xsec: the cross section evaluation code

"Bringing the power of Gaussian processes to global fits"

Jeriek Van den Abeele

in collaboration with A. Buckley, I.A.V. Holm, A. Kvellestad, A. Raklev, P. Scott and J.V. Sparre

Nordic Winter School on Particle Physics and Cosmology
Skeikampen – January 4, 2019

UiO : University of Oslo

Have you met SUSY?

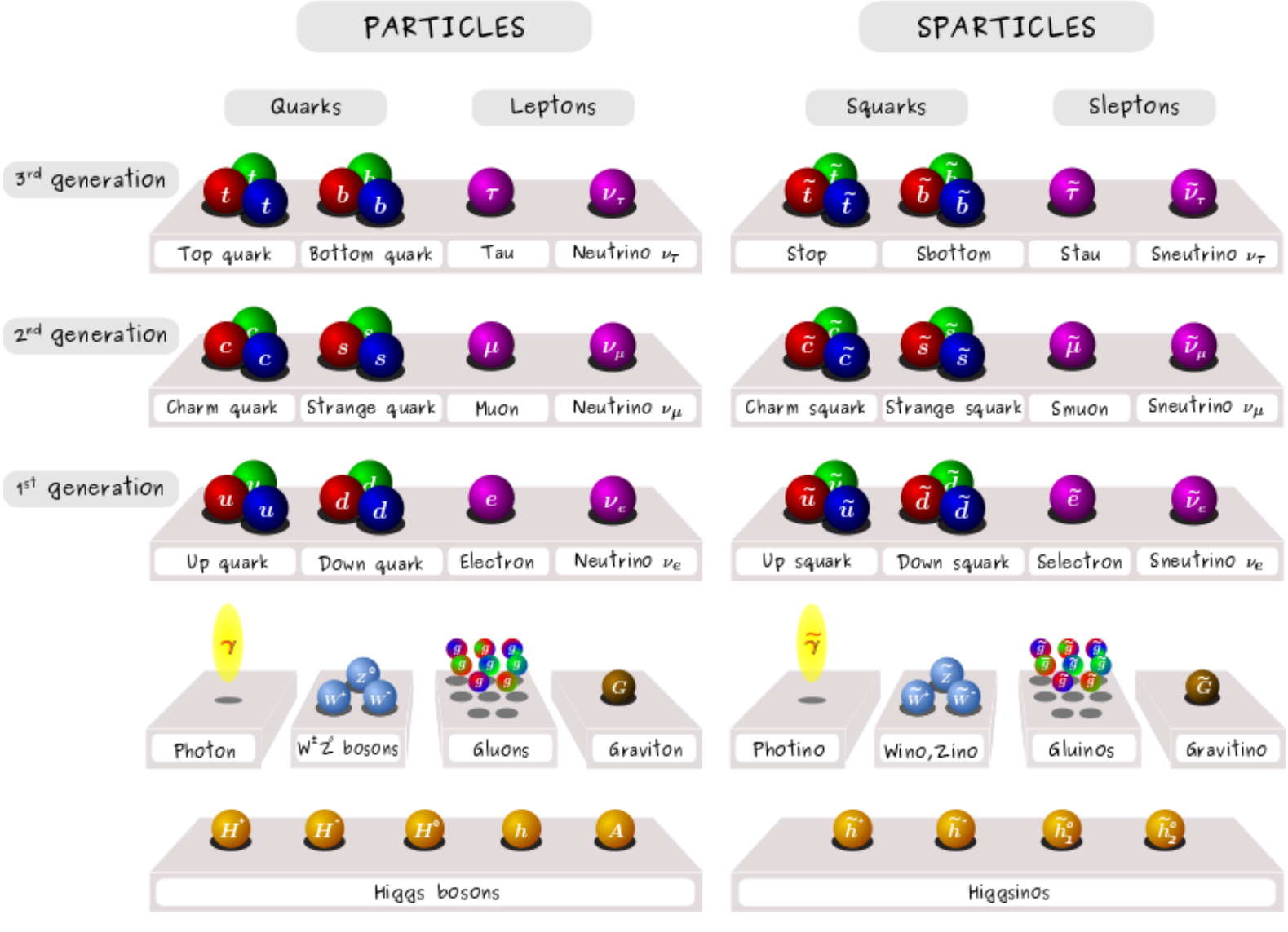


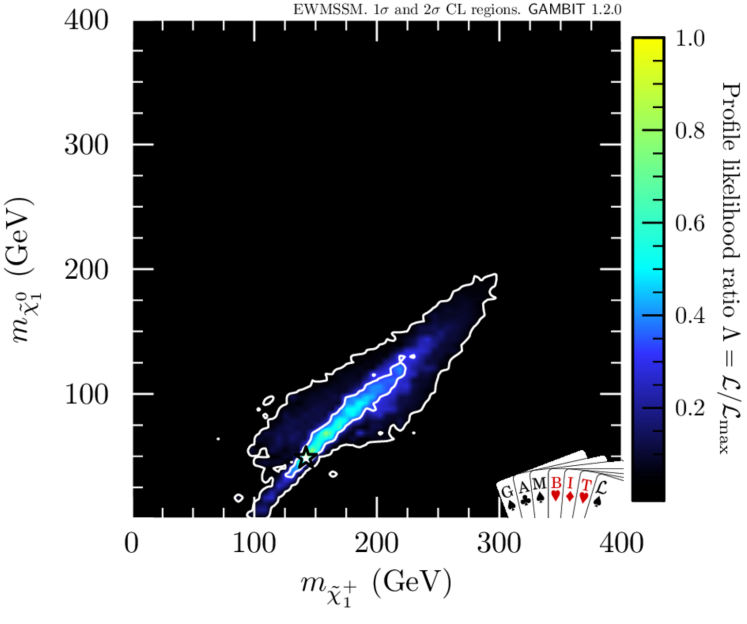
Image: Claire David (CERN)

Global fits and the need for speed

Global fits consistently compare theories to all experimental results

- High-dimensional parameter space, e.g. MSSM-24
