GUTPC 2025

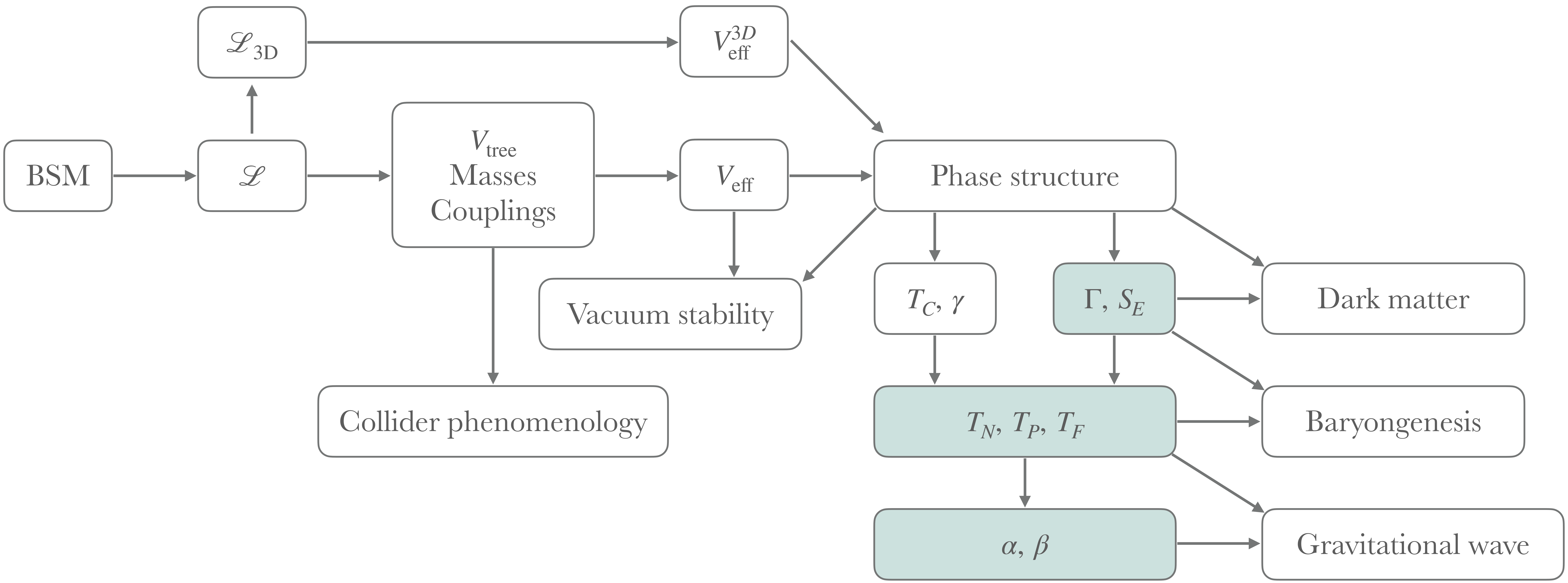
Enhancing Phase Transition Calculations through Polynomial Fitting and Neural Network Approximation

Yang Zhang, Henan Normal University

张阳，河南师范大学理论物理中心

In collaboration with Ligong Bian, Hongxin Wang, Yang Xiao, Ji-Chong Yang, Jin Min Yang

Introduction



Transition parameters

- Bubble nucleation rate

$$\frac{\Gamma}{V} = A(T)e^{-S_E(T)/T}[1 + \mathcal{O}(\hbar)]$$

- Nucleation temperature

$$N(T) = \int_{T_{\rm nuc}}^{T_{\rm tra}} \frac{\mathrm{d}T}{T} \frac{\Gamma(T)}{H(T)^4} = 1$$

