Brief DeepXS Manual

DeepXS is a tool that estimates next-to-leading order cross sections for the production of charginos and neutralinos at the LHC in the pMSSM-19 using a combination of trained neural networks. To use DeepXS a user should provide an SLHA file or an array containing the necessary information in a format specified below. Additionally, one can configure DeepXS via command line arguments given below. The general syntax can be summarized as follows:

python main.py [-pairs *pairs*] [-return LO|NLO|both] [-stream 0|1|2] [-array 0|1] -fn *filename*

Explanation of arguments:

```
-pairs (default: 0)

Specifies the cross section to predict. The value is an integer
between 0 and 14. The corresponding configurations are as follows:
```

0: all
1: $\tilde{\chi}_1^+/\tilde{\chi}_1^-$
2: $\tilde{\chi}_2^0/\tilde{\chi}_2^0$
3: $\tilde{\chi}_2^0/\tilde{\chi}_1^+$
4: $\tilde{\chi}_2^0/\tilde{\chi}_1^-$

```
5: 1 & 2
6: 1 & 3
7: 1 & 4
8: 2 & 3
9: 2 & 4
10: 3 & 4
11: 1, 2 & 3
12: 1, 2 & 4
13: 1, 3 & 4
14: 2, 3 & 4

-return (default: both)

LO: only returns the cross section at leading order.

NLO: only returns the cross section at next-to-leading order.

both: returns the LO and NLO cross section

-stream (default: 0)

0: no streaming

1: input via an SLHA file provided with -fn.

2: calculating the cross section for every file in ./SLHA_dump/

-array (default: 0)

0: no array

1: Taking the input values from a file specified with -fn

-fn (no default)

Requires you to specify a filename: either an SLHA file or .txt/.csv
```

Warning: When you provide the input via arrays the files must contain the input in a certain shape. Moreover, if you use the array option, you can currently only predict one pair at a time.

The general shape of the array input is that one point in the parameter space is given per row while the components are separated by ‘ ‘. For the different pairs the order of the components must be as follows:

$$\boldsymbol{\tilde{\chi}_1^+/\tilde{\chi}_1^-}$$

$$U_{11}, U_{21}, V_{11}, V_{21}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{q}_1}, m_{\tilde{q}_2}, \dots, m_{\tilde{q}_8}, m_{\tilde{g}}$$

$$\boldsymbol{\tilde{\chi}_2^0/\tilde{\chi}_2^0}$$

$$U_{11}, U_{21}, V_{11}, V_{21}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{q}_1}, m_{\tilde{q}_2}, \dots, m_{\tilde{q}_8}, m_{\tilde{g}}$$

$$\boldsymbol{\tilde{\chi}_2^0/\tilde{\chi}_1^\pm}$$

$$U_{11}, U_{21}, V_{11}, V_{21}, N_{21}, N_{22}, N_{23}, N_{24}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{\chi}_2^0}, m_{\tilde{q}_1}, m_{\tilde{q}_2}, \dots, m_{\tilde{q}_8}, m_{\tilde{g}}$$

The squark masses $m_{\tilde{q}_1}, m_{\tilde{q}_2}, \dots, m_{\tilde{q}_8}$ refer to the PDG codes 1000001, 1000002, 2000001, 2000002, 1000003, 2000003, 1000004 and 2000004 in ascending order.

In case you are interested in a decoupled spectrum, keep in mind that the neural networks we have trained have only seen masses of about 5 TeV and neural network extrapolation is currently not reliable, and you might encounter unphysical predictions such as negative K-factors etc.