Karlsruher Institut für Technologie
Institute for Theoretical Physics

Diving deep into the (N)MSSM potential: ML-enhanced analysis of vacuum stability

Work in collaboration with Fabio Campello and Georg Weiglein

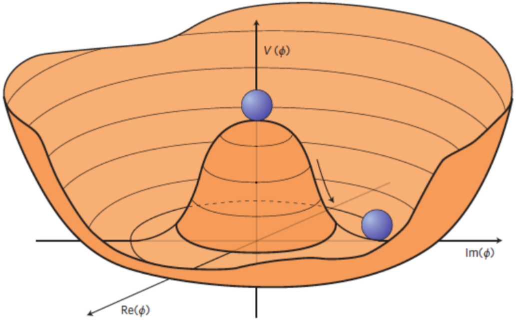
SUSY 2024 at the IFT in Madrid

June 14th 2024

Thomas Biekötter

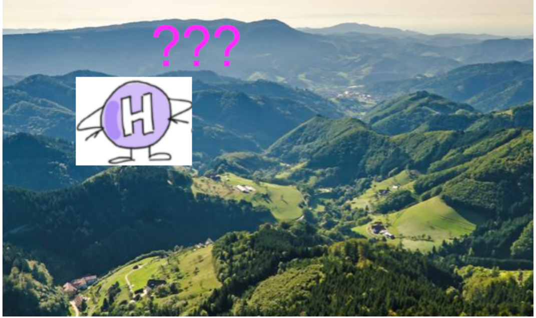
Vacuum stability: SM vs. Susy

Standard Model



[1504.07217]

Supersymmetry



Vacuum stability analysis

Idea: In models with extended scalar sectors the EW vacuum might not correspond to the global minimum of the potential.

→ The EW vacuum is **not stable**

→ If the EW vacuum is **short-lived**[1] a parameter point is **unphysical**

→ **Constraints** on the parameter space of the model

Vacuum-stability constraints in the MSSM and the NMSSM

We improve the computation of vacuum decay rates using neural networks

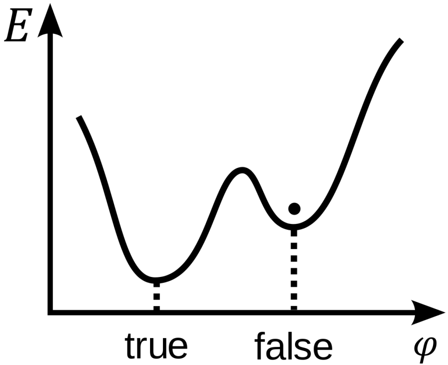
[1] in comparison to the age of the Universe

Vacuum stability analysis

Karlsruher Institut für Technologie

Step 1: Is the EW vacuum meta-stable?

→ Determination of all stationary points of the scalar potential below the EW minimum.



Vacuum stability analysis

Step 1: Is the EW vacuum meta-stable?

→ Determine all stationary points of the scalar potential below the EW minimum.

Step 2: What is the lifetime of the EW vacuum?

→ Compute the vacuum decay rates for each possible transition

$$\frac{\Gamma}{V} = K \mathrm{e}^{-S_E}$$

[Coleman, 1977]

S_E: Euclidean bounce action

Vacuum stability analysis

Step 1: Is the EW vacuum meta-stable?

→ Determine all stationary points of the scalar potential below the EW minimum.

Step 2: What is the lifetime of the EW vacuum?

→ Compute the vacuum decay rates for each possible transition

Step 3: Is the EW vacuum sufficiently long-lived?

→ Compare the smallest inverse decay rate (lifetime) to the age of the universe

$S_E < 390$: Short-lived EW vacuum, unphysical

$390 < S_E < 440$: Uncertain fate of the EW vacuum

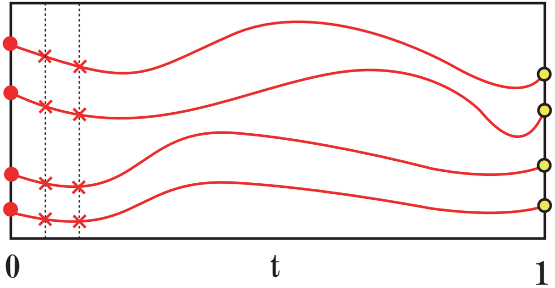
$S_E > 440$: Long-lived (meta-stable) EW vacuum, physical

