

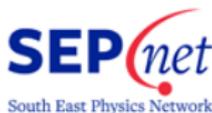
# Hunting for Minimal Walking Technicolor using $Z'$ searches at the LHC

Azaria Coupe

University of Southampton

11th May 2017

Collaborators: Alexander Belyaev, Mads Frandsen, Claire Shepherd

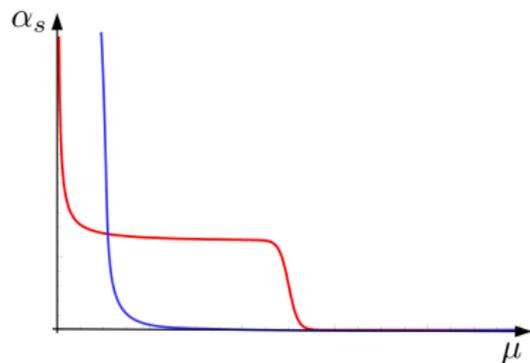


# Project Overview

- We are exploring **Walking Technicolor** (WTC) as a dynamical alternative to spontaneous EWSB
- WTC also addresses hierarchy problem and has a consistent Higgs boson-like *composite* particle
- WTC has interesting phenomenology - exploring heavy neutral resonances using **CalcHEP**
- Constraining the WTC parameter space - calculating  $\sigma^{DY}(Z'/Z'')$  through **HEPMDB** interface
- Random scan of cross section in 4-D parameter space using C code running over CalcHEP model

# WTC Overview

WTC is a strongly interacting theory with interacting bound states of techni-fermions at TeV scale.



Coupling  $\alpha_S$  'walks' between scale of Technicolor,  $\Lambda_{TC}$  up to some Extended TC scale  $\Lambda_{ETC}$  (red)

NMWT has global symmetry  $SU(2)_L \times SU(2)_R \times U(1)_Y$ .

We have **two gauge triplets** in TC sector under *hidden local symmetry* in the EFT. Physical particles are  $Z'$ ,  $W'^{\pm}$  and  $Z''$ ,  $W''^{\pm}$ .

# Setup of NMWT

NMWT is encoded into the chiral Lagrangian (low energy description)

$$\begin{aligned}\mathcal{L}_{boson} = & -\frac{1}{2}\text{Tr}[\tilde{W}_{\mu\nu}\tilde{W}^{\mu\nu}] - \frac{1}{4}\tilde{B}_{\mu\nu}\tilde{B}^{\mu\nu} - \frac{1}{2}\text{Tr}[F_{L\mu\nu}F_L^{\mu\nu} + F_{R\mu\nu}F_R^{\mu\nu}] \\ & + m^2\text{Tr}[C_{L\mu}^2 + C_{R\mu}^2] + \frac{1}{2}\text{Tr}[D_\mu M D^\mu M^\dagger] - \tilde{g}^2 r_2 \text{Tr}[C_{L\mu} M C_{R\mu}^\dagger M^\dagger] \\ & - \frac{i\tilde{g}r_3}{4}\text{Tr}[C_{L\mu}(M D^\mu M^\dagger - D^\mu M M^\dagger) + C_{R\mu}(M^\dagger D^\mu M - D^\mu M^\dagger M)] \\ & + \frac{\tilde{g}^2 s}{4}\text{Tr}[C_{L\mu}^2 + C_{R\mu}^2]\text{Tr}[M M^\dagger] + \frac{\mu^2}{2}\text{Tr}[M M^\dagger] - \frac{\lambda}{4}\text{Tr}[M M^\dagger]^2\end{aligned}\quad (1)$$

Key constructs in equation 1:

$$C_{L\mu} \equiv A_{L\mu} - \frac{g}{\tilde{g}}\tilde{W}_\mu, \quad C_{R\mu} \equiv A_{R\mu} - \frac{g'}{\tilde{g}}\tilde{B}_\mu, \quad (2)$$

Lagrangian parameters:  $m, r_2, r_3, s, \tilde{g}$ .

