Finding bounce solutions with neural networks

Fabio Campello

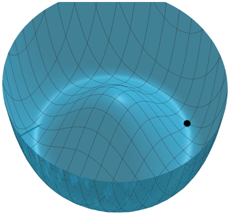
Thesis Defense Presentation
19.09.2024

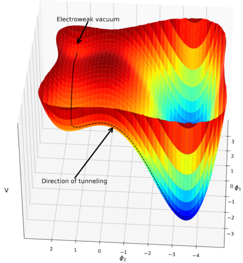
Supervisor: Prof. Dr. Georg Weiglein
Co-supervisor: Dr. Thomas Biekötter

Content

Introduction

- Spontaneous electroweak symmetry breaking in the early universe
- Universe settles in minimum with non zero vacuum expectation value
- Not generally global minimum in extended scalar sectors, tunneling possible
- Closest/deepest minimum not generally the most dangerous
- Compare EWV lifetime to age of the universe
- Important constraints on extended scalar sectors





False Vacuum Decay

- Random fluctuations form bubbles of the true vacuum
- Bubbles expand to convert the entire universe to true vacuum

$$\rho \equiv \sqrt{\tau^2 + |\vec{x}|^2}$$

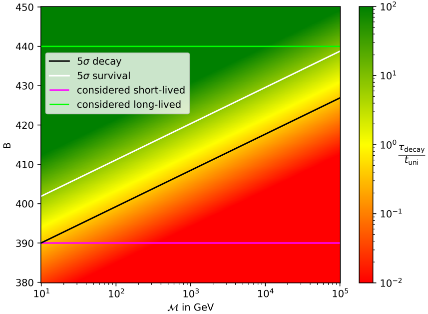
Equation of motion
$$\frac{d^2\phi}{d\rho^2} + \frac{3}{\rho}\frac{d\phi}{d\rho} - \nabla V(\phi) = 0$$

With boundary conditions
$$\left.\frac{d\phi}{d\rho}\right|_{\rho=0} = 0 \quad \phi_B(\infty) = \phi_f$$

S. Coleman. "Fate of the false vacuum: Semiclassical theory". In: *Phys. Rev. D* (10 May 1977).

False Vacuum Decay

$$B[\phi] = 2\pi^2 \int_0^\infty \mathrm{d}\rho\rho^3 \left[\frac{1}{2}\left(\frac{\mathrm{d}\phi}{\mathrm{d}\rho}\right)^2 + V(\phi)\right], \quad \Gamma \propto \mathcal{M}^4 e^{-B}$$



$$\begin{aligned} B > 440 &\quad \text{long-lived} \\ 390 \geq B \geq 440 &\quad \text{uncertain} \\ B < 390 &\quad \text{short-lived} \end{aligned}$$

W. G. Hollik, G. Weiglein, and J. Wittbrodt. "Impact of vacuum stability constraints on the phenomenology of supersymmetric models". In: *Journal of High Energy Physics* 3 (Mar. 2019).

False Vacuum Decay

Split the functional into two parts

$$B[\phi] = B_{\mathrm{kin}}[\phi] + B_{\mathrm{pot}}[\phi]$$

$$B_{\mathrm{kin}}[\phi] = 2\pi^2 \int_0^\infty d\rho \rho^3 \frac{1}{2}\left(\frac{d\phi}{d\rho}\right)^2, \quad B_{\mathrm{pot}}[\phi] = 2\pi^2 \int_0^\infty d\rho \rho^3 V(\phi)$$

By change of variables $\rho \to a\rho$ one finds

$$\left.\frac{\delta B}{\delta a}\right|_{a=1} = 0 \quad \Rightarrow \quad B_{\mathrm{pot}}[\phi_B] = -\frac{1}{2} B_{\mathrm{kin}}[\phi_B]$$

