IMPROVING CALCULATIONS OF $g-2$ AND EDMS TO ONE-LOOP LEVEL

HANNAH DAY

SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

cnrs

LPTHE
LABORATOIRE DE PHYSIQUE THEORIQUE ET HAUTES ENERGIES

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

THE NEED FOR NEW PHYSICS

- Dark matter
- Matter-antimatter asymmetry
- Neutrino masses
- Muon $g - 2$?
  - Fermilab experiment placed discrepancy between experiment and SM at 4.6σ
  - Still possible it is an artifact of experimental uncertainty
  - May be explained by new lattice QCD method of performing SM calculation
  - Strong possibility it is an indication of new physics

THE TROUBLE WITH CALCULATING OBSERVABLES

Standard Method:

- Normally extract observables by working within a single (effective) theory
- Extremely inefficient when considering multiple theories or observables

Solution:

- Automate calculations from a generic theory
- Input theory parameters and desired observable, output result
- This is what `SARAH` was designed to do

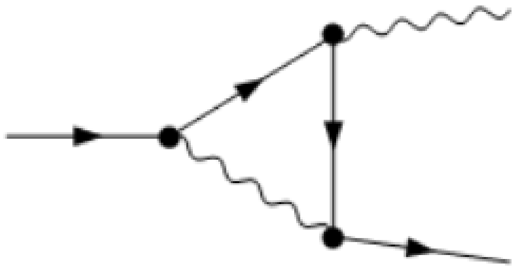
This project:

- Improve the way `SARAH` performs RG running for low-energy observables

OUTLINE

- How to extract $g-2$ and EDMs from a matrix element
- How to automate calculations
- Improvements to SARAH
- Results
- Conclusion

EXTRACTING $g-2$ AND THE EDM



Matrix element $\mathcal{M} = \mathcal{M}^\mu \epsilon^*_\mu$

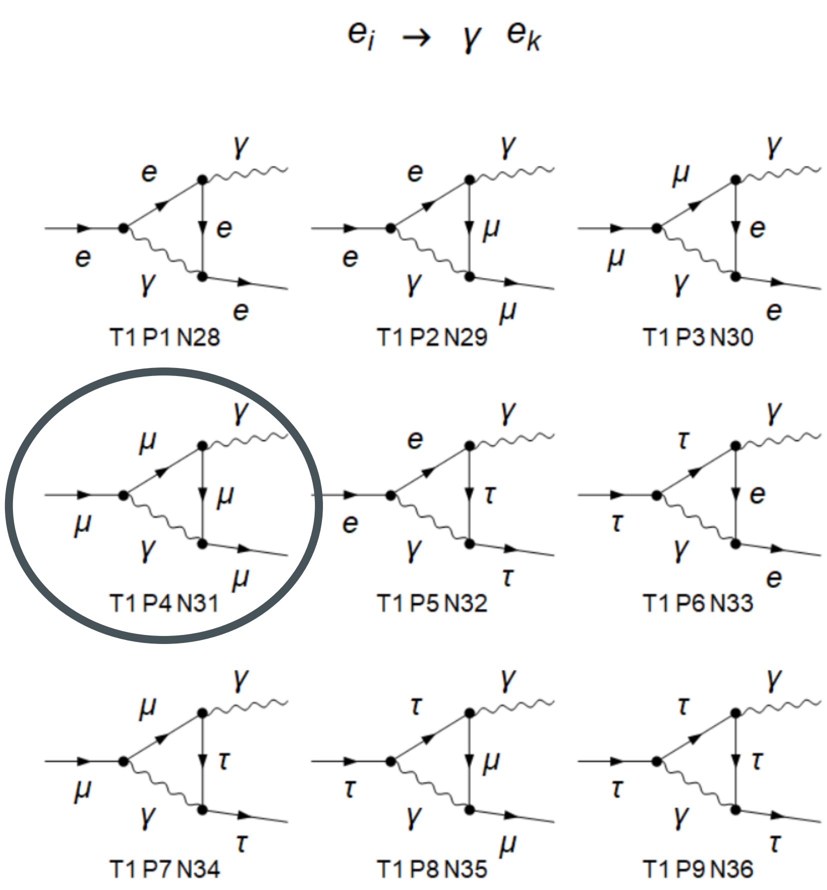
$$i\mathcal{M}^\mu = (-ie)\bar{u}(p_f)\Big[\gamma^\mu \underbrace{F_1(q^2)}_{\text{charge renormalization}} + \frac{i\sigma^{\mu\nu}}{2m_e} q_\nu \underbrace{F_2(q^2)}_{g-2} + \frac{i\sigma^{\mu\nu}}{2m_e}\gamma^5 q_\nu \underbrace{F_3(q^2)}_{\text{EDM}} + \frac{1}{2m_e}\left(q^\mu - \frac{q^2}{2m_e}\gamma^\mu\right)\gamma^5 \underbrace{F_4(q^2)}_{\text{anapole}}\Big] u(p_i)$$