Rephrase parameters in terms of vector-vector and axial-vector masses and decay constants

$$M_V^2 = m^2 + \frac{\tilde{g}^2(s - r_2)v^2}{4}, \quad M_A^2 = m^2 + \frac{\tilde{g}^2(s + r_2)v^2}{4}. \quad (3)$$

Use the Weinberg Sum Rules to connect effective  $\mathcal{L}$  to underlying theory

$$S = 4\pi \left[ \frac{F_V^2}{M_V^2} - \frac{F_A^2}{M_A^2} \right], \quad (4)$$

where  $F_V, F_A$  are vector and axial decay constants.  
NMWT defined by 4-D parameter space

$$\boxed{M_A, \quad \tilde{g}, \quad S, \quad s.} \quad (5)$$

## Current progress

- Updated LanHEP model for Next to Minimal Walking Technicolor
- Replacing external code for Mass Matrix Diagonalisation with inbuilt SLHA diagonalisation functions in LanHEP
- Optimising field definitions in LanHEP in preparation for NMWT with  $N_D > 1$  techni-fermion doublets
- Ran C code scan to calculate  $Z'/Z''$  properties (widths, branchings, couplings etc.) across 4-D parameter space grid
- Used HEPMDB interface to calculate  $\sigma(pp \rightarrow Z'/Z'' \rightarrow e^+e^-)$  across same grid
- Writing/running C code to create  $\sim 10\text{M}$  point random scan over 4-D space

# Updating NMWT model - SLHA+ Diagonalisation

EW and TC gauge particles mix, gauge mixing matrix is

$$\mathcal{M}_N^2 = \begin{pmatrix} \frac{g_1^2}{4}(f^2 + av^2) & -\frac{g_1 g_2}{4}bv^2 & \frac{g_1 \tilde{g}}{4\sqrt{2}}(f^2 + cv^2) & -\frac{g_1 \tilde{g}}{4\sqrt{2}}(f^2 + dv^2) \\ -\frac{g_1 g_2}{4}bv^2 & \frac{g_2^2}{4}(f^2 + av^2) & -\frac{g_2 \tilde{g}}{4\sqrt{2}}(f^2 + cv^2) & -\frac{g_2 \tilde{g}}{4\sqrt{2}}(f^2 + dv^2) \\ \frac{g_1 \tilde{g}}{4\sqrt{2}}(f^2 + cv^2) & -\frac{g_2 \tilde{g}}{4\sqrt{2}}(f^2 + cv^2) & \frac{\tilde{g}^2}{4}(f^2 - dv^2) & 0 \\ -\frac{g_1 \tilde{g}}{4\sqrt{2}}(f^2 + dv^2) & -\frac{g_2 \tilde{g}}{4\sqrt{2}}(f^2 + dv^2) & 0 & \frac{\tilde{g}^2}{4}(f^2 + dv^2) \end{pmatrix} \quad (6)$$

where  $a = (1 - r_3 + s)$ ,  $b = (1 + r_2 - r_3)$ ,  $c = (r_2 - r_3 + s)$  and  $d = (-r_2 + s)$ .

Original NMWT model has external  $rotate_f.f$  functions that rotate  $\mathcal{M}_N^2$  and  $\mathcal{M}_G^2$ . These can be replaced by **SLHA+** diagonalisation procedure.