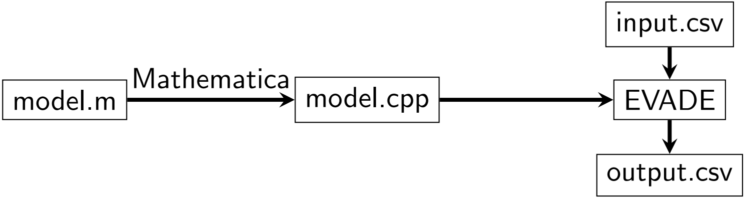
$$\underbrace{B[\phi_B]}_{B_1} = \underbrace{\frac{B_{\mathrm{kin}}[\phi_B]}{2}}_{B_2} = \underbrace{-B_{\mathrm{pot}}[\phi_B]}_{B_3}$$

$$\mathcal{L}_{\mathrm{bc}}[\phi] = |B_1 - B_2| + |B_1 - B_3| + |B_2 - B_3|$$

P. Athron et al. "Cosmological phase transitions: From perturbative particle physics to gravitational waves". In: *Progress in Particle and Nuclear Physics* (Feb. 2024).

EVADE

- Works for any renormalisable Higgs potential at zero temparature and tree-level, quartic potential of n fields
- Find all *extrema* using polynomial homotopy continuation, up to 3^n
- Straight path approximation, $V(\varphi, \hat{\varphi}) = \lambda(\hat{\varphi})\varphi^4 - A(\hat{\varphi})\varphi^3 + m^2(\hat{\varphi})\varphi^2$
- Use semianalytic result, $B = B(\lambda, A, m)$

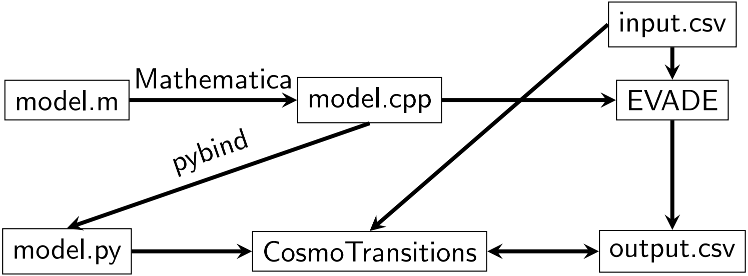


Path deformation algorithm

- Start with initial guess
- Split equation into parallel and perpendicular to the path
- Solve the parallel (one dimensional) equation using overshoot/undershoot
- Calculate normal forces on the path
- Deform the path based on the normal forces
- Repeat until normal forces vanish
- Implemented in the Python package CosmoTransitions

C. L. Wainwright. "CosmoTransitions: Computing cosmological phase transition temperatures and bubble profiles with multiple fields". In: *Computer Physics Communications* 9 (Sept. 2012).

Adding path deformation to EVADE



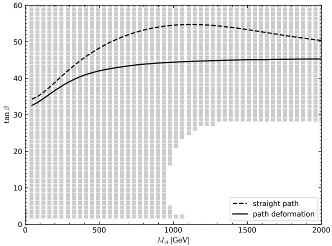
- Minimal additional setup
- Combines efficiency of EVADE with increased accuracy of using path deformation

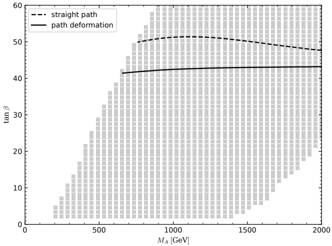
https://gitlab.com/fcampello/EVADE/-/tree/develop?ref_type=heads.