- Quick calculation of **next-to-leading order cross sections** needed!
- Existing tools have drawbacks

  (Prospino: slow, NLL-fast: limited)

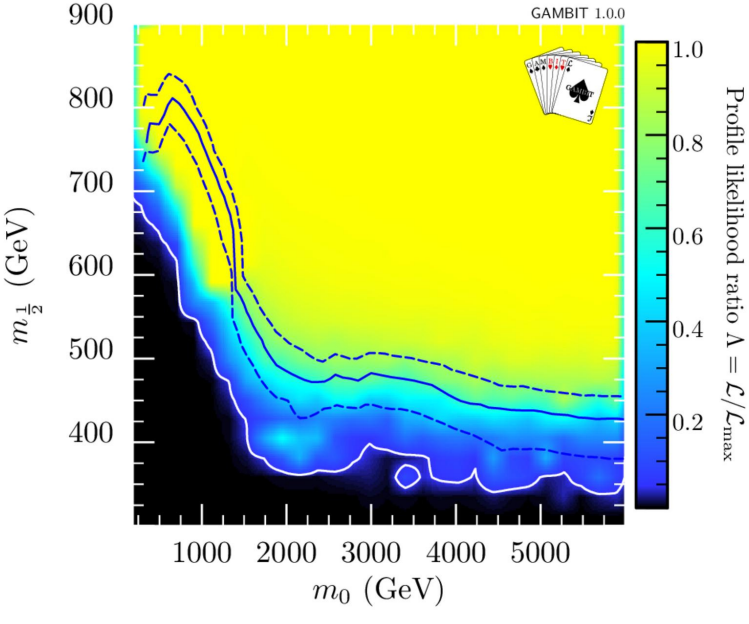


Global fits and the need for speed

Global fits consistently compare theories to all experimental results

- High-dimensional parameter space, e.g. MSSM-24
- Quick calculation of **next-to-leading order cross sections** needed!
- Existing tools have drawbacks

  (Prospino: slow, NLL-fast: limited)



Gaussian Processes 101 (non-parametric regression)

Assume: the data is a sample from a multivariate **Gaussian distribution**.