# SLHA+ Diagonalisation - LanHEP

```
% SLHA+ functions
external_func(rDiagonal,*).
external_func(MassArray,*).
external_func(initDiagonal,0).
external_func(MixMatrix,*).
external_func(MixMatrixU,*).

parameter zero = initDiagonal.

parameter nDiag = rDiagonal(4,VN11,VN12,VN13,VN14,
                             VN22,VN23,VN24,
                             VN33,VN34,
                             VN44).

parameter masssph = MassArray(nDiag,1).
parameter massZ   = MassArray(nDiag,2).
parameter massZp  = MassArray(nDiag,3).
parameter massZpp = MassArray(nDiag,4).

parameter Mph = sqrt(fabs(masssph)),
           MZ_d = sqrt(fabs(massZ)).

parameter MZp = sqrt(fabs(massZp)),
           MZpp = sqrt(fabs(massZpp)).

_i=1-4 in _j=1-4 in parameter ZpD_i_j=MixMatrix(nDiag,_i,_j).

OrthMatrix ({{ZpD11,ZpD12,ZpD13,ZpD14},
             {ZpD21,ZpD22,ZpD23,ZpD24},
             {ZpD31,ZpD32,ZpD33,ZpD34},
             {ZpD41,ZpD42,ZpD43,ZpD44}}).
```

# Current Challenges - SLHA+

Implementing diagonalisation as shown reproduces correct mass spectrum for gauge particles - success!

However, widths and branching ratios **not** correct - comparison of old and new versions at parameter space values of  $M_A = 500\text{GeV}$ ,  $\tilde{g} = 5$ ,  $S = 0.1$ ,  $s = 0.0$ ;

```
Particle Information
Patricle 00(00), PDG = 1000023, Mass= 5.026E+02 Width=1.59E-01
Quantum numbers: spin=1, charge(e1.)=0 color=1
Branchings & Decay channels:
6.28E-05 00 -> W+,W-
1.92E-17 00 -> A,H
6.42E-02 00 -> Z,H
6.11E-02 00 -> n1,N1
3.07E-02 00 -> e1,E1
6.11E-02 00 -> n2,N2
3.07E-02 00 -> e2,E2
6.11E-02 00 -> n3,N3
3.07E-02 00 -> e3,E3
1.04E-01 00 -> u,U
1.35E-01 00 -> d,D
1.04E-01 00 -> c,C
1.35E-01 00 -> s,S
4.68E-02 00 -> t,T
1.35E-01 00 -> b,B
```

```
Particle Information
Patricle 00(00), PDG = 1000023, Mass= 5.026E+02 Width=2.74E+01
Quantum numbers: spin=1, charge(e1.)=0 color=1
Branchings & Decay channels:
3.33E-01 00 -> W+,W-
1.07E-01 00 -> A,H
1.85E-01 00 -> Z,H
2.75E-04 00 -> n1,N1
2.75E-04 00 -> e1,E1
2.75E-04 00 -> n2,N2
2.75E-04 00 -> e2,E2
2.75E-04 00 -> n3,N3
2.93E-04 00 -> e3,E3
8.24E-04 00 -> u,U
8.24E-04 00 -> d,D
8.53E-04 00 -> c,C
8.25E-04 00 -> s,S
3.68E-01 00 -> t,T
1.14E-03 00 -> b,B
```

SLHA+ version of the model (right) needs extensions to diagonalisation *.lhep* file to match procedure in *rotate<sub>f</sub>*.

## Extending to $N_D > 1$

To write LanHEP model of WTC with  $SU(2N_D)_L \otimes SU(2N_D)_R$  where  $N_D > 1$ , need to optimise LanHEP model to calculate with increased number of fields/lagrangian terms.

Original

```
let AL1 = {('V0+'+'A0+'+'V0-'+'A0-')/2, i*('V0+'+'A0+'-'+'V0-'+'A0-')/2, ('V03+'+'A03')/Sqrt2}.
```

```
let AR1 = {('V0+'+'A0+'+'V0-'+'A0-')/2, i*('V0+'+'A0+'-'+'V0-'+'A0-')/2, ('V03+'+'A03')/Sqrt2}.
```

Optimised

```
_x=1-4 in let ALo__x = {('Wpf-__x')/Sqrt2,('Zpf__x')/Sqrt2,('Wpf+__x')/Sqrt2}.
```

```
_x=1-4 in let ARo__x = {('Wppf-__x')/Sqrt2,('Zppf__x')/Sqrt2,('Wppf+__x')/Sqrt2}.
```

Optimised field definitions (right) reduce LanHEP run time by factor of  $\sim 16$ .