- Percolation temperature

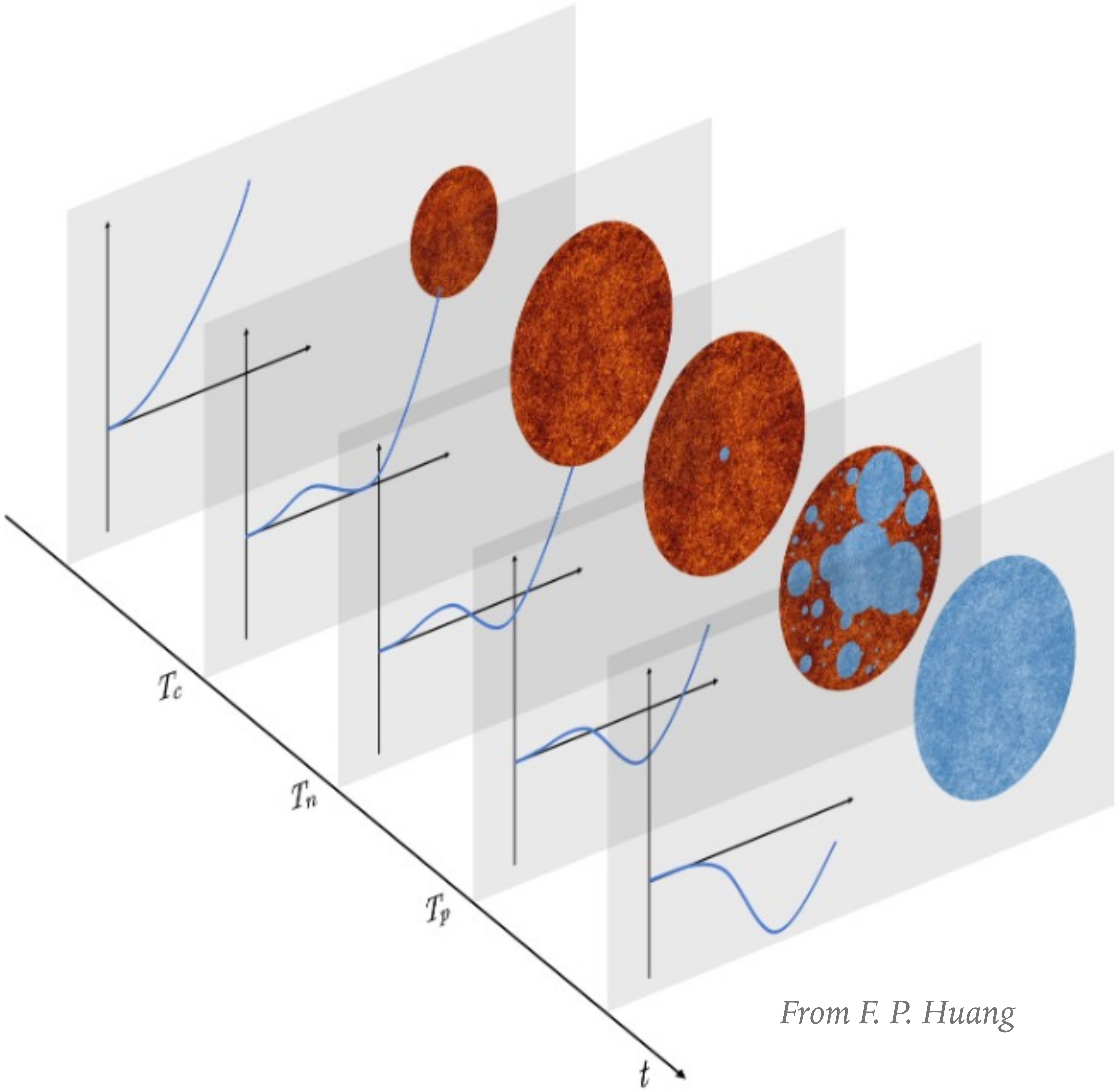
$$P(T_{\rm per}) = \exp\left[-\frac{64\pi}{3}\xi^4 \int_{T_{\rm per}}^{T_{\rm tra}} \mathrm{d}T' \frac{\Gamma(T')}{T'^6}\left(\frac{1}{T_{\rm per}} - \frac{1}{T'}\right)^3\right] = 70\,\%$$

- Inverse duration of the phase transition

$$\beta(T) = \frac{d}{dt}\left[\frac{S_E(T)}{T}\right] = TH(T)\frac{d}{dT}\left[\frac{S_E(T)}{T}\right]$$

- Ratio of latent heat to radiation density

$$\alpha = \frac{D\theta}{\pi^2 g_* T_*^4/30} = \frac{1}{\pi^2 g_* T_*^4/30}\left(V(\phi) - \frac{T}{4}\frac{\partial V(\phi, T)}{\partial T}\right)\Bigg|_{\phi_t}^{\phi_f}$$



From F. P. Huang

Transition parameters

- Bubble nucleation rate

$$\frac{\Gamma}{V} = A(T)e^{-S_E(T)/T}[1 + \mathcal{O}(\hbar)]$$

- Nucleation temperature

$$N(T) = \int_{T_{\rm nuc}}^{T_{\rm tra}} \frac{\mathrm{d}T}{T}\frac{\Gamma(T)}{H(T)^4} = 1$$

