A JAX-based EFT likelihood

Aleks Smolkovič

Jožef Stefan Institute

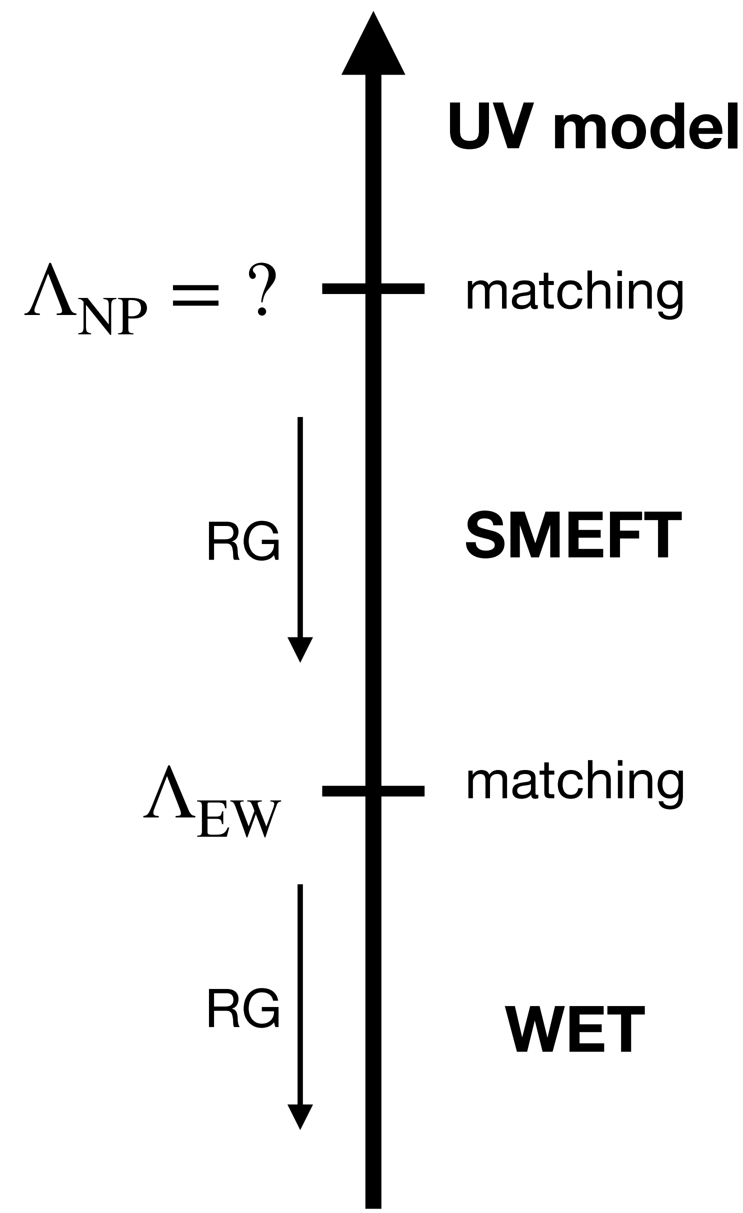
SMEFT

Nucl.Phys.B 268 (1986) 621-653
Phys.Rept. 793 (2019) 1-98
Rev.Mod.Phys. 96 (2024) 1, 015006
JHEP 10 (2010) 085, *JHEP* 10 (2013) 087
JHEP 01 (2014) 035, *JHEP* 04 (2014) 159
JHEP 03 (2018) 016, *JHEP* 10 (2019) 197

- SM fields and symmetries
- Scale separation, linearly realized EWSB
- Higher-dimensional operators encode short-distance NP

$$\mathscr{L} = \mathscr{L}_{\mathrm{SM}} + \sum_{Q} \frac{C_Q}{\Lambda_Q^{[Q]-4}} Q$$

- No preferred BSM model-building direction
- SM works well as a low-energy limit
- Experiments headed towards the precision era
- A global SMEFT likelihood can be ‘recycled’ for reinterpretation in concrete models
  -> in general, no flavor assumption should be made

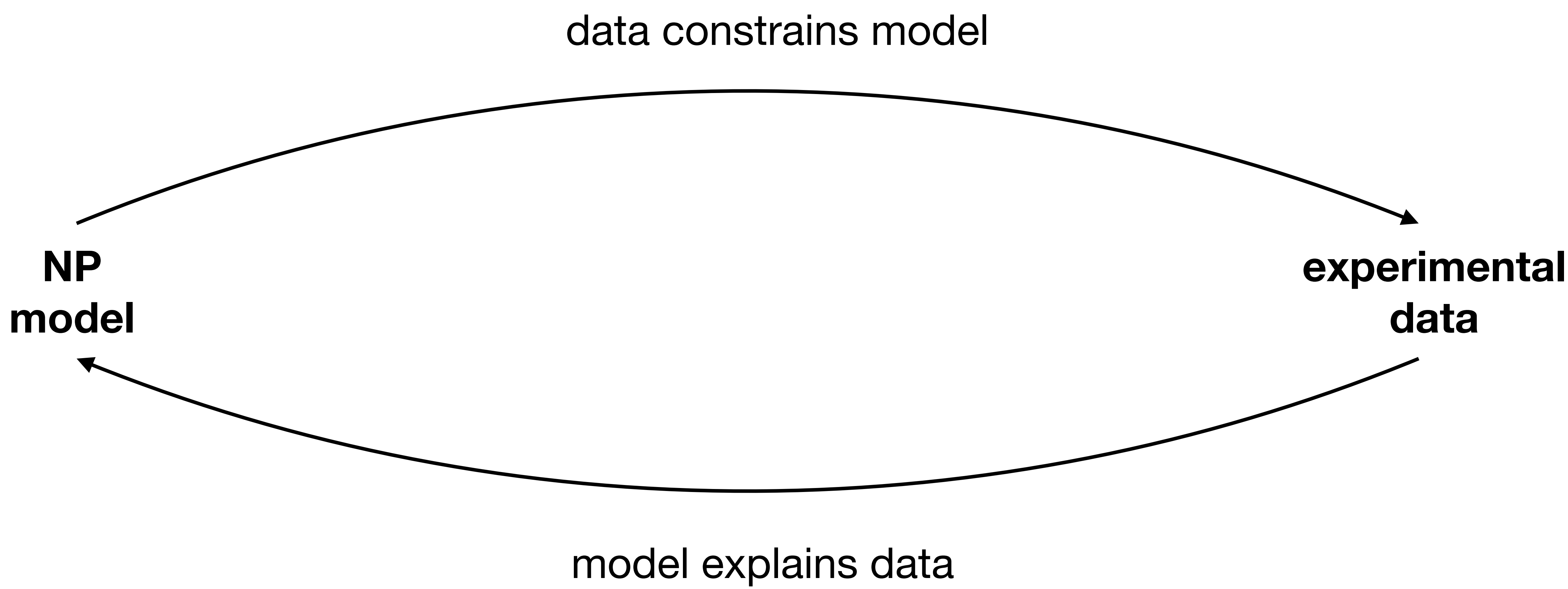


Challenge:

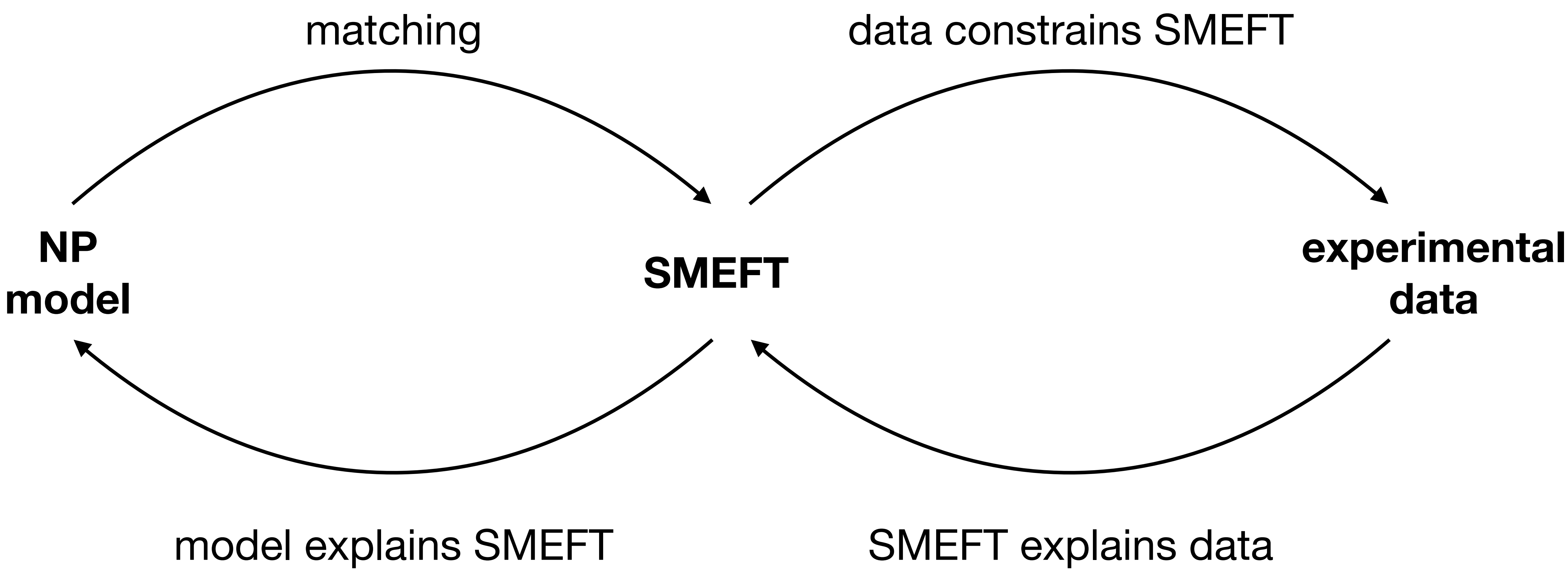
Large number of parameters (2499@dim 6 (dB=dL=0) for 3 gen, flavor!)
SMEFT operators will impact observables from vastly different classes