# Calculating with CalcHEP

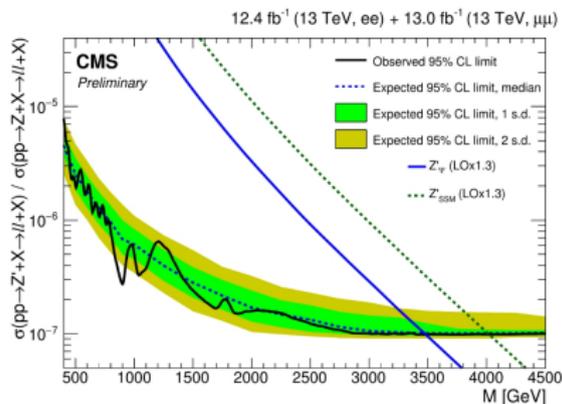
CalcHEP can be utilised through user defined C code - using this method to calculate properties of  $Z'$  and  $Z''$  in our NMWT model.

Built in CalcHEP functions can be written in to user scan to collect desired properties

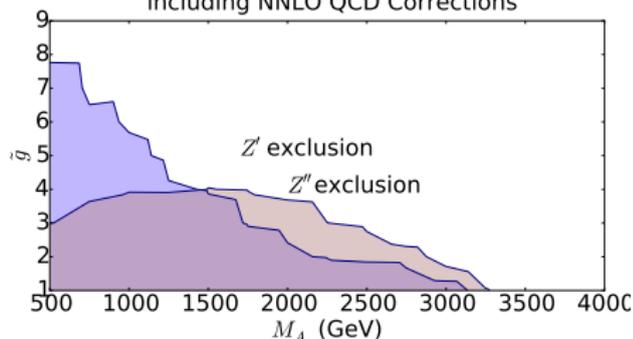
CalcHEP function	Output
<code>assignValW("Parameter name", Param)</code>	Parameter Value
<code>pWidth("Particle name", &amp;branchings)</code>	$\Gamma_{Particle}$
<code>findBr(branchings,"X,Y")</code>	$\text{Br}(P \rightarrow X, Y)$
<code>findValW("Constraint name")</code>	Constraint value

# HEPMDB and IRIDIS

We calculate the theoretical cross section  $\sigma(pp \rightarrow Z'/Z'' \rightarrow e^+e^-)$  across the 4-D NMWT parameter space using **IRIDIS4**, through the **HEPMDB** interface.



Exclusion on  $M_A, \tilde{g}$  from  $pp \rightarrow Z'/Z'' \rightarrow l^+l^-$   
for  $\sqrt{s}=13\text{TeV}$  at  $S=0.1, s=0$ ,  
including NNLO QCD Corrections



Exclusion region defined by region where  $\sigma^{theory} > \sigma^{exp}$ ,  $Z'$  and  $Z''$  exclude complementary regions of the NMWT parameter space.

## Calculating with CalcHEP

We want to repeat previous  $\sigma(pp \rightarrow Z'' \rightarrow e^+e^-)$  calculation for  $\sim 10\text{M}$  randomly generated points in the 4-D parameter space.

Can significantly reduce calculation time by splitting  $\sigma$  into parts;

$$\sigma(pp \rightarrow Z' \rightarrow e^+e^-) = Br(Z' \rightarrow e^+e^-)(c_u^2 + c_d^2) \sum_{q\bar{q}=u\bar{u}}^{b\bar{b}} w(PDF)_{q\bar{q}}. \quad (7)$$

PDF-functions for each subprocess can be extracted by finding coefficients of

$$w(PDF)_{q\bar{q}} = \sigma_{q\bar{q}}(M_{Z'}) = a \exp^{-bM_{Z'}} + c. \quad (8)$$

Calculate couplings  $c_u, c_d$ , branching ratio, and  $w(PDF)_{q\bar{q}}$  then reconstruct 7 in C code.