- Percolation temperature

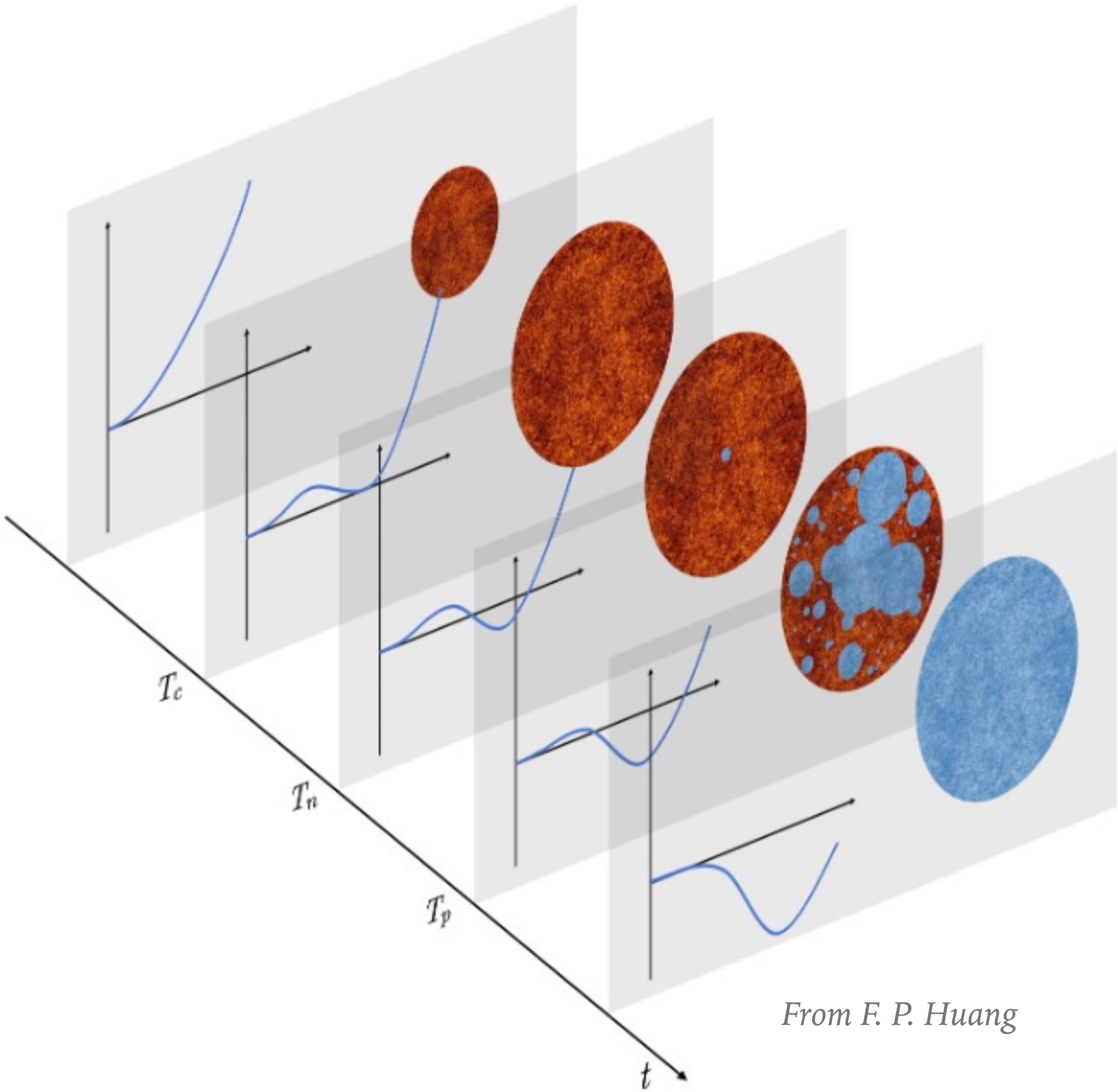
$$P(T_{\rm per}) = \exp\left[-\frac{64\pi}{3}\xi^4\int_{T_{\rm per}}^{T_{\rm tra}} \mathrm{d}T' \frac{\Gamma(T')}{T'^6}\left(\frac{1}{T_{\rm per}} - \frac{1}{T'}\right)^3\right] = 70\,\%$$

- Inverse duration of the phase transition

$$\beta(T) = \frac{d}{dt}\left[\frac{S_E(T)}{T}\right] = TH(T)\frac{d}{dT}\left[\frac{S_E(T)}{T}\right]$$