Need to convert from $\sigma^{\mu\nu}$ to γ^μ in order to match output of `SARAH`

→ Use Dirac equations and γ identities to obtain

$$\mathcal{M} = -e\bar{u}(p_f)\epsilon^*_\mu \left[\gamma^\mu\left(F_1(q^2) - F_2(q^2)\right) + \frac{p_i^\mu}{m} F_2(q^2) + \frac{p_i^\mu}{m}\gamma^5 F_3(q^2) - \frac{q^2}{4m^2}\gamma^\mu\gamma^5 F_4(q^2)\right] u(p_i)$$

EXAMPLE: CALCULATING MUON $g - 2$



Simplified `FormCalc` output:

$$\mathcal{M} = \tfrac{1}{\pi}\alpha e\Big((\bar{u}_2\gamma^\mu\epsilon^*_\mu u_1)\big(\tfrac{5}{4} + \tfrac{1}{2}\ln\tfrac{\lambda^2}{m_e^2}\big) - (\bar{u}_2 u_1)\tfrac{(\epsilon^*\cdot k_1)}{2m_e}\Big)$$

Compare with:

$$\mathcal{M} \supset -e\bar{u}_2\epsilon^*_\mu\frac{k_1^\mu}{m}F_2(q^2)u_1$$

Result:

$$F_2(0) = \frac{\alpha}{2\pi}$$

SARAH: A SPECTRUM GENERATOR GENERATOR

- Takes Mathematica file as input
- Outputs Fortran code used to generate spectrum of particles
- Has been extended to include many pre-written input files
- Includes a variety of observables as additional output
- Interfaces with several other codes

- Use for this project: calculating low-energy observables

SARAH: OLD METHOD

- Pre-SARAH uses FeynArts and FormCalc to calculate matrix elements
- Pre-SARAH outputs a SARAH input file
- SARAH uses SPheno library to run RGEs between GUT scale and Z mass
- EDM and $g-2$ are output by SPheno

Problems:

- Pre-SARAH designed for flavor-violating processes
  - only calculates dipole and four-fermion diagrams
- SPheno cannot run down to observable scale and performs calculation in full theory
  - introduces log corrections from heavy particles not present at WET scale

A NEW PACKAGE: `SARAH-PRELUDE`

Input parameters:

- Types of particles used for external legs
- List of particles and topologies to be excluded

Calculations:

- Generate possible diagrams with `FeynArts`
- Evaluate amplitudes in terms of PV integrals with `FormCalc`
- Extract Lorentz structures present in amplitudes

Output:

- Information about couplings to Mathematica package file
- List of Lorentz structures to Fortran module file

MATCHING OPERATORS OF INPUT THEORY TO SMEFT

- SMEFT is generalized version of SM valid up to matching scale
- Contains basis of 56 operators, 7 of which are relevant for $g-2$ / EDMs

Tree level

$$\mathcal{O}_{eW} = (\bar{l}^i_L \sigma^a \sigma^{\mu\nu} e_R) H^i W^a_{\mu\nu}$$
$$\mathcal{O}_{eB} = (\bar{l}^i_L \sigma^{\mu\nu} e_R) H^i B_{\mu\nu}$$