# Generic $\sigma(2 \rightarrow 2)$ C Code

```
static double integrand(double *x, double w)
{
    double cs=2*x[0]-1; // cosine of scattering angle
    double J=2; // Jacobian
    REAL sn=sqrt(1-cs*cs);
    REAL pin=Pcm*sqrt(x[1]*x[2]);
    REAL e1=sqrt(m[0]*m[0]+pin*pin),e2=sqrt(m[1]*m[1]+pin*pin);
    REAL s_hat= pow(e1+e2,2);
    REAL ms=m[2]+m[3],md=m[2]-m[3];
    REAL pout=sqrt((s_hat - ms*ms) * (s_hat - md*md))/(2*sqrt(s_hat));
    REAL pvec[16];
    for(int i=0;i<16;i++) pvec[i]=0;
    // p1
    pvec[0]= e1;
    pvec[3]= pin;
    // p2
    pvec[4]= e2;
    pvec[7]= -pin;
    // p3
    pvec[8]= sqrt(pout*pout + m[2]*m[2]);
    pvec[10]= pout*sn;
    pvec[11]= pout*cs;
    if(pvec[10]<PtCut) return 0;
    // p4
    pvec[12]= sqrt(pout*pout + m[3]*m[3]);
    pvec[14]= -pvec[10];
    pvec[15]= -pvec[11];

    double Scale=sqrt(s_hat);
    Scale=100;
    double GG=sqrt(alphaspdf_(&Scale)*4*M_PI);
    double totcoef=3.8937966E8 * pln / (32.0 * M_PI * pout * s_hat);
    int err=0;
    double dSigmaCos=totcoef*cc->interface->sqme(1,GG,pvec,NULL,&err);
    double f1=0,f2=0,ff[14];
```

```
    evolvpdf_(&(x[1]),&Scale,ff);
    switch(pdg[0])
    {
        case 2 : f1=ff[8]; break;
        case 1 : f1=ff[7]; break;
        case 3 : case -3 : f1=ff[9]; break;
        case 4 : case -4 : f1=ff[10]; break;
        case 5 : case -5 : f1=ff[11]; break;
        case 21 : case -21 : f1=ff[6]; break;
        case -1 : f1=ff[5]; break;
        case -2 : f1=ff[4]; break;
    }

    evolvpdf_(&(x[2]),&Scale,ff);
    switch(pdg[1])
    {
        case 2 : f2=ff[8]; break;
        case 1 : f2=ff[7]; break;
        case 3 : case -3 : f2=ff[9]; break;
        case 4 : case -4 : f2=ff[10]; break;
        case 5 : case -5 : f2=ff[11]; break;
        case 21 : case -21 : f2=ff[6]; break;
        case -1 : f2=ff[5]; break;
        case -2 : f2=ff[4]; break;
    }

    return f1/x[1]*f2/x[2]*dSigmaCos*J;
}
```