Efficient BSM phenomenology

Efficient BSM phenomenology

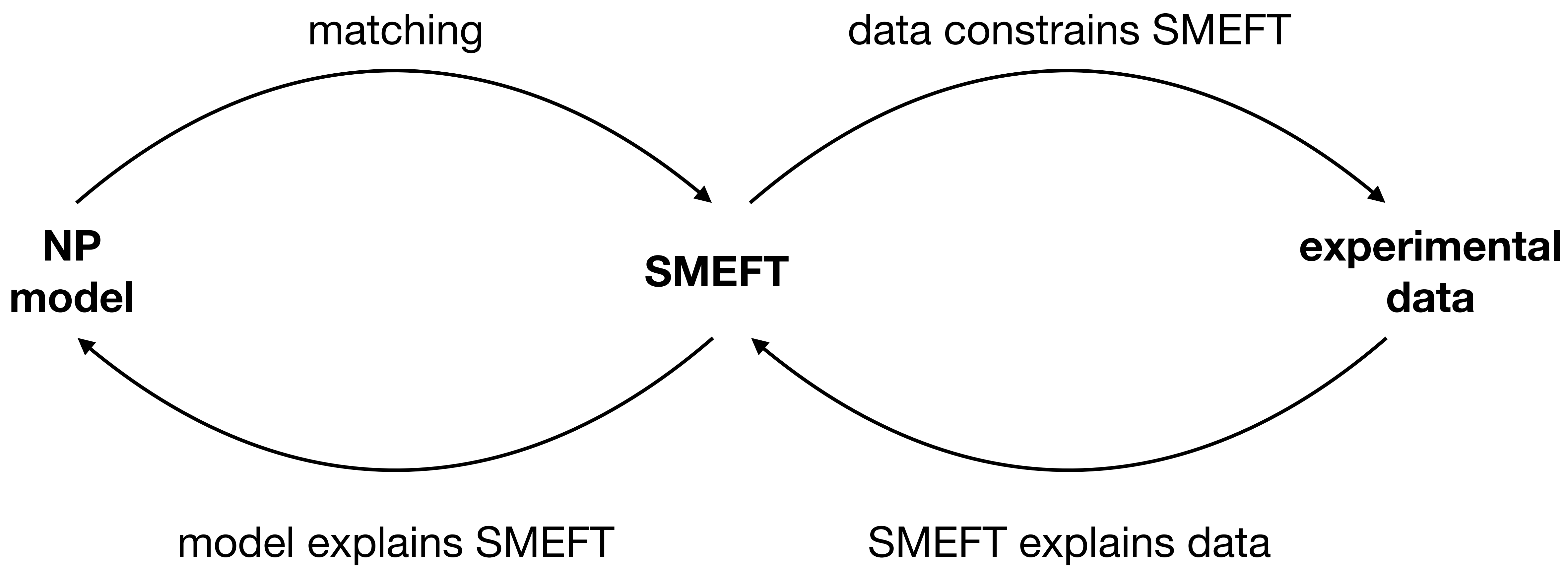


Efficient BSM phenomenology



Efficient BSM phenomenology

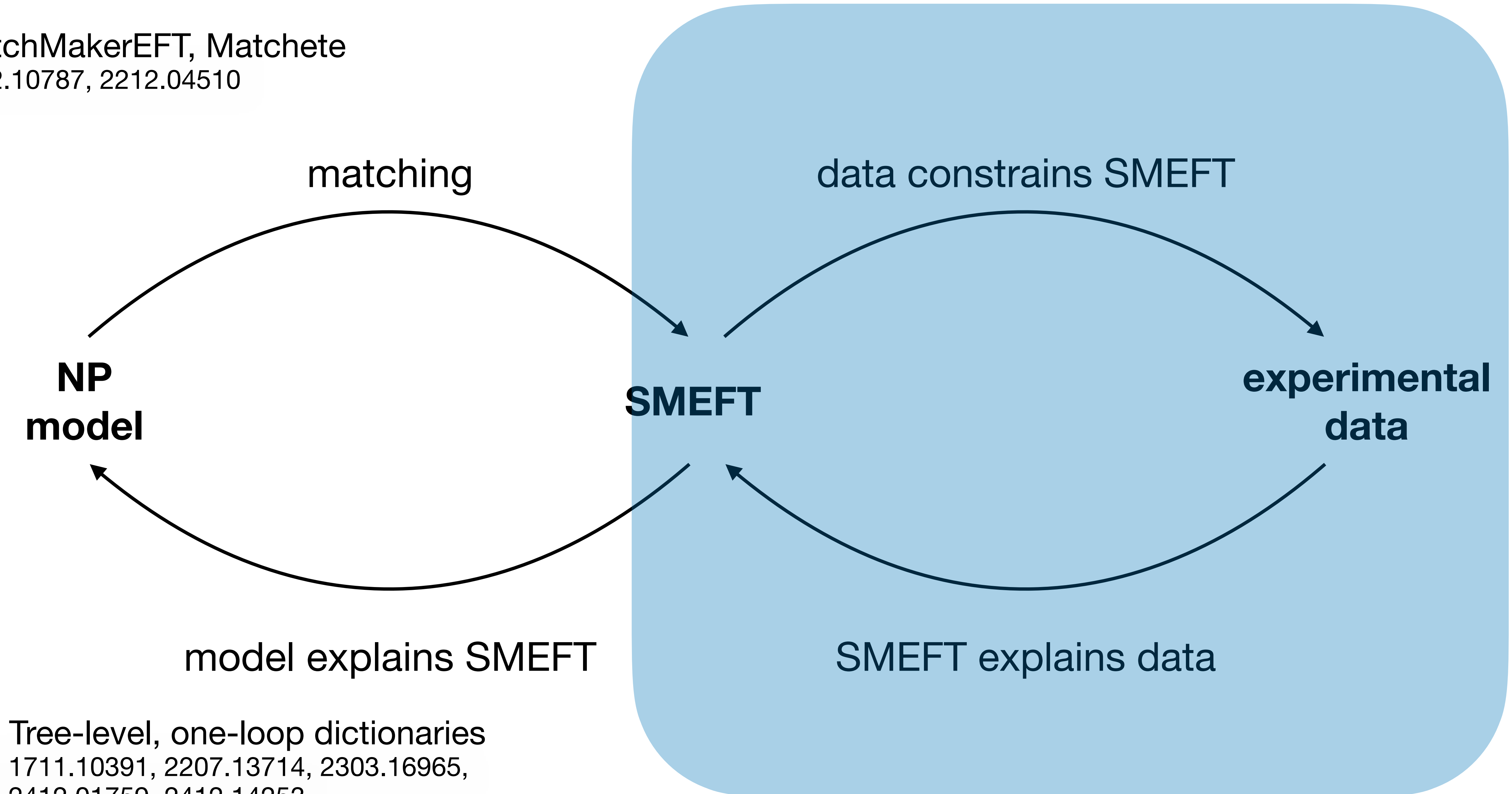
MatchMakerEFT, Matchete
2112.10787, 2212.04510



Tree-level, one-loop dictionaries
1711.10391, 2207.13714, 2303.16965, 2412.01759, 2412.14253

Efficient BSM phenomenology

MatchMakerEFT, Matchete
2112.10787, 2212.04510



Tree-level, one-loop dictionaries
1711.10391, 2207.13714, 2303.16965, 2412.01759, 2412.14253

Global, flavorful SMEFT likelihood

How can we make it as efficient as possible?

Building blocks of a global likelihood

- A diverse set of observables $\overrightarrow{O}$
- State-of-the-art theory predictions $\overrightarrow{O}_{\text{th}}(\overrightarrow{C}, \vec{\theta})$
- Latest experimental measurements, defining $\mathscr{L}_{\text{exp}}$

$\overrightarrow{C}$ - Wilson coefficients

$\vec{\theta}$ - nuisance parameters

Building blocks of a global likelihood

- A diverse set of observables $\vec{O}$
- State-of-the-art theory predictions $\vec{O}_{\rm th}(\vec{C}, \vec{\theta})$
- Latest experimental measurements, defining $\mathscr{L}_{\rm exp}$

$\vec{C}$ - Wilson coefficients
$\vec{\theta}$ - nuisance parameters

Likelihood:

$$\mathscr{L}(\vec{C}, \vec{\theta}) = \prod_i \mathscr{L}^i_{\rm exp}\left(\vec{O}_{\rm exp}, \vec{O}_{\rm th}(\vec{C}, \vec{\theta})\right) \times \mathscr{L}_\theta(\vec{\theta})$$

J. Aebischer, J. Kumar, P. Stangl, D. Straub, *Eur.Phys.J.C* 79 (2019) 6, 509

- Various $\vec{O}$ should be treated consistently

E.g. flavor observables in terms of WET WCs at low-energy scales
various high-energy observables in terms of SMEFT WCs at various scales

Crucially relies on RGE&matching computations
1308.2627, 1310.4838, 1312.2014, 1709.04486, 1908.05295, …

- Obtaining a nuisance-free likelihood $\mathscr{L}(\vec{C})$ for parameter estimation, hypothesis testing:

Marginalization/profiling computationally expensive

With clever approximations we can speed things up

Building blocks of a global likelihood

flavio:
1810.08132

- Theory predictions: flavor, EWPO, Higgs, beta decays, EDMs, DY tails, …
- Large database of latest measurements
- Allows for constructing likelihoods

Building blocks of a global likelihood

flavio:
1810.08132

- Theory predictions: flavor, EWPO, Higgs, beta decays, EDMs, DY tails, …
- Large database of latest measurements
- Allows for constructing likelihoods

wilson + WCxf
1712.05298, 1804.05033

- SMEFT RG evolution, SMEFT to WET matching, WET RG evolution
- WC exchange format, conventions for bases, interfacing

Building blocks of a global likelihood

flavio:

1810.08132

- Theory predictions: flavor, EWPO, Higgs, beta decays, EDMs, DY tails, …
- Large database of latest measurements
- Allows for constructing likelihoods

wilson + WCxf

1712.05298, 1804.05033

- SMEFT RG evolution, SMEFT to WET matching, WET RG evolution
- WC exchange format, conventions for bases, interfacing

smelli - the **SME**FT **li**keli**h**ood

1810.07698, 2012.12211

- Observables and measurements consistently included
- Flexible, uses **flavio** for predictions, **wilson** for RGs
- Approximate nuisance-free likelihood provided

Observables split into two categories:

$$L_{\mathrm{SMEFT}}(\vec{C}) = \prod_{i \in 1.} L_{\exp}\left(\vec{O}_i^{\exp}, \vec{O}_i^{\mathrm{th}}\left(\vec{C}, \vec{\theta}_0\right)\right) \prod_{i \in 2.} \widetilde{L}_{\exp}\left(\vec{O}_i^{\exp}, \vec{O}_i^{\mathrm{th}}\left(\vec{C}, \vec{\theta}_0\right)\right)$$

1. Negligible theory uncertainties (can be non-Gaussian exp. likelihood)
2. Gaussian approximation
   -> covariances computed “once and for all”, but SM th. unc. assumed

$$-2\ln \widetilde{L}_{\exp} = \vec{x}^T (C_{\exp} + C_{\mathrm{th}})^{-1} \vec{x}\,,$$