- Ratio of latent heat to radiation density

$$\alpha = \frac{D\theta}{\pi^2 g_* T_*^4/30} = \frac{1}{\pi^2 g_* T_*^4/30}\left(V(\phi) - \frac{T}{4}\frac{\partial V(\phi,T)}{\partial T}\right)\Bigg|_{\phi_t}^{\phi_f}$$



From F. P. Huang

Bounce action

- Bounce action

$$S_E = \mathcal{S}_{d-1} \int_0^\infty \rho^{d-1} \left(\frac{1}{2} \dot{\phi}^2 + V(\phi) \right)$$

where ϕ satisfies

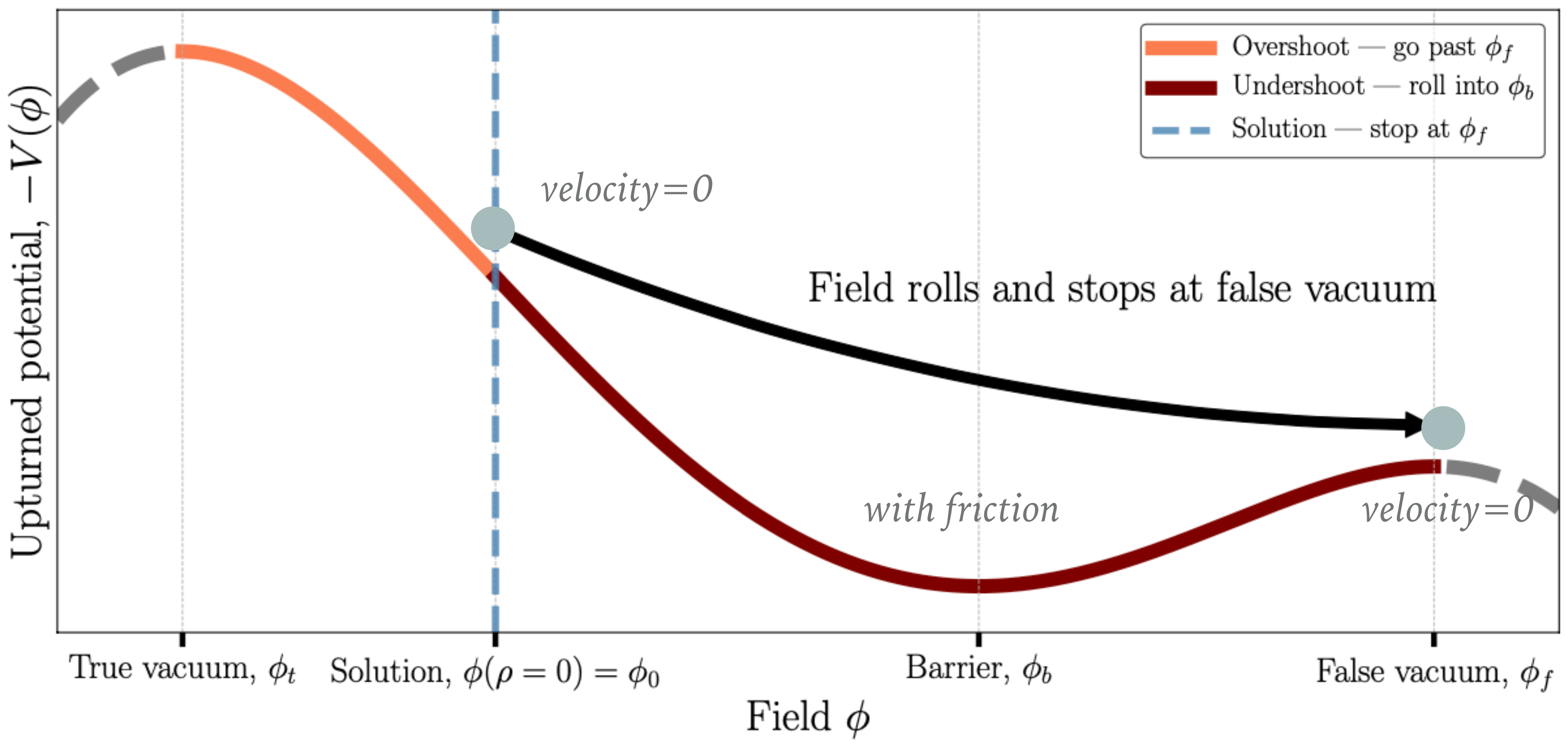
$$\frac{d^2\phi(\rho)}{d\rho^2} + \frac{\alpha}{\rho} \frac{d\phi(\rho)}{d\rho} = \Delta V(\phi)$$