# Generic $\sigma(2 \rightarrow 2)$ C Code

```
int main(void)
{ int err;

// Specify model for work: Model directory and model number.
  setModel("models" ,2 );

// SQME code generation
  cc= newProcess("u,U->m,M");

// Calculation of public constraints
  err=calcMainFunc();
  if(err) { printf("Can not calculate constrained parameter %s\n",varNames[err]);return err;}

  err=passParameters(cc);
  if(err) { printf("Can not calculate constrained parameter %s\n",cc->interface->varName[err]); return err;}

/* find masses and PDG codes */
  for(int i=0;i<4;i++) cc->interface->pinf(1,i+1,m+l,pdg+l);

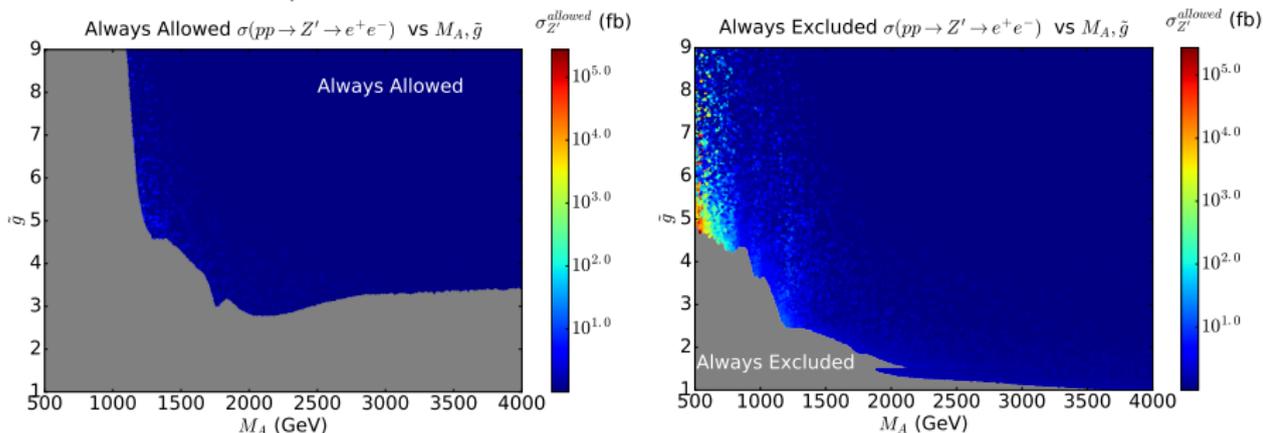
  Pcm=6500; // Energy of LHC Run 2
  PtCut=200; // Cut for transverse momentum

  char * pdfName="cteq6l1";
  initpdfsetbyname(pdfName, strlen(pdfName));
  { double x=0.1,q=100,ff[14]; evolvepdf_( &x,&q,ff);alphaspdf_(&q); } // to avoid PDF initialisation in multi-core calcula
// integration by Vegas
  vegasGrid *vegPtr=vegas_init(3/*dimension of integration*/,integrand,50);
// First Vegas call
  double ti,dti;
  vegas_int(vegPtr, 10000 // number of calls
            , 1.5 // grid improving parameter
            , 4 // number of processors
            , &ti // evaluated integral
            , &dti // statistical uncertainty
            );
  printf("Cross section %.2E +/- %.1E\n",ti,dti);
// Vegas call with improved grid
  vegas_int(vegPtr, 10000, 1.5,4, &ti , &dti);
  printf("Cross section %.2E +/- %.1E\n",ti,dti);

  return 0;
}
```

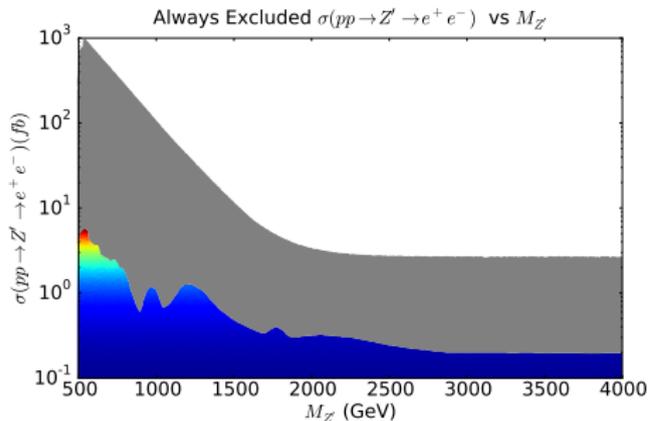
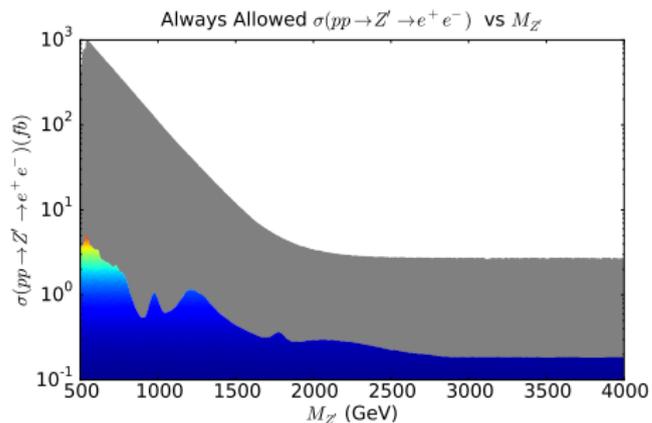
# AA and AE: $M_A - \tilde{g}$