The covariance matrix controls smoothness.

Assume it is given by a **kernel function**, like

$$k(x_1, x_2) = \exp\left(-\frac{|x_2 - x_1|^2}{2l^2}\right)$$

Bayesian approach to estimate $y_* = f(x_*)$:

prior over functions

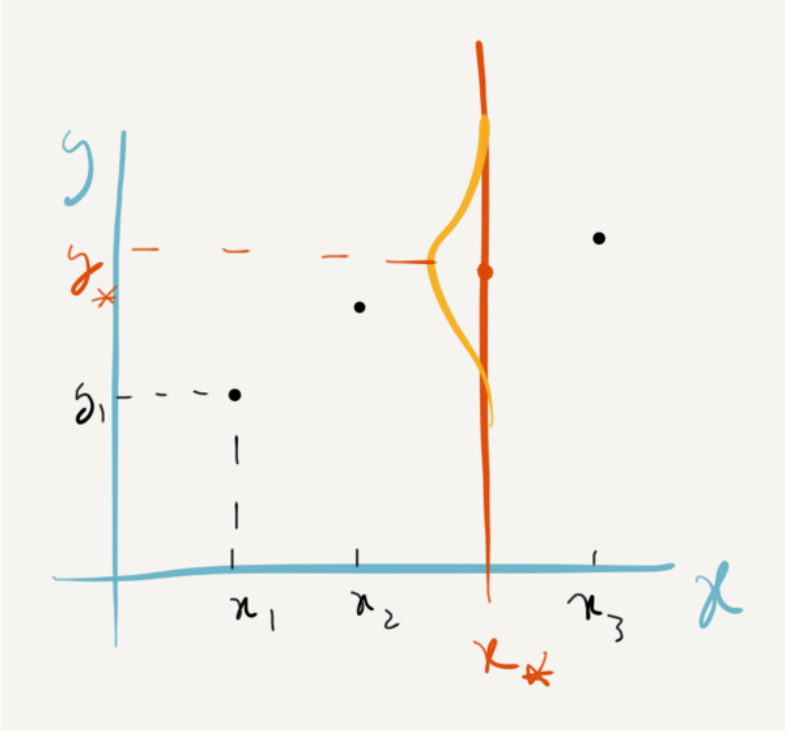
$$\begin{pmatrix} \mathbf{y} \\ y_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix}\right)$$

$$\begin{aligned} K_{ij} &= k(x_i, x_j) \\ K_{*i} &= k(x_*, x_i) \\ K_{**} &= k(x_*, x_*) \end{aligned}$$

data

posterior over functions

$$y_* | \mathbf{y} \sim \mathcal{N}(\underbrace{K_* K^{-1} \mathbf{y}}_{\text{mean}}, \underbrace{K_{**} - K_* K^{-1} K_*^T}_{\text{covariance}})$$