One-loop level

From RGEs:

$$\mathcal{O}_{luqe} = (\bar{l}^i_L u_R)\varepsilon_{ij}(\bar{Q}^j_L e_R)$$
$$\mathcal{O}_{W\overset{(\sim)}{W}} = |H|^2 W^{a\mu\nu} \overset{(\sim)}{W}{}^a_{\mu\nu}$$
$$\mathcal{O}_{B\overset{(\sim)}{B}} = |H|^2 B^{\mu\nu} \overset{(\sim)}{B}_{\mu\nu}$$
$$\mathcal{O}_{W\overset{(\sim)}{B}} = (H^\dagger \sigma^a H) W^{a\mu\nu} \overset{(\sim)}{B}_{\mu\nu}$$

Finite contribution:

$$\mathcal{O}_{\overset{(\sim)}{W}} = \varepsilon_{abc} \overset{(\sim)}{W}{}^{a\nu}_{\mu} W^{b\rho}_{\nu} W^{c\mu}_{\rho}$$

- Rewrite in broken phase for below WET

SARAH: NEW METHOD

- SARAH-Prelude supplements pre-SARAH
- Outputs of SARAH-Prelude and pre-SARAH are used as input for SARAH
- SARAH outputs Fortran code, which uses
  - Spheno: Run RGEs to matching scale and match HET to SMEFT
  - wilson: Run to electroweak scale, match SMEFT to LEFT, then run to observable scale
  - Flavio: Output EDM and g-2 observables

PRELIMINARY RESULTS

- Test two models with tree-level operators
  - Well-known model (MSSM) to ensure results are as expected
  - Simple model (ProtoSlepton) to more clearly see effects
- Vary over energy range above Z boson mass
- Compare results with and without Z boson contribution
  - Z boson is main contribution to RG running

PRELIMINARY RESULTS: MSSM

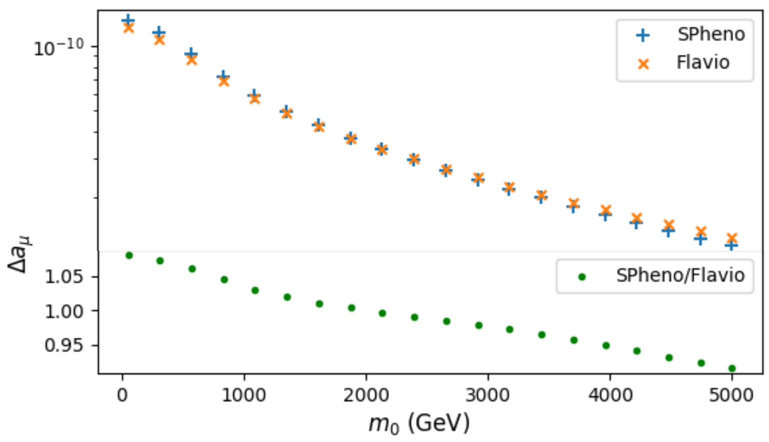
Expected:

- `Flavio` result diverges as energy increases
- `Flavio` contributes extra factor of about 10% from running

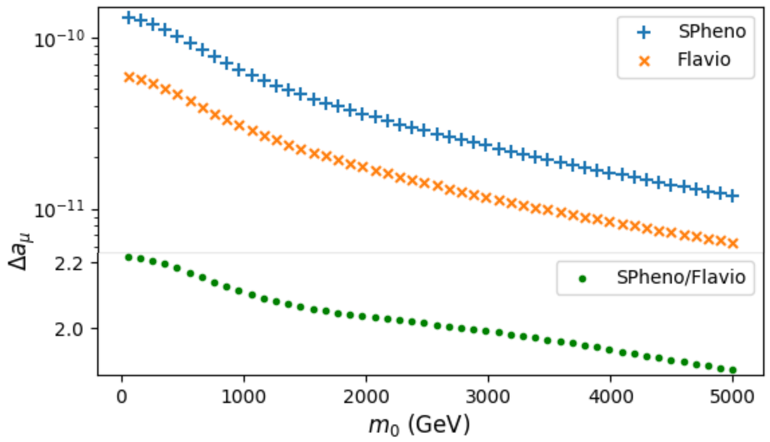
Actual:

- `Flavio` result is multiplied by 0.48
- Difference from `SPheno` is too large
- Waiting for `Wilson` authors to fix bug

