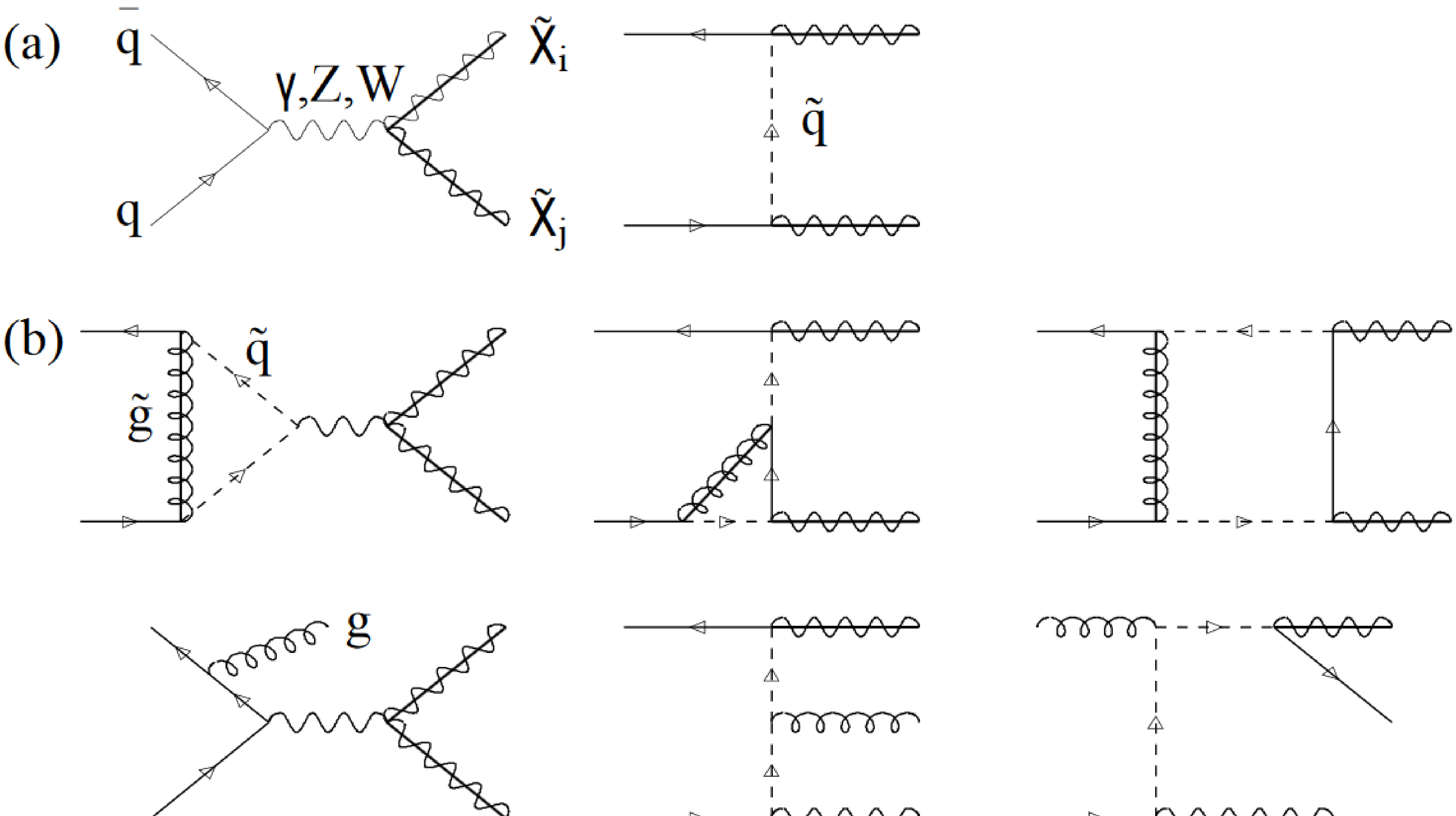
→ This leads to a negative prediction $\sigma = -0.0199904$!

Summary and Conclusion

- We can predict the chargino pair production cross-section such that we satisfy the mission requirements with the decomposing approach at LO and the direct approach for the K factor.
- We can predict all the other pair production cross-sections with the direct approach given sufficient data such that we satisfy the mission requirements.
- Should we have access to suitable resources, the direct approach should be combined with active learning. This is what we are doing right now.
- We now create the predictive tool for all electroweakino pairs, including scale dependence and for several pdfs, try to fix decomposing approach. The tool will be made available via PhenoAI (see Bob Stienen‘s talk)
- Still a debate: how do we treat and present the maximum errors? What are mission requirements that make sense? How to deal with uncertainties?

Thank you for your attention!

Feynman Diagrams



LO decomposition: Neutralino-Chargino

$$C_1 = \sqrt{2} V_{i1} N_{j2} - V_{i2} N_{j4},$$
$$C_2 = \sqrt{2} U_{i1} N_{j2} + U_{i2} N_{j3},$$
$$C_3 = V_{i1} \left(N_{j1} + 3 \frac{c_w}{s_w} N_{j2} \right),$$
$$C_4 = U_{i1} \left(N_{j1} - 3 \frac{c_w}{s_w} N_{j2} \right).$$

C_1 and C_2 come from the s-channel diagram with a W-exchange and correspond to the left- and right-handed component of the chargino-neutralino-gauge boson coupling.

C_3 and C_4 come from the t- and u-channel diagram with a squark-exchange and correspond to the product of the quark-squark-chargino/neutralino couplings.