[W. Hollik, G. Weiglein, J. Wittbrodt: 1812.04644]

Homotopy continuation method

Solutions of a system of polynomial equations can be found using homotopy continuation methods.
→ Can be used to determine all stationary points $\vec{F}(\vec{z}) = 0$ of tree-level scalar potentials

$$\vec{H}(\vec{z}, t) = (1 - t) \cdot \vec{F}(\vec{z}) + t \cdot \vec{g}(\vec{z}) = 0 \ , \quad \text{solutions of } \vec{g}(\vec{z}) = 0 \text{ are known.}$$

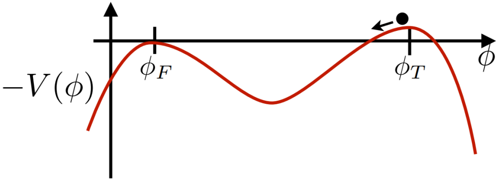


[Kobayashi, Wynn, 1310.6515]

We use `HOM4PS2 v.2`: Polyhedral homotopy continuation [Lee, Li, Tsai, 2008]

The bounce solution: 1 field

The bounce solution is the key object describing the vacuum transition.



[C.L. Wainwright, 1109.4189]

$$\frac{d^2\vec{\phi}}{d\rho^2} + \frac{3}{\rho}\frac{d\vec{\phi}}{d\rho} = \vec{\nabla}_\phi V(\vec{\phi})$$

with $\rho = r^2 - t^2$

Boundary conds.:

$$\vec{\phi}(\infty) = \vec{\phi}_F \ , \qquad \left.\frac{d\vec{\phi}}{d\rho}\right|_{\rho=0} = 0$$

In one dimensions this can be solved using the **overshoot-undershoot method**.

Shooting methods: [e.g. Stoer, Bulirsch, 1972]

$$\textbf{Bounce action:}\ S_E = \frac{2\pi^2}{\Gamma(2)} \int \rho^3 d\rho \left[\frac{1}{2}(\dot{\vec{\phi}}(\rho))^2 + \Delta V(\vec{\phi}(\rho))\right]$$

Bounce solution: Multi-field case

Important obstacle in many dimensions is **finding the correct tunneling path** in field space.

Method 1: Straight path approximation
Compute the bounce action assuming that the true tunneling path is well approximated by a straight path connecting the true and false minima.

Method 2: Path deformation algorithm
Iterative procedure that deforms the straight path into the correct path by minimizing *perpendicular forces* along the path (`cosmoTransitions`).

Method 3: Using neural network (NN) to solve differential equations
Transform the problem of solving a system of differential equations into a minimization problem. Then use NN to minimize the loss function.

Methods 2 and 3 now implemented into a new version of `EVADE`.

Many public codes: `cosmoTransitions`, `AnyBubble`, `FindBounce`, `SimpleBounce`, `BSMPT-v.3`, ...

Neural-network method

Bounce equations are a system of coupled differential equations: [Callan, Coleman, Curtis, 1977]

$$\frac{d^2\vec{\phi}}{d\rho^2} + \frac{3}{\rho}\frac{d\vec{\phi}}{d\rho} - \vec{\nabla}_\phi V(\vec{\phi}) = 0 \qquad \text{BC: } \vec{\phi}_B(\infty) = \vec{\phi}_{\text{false}} \quad \text{and} \quad \left.\frac{d\vec{\phi}_B}{d\rho}\right|_{\rho=0} = 0$$

Loss function L for neural network with set of weights and biases $\{w, b\}$, simplest way:

Loss = "squared sum" of discretized diff. eqs. + squared BC

$$\mathcal{L}(\{w,b\}) = \sum_{i=1}^{n_\rho}\left(\left.\frac{d^2\phi}{d\rho^2}\right|_{\rho_i} + \frac{3}{\rho_i}\left.\frac{d\phi}{d\rho}\right|_{\rho_i} - \nabla V(\phi(\rho_i))\right)^2 + n_\rho\left[\left(\left.\frac{d\phi}{d\rho}\right|_{\rho=0}\right)^2 + \left(\phi(\rho_{\text{max}}) - \phi_{\text{false}}\right)^2\right]$$