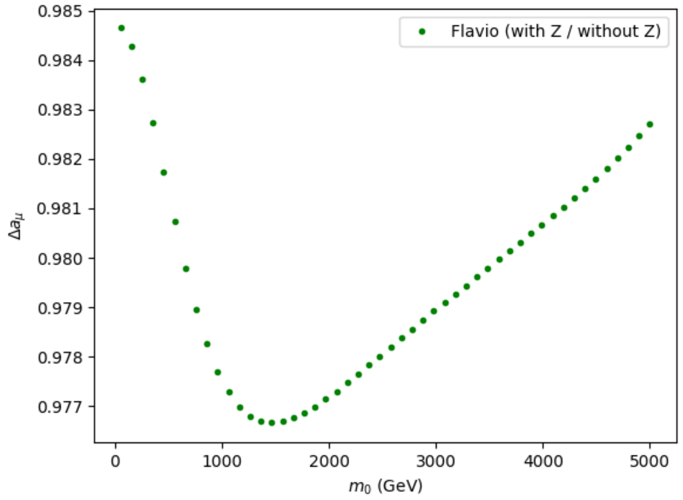
Only SMEFT running



Full running



Z contribution



PRELIMINARY RESULTS: PROTOSLEPTON MODEL

Couplings			Particles		
Symbol	Definition	SM?	Symbol	Definition	SM?
Y_d	Yukawa coupling to down-type quarks	Yes	H	SM Higgs doublet	Yes
Y_e	Yukawa coupling to leptons	Yes	d_R, e_R, u_R	SM right-handed singlets	Yes
Y_u	Yukawa coupling to up-type quarks	Yes	q, l	SM left-handed doublets	Yes
T_e	ProtoSlepton coupling to sleptons	No	S_{e_L}	ProtoSlepton left-handed doublet	No
Y_{LB}	ProtoSlepton coupling between left-handed sleptons and leptons	No	S_{e_R}	ProtoSlepton right-handed singlet	No
Y_{RB}	ProtoSlepton coupling between right-handed sleptons and leptons	No	$\widetilde{B}^0$	ProtoSlepton fermion field	No
M_B	Mass of ProtoSlepton fermion field	No			

$$\mathcal{L} = -(Y_d \overline{H} \cdot \overline{d}_R \cdot q + Y_e \overline{H} \cdot \overline{e}_R \cdot l + Y_u \overline{u}_R \cdot q \cdot H + T_e S_{e_L} \cdot \overline{H} \cdot S_{e_R} + Y_{LB} \overline{S}_{e_L} \cdot l \cdot \widetilde{B}^0 + Y_{RB} \overline{S}_{e_R} \cdot \overline{e}_R \cdot \widetilde{B}^0 + \frac{M_B}{2} \widetilde{B}^0 \cdot \widetilde{B}^0)$$

- Simplified version of MSSM
  - Contains only sleptons and one vector
- Fewer new particles means cleaner results

PRELIMINARY RESULTS: PROTOSLEPTON MODEL

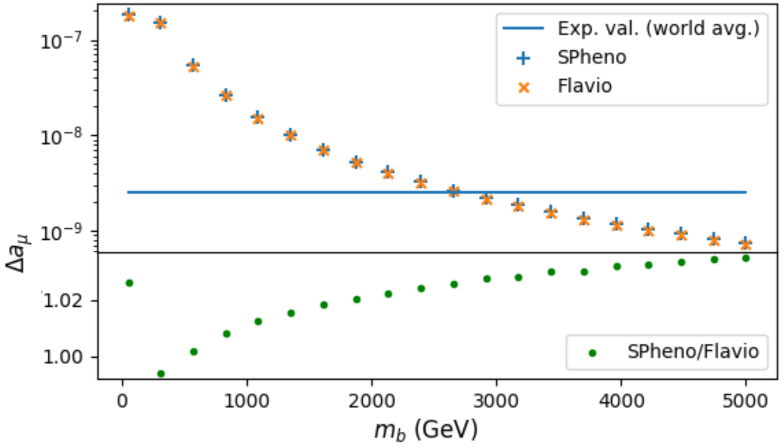
Expected:

- `Flavio` result diverges as energy increases
- `Flavio` contributes extra factor of about 10% from running