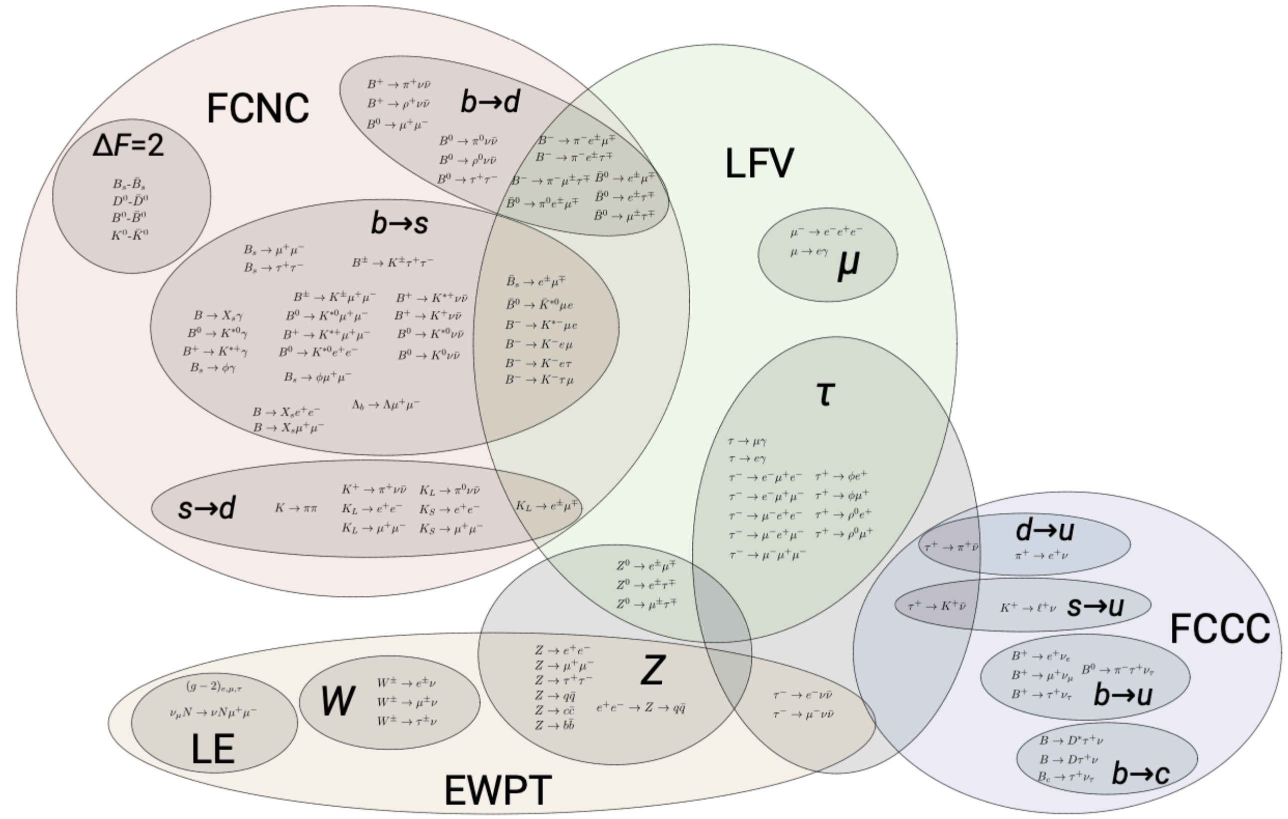
$$\vec{x} = \vec{O}_i^{\exp} - \vec{O}_i^{\mathrm{th}}\,.$$

J. Aebischer, J. Kumar, P. Stangl, D. Straub,
Eur.Phys.J.C 79 (2019) 6, 509

smelli v1

- Flavor observables from various sectors
- EWPT

~250 observables



smelli v2

- Updated flavor observables
- Beta decays
- Higgs signal strengths for various decay and production channels
- $e^+e^- \to W^+W^-$ pair production at LEP-2

~500 observables

Gonzalez-Alonso, Naviliat-Cuncic, Severijns, arXiv:1803.08732

Falkowski, Straub, arXiv:1911.07866

Long term goal: truly global likelihood, adding more observables
-> obs. predictions, then RGs are the bottleneck

introducing

jelli

a **J**AX-based **E**FT **l**ike**li**hood

AS, P. Stangl, 25xx.soon

Data:

- observables
- measurements

→

jelli

→

Highly optimized likelihood

Data:

- observables
- measurements

jelli

Highly optimized likelihood

- open-source python package
- built from scratch, using JAX
- uses Just-In-Time (**JIT**) compilation
- **autodiff:** differentiable likelihood
- **backwards compatible** with the smelli **user interface** + new features
- **fast**: **few ms** per parameter point
- takes any data in predefined format
- already includes RGs, matching

High performance array computing

JAX is a Python library for accelerator-oriented array computation and program transformation, designed for high-performance numerical computing and large-scale machine learning.

jax.readthedocs.io

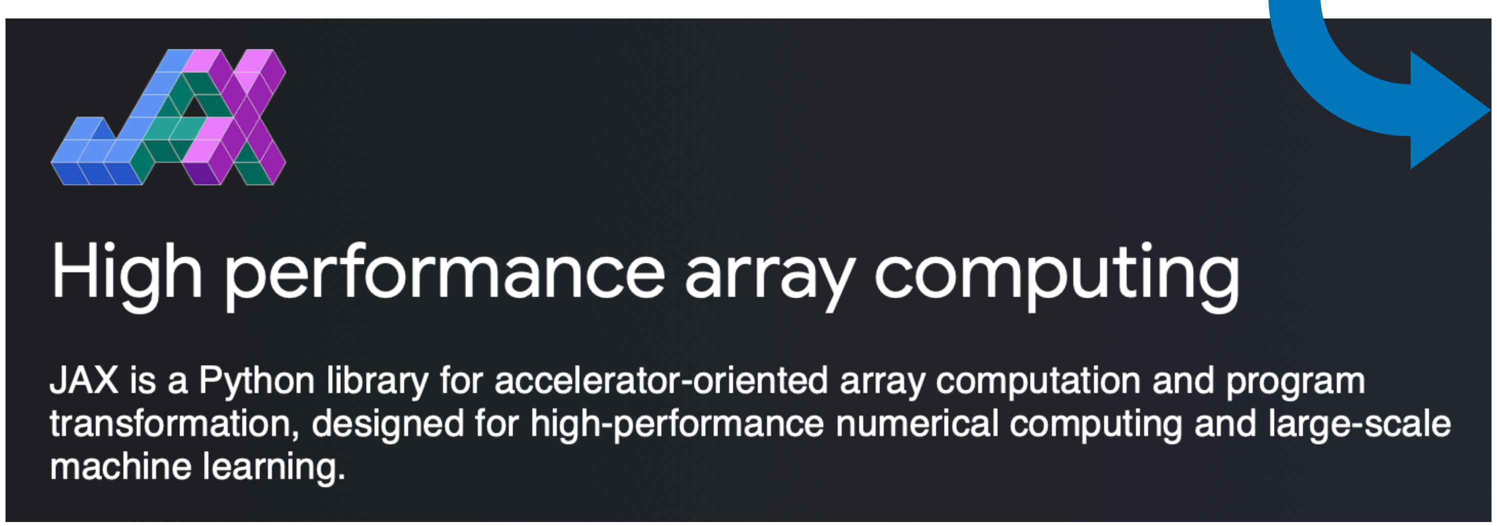
- observable .json files for predictions
- measurement .json files for experimental likelihoods
- **can be any data in predefined format**
- ongoing work within the LHC EFT WG to standardise the data format for observable predictions **(see this talk by Ken Mimasu)**
  -> will allow to exchange observable predictions between different
  likelihood codes, collaborations
- later in this talk we will discuss a *jellified* smelli 3.0 - example usecase of jelli

Data:
- observables
- measurements

jelli

Highly optimized likelihood

- open-source python package
- built from scratch, using JAX
- uses Just-In-Time (**JIT**) compilation
- **autodiff:** differentiable likelihood
- **backwards compatible** with the smelli **user interface** + new features
- **fast**: **few ms** per parameter point
- takes any data in predefined format
- already includes RGs, matching



jax.readthedocs.io

Theory predictions and uncertainties

Altmannshofer, Stangl, 2103.13370

Theory predictions and uncertainties

- amplitudes $\mathscr{A}$ computed to first order in dim. 6 expansion -> linear in WCs
- observables depend on $|\mathscr{A}|^2$ -> second-order polynomial in WCs p_i
- $O_k = f_k(p_1, \ldots, p_n)$ - observable as a function of polynomials

important example: branching ratios/cross sections directly $O_k = f_k(p_1) = p_1$

Theory predictions and uncertainties

- amplitudes $\mathscr{A}$ computed to first order in dim. 6 expansion -> linear in WCs
- observables depend on $|\mathscr{A}|^2$ -> second-order polynomial in WCs p_i
- $O_k = f_k(p_1, \ldots, p_n)$ - observable as a function of polynomials

important example: branching ratios/cross sections directly $O_k = f_k(p_1) = p_1$

Then write each polynomial as

$$p_i = \vec{p}_i \cdot \overrightarrow{V}_i$$

Vector of polynomial coefficients

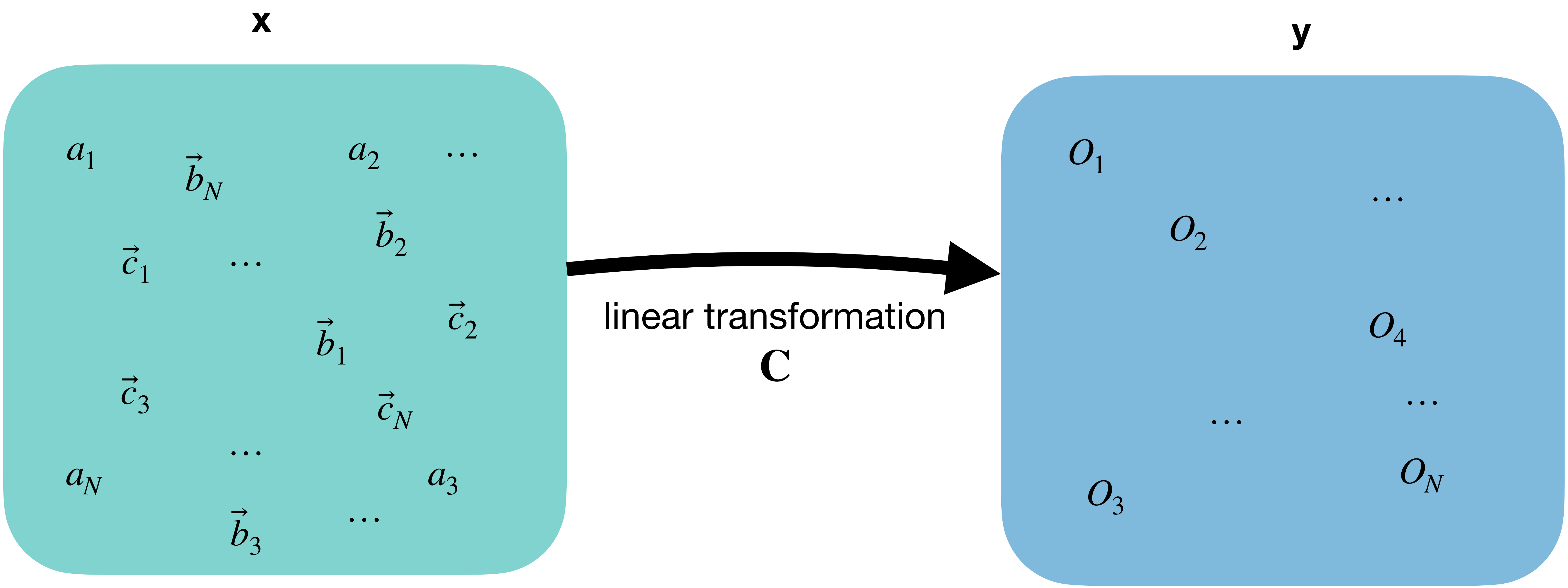
$$\vec{p}_i^{\,T} = (a_i, \vec{b}_i^{\,T}, \vec{c}_i^{\,T})$$

Vector of WCs

$$\overrightarrow{V}_i^T = (1, C_1, \ldots, C_k, \ldots, C_l, \ldots, \ldots, C_k^2, \ldots, C_k C_l, \ldots C_l^2, \ldots)$$

- NP-independent
- encodes dependence on nuissance parameters, these can be sampled or fixed
- we compute an ensemble of poly. coefficients for all polynomials
  -> obtain mean values and covariance matrix for all polynomial coefficients

Theory predictions and uncertainties



space of polynomial coefficients

mean values: μ_x

covariance matrix: $\mathbf{\Sigma}_x$

space of observables

$$\mu_y = \mathbf{C}\mu_x$$
$$\mathbf{\Sigma}_y = \mathbf{C}\mathbf{\Sigma}_x\mathbf{C}^T$$