[Piscopo, Spannowsky, Waite, 1902.05563]

Neural network: Trained to minimze $\mathcal{L}$ (adam optimizer)

One neuron in input layer (ρ) and $N = \dim(\vec{\phi})$ neurons in output layer

Depending on model 3 to 5 inner layers with 20 to 50 neurons each

CPU and GPU implementation (`tensorflow`, `CUDA`, `XLA`)

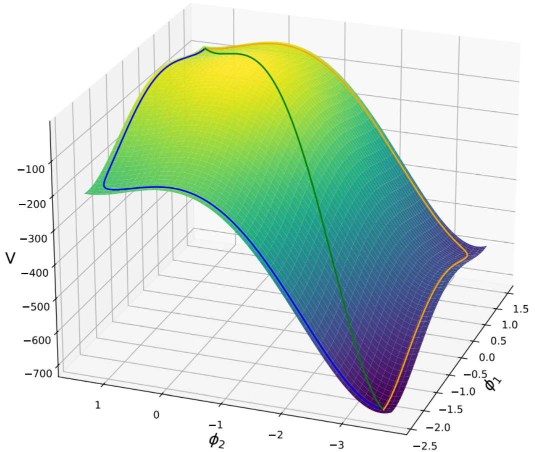
Toy model: 2 singlet fields

$$V_{\text{toy}} = \sum_{i=1}^{N=2} \lambda_i \phi_i^4 + A_i \phi_i^3 - m_i^2 \phi_i^2$$

Straight-path approximation →

Path-deformation algorithm →

Neural-network approach →



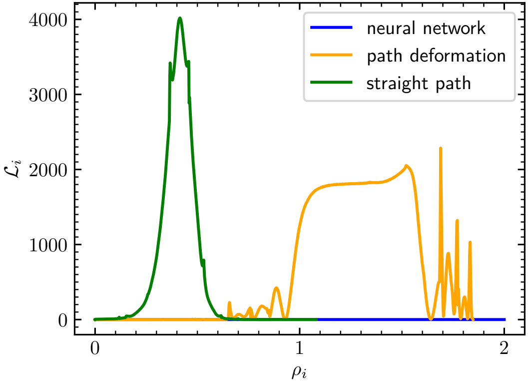
Only the neural-network approach is able to determine the correct tunneling path

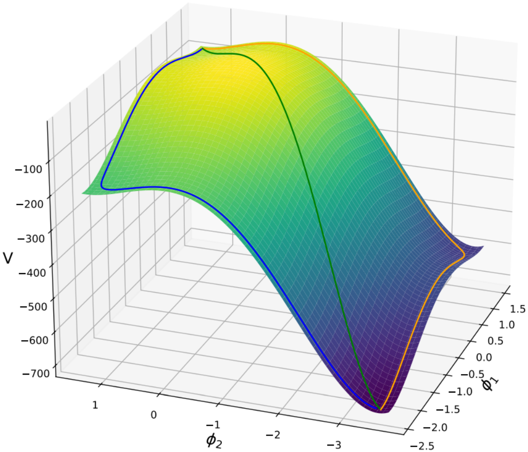
	B_1	B_2	B_3	$\mathcal{L}_{\text{bc}}$
Neural network	107.24	107.26	107.28	0.0915
Straight path	47.78	47.57	47.37	0.8215
Path deformation	−15332	3060	21452	73570

$\mathcal{L}_{\text{bc}}$: quality measure (smaller is better)

[TB, F. Campello, G. Weiglein, tbp]