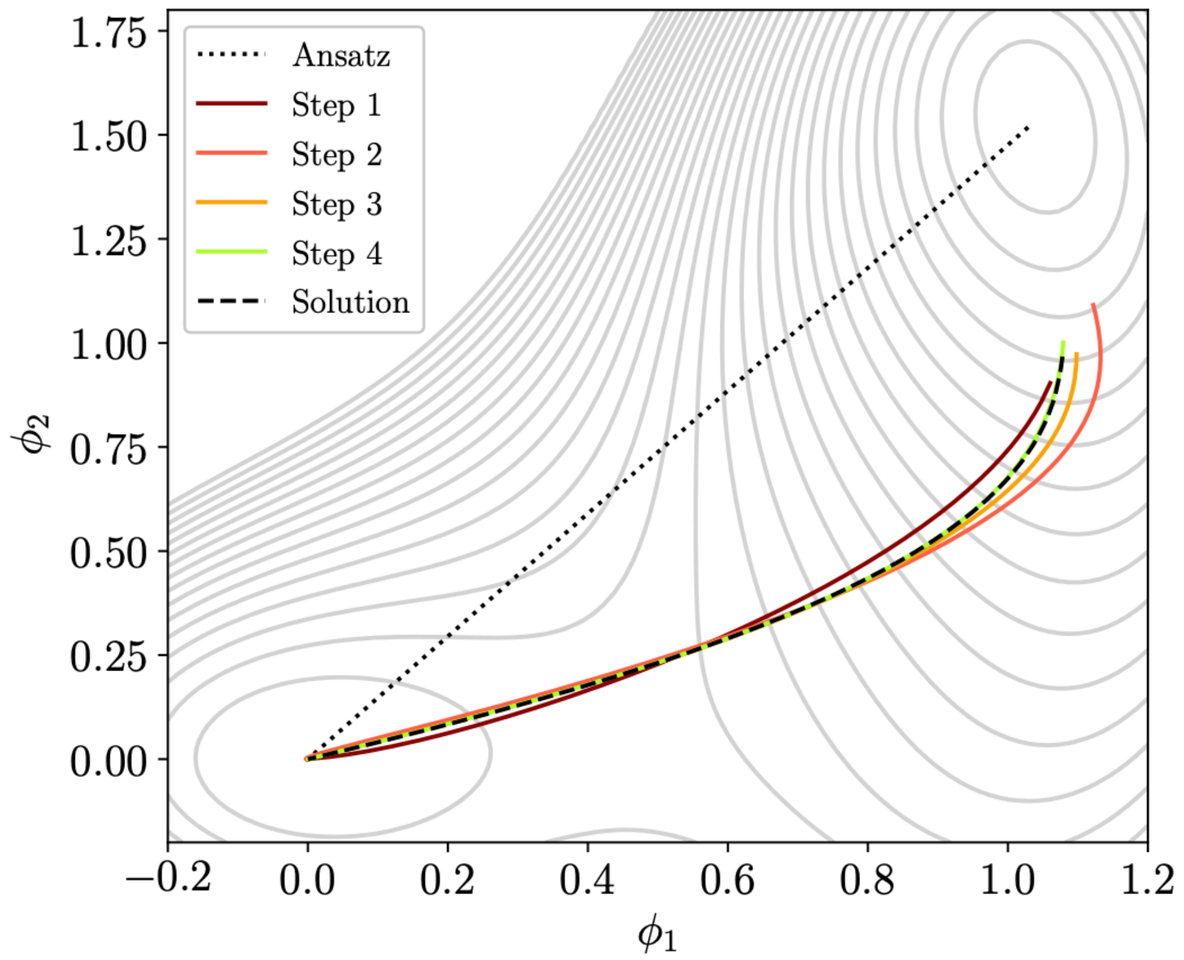
with boundary conditions

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

$$\phi(\rho \to \infty) = \phi_f$$

The under/over-shooting method





From Bubbleprofiler

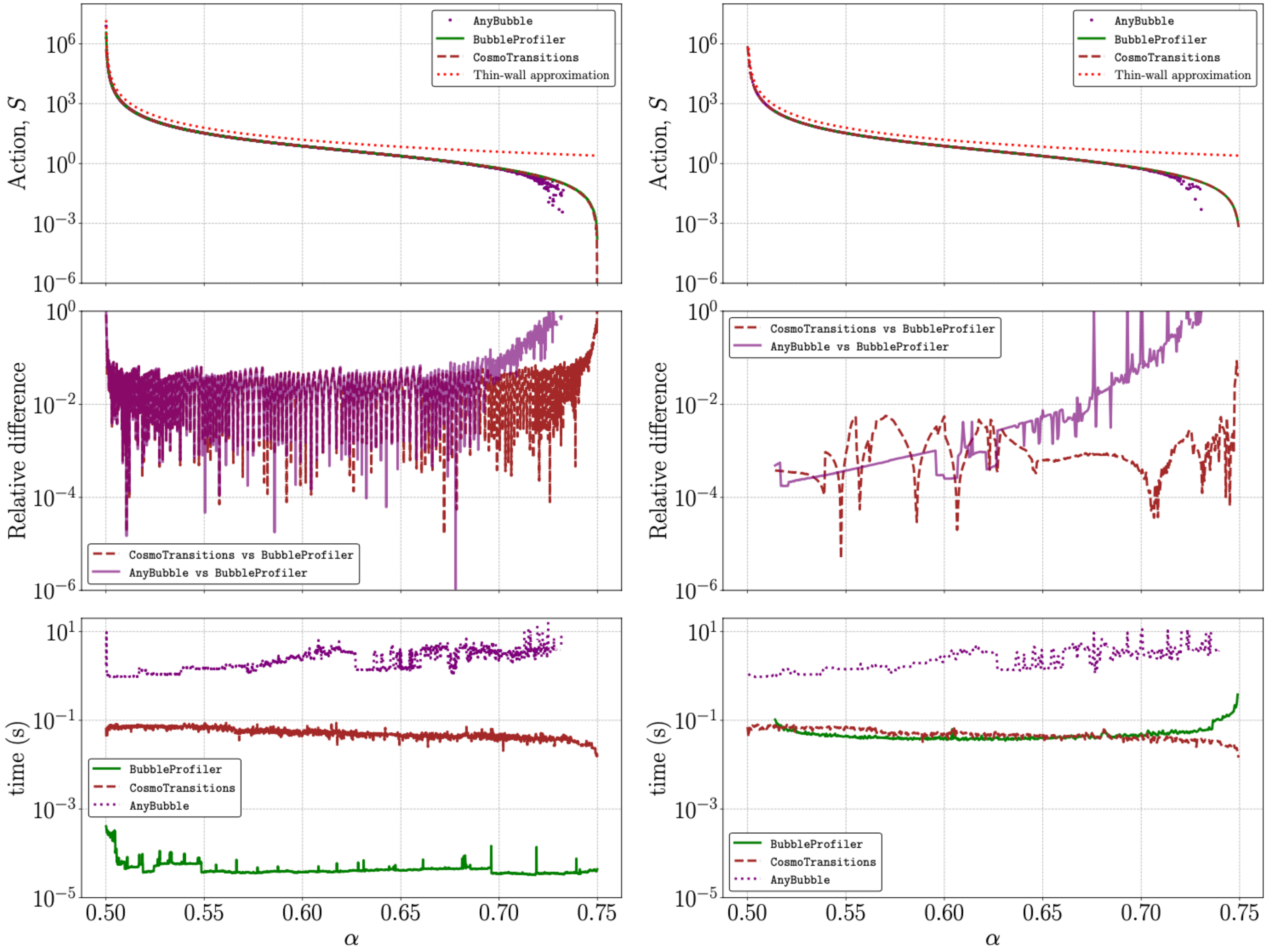
	Action			Time (s)		
# fields	BP	CT	AB	BP	CT	AB
1	54.1	52.6	52.4	0.051	0.066	1.285
2	20.8	21.1	20.8	0.479	0.352	7.473
3	22.0	22.0	22.0	0.964	0.215	25.209
4	55.9	56.4	55.9	1.378	0.255	54.258
5	16.3	16.3	16.3	2.958	0.367	305.531
6	24.5	24.5	24.4	4.853	0.337	830.449
7	36.7	36.6	36.7	6.754	0.375	1430.892
8	46.0	46.0	46.0	10.014	0.409	1805.713

Bounce action

$$V(\chi) = \frac{-4\alpha + 3}{2}\chi^2 - \chi^3 + \alpha\chi^4$$

- Tools for calculating bounce action
  - CosmoTransitions
  - PhaseTracer2
  - BSMPT3
  - AnyBubble
  - BubbleProfiler
  - FindBounce
  - SimpleBounce
  - … …