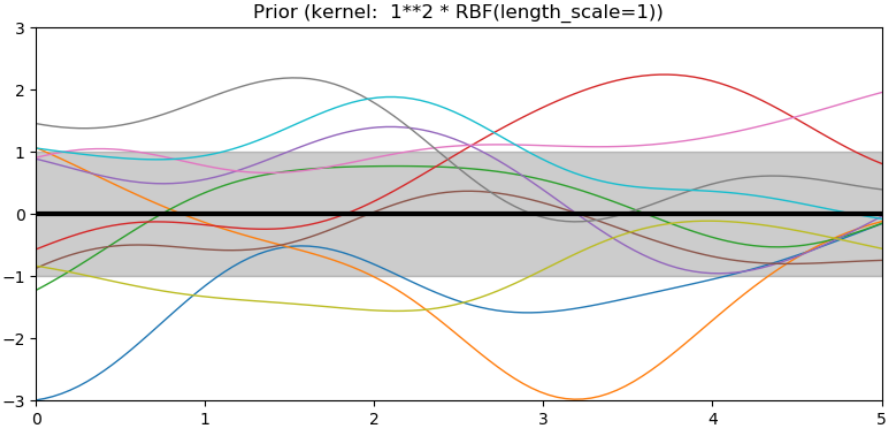
Gaussian Processes 101 (non-parametric regression)

Assume: the data is a sample from a multivariate **Gaussian distribution**.

The covariance matrix controls smoothness.

Assume it is given by a **kernel function**, like

$$k(x_1, x_2) = \exp\left(-\frac{|x_2 - x_1|^2}{2l^2}\right)$$



Bayesian approach to estimate $y_* = f(x_*)$:

prior over functions

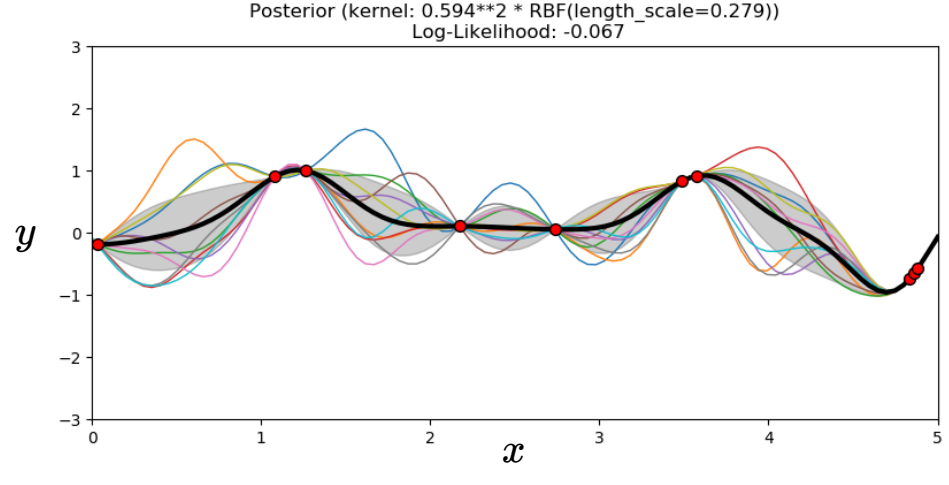
$$\begin{pmatrix} \mathbf{y} \\ y_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix}\right)$$

$$K_{ij} = k(x_i, x_j)$$
$$K_{*i} = k(x_*, x_i)$$
$$K_{**} = k(x_*, x_*)$$

data

posterior over functions

$$y_* | \mathbf{y} \sim \mathcal{N}(\underbrace{K_* K^{-1} \mathbf{y}}_{\text{mean}},\ \underbrace{K_{**} - K_* K^{-1} K_*^T}_{\text{covariance}})$$



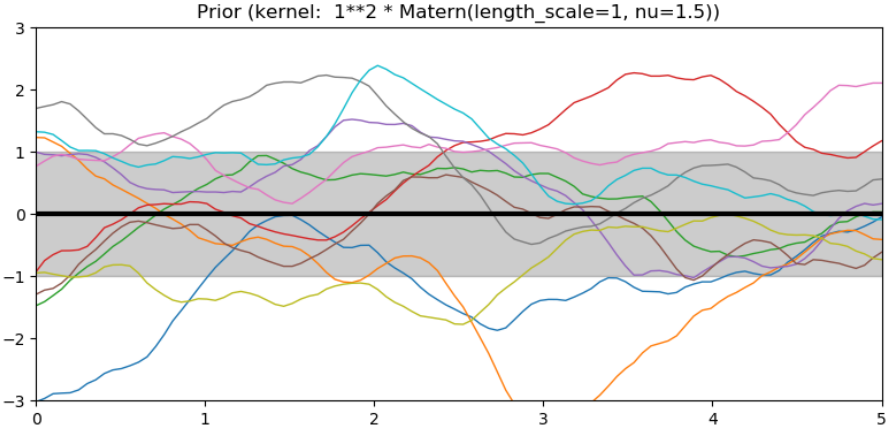
Gaussian Processes 101 (non-parametric regression)