Altmannshofer, Stangl, 2103.13370

Then mean polynomial predictions are computed as:

$$p_i = \vec{p}_i \cdot \overrightarrow{V}_i$$

$$\overrightarrow{V}_i^T = (1, C_1, \ldots, C_k, \ldots, C_l, \ldots, \ldots, C_k^2, \ldots, C_k C_l, \ldots C_l^2, \ldots)$$

$$\underbrace{\begin{pmatrix} \overrightarrow{V}_1^T & & & \\ & \overrightarrow{V}_2^T & & \\ & & \ddots & \\ & & & \overrightarrow{V}_N^T \end{pmatrix}}_{\text{matrix } \mathbf{C}}$$

$$\begin{array}{cl} \left(\begin{array}{c} \overrightarrow{\mu}_1 \\ \hline \overrightarrow{\mu}_2 \\ \hline \vdots \\ \hline \overrightarrow{\mu}_N \end{array}\right) & \begin{array}{l} \text{poly. 1} \\ \text{poly. 2} \\ \\ \text{poly. N} \end{array} \end{array}$$

$\overrightarrow{\mu}_i$ - mean values of poly. coefficients

The covariance matrix can be computed as:

Altmannshofer, Stangl, 2103.13370

cov. matrix of poly. 1 coefficients

cov. matrix between poly. 1 and poly. N coefficients

$$\underbrace{\mathbf{\Sigma}}_{\text{cov. matrix in obs. space}} = \mathbf{C} \underbrace{\begin{pmatrix} \mathrm{cov}(1,1) & \cdots & \mathrm{cov}(1,N) \\ & \ddots & \\ & \cdots & \mathrm{cov}(N,N) \end{pmatrix}}_{\text{cov. matrix in poly. coeff. space}} \mathbf{C}^T$$

The covariance matrix can be computed as:

Altmannshofer, Stangl, 2103.13370

cov. matrix of poly. 1 coefficients

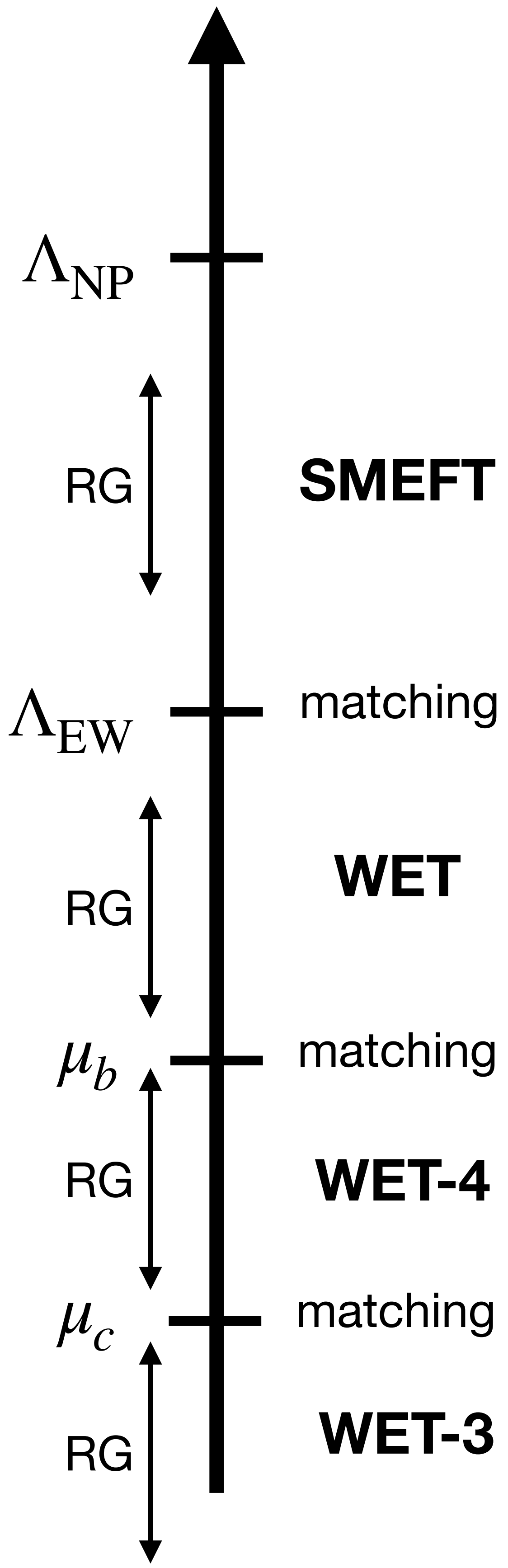
cov. matrix between poly. 1 and poly. N coefficients

$$\underbrace{\boldsymbol{\Sigma}}_{\text{cov. matrix in obs. space}} = \mathbf{C} \underbrace{\left(\begin{array}{c:c:c} \text{cov}(1,1) & \cdots & \text{cov}(1,N) \\ \hdashline & \ddots & \\ \hdashline & \cdots & \text{cov}(N,N) \end{array}\right)}_{\text{cov. matrix in poly. coeff. space}} \mathbf{C}^T$$

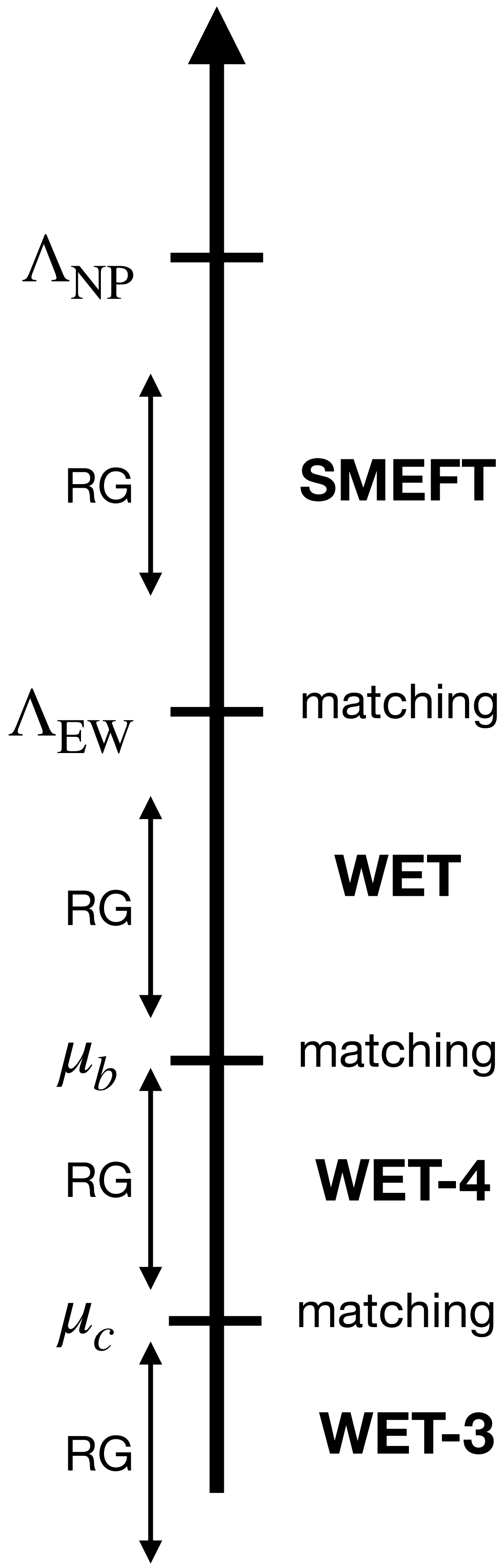
$$\boldsymbol{\Sigma}_{ij} = \vec{V}_i^T \text{cov}(i,j) \vec{V}_j$$

NP-dependent th. cov. matrix!

RGs, matching, translation



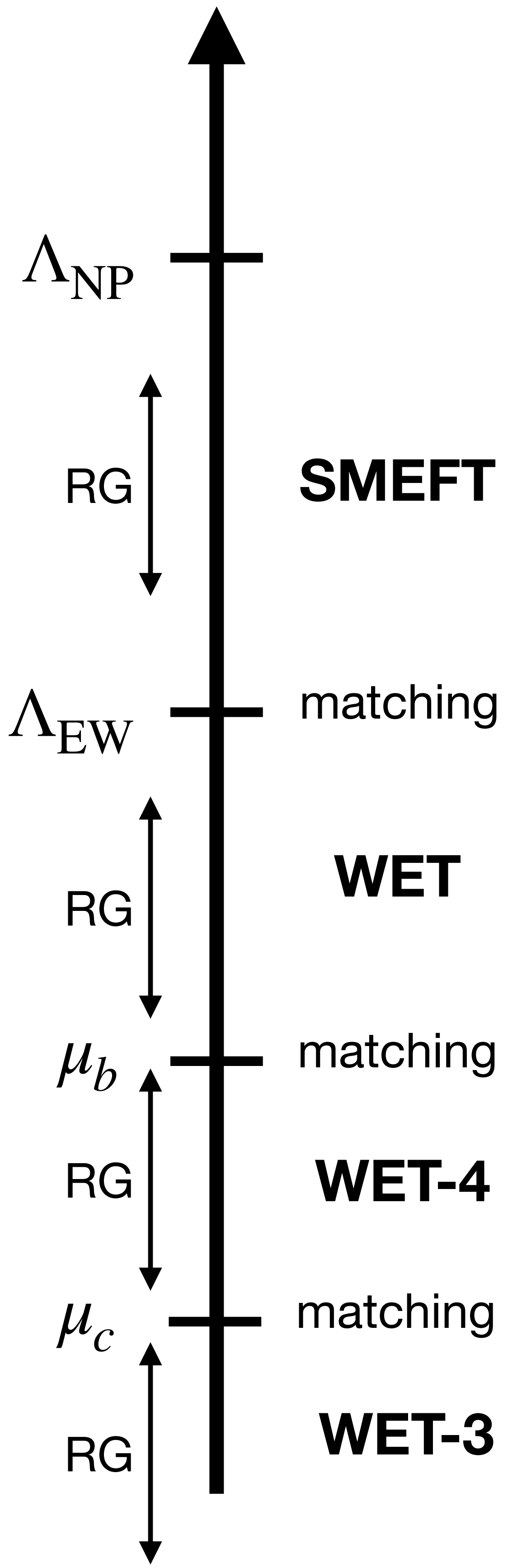
RGs, matching, translation



For users:

- Adding new observables at various scales in these EFTs does not require any intervention or recomputation, all is automatically taken care of by jelli
- Pheno can be done in any EFT at any scale

RGs, matching, translation



For users:

- Adding new observables at various scales in these EFTs does not require any intervention or recomputation, all is automatically taken care of by jelli
- Pheno can be done in any EFT at any scale

Internally we provide:

- evolution matrices from/to various scales for each EFT
- translators for different bases of each EFT
- matching matrices for subsequent EFTs

computed using wilson*, **for full EFTs, as defined in** WCxf

e.g. SMEFT Warsaw/Warsaw up, WET JMS/flavio

- Efficient methods for evaluation
- Fast interpolation available for scales not (yet) precomputed

*internal version, will be made public

Upshot:

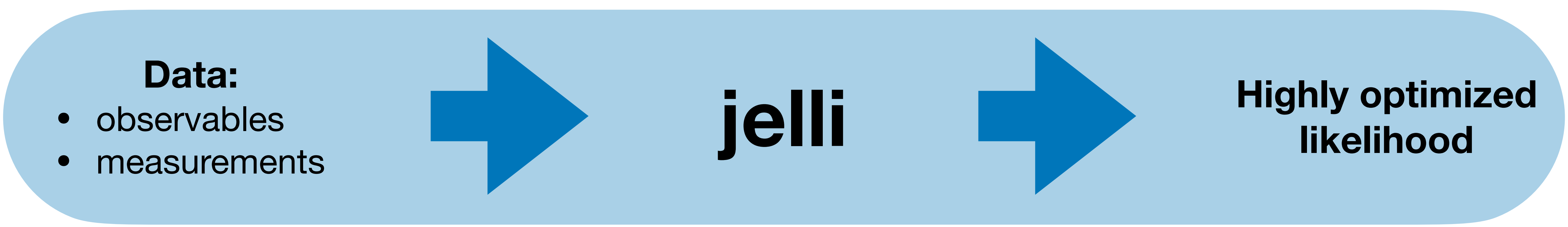
- observable predictions,
- computation of theoretical cov. matrix in presence of NP,
- RG evolution, matching, translating,
- computation of likelihoods

all turn into fast linear algebra operations

-> jelli leverages JAX for optimal handling of large arrays

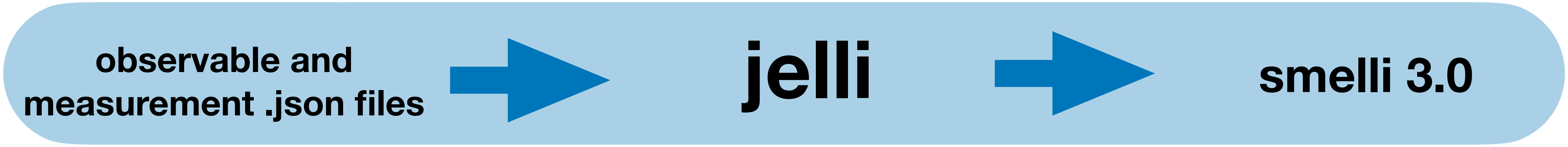
-> we implement a JAX-friendly statistics module for handling of all likelihoods

-> jelli combines them into a highly efficient global likelihood

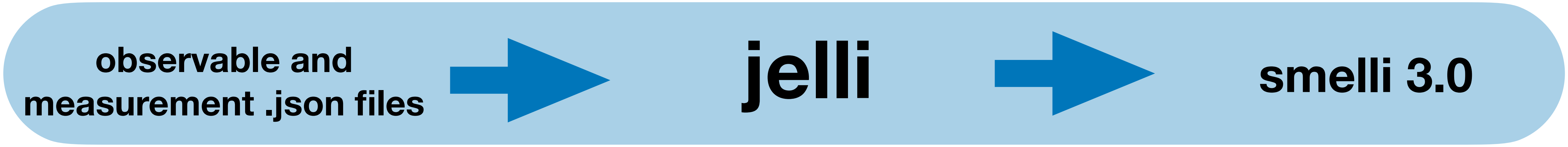


smelli 3.0 - a *jellified* version with new features and observables

smelli 3.0 - a *jellified* version with new features and observables



smelli 3.0 - a *jellified* version with new features and observables



New observables:

- Major: neutral and charged current Drell-Yan tails
- Many updates
- $\mathcal{O}(1.5\mathrm{k})$ observables

Greljo, Salko, AS, Stangl, 2212.10497
Altmannshofer, Stangl, 2103.13370
Crivellin, Kirk, Kitahara, Mescia, 2212.06862
Greljo, Salko, AS, Stangl, 2306.09401

smelli 3.0 - a *jellified* version with new features and observables

observable and measurement .json files → **jelli** → **smelli 3.0**

New observables:

- Major: neutral and charged current Drell-Yan tails
- Many updates
- $\mathcal{O}(1.5\mathrm{k})$ observables

Greljo, Salko, AS, Stangl, 2212.10497
Altmannshofer, Stangl, 2103.13370
Crivellin, Kirk, Kitahara, Mescia, 2212.06862
Greljo, Salko, AS, Stangl, 2306.09401

Theory predictions:

- All observables expressed as (functions of) polynomials of WCs -> .json files
- We precompute these from flavio with internal code, will be public in future
- Significant speed up at the cost of some flexibility compared to smelli 2

smelli 3.0 - a *jellified* version with new features and observables

observable and measurement .json files → **jelli** → **smelli 3.0**

New observables:

- Major: neutral and charged current Drell-Yan tails
- Many updates
- $\mathcal{O}(1.5\mathrm{k})$ observables

Greljo, Salko, AS, Stangl, 2212.10497
Altmannshofer, Stangl, 2103.13370
Crivellin, Kirk, Kitahara, Mescia, 2212.06862
Greljo, Salko, AS, Stangl, 2306.09401

Theory predictions:

- All observables expressed as (functions of) polynomials of WCs -> .json files
- We precompute these from flavio with internal code, will be public in future
- Significant speed up at the cost of some flexibility compared to smelli 2

Theory uncertainties:

- Theory cov. matrix precomputed in the space of poly. coeffs. -> NP dependent th. cov. mat.
- Allows inclusion of new observables, e.g. EDMs
- SM-only cov. matrix is a limitting case (approach taken in smelli 2) - faster evaluation

smelli 3.0 - a *jellified* version with new features and observables

observable and measurement .json files → **jelli** → **smelli 3.0**

Greljo, Salko, AS, Stangl, 2212.10497
Altmannshofer, Stangl, 2103.13370
Crivellin, Kirk, Kitahara, Mescia, 2212.06862
Greljo, Salko, AS, Stangl, 2306.09401

New observables:

- Major: neutral and charged current Drell-Yan tails
- Many updates
- $\mathcal{O}(1.5\mathrm{k})$ observables

Theory predictions:

- All observables expressed as (functions of) polynomials of WCs -> .json files
- We precompute these from flavio with internal code, will be public in future
- Significant speed up at the cost of some flexibility compared to smelli 2

Theory uncertainties:

- Theory cov. matrix precomputed in the space of poly. coeffs. -> NP dependent th. cov. mat.
- Allows inclusion of new observables, e.g. EDMs
- SM-only cov. matrix is a limitting case (approach taken in smelli 2) - faster evaluation

RGs:

- Automatically taken care of by jelli using info from above .json files

Measurements

- .json files defining experimental likelihoods exported from flavio database

smelli 3.0 - a *fast* version

smelli 3.0 - a *fast* version

```
plotdata_smelli = gls.plot_data_2d(
    wc_func,
    scale=1000.0,
    x_min=-4.0, x_max=1.0,
    y_min=-2.0, y_max=1.5,
    steps=20
)
```

✓ 60m 48.8s **(smelli 2)**

- **Goal:** scan over 2 directions in the SMEFT parameter space
- The directions are defined in wc_func
- The **UV scale** is set to 1 TeV
- The **range and step** of the two dimensions is set
- The result is the **sampled global likelihood** (and sublikelihoods)
- In the background: RG evolving and matching to all the relevant EFTs and scales for observables, computing predictions, computing all the likelihoods

smelli 3.0 - a *fast* version

(preliminary version!)

```
plotdata_smelli = gls.plot_data_2d(
    wc_func,
    scale=1000.0,
    x_min=-4.0, x_max=1.0,
    y_min=-2.0, y_max=1.5,
    steps=20
)
```

✓ 60m 48.8s

(smelli 2)

```
plotdata_jelli = gl.plot_data_2d(
    wc_func,
    scale=1000.0,
    x_min=-4.0, x_max=1.0,
    y_min=-2.0, y_max=1.5,
    steps=20
)
```

✓ 1.5s

(smelli 3)

- **Goal:** scan over 2 directions in the SMEFT parameter space
- The directions are defined in wc_func
- The **UV scale** is set to 1 TeV
- The **range and step** of the two dimensions is set
- The result is the **sampled global likelihood** (and sublikelihoods)
- In the background: RG evolving and matching to all the relevant EFTs and scales for observables, computing predictions, computing all the likelihoods

smelli 3.0 - a *fast* version

(preliminary version!)

```python
plotdata_smelli = gls.plot_data_2d(
    wc_func,
    scale=1000.0,
    x_min=-4.0, x_max=1.0,
    y_min=-2.0, y_max=1.5,
    steps=20
)
```

✓ 60m 48.8s

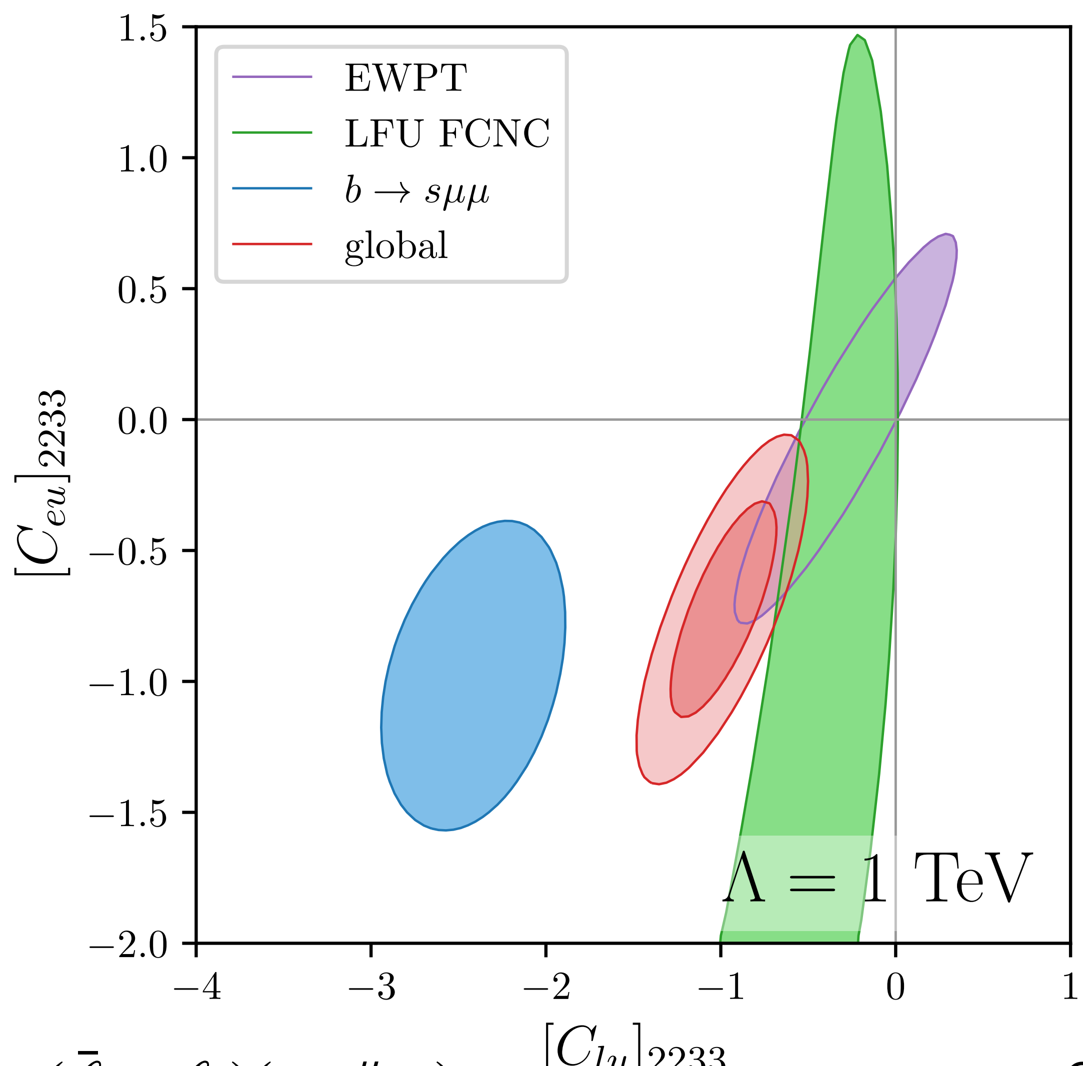
(smelli 2)

```python
plotdata_jelli = gl.plot_data_2d(
    wc_func,
    scale=1000.0,
    x_min=-4.0, x_max=1.0,
    y_min=-2.0, y_max=1.5,
    steps=20
)
```