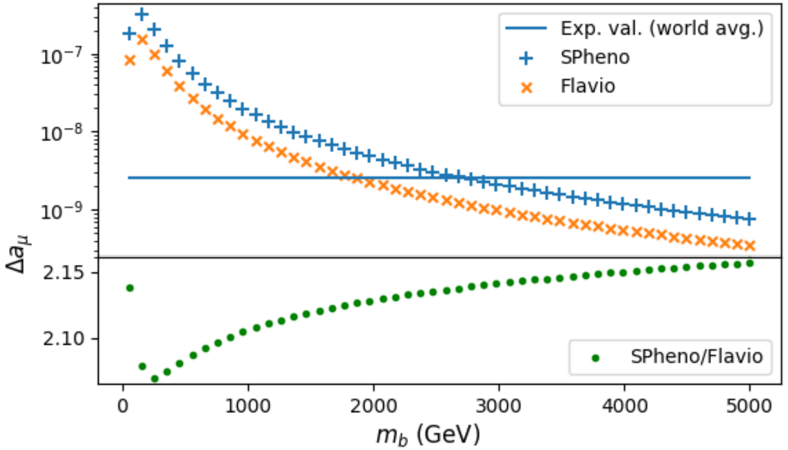
Actual:

- `Flavio` result is multiplied by 0.48
- Difference from `SPheno` is too large
- Waiting for `Wilson` authors to fix bug

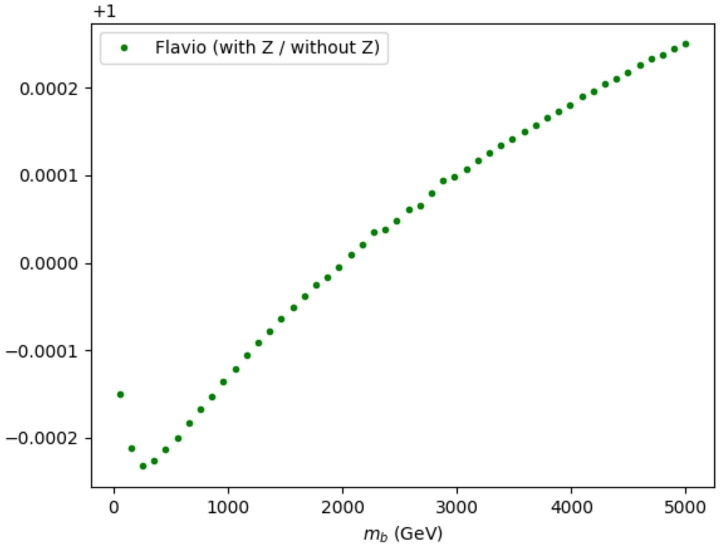
Only SMEFT running



Full running



Z contribution



ONGOING WORK

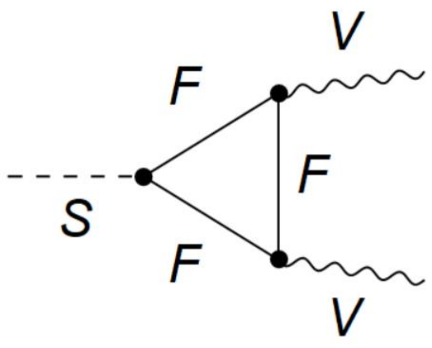
- Ensure RG running is being performed correctly by `wilson`
- Fully replace `pre-SARAH` with `SARAH-Prelude`
- Apply method to calculation of EDMs
- Expand results to include one-loop contributions
  - Introduces new contributions to RG running compared to `SPheno`
  - First time this will have been done at one-loop level generically (that we are aware of)
- Apply results to models proposed to explain latest $g-2$ result

CONCLUSION

- Calculations of $g-2$ and EDMs are extremely important for BSM physics
- `SARAH` is able to perform these calculations completely generically
- Using `Flavio` allows us to account for running effects more accurately
- Replacing `Pre-SARAH` with `SARAH-Prelude` allows us to include a larger number of operators (eventually any)
- Work has potential to be expanded much further so it can be used for a vast array of problems