Assume: the data is a sample from a multivariate **Gaussian distribution**.

The covariance matrix controls smoothness.

Assume it is given by a **kernel function**, like

$$k(x_1, x_2) = \exp\left(-\frac{|x_2 - x_1|^2}{2l^2}\right)$$



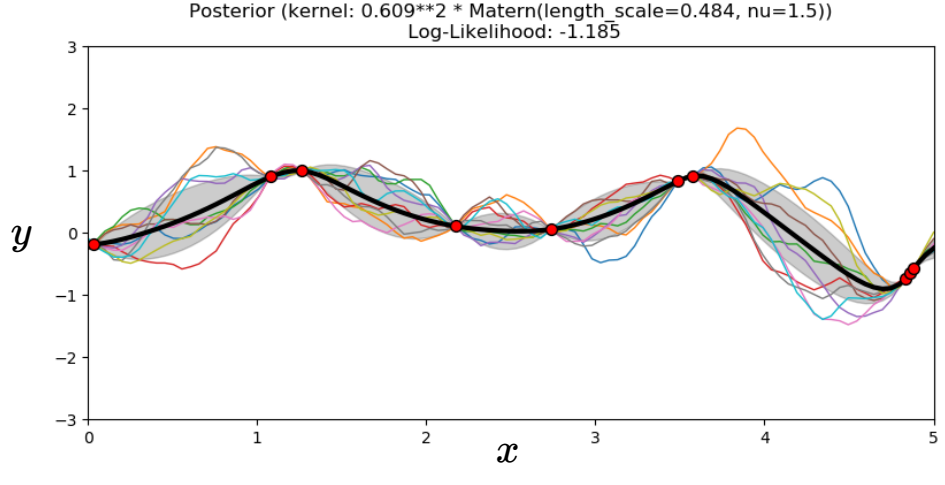
Bayesian approach to estimate $y_* = f(x_*)$:

prior over functions

$$\begin{pmatrix} \mathbf{y} \\ y_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix}\right)$$

$$K_{ij} = k(x_i, x_j)$$
$$K_{*i} = k(x_*, x_i)$$
$$K_{**} = k(x_*, x_*)$$

↓ ***data***



posterior over functions

$$y_* | \mathbf{y} \sim \mathcal{N}(\underbrace{K_* K^{-1} \mathbf{y}}_{\text{mean}},\ \underbrace{K_{**} - K_* K^{-1} K_*^T}_{\text{covariance}})$$