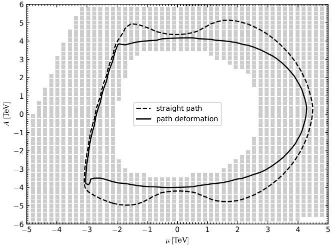
EVADE path deformation results

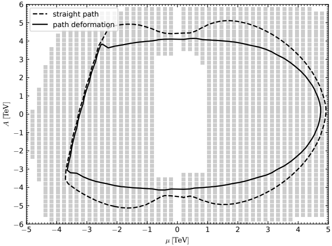
MSSM

NMSSM









- Path deformation algorithm leads to stricter exclusion limits

Neural network approach - Definitions

Train neural network to predict the tunneling path $\phi(\rho)$
Choose a ρ_{max} and discretize the path into n_ρ steps
Define the loss: $\mathcal{L} = \mathcal{L}_{\mathrm{eq}} + n_\rho \mathcal{L}_{\mathrm{b}}$

$$\mathcal{L}_{\mathrm{eq}} = \sum_i \left(\frac{d^2\phi(\rho_i)}{d\rho^2} + \frac{3}{\rho_i}\frac{d\phi(\rho_i)}{d\rho} - \nabla V(\phi(\rho_i)) \right)^2$$

$$\mathcal{L}_{\mathrm{b}} = \left(\frac{d\phi(\rho_0)}{d\rho} \right)^2 + (\phi(\rho_{\mathrm{max}}) - \phi_{\mathrm{f}})^2$$

Neural network prefers values between 0 and 1
$\Rightarrow \phi(\rho) = \phi_{\mathrm{t}} + \mathrm{NN}(\rho)(\phi_{\mathrm{f}} - \phi_{\mathrm{t}})$

Neural network approach - Training process

$$\mathcal{L}_{\text{eq}} = \sum_i \left(\frac{d^2\phi(\rho_i)}{d\rho^2} + \frac{3}{\rho_i}\frac{d\phi(\rho_i)}{d\rho} - \nabla V(\phi(\rho_i)) \right)^2$$
$$\mathcal{L}_{\text{b}} = \left(\frac{d\phi(\rho_0)}{d\rho} \right)^2 + (\phi(\rho_{\text{max}}) - \phi_{\text{f}})^2$$

- Equation has a trivial solution $\phi(\rho) = \phi_{\text{f}}$
- Random initialization $\rightarrow$ Network finds trivial solution
- Solution: Two step training process

Randomly initialize the network, then train a few epochs on
$\mathcal{L}_{\text{init}} = \sum_i (NN(\rho_i) - \frac{\rho_i}{\rho_{\text{max}}})^2$

Continue training with the correct Loss function

Neural network approach - Implementation overview

- Option 1: Use only existing TensorFlow operations via python
- Option 2: Write custom TensorFlow operation in C++

Python option:

- Easier to implement, no additional code to compile
- Use automatic differentiation of TensorFlow
- C++ model files can not be used
- Graph creation for complicated models is very slow

C++ option:

- C++ for Loss and its gradient, separate for CPU and GPU
- Need to use finite differences
- Use already available C++ model files
- Overall better performance

Neural network approach - Hyperparameter

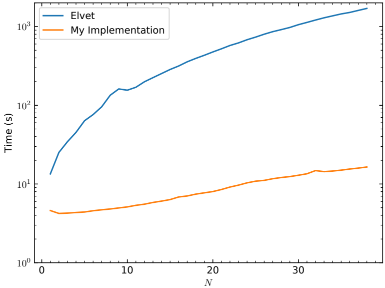
	Toy model	MSSM/NMSSM
Hidden layers	5	6
Neurons per hidden layer	10	50
Activation function	sigmoid	sigmoid
Initialization epochs	10^3, $2 \cdot 10^3$	$2 \cdot 10^4$
Initialization learning rate	10^{-2}	10^{-2}
Main epochs	$5 \cdot 10^3$, $3 \cdot 10^4$	10^6
Main learning rate	10^{-3}	10^{-4}
ρ_1	10^{-4}	10^{-4}
n_ρ	10^3	$5 \cdot 10^2$

- small networks
- high number of epochs
- large impact of the learning rate

Neural network approach - Results - Toy Model

- Toy model: $V = \sum_{i=0}^{N} \lambda_i x_i^4 - A_i x_i^3 + m_i x_i^2$
- $\lambda_i, A_i, m_i > 0$ randomly generated
- Compute tunneling from highest to lowest minimum