✓ 1.5s

(smelli 3)

$(\bar{e}_2\gamma_\mu e_2)(\bar{u}_3\gamma^\mu u_3)$



$(\bar{\ell}_2\gamma_\mu \ell_2)(\bar{u}_3\gamma^\mu u_3)$

Assuming NP in top RG effects crucial

Celis, Fuentes-Martin, Vicente, Virto, 1704.05672
Kamenik, Soreq, Zupan, 1704.06005
Camargo-Molina, Celis, Faroughy, 1805.04917
Garosi, Marzocca, Sanchez, Stanzione, 2310.00047

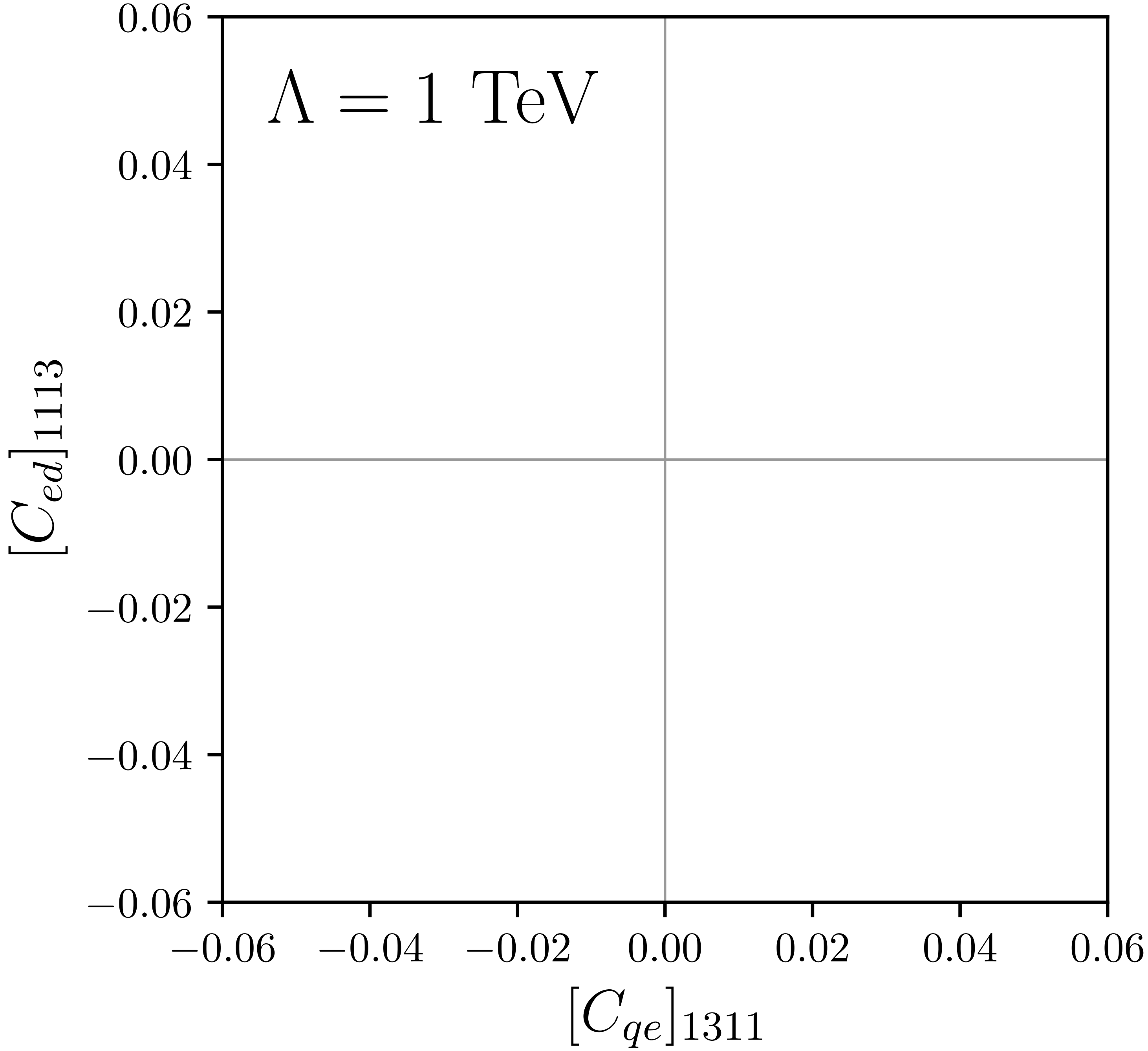
Interplay in a flavorful global likelihood

Similar to Greljo, Salko, AS, Stangl; 2212.10497

Interplay in a flavorful global likelihood

Similar to Greljo, Salko, AS, Stangl; 2212.10497

Example assuming NP only in $bdee$



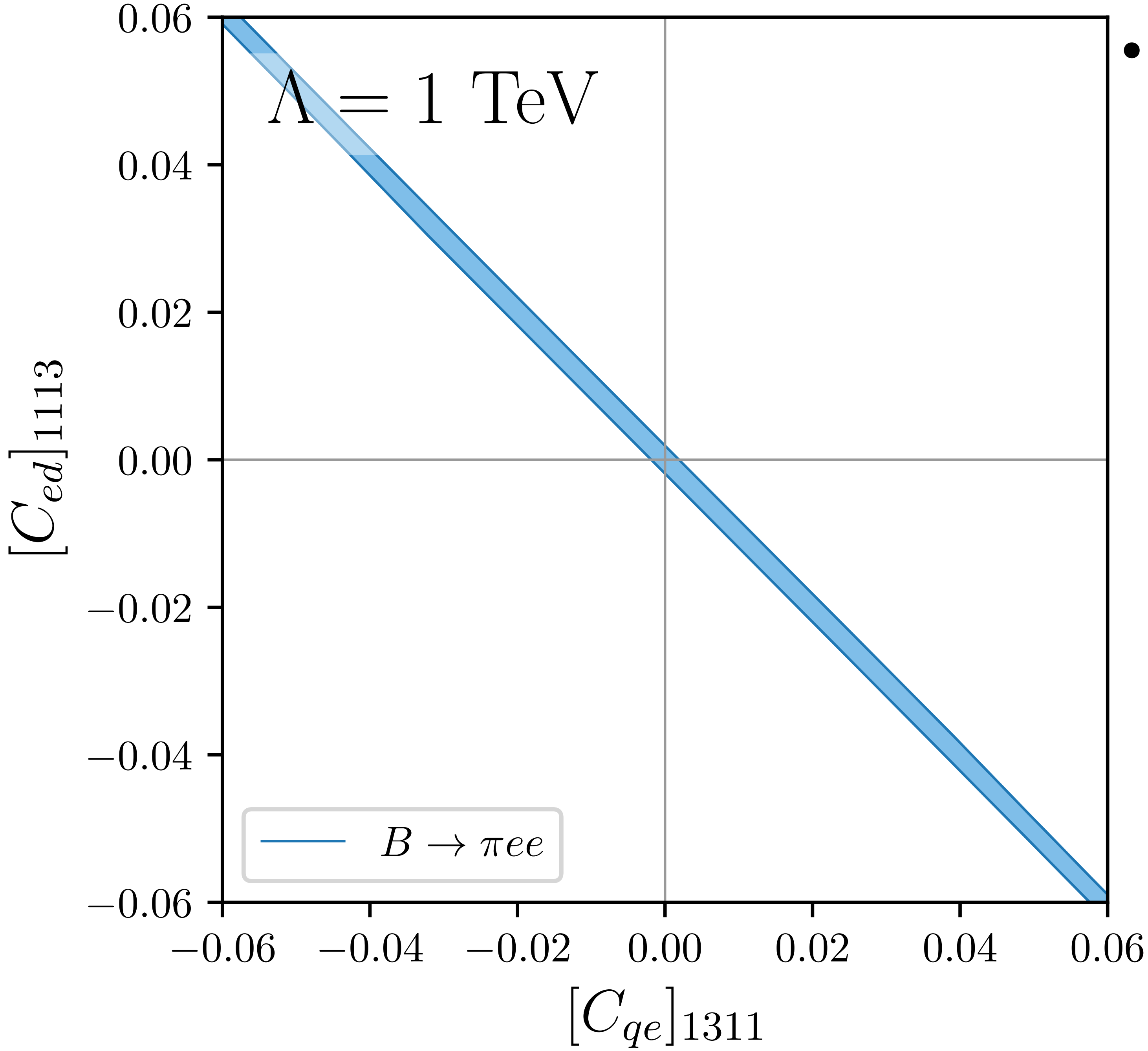
$Q_{ed} = (\bar{e}\gamma_\mu e)(\bar{d}\gamma^\mu d)$