Toy model: 2 singlet fields





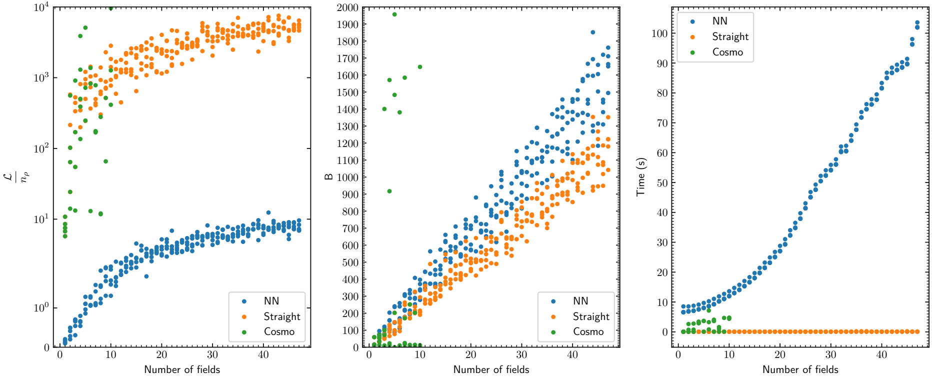
If $\mathcal{L}_i = \mathcal{L}(\rho_i) > 0$, the predicted solution $\vec{\phi}(\rho)$ does not satisfy the bounce equations at $\rho = \rho_i$

$\rightarrow$ Neural network found correct solution

[TB, F. Campello, G. Weiglein, tbp]

Toy model: many singlet fields

Loss | Bounce action | Runtime (NN on GPU)



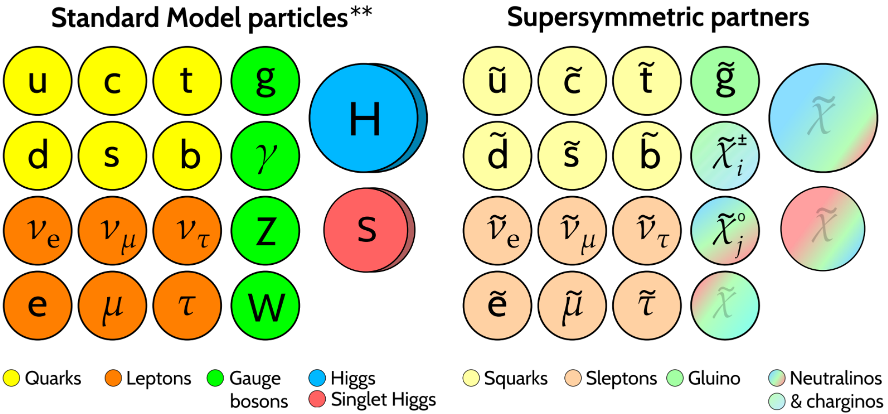
[TB, F. Campello, G. Weiglein, tbp]

For sizeable number of fields only the neural network is able to determine the bounce solution

$\rightarrow$ suitable method to analyze vacuum stability in SUSY models

The NMSSM

NMSSM: Next-to Minimal Supersymmetric Standard Model: MSSM + singlet



[Slide from Christoph Borschenksy]

Scalar potential of the NMSSM

Susy gives a recipe for constructing the scalar potential:

$$V = F + D + V_{\rm soft}$$

Scalar potential of the NMSSM

Susy gives a recipe for constructing the scalar potential:

$$V = F + D + V_{\text{soft}}$$

F-term contributions from the Superpotential:

$$W = \frac{1}{3}\kappa S^3 + \lambda S H_u \cdot H_d + y_t Q_L \cdot H_u \bar{t}_R + y_b H_d \cdot Q_L \bar{b}_R + y_\tau H_d \cdot L_L \bar{\tau}_R$$

$$F = \sum_\phi |\partial_x W|^2 \ , \quad \phi \in \{h_u^0, h_u^+, h_d^0, h_d^-, \tilde{t}_L, \tilde{b}_L, \tilde{\tau}_L, \tilde{\nu}_L, \tilde{t}_R^*, \tilde{b}_R^*, \tilde{\tau}_R^*\}$$

Scalar potential of the NMSSM

Karlsruher Institut für Technologie

Susy gives a recipe for constructing the scalar potential:

$$V = F + D + V_{\text{soft}}$$

D-term contributions from the gauge structure:

$$D = D_{U(1)_Y} + D_{SU(2)_L} + D_{SU(3)_c}$$

$$D_{U(1)_Y} = \frac{g_1^2}{8}\Big(\sum_{\phi} Y_\phi |\phi|^2\Big)^2$$

$$D_{SU(2)_L} = \frac{g_2^2}{8}\sum_{\Phi_i}\sum_{\Phi_j} 2(\Phi_i^\dagger\Phi_j)(\Phi_j^\dagger\Phi_i) - (\Phi_i^\dagger\Phi_i)(\Phi_j^\dagger\Phi_j)$$

$$D_{SU(3)_c} = \frac{g_3^2}{6}\Big(|\tilde{t}_L|^2 - |\tilde{t}_R|^2 + |\tilde{b}_L|^2 - |\tilde{b}_R|^2\Big)^2$$

Scalar potential of the NMSSM

Karlsruher Institut für Technologie

Susy gives a recipe for constructing the scalar potential:

$$V = F + D + V_{\text{soft}}$$

Contributions from soft Susy breaking:

$$\begin{aligned} V_{\mathsf{soft}} = {} & m_S^2 s^\dagger s + m_{H_u}^2 h_u^\dagger h_u + m_{H_d}^2 h_d^\dagger h_d + \left(A_\lambda s h_u \cdot h_d + \frac{1}{3} A_\kappa s^3 + \text{h.c.} \right) \\ & + m_{Q_3}^2 \tilde{Q}_L^\dagger \tilde{Q}_L + m_{L_3}^2 \tilde{L}_L^\dagger \tilde{L}_L + m_{U_3}^2 |\tilde{t}_R|^2 + m_{D_3}^2 |\tilde{b}_R|^2 + m_{E_3}^2 |\tilde{\tau}_R|^2 \\ & + \left(y_t A_t \tilde{t}_R^* \tilde{Q}_L \cdot h_u + y_b A_b \tilde{b}_R^* h_d \cdot \tilde{Q}_L + y_\tau A_\tau \tilde{\tau}_R^* h_d \cdot \tilde{L}_L + \text{h.c.} \right) \ , \end{aligned}$$

Scalar potential of the NMSSM

Karlsruher Institut für Technologie

Susy gives a recipe for constructing the scalar potential:

$$V = F + D + V_{\text{soft}}$$

Physical EW vacuum (the one we want to be in):

$$H_d = \begin{pmatrix} (v_d + h_d + \mathrm{i}a_d)/\sqrt{2} \\ h_d^+ \end{pmatrix} \ , \quad H_d = \begin{pmatrix} h_u^+ \\ (v_u + h_u + \mathrm{i}a_u)/\sqrt{2} \end{pmatrix} \ , \quad s = (v_s + h_s + \mathrm{i}a_s)/\sqrt{2}$$

$$\text{with} \quad v_d, v_u, v_s \in \mathbb{R} \quad \text{and} \quad v^2 = v_d^2 + v_u^2 = 246^2 \ \text{GeV}^2$$

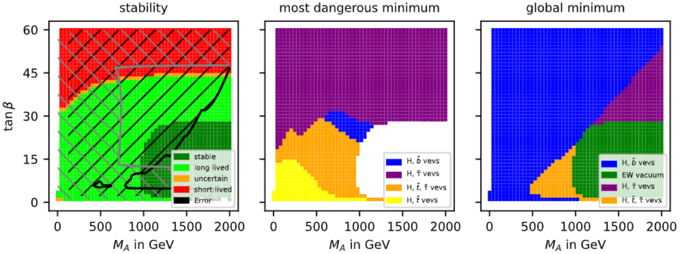
BUT: V is a function in many (field) dimensions with many parameters

$\rightarrow$ In general there can be several local (dangerous) minima below the EW minimum

$M_h^{125}(\tilde{\tau})$ benchmark scenario

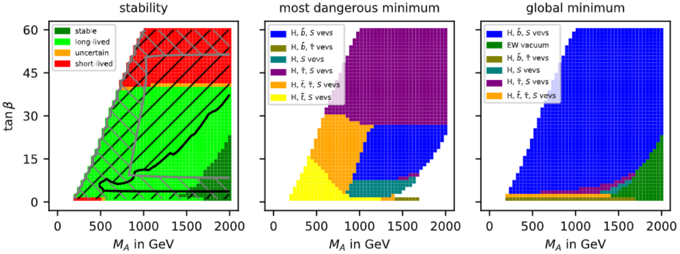
Karlsruher Institut für Technologie

MSSM:



NMSSM:

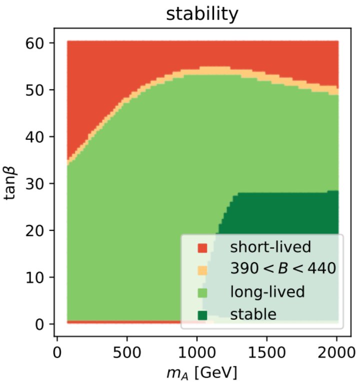
$\lambda = \kappa = 0.1$



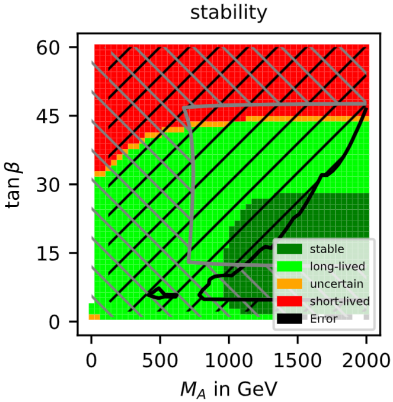
[TB, F. Campello, G. Weiglein, tbp]

$M_h^{125}(\tilde{\tau})$ benchmark scenario

Comparison: straight-path approximation $\leftrightarrow$ true tunneling path



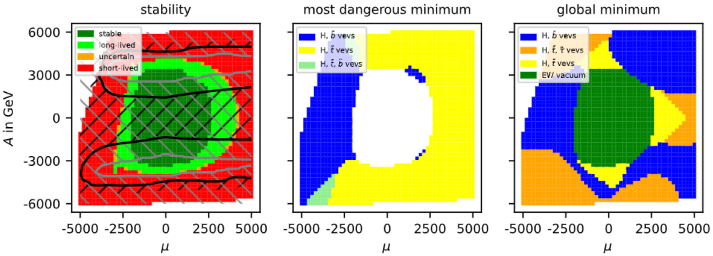
[W. Hollik, G. Weiglein, J. Wittbrodt, 1812.04644]



[TB, F. Campello, G. Weiglein, tbp]

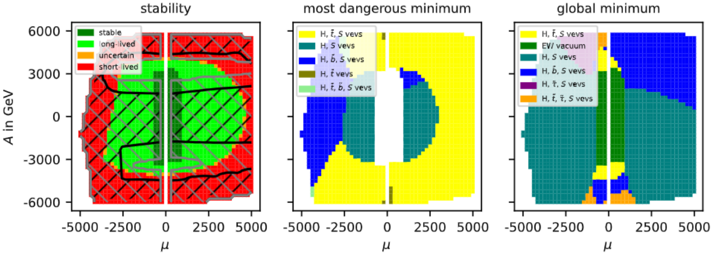
$M_h^{125}(A)$ benchmark scenario

MSSM:



NMSSM:

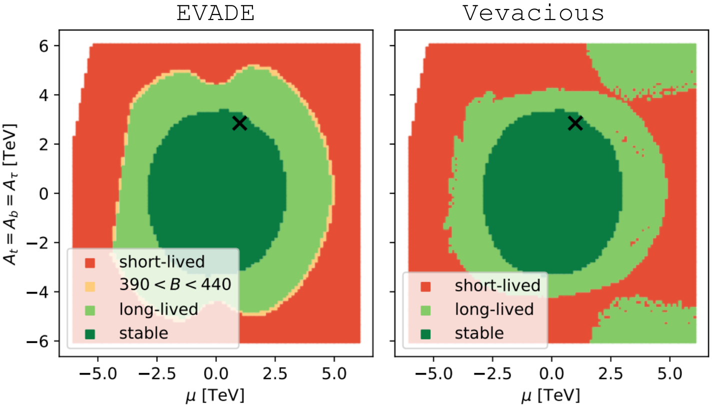
$\lambda = \kappa = 0.1$

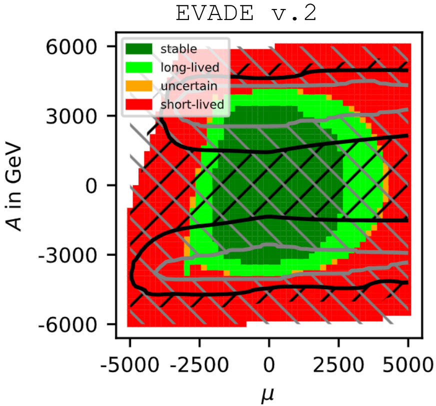


[TB, F. Campello, G. Weiglein, tbp]

$M_h^{125}(A)$ benchmark scenario

Karlsruher Institut für Technologie





[W. Hollik, G. Weiglein, J. Wittbrodt, 1812.04644]

[TB. F. Campello, G. Weiglein, tbp]

EVADE v.2: attempts to find bounce with path-deformation algorithm (CosmoTransitions)
if this fails switch to neural-network approach

Conclusions

Analysis of vacuum is difficult but worth the effort
$\rightarrow$ constrains the way in which BSM theories might manifest themselves at colliders

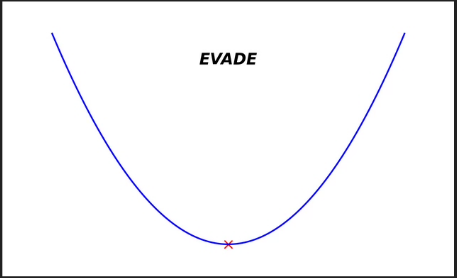
Meta-stable EW vacua are in most cases sufficiently long-lived
$\rightarrow$ demanding a global EW minimum is too restrictive, especially in the NMSSM

New method for computing bounce solutions using neural networks
$\rightarrow$ Reliable determination of tunneling path for $\mathcal{O}(10-100)$ fields
$\rightarrow$ Can find the bounce solution in cases where the path-deformation algorithm fails
$\rightarrow$ Drawback: longer runtime of about 10 s for 10 fields and about 100 s for 50 fields

CPU and GPU implementation of neural-network method in new version of `EVADE`

A logo for EVADE

ChatGPT-4: “graphic design is my passion”



Here's a simple conceptual sketch of the logo for "EVADE":

- The **U-shaped curve** represents the potential well, indicating stability.
- The **red dot** at the bottom symbolizes an atomic orbit, highlighting the focus on particle physics and vacuum stability.
- The software name **"EVADE"** is placed above the well in a sleek, modern font.

This design visually conveys the software's purpose of verifying vacuum stability in theories beyond the Standard Model.

Benchmark scenarios

BP	$\tan\beta$	μ	M_A	m_{L_3,e_3}	X_t	A_τ	A
M_h^{125}	$[0, 60]$	1000	$[0, 2000]$	2000	2800	$A_\tau = A$	$A(X_t, \mu, \tan\beta)$
$\mathrm{M}_h^{125}(\tilde{\tau})$	$[0, 60]$	1000	$[0, 2000]$	350	2800	800	$A(X_t, \mu, \tan\beta)$
$\mathrm{M}_h^{125}(A)$	20	$[-5000, 5000]$	1500	2000	$X_t(A, \mu, \tan\beta)$	$A_\tau = A$	$[-6000, 6000]$

Table 2: Parameter values for the M_h^{125} and $\mathrm{M}_h^{125}(\tilde{\tau})$ scenarios as defined in Ref. [65] and the $\mathrm{M}_h^{125}(A)$ scenario defined in Ref. [33]. The remaining MSSM parameters are set equally in all scenarios: $m_{Q_3,u_3,d_3} = 1500$, $M_{1,2} = 1000$, $M_3 = 2500$. In the first row, we also set $A \equiv A_t = A_b = A_\tau$ and $A = X_t + \mu/\tan\beta$. In the second row we use the same relations with the exception of $A_\tau = 800$. In the third row, A is varied and X_t is derived from the relation given above. In the NMSSM we additionally set $A_\kappa = -100$, $\mu_{\mathrm{eff}} = \mu$ and $\lambda = \kappa = 0.1$, such that the singlet vev v_S varies according the relation shown in Eq. (35). All dimensionful parameters are given in GeV.

Ref. [65]: [E. Bagnaschi et al., 1808.07542]

Ref. [33]: [W. Hollik, G. Weiglein, J. Wittbrodt, 1812.04644]