From Bubbleprofiler



(a) Shooting method **(b)** Perturbative algorithm

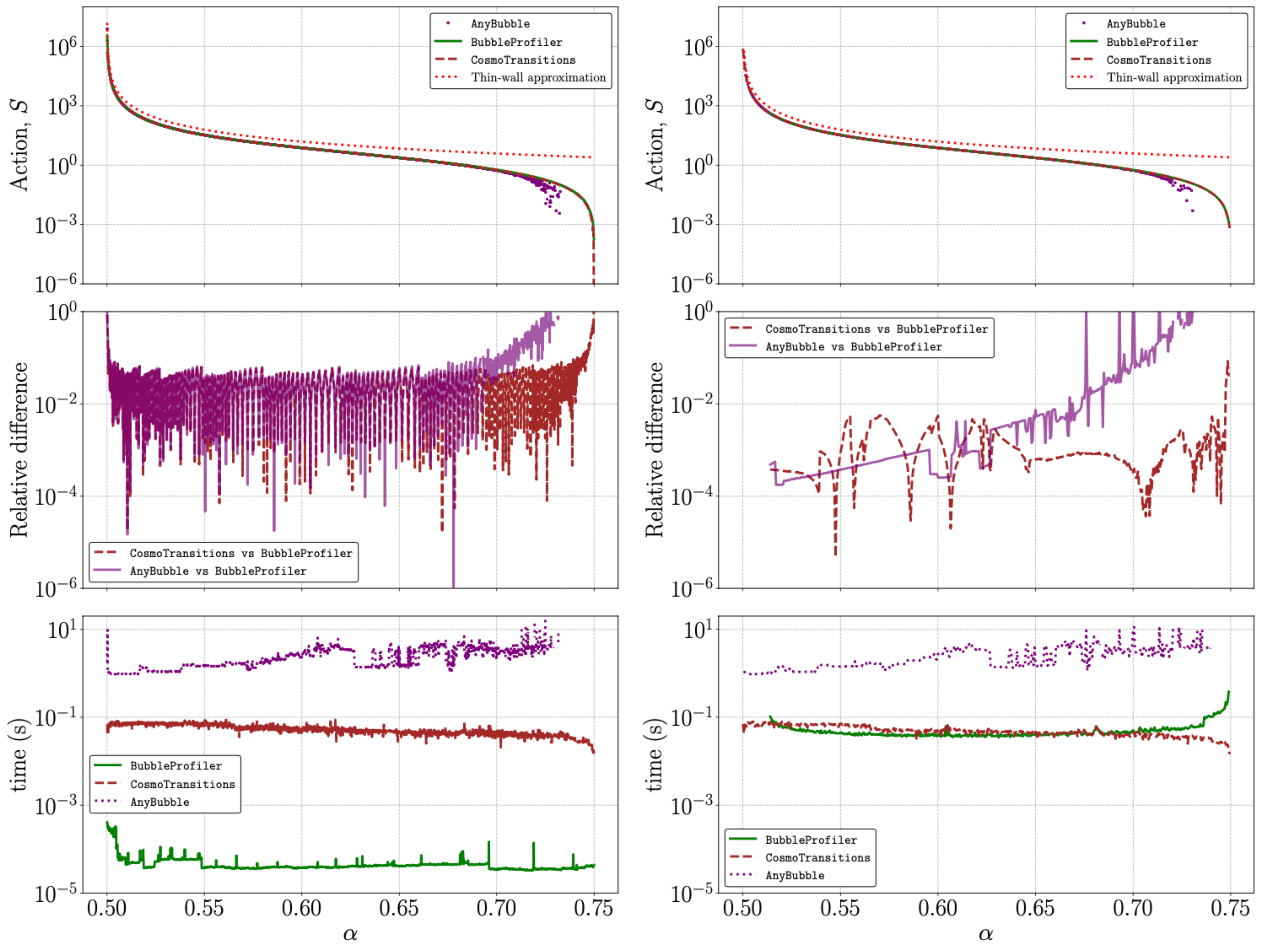
Bounce action

$$V(\chi) = \frac{-4\alpha + 3}{2}\chi^2 - \chi^3 + \alpha\chi^4$$

- Tools for calculating bounce action
  - CosmoTransitions
  - PhaseTracer2
  - BSMPT3
  - AnyBubble
  - BubbleProfiler
  - FindBounce
  - SimpleBounce
  - … …

We are not adding a new tool here.

From Bubbleprofiler



(a) Shooting method

(b) Perturbative algorithm

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \frac{f(x+h) - f(x-h)}{2h}$$

Transition parameters

- Bubble nucleation rate

$$\frac{\Gamma}{V} = A(T)e^{-S_E/T}[1 + \mathcal{O}(\hbar)]$$

- Nucleation temperature

$$N(T) = \int_{T_{\rm nuc}}^{T_{\rm tra}} \frac{\mathrm{d}T}{T} \frac{\Gamma(T)}{H(T)^4} = 1$$

- Percolation temperature

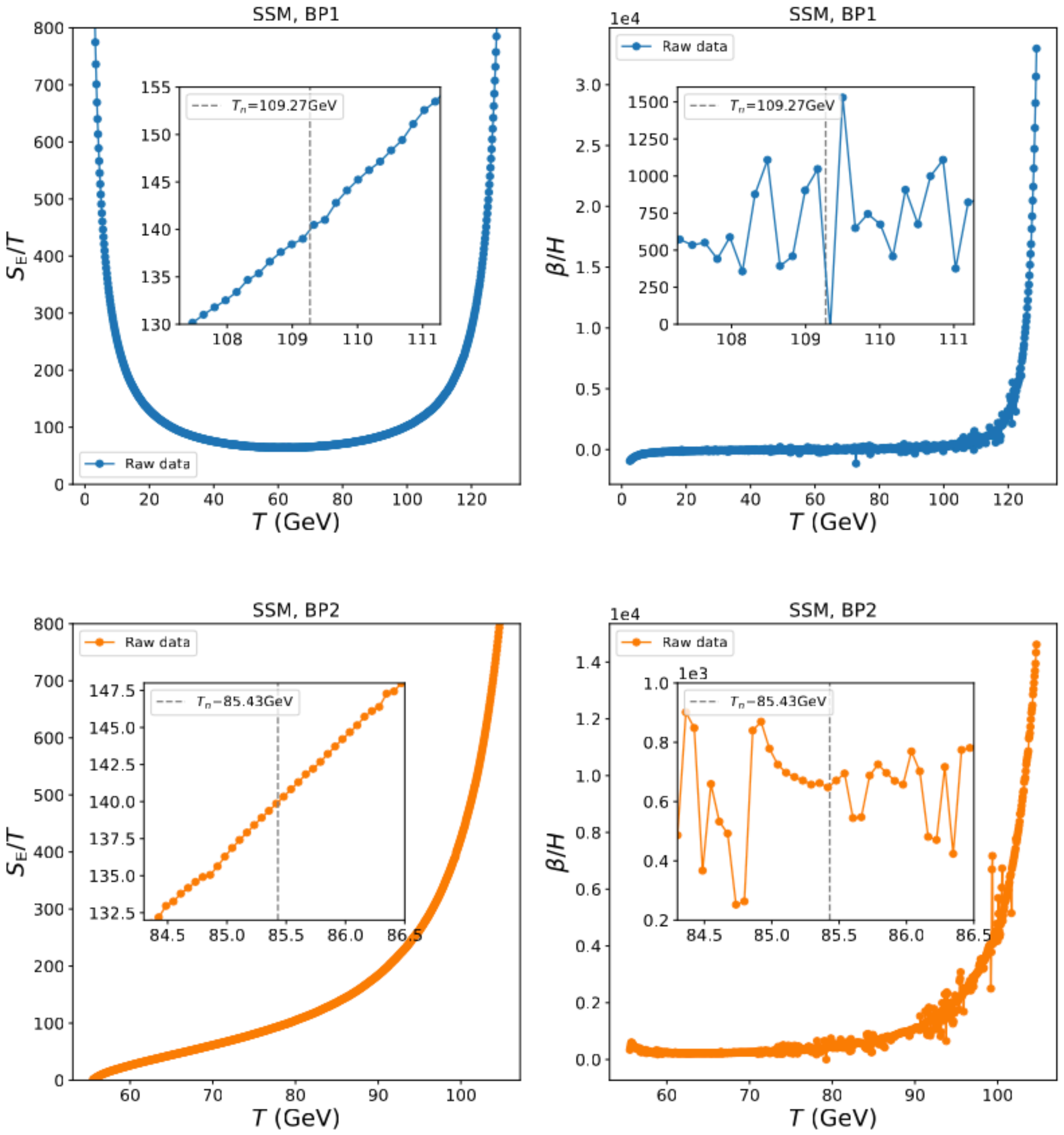
$$P(T_{\rm per}) = \exp\left[-\frac{64\pi}{3}\xi^4 \int_{T_{\rm per}}^{T_{\rm tra}} \mathrm{d}T' \frac{\Gamma(T')}{T'^6}\left(\frac{1}{T_{\rm per}} - \frac{1}{T'}\right)^3\right]$$

- Inverse duration of the phase transition