$Q_{qe} = (\bar{q}\gamma_\mu q)(\bar{e}\gamma^\mu e)$

Interplay in a flavorful global likelihood

Similar to Greljo, Salko, AS, Stangl; 2212.10497

Example assuming NP only in $bdee$



$Q_{ed} = (\bar{e}\gamma_\mu e)(\bar{d}\gamma^\mu d)$

$Q_{qe} = (\bar{q}\gamma_\mu q)(\bar{e}\gamma^\mu e)$

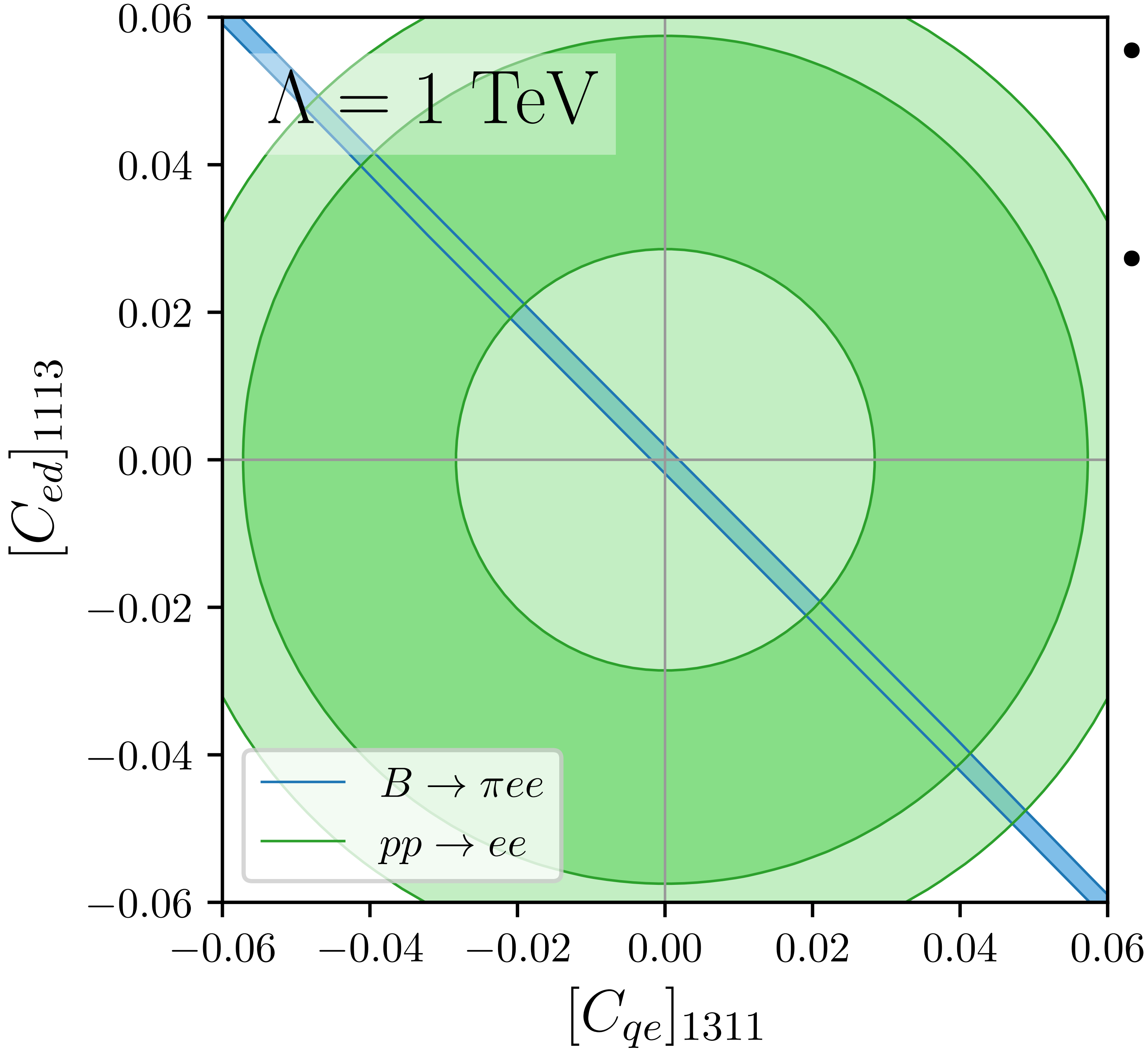
Note:

- Narrow flat direction in $B \to \pi ee$
  -> need many points in this plane
  -> no problem, sampling is fast!

Interplay in a flavorful global likelihood

Similar to Greljo, Salko, AS, Stangl; 2212.10497

Example assuming NP only in $bdee$



Note:

- Narrow flat direction in $B \to \pi ee$
  -> need many points in this plane
  -> no problem, sampling is fast!
- Complementarity with DY tails
  -> same contact interactions contribute
  to complementary processes
  -> helps close the fit

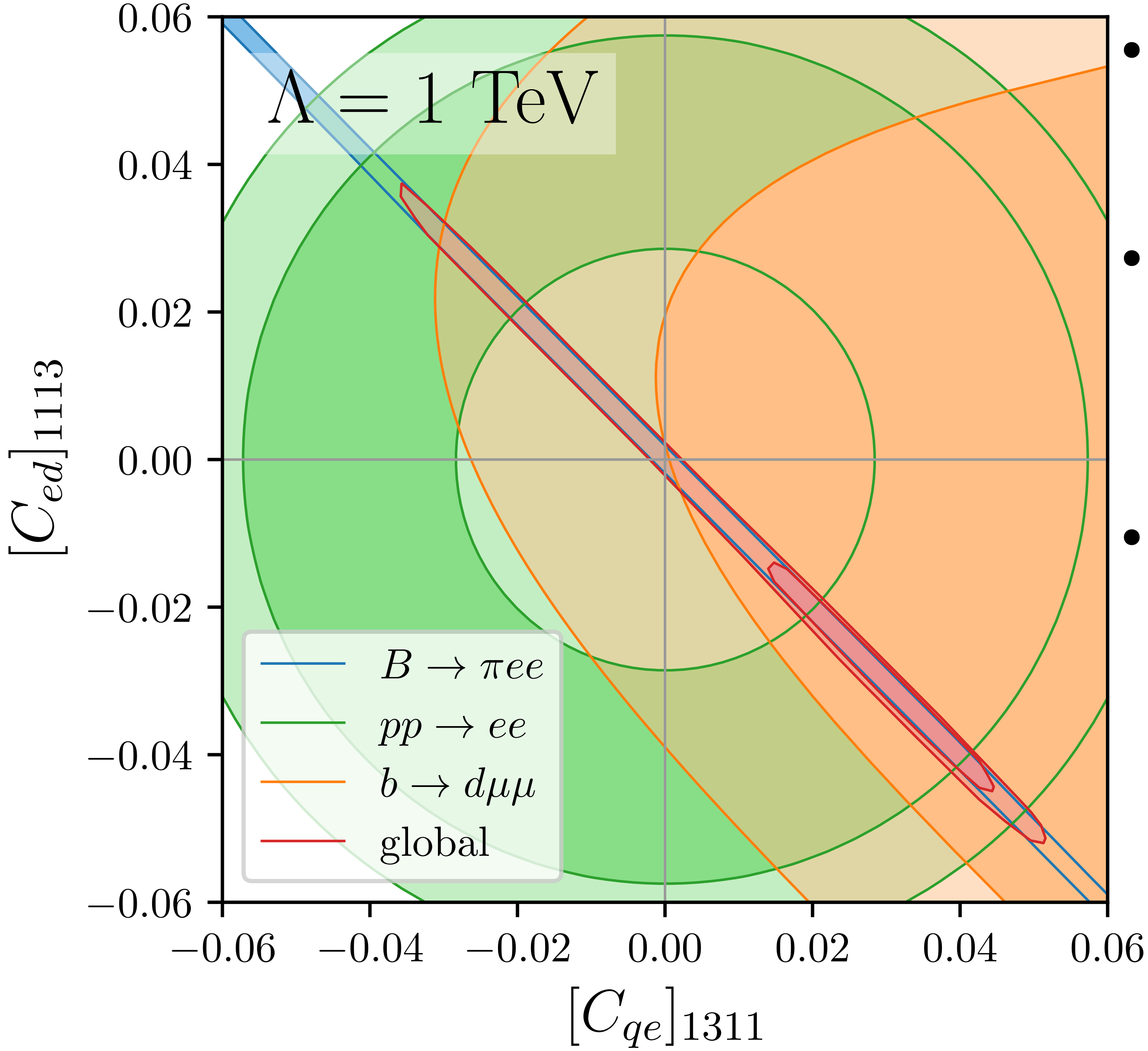
$$Q_{ed} = (\bar{e}\gamma_\mu e)(\bar{d}\gamma^\mu d)$$

$$Q_{qe} = (\bar{q}\gamma_\mu q)(\bar{e}\gamma^\mu e)$$

Interplay in a flavorful global likelihood

Similar to Greljo, Salko, AS, Stangl; 2212.10497

Example assuming NP only in $bdee$



Note:

- Narrow flat direction in $B \to \pi ee$
  -> need many points in this plane
  -> no problem, sampling is fast!
- Complementarity with DY tails
  -> same contact interactions contribute to complementary processes
  -> helps close the fit
- RG into better measured $b \to d\mu\mu$
  -> more exp info: $B^0 \to \mu\mu, B_s \to K^{*0}\mu\mu, B \to \pi\mu\mu$
  -> important even if gauge coupling suppressed

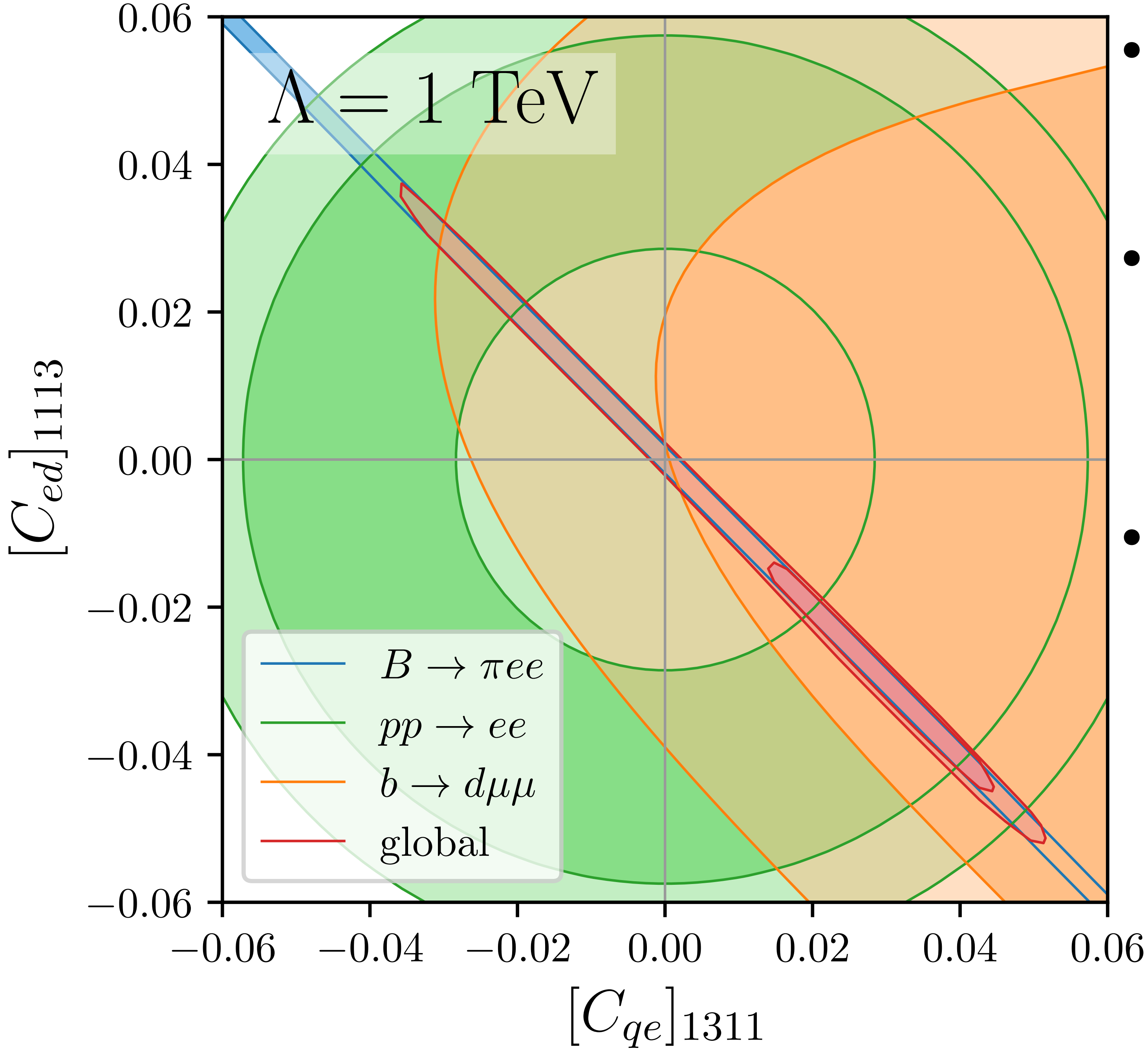
$Q_{ed} = (\bar{e}\gamma_\mu e)(\bar{d}\gamma^\mu d)$

$Q_{qe} = (\bar{q}\gamma_\mu q)(\bar{e}\gamma^\mu e)$

Interplay in a flavorful global likelihood

Similar to Greljo, Salko, AS, Stangl; 2212.10497

Example assuming NP only in $bdee$



Note:

- Narrow flat direction in $B \to \pi ee$
  -> need many points in this plane
  -> no problem, sampling is fast!
- Complementarity with DY tails
  -> same contact interactions contribute to complementary processes
  -> helps close the fit
- RG into better measured $b \to d\mu\mu$
  -> more exp info: $B^0 \to \mu\mu, B_s \to K^{*0}\mu\mu, B \to \pi\mu\mu$
  -> important even if gauge coupling suppressed

$Q_{ed} = (\bar{e}\gamma_\mu e)(\bar{d}\gamma^\mu d)$

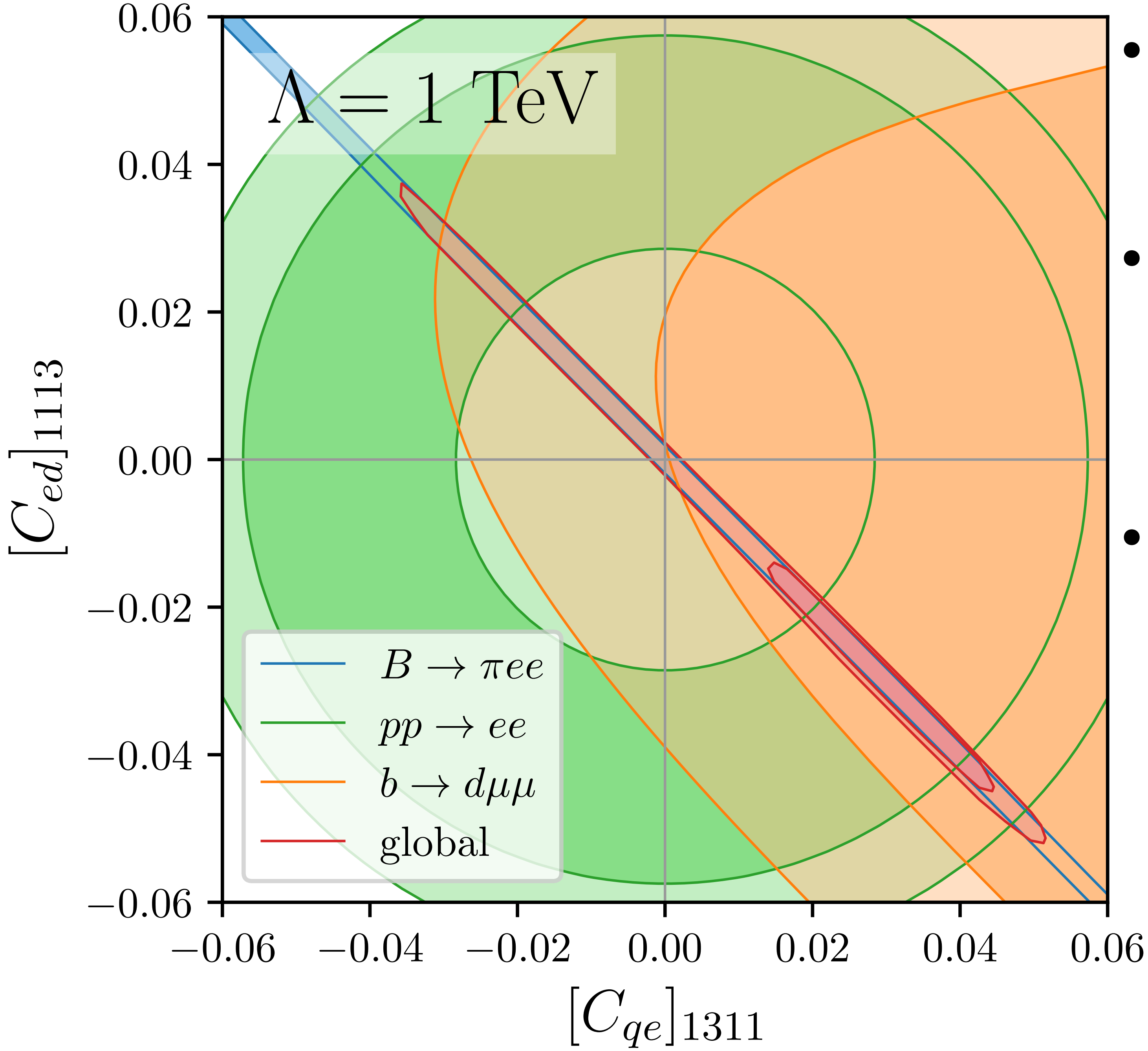
$Q_{qe} = (\bar{q}\gamma_\mu q)(\bar{e}\gamma^\mu e)$

~20s for this (70x70) plot on a single core on a laptop

Interplay in a flavorful global likelihood

Similar to Greljo, Salko, AS, Stangl; 2212.10497

Example assuming NP only in $bdee$



Note:

- Narrow flat direction in $B \to \pi ee$
  -> need many points in this plane
  -> no problem, sampling is fast!
- Complementarity with DY tails
  -> same contact interactions contribute
  to complementary processes
  -> helps close the fit
- RG into better measured $b \to d\mu\mu$
  -> more exp info:
  $B^0 \to \mu\mu, B_s \to K^{*0}\mu\mu, B \to \pi\mu\mu$
  -> important even if gauge
  coupling suppressed

This assumes Warsaw down-diagonal basis

$Q_{ed} = (\bar{e}\gamma_\mu e)(\bar{d}\gamma^\mu d)$

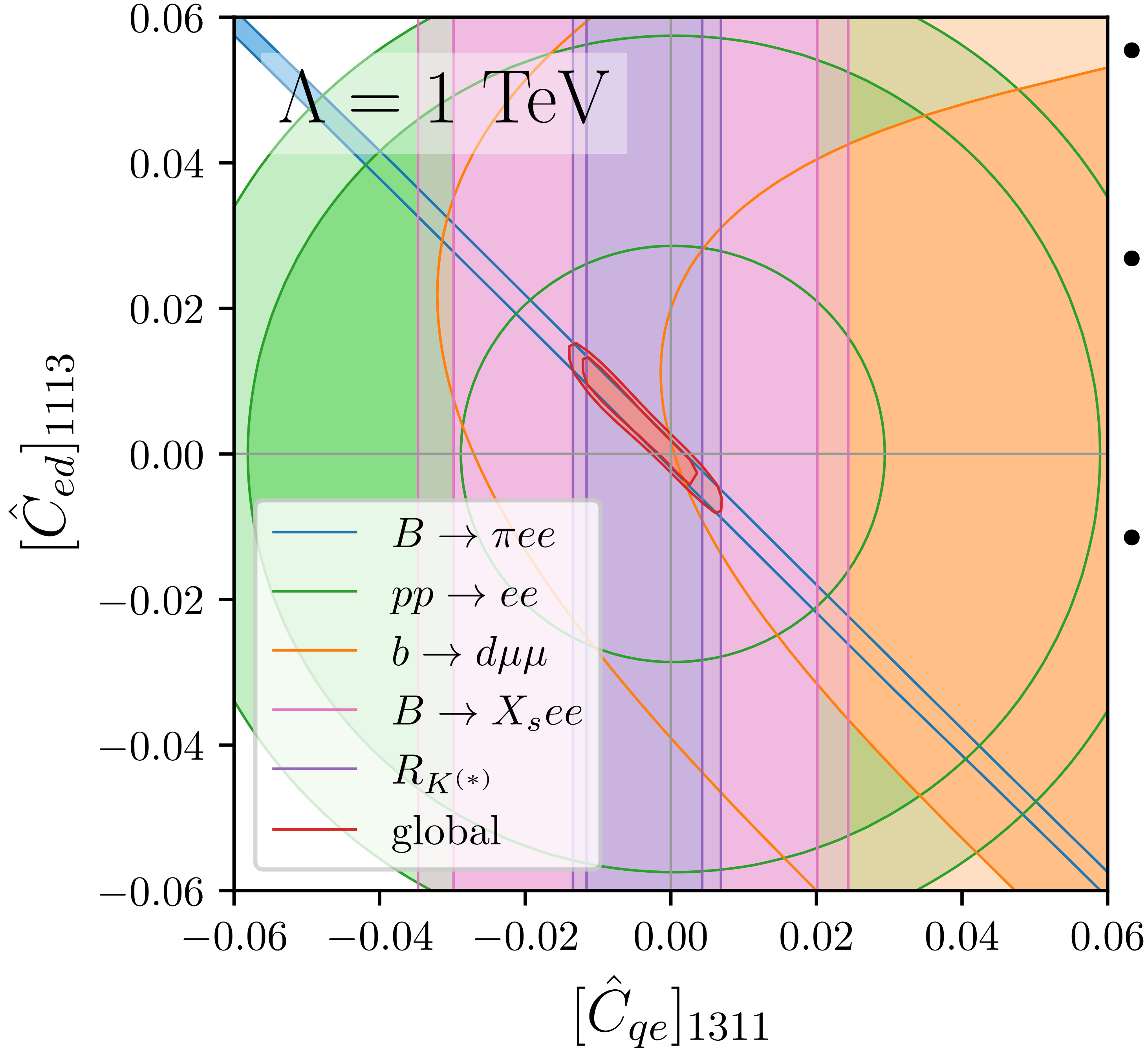
$Q_{qe} = (\bar{q}\gamma_\mu q)(\bar{e}\gamma^\mu e)$

~20s for this (70x70) plot on a single core on a laptop

Interplay in a flavorful global likelihood

Similar to Greljo, Salko, AS, Stangl; 2212.10497

Example assuming NP only in $bdee$



$$Q_{ed} = (\bar{e}\gamma_\mu e)(\bar{d}\gamma^\mu d)$$
$$Q_{qe} = (\bar{q}\gamma_\mu q)(\bar{e}\gamma^\mu e)$$

Note:

- Narrow flat direction in $B \to \pi ee$
  -> need many points in this plane
  -> no problem, sampling is fast!
- Complementarity with DY tails
  -> same contact interactions contribute to complementary processes
  -> helps close the fit
- RG into better measured $b \to d\mu\mu$
  -> more exp info: $B^0 \to \mu\mu, B_s \to K^{*0}\mu\mu, B \to \pi\mu\mu$
  -> important even if gauge coupling suppressed

Warsaw up-diagonal basis:

- Obtain additional constraints from $b \to see$, as $V_{\text{CKM}} \neq 1$

~20s for this (70x70) plot on a single core on a laptop

W.i.p.: efficient interface with matching tools

UV model

MATCHETE

model exchange .json file, containing

- model parameters, masses
- various (flavour) index contractions
- mass functions depending on loop functions and masses
- one-loop matching results in Warsaw basis

jelli

fast likelihood as a function of model parameters

Summary

- Model-independent approach to heavy NP
  -> tools for global analyses indispensable
  -> open-source, this is a community effort

- Complicated data analyses done in the SMEFT parameter space
  -> relative importance of data can be assessed
  -> high complementarity between various observables
  -> important RG effects captured
  -> reinterpretation in concrete heavy NP models possible, to be fully general no flavor assumption should be made

- We introduce **jelli**: a **J**AX-based **E**FT **l**ike**li**hood
  -> built from scratch, **differentiable** high-dim. EFT likelihood, **fast**, with powerful new features (JIT, autodiff, …)
  -> fully built on **JAX**, from user interface to observable predictions, stats, RGs, …
  -> general and modular, **supports any data in proposed format**
  -> backwards compatible with the smelli UI
  -> **smelli v3.0** running on jelli, with new features, new and updated observables, a **flavorful global likelihood with no restrictions on flavor structure**
  -> efficient interface with matching tools w.i.p.

to be released soon!

Thank you