SARAH-PRELUDE OUTPUT: MATHEMATICA PACKAGE

Includes information about each coupling in each topology




```
GenerateSMEFTprocess[SVV][processName_,S1_,V1_,V2_]:=Block[{CurrentTop, FunctionName, NumStructs},

 numberOfSMEFTStructures[processName]=3;


(*Diagram No. 1*)

CurrentTop={{{FieldToInsert[F, 2], FieldToInsert[F, 3], External[3]},
             {AntiField[FieldToInsert[F, 3]], FieldToInsert[F, 1], External[2]},
             {AntiField[FieldToInsert[F, 1]], AntiField[FieldToInsert[F, 2]], External[1]}},
             {External[1] -> S1, External[2] -> V1, External[3] -> V2,
              Internal[1] -> FieldToInsert[F, 1], Internal[2] -> FieldToInsert[F, 2], Internal[3] -> FieldToInsert[F, 3],
              Index[1] -> gt1, Index[2] -> gt2, Index[3] -> gt3}};
FunctionName="SMEFT_SVV_top1_FFF";
AddDiagramsSMEFT[processName, CurrentTop, FunctionName];


(*Diagram No. 2*)
```

`SARAH-Prelude` output: Fortran Module

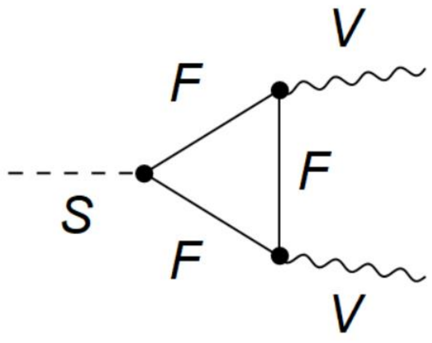
```
! --------------------
! Diagram No. 1
! --------------------

Subroutine SMEFT_SVV_top1_FFF(mExt1, mExt12, mExt2, mExt22, mExt3, mExt32, &
                        & mLoop1, mLoop12, mLoop2, mLoop22, mLoop3, mLoop32, &
                        & coup1L, coup1R, coup2L, coup2R, coup3L, coup3R, amps)
Implicit None
Real(dp), Intent(in) :: mExt1, mExt12, mExt2, mExt22, mExt3, mExt32, &
                        & mLoop1, mLoop12, mLoop2, mLoop22, mLoop3, mLoop32
Complex(dp), Intent(out) :: amps(3)
Complex(dp), Intent(in) :: coup1L, coup1R, coup2L, coup2R, coup3L, coup3R
Complex(dp) :: int1, int2, int3, int4, int5, int6, int7

amps = 0._dp

! Necessary loop integrals

int1=SA_B0(mExt32, mLoop22, mLoop32)
int2=SA_Cget("C0", mExt12, mExt32, mExt22, mLoop12, mLoop22, mLoop32)
int3=SA_Cget("C00", mExt12, mExt32, mExt22, mLoop12, mLoop22, mLoop32)
int4=SA_Cget("C1", mExt12, mExt32, mExt22, mLoop12, mLoop22, mLoop32)
int5=SA_Cget("C11", mExt12, mExt32, mExt22, mLoop12, mLoop22, mLoop32)
int6=SA_Cget("C12", mExt12, mExt32, mExt22, mLoop12, mLoop22, mLoop32)
int7=SA_Cget("C2", mExt12, mExt32, mExt22, mLoop12, mLoop22, mLoop32)
```




```
! Coefficients of Lorentz structures

! Coefficient of Eps[ec[2], ec[3], k[1], k[2]]
amps(1)= &
& -2*coup1L*coup2L*coup3L*int2*mLoop1 &
& +2*coup1R*coup2R*coup3R*int2*mLoop1 &
& -2*coup1L*coup2L*coup3L*int4*mLoop1 &
& +2*coup1R*coup2R*coup3R*int4*mLoop1 &
& -2*coup1L*coup2L*coup3L*int7*mLoop1 &
& +2*coup1R*coup2R*coup3R*int7*mLoop1 &
& -2*coup1R*coup2L*coup3L*int4*mLoop2 &
& +2*coup1L*coup2R*coup3R*int4*mLoop2 &
& +2*coup1L*coup2R*coup3L*int7*mLoop3 &
& -2*coup1R*coup2L*coup3R*int7*mLoop3

! Coefficient of Pair[ec[2], ec[3]]
amps(2)= &
```