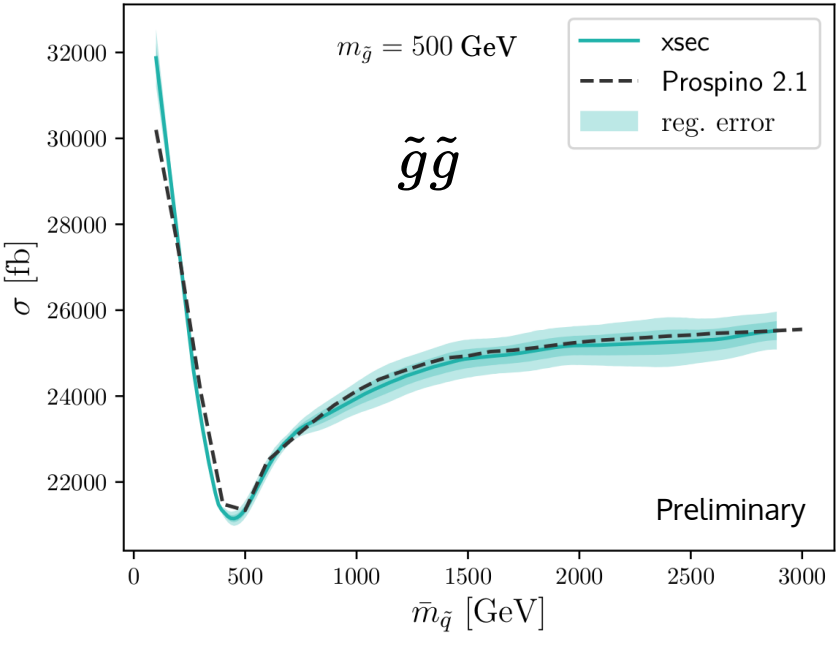
xsec: the cross section evaluation code

Fast estimate of SUSY (strong) production **cross sections**:

$$\tilde{g}\tilde{g},\ \tilde{g}\tilde{q}_i,\ \tilde{q}_i\tilde{q}_j^{(*)},\ \tilde{b}_i\tilde{b}_i^*,\ \tilde{t}_i\tilde{t}_i^*$$

Parameters: $m_{\tilde{g}},\ m_{\tilde{q}_i},\ \bar{m}_{\tilde{q}},\ m_{\tilde{b}_i},\ m_{\tilde{t}_i},\ \theta_b,\ \theta_t$

- Computation of **uncertainties** from regression, scale, PDF and α_s
- Using pre-trained, distributed Gaussian processes
- Simple interface in Python ❤
- Soon public on GitHub!



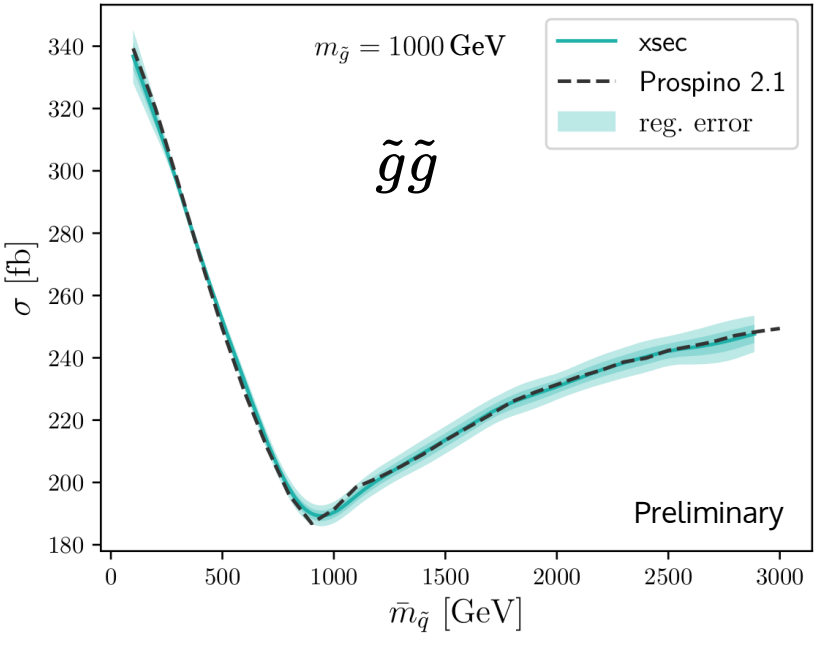
xsec: the cross section evaluation code

Fast estimate of SUSY (strong) production **cross sections**:

$$\tilde{g}\tilde{g},\ \tilde{g}\tilde{q}_i,\ \tilde{q}_i\tilde{q}_j^{(*)},\ \tilde{b}_i\tilde{b}_i^*,\ \tilde{t}_i\tilde{t}_i^*$$

Parameters: $m_{\tilde{g}},\ m_{\tilde{q}_i},\ \bar{m}_{\tilde{q}},\ m_{\tilde{b}_i},\ m_{\tilde{t}_i},\ \theta_b,\ \theta_t$

- Computation of **uncertainties** from regression, scale, PDF and α_s
- Using pre-trained, distributed Gaussian processes
- Simple interface in Python ❤
- Soon public on GitHub!