Using IRIDIS we scan  $\sim 10\text{M}$  points to set up Always Allowed and Always Excluded regions by layering excluded points on top of allowed for AA, and vice versa for AE.



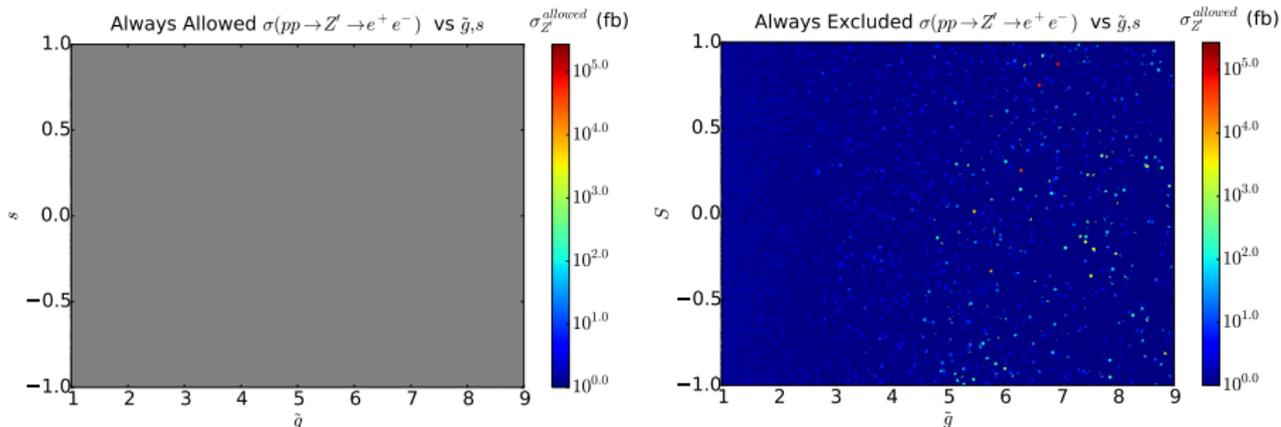
Current AA and AE regions based on  $Z'$  cross section.

# AA and AE: $\sigma$ vs. $M_{Z'}$



# Implications for $s$ dependence

AA and AE regions in the  $s - \tilde{g}$  parameter space have been constructed and show unexpected behaviour,



Expect factor  $> 2$  increase in  $\text{Br}(Z' \rightarrow e^+ e^-)$  due to  $s$  in certain parameter space region - would lead to some region that is always excluded. What is compensating for this increase due to  $s$ ?

## Prospects/Future Work

- Correct the SLHA+ diagonalisation procedure of NMWT LanHEP model to reproduce correct branching ratios
- Develop NMWT with  $N_D = 2$  model in LanHEP
- Produce combined  $Z'$ ,  $Z''$  random scan AA and AE plots
- Explore  $s$  parameter phenomenology further
- Predicted exclusions for increased LHC luminosity