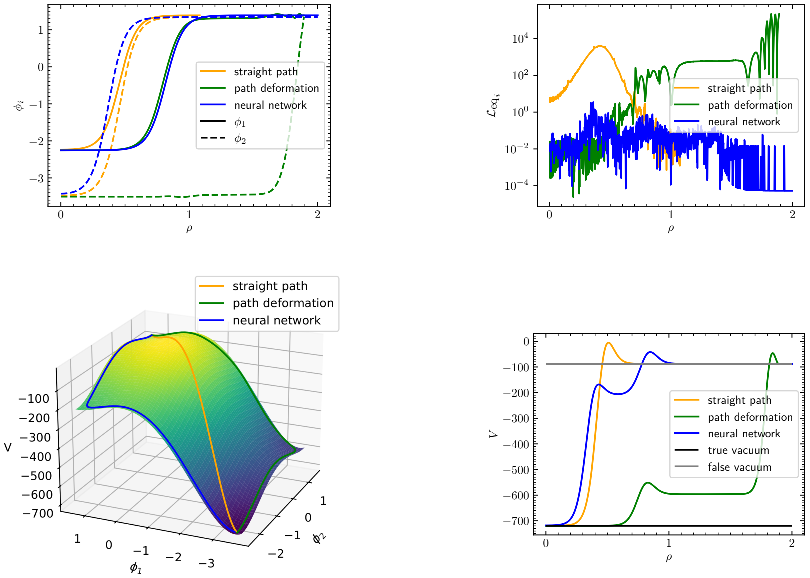
Comparison of Elvet, a general purpose differential equation solver using neural networks, and my implementation which is specialized to the bounce equation



- The specialized implementation is significantly faster.

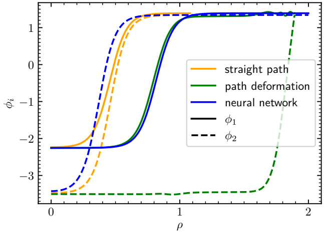
J. Y. Araz, J. C. Criado, and M. Spannwosky. *Elvet – a neural network-based differential equation and variational problem solver*. 2021. arXiv: 2103.14575 [cs.LG].

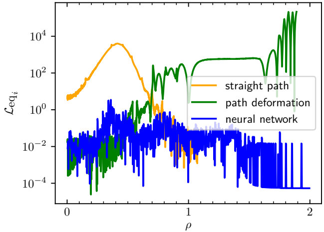
Neural network approach - Results - Toy Model



- Path deformation algorithm converges to the wrong side

Neural network approach - Results - Toy Model

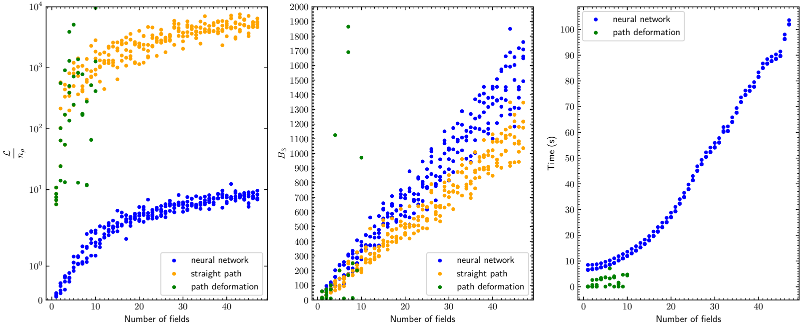




	B_1	B_2	B_3	$\mathcal{L}_{\mathrm{bc}}$
Straight path	47.78	47.57	47.37	0.8215
Path deformation	-15332	3060	21452	73570
Neural network	107.24	107.26	107.28	0.0915

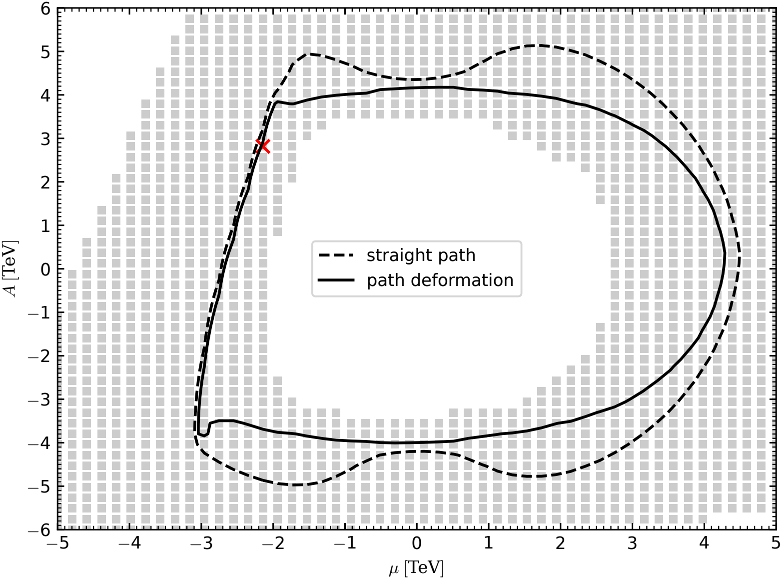
- Path deformation algorithm converges to the wrong side

Neural network approach - Results - Toy Model

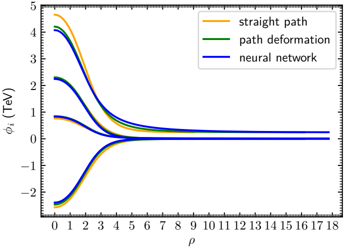


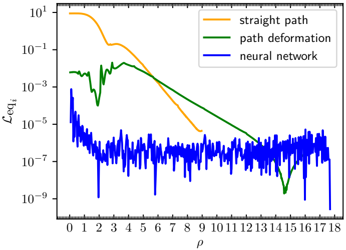
- The neural network is able to find the bounce solution consistently up to very high numbers of fields.

Neural network approach - Results - MSSM



Neural network approach - Results - MSSM

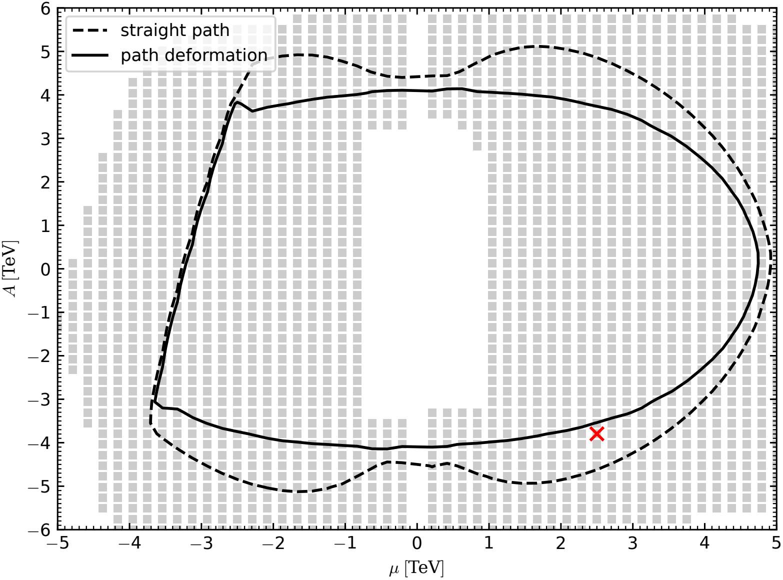




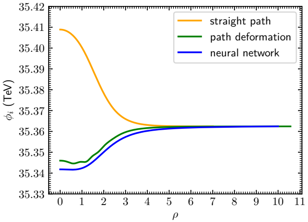
	B_1	B_2	B_3	$\mathcal{L}_{\mathrm{bc}}$
Straight path	499.55	499.55	499.55	0.01
Path deformation	413.04	410.65	408.26	9.56
Neural network	378.20	379.56	380.92	5.44

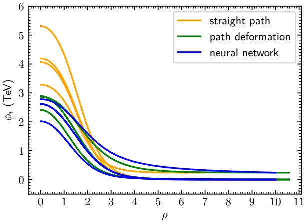
- Path deformation result is in the uncertainty region
- Excluded by neural network

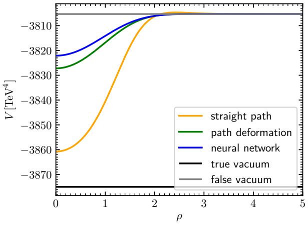
Neural network approach - Results - NMSSM

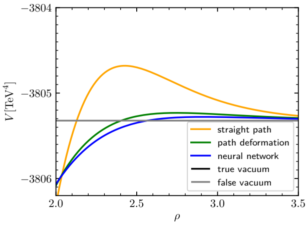


Neural network approach - Results - NMSSM



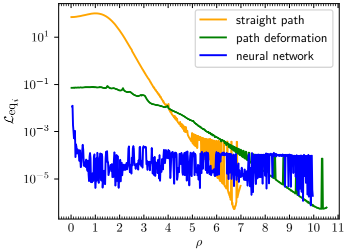


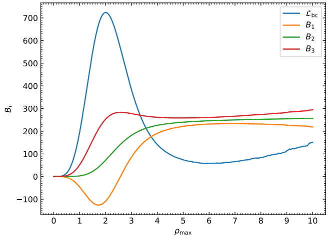




- Straight path approximation not applicable
- $\phi_i(\rho) = \phi_{i\,\mathrm{false}} + 5(\phi_{i\,\mathrm{true}} - \phi_{i\,\mathrm{false}})\,\mathrm{NN}_i(\rho)$

Neural network approach - Results - NMSSM





	B_1	B_2	B_3	$\mathcal{L}_{\mathrm{bc}}$
Straight path	720.73	730.49	740.24	39.03
Path deformation	235.58	285.60	335.62	200.08
Neural network	218.65	256.65	294.65	152.00

- Straight path approximation bounce too high
- Excluded by path deformation and neural network

Summary and outlook

The computation of the bounce action in EVADE was improved with important effects on the vacuum stability analysis and the resulting limits.

- Added path deformation to EVADE via CosmoTransitions
- Enables large scale parameter scans with improved accuracy

- Added neural network based bounce action solver to EVADE
- Neural network outperforms path deformation in accuracy of the solution
- Ability to handle $\mathcal{O}(50)$ scalar fields was shown, even higher numbers expected on more powerful hardware
- Demonstrated the significance of an accurate computation of the bounce solution
- High computational cost
- Tuning of hyper parameters can be necessary

Outlook

- Improve determination of stationary points to make full use of the neural network capabilities
- Finite temperature

---

https://gitlab.com/fcampello/EVADE/-/tree/develop?ref_type=heads.

References

[1] https://gitlab.com/fcampello/EVADE/-/tree/develop?ref_type=heads.

[2] J. Y. Araz, J. C. Criado, and M. Spannwosky. *Elvet – a neural network-based differential equation and variational problem solver*. 2021. arXiv: 2103.14575 [cs.LG].

[3] P. Athron et al. "Cosmological phase transitions: From perturbative particle physics to gravitational waves". In: *Progress in Particle and Nuclear Physics* 135 (Feb. 2024), p. 104094.

[4] S. Coleman. "Fate of the false vacuum: Semiclassical theory". In: *Phys. Rev. D* 15 (10 May 1977), pp. 2929–2936.

[5] W. G. Hollik, G. Weiglein, and J. Wittbrodt. "Impact of vacuum stability constraints on the phenomenology of supersymmetric models". In: *Journal of High Energy Physics* 2019.3 (Mar. 2019).

[6] C. L. Wainwright. "CosmoTransitions: Computing cosmological phase transition temperatures and bubble profiles with multiple fields". In: *Computer Physics Communications* 183.9 (Sept. 2012), pp. 2006–2013.

Backup - Benchmark scenarios

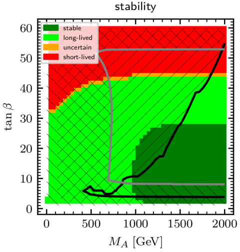
Scenario	$\tan\beta$	μ	M_1	M_2	M_A
M_h^{125}	$[0,60]$	1000	1000	1000	$[0,2000]$
$\mathrm{M}_h^{125}(\tilde{\tau})$	$[0,60]$	1000	180	300	$[0,2000]$
$\mathrm{M}_h^{125}(A)$	20	$[-5000,5000]$	1000	1000	1500

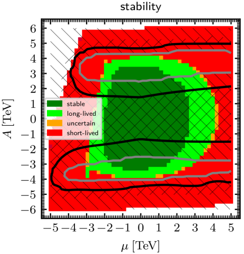
Scenario	m_{L_3,e_3}	X_t	A_τ	A
M_h^{125}	2000	2800	$A_\tau = A$	$A(X_t,\mu,\tan\beta)$
$\mathrm{M}_h^{125}(\tilde{\tau})$	350	2800	800	$A(X_t,\mu,\tan\beta)$
$\mathrm{M}_h^{125}(A)$	2000	$X_t(A,\mu,\tan\beta)$	$A_\tau = A$	$[-6000,6000]$

$$m_{Q_3,u_3,d_3} = 1500, \quad M_3 = 2500$$
$$A \equiv A_t = A_b = A_\tau, \quad A = X_t + \mu/\tan\beta$$
$$A_\kappa = -100, \quad \mu_{\mathrm{eff}} = \mu, \quad \lambda = \kappa = 0.1$$

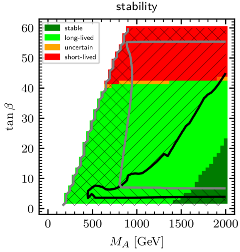
Backup - EVADE path deformation results

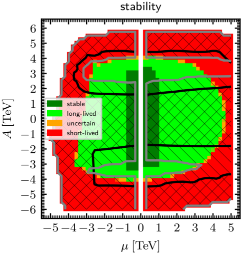
MSSM



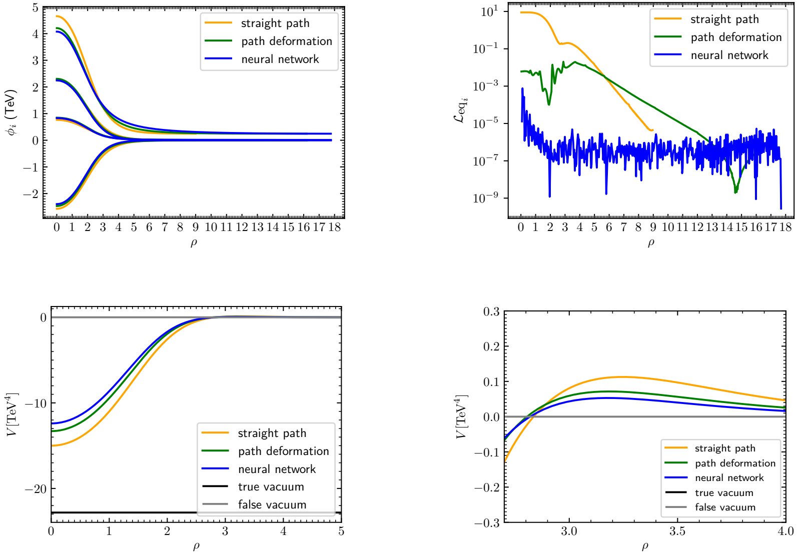


NMSSM





Backup - Neural network approach - Results - MSSM



- Neural network path has the lowest loss and barrier.