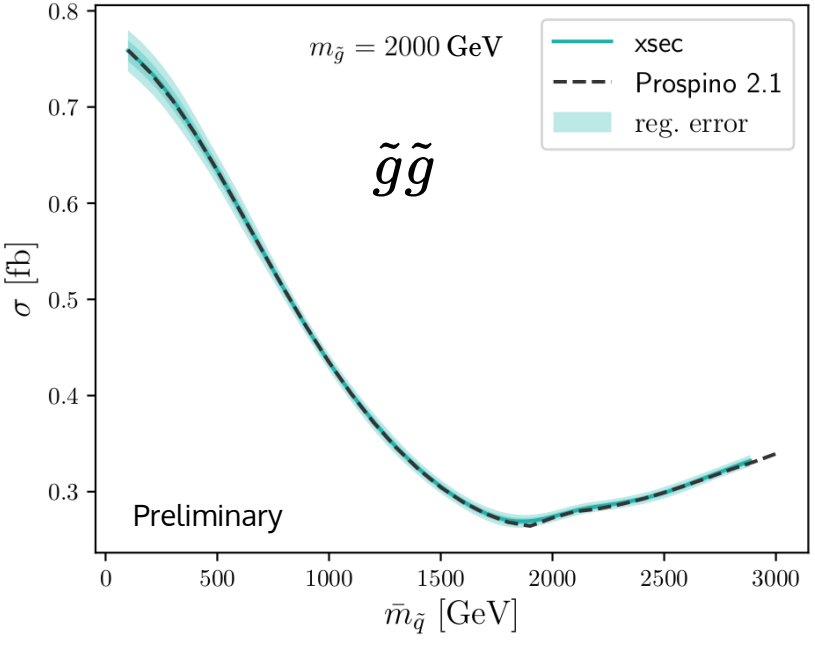
xsec: the cross section evaluation code

Fast estimate of SUSY (strong) production **cross sections**:

$$\tilde{g}\tilde{g},\ \tilde{g}\tilde{q}_i,\ \tilde{q}_i\tilde{q}_j^{(*)},\ \tilde{b}_i\tilde{b}_i^*,\ \tilde{t}_i\tilde{t}_i^*$$

Parameters: $m_{\tilde{g}},\ m_{\tilde{q}_i},\ \bar{m}_{\tilde{q}},\ m_{\tilde{b}_i},\ m_{\tilde{t}_i},\ \theta_b,\ \theta_t$

- Computation of **uncertainties** from regression, scale, PDF and α_s
- Using pre-trained, distributed Gaussian processes
- Simple interface in Python ❤
- Soon public on GitHub!



xsec: the cross section evaluation code

Fast estimate of SUSY (strong) production **cross sections**:

$$\tilde{g}\tilde{g},\ \tilde{g}\tilde{q}_i,\ \tilde{q}_i\tilde{q}_j^{(*)},\ \tilde{b}_i\tilde{b}_i^*,\ \tilde{t}_i\tilde{t}_i^*$$

Parameters: $m_{\tilde{g}},\ m_{\tilde{q}_i},\ \bar{m}_{\tilde{q}},\ m_{\tilde{b}_i},\ m_{\tilde{t}_i},\ \theta_b,\ \theta_t$

- Computation of **uncertainties** from regression, scale, PDF and α_s
- Using pre-trained, distributed Gaussian processes
- Simple interface in Python ❤
- Soon public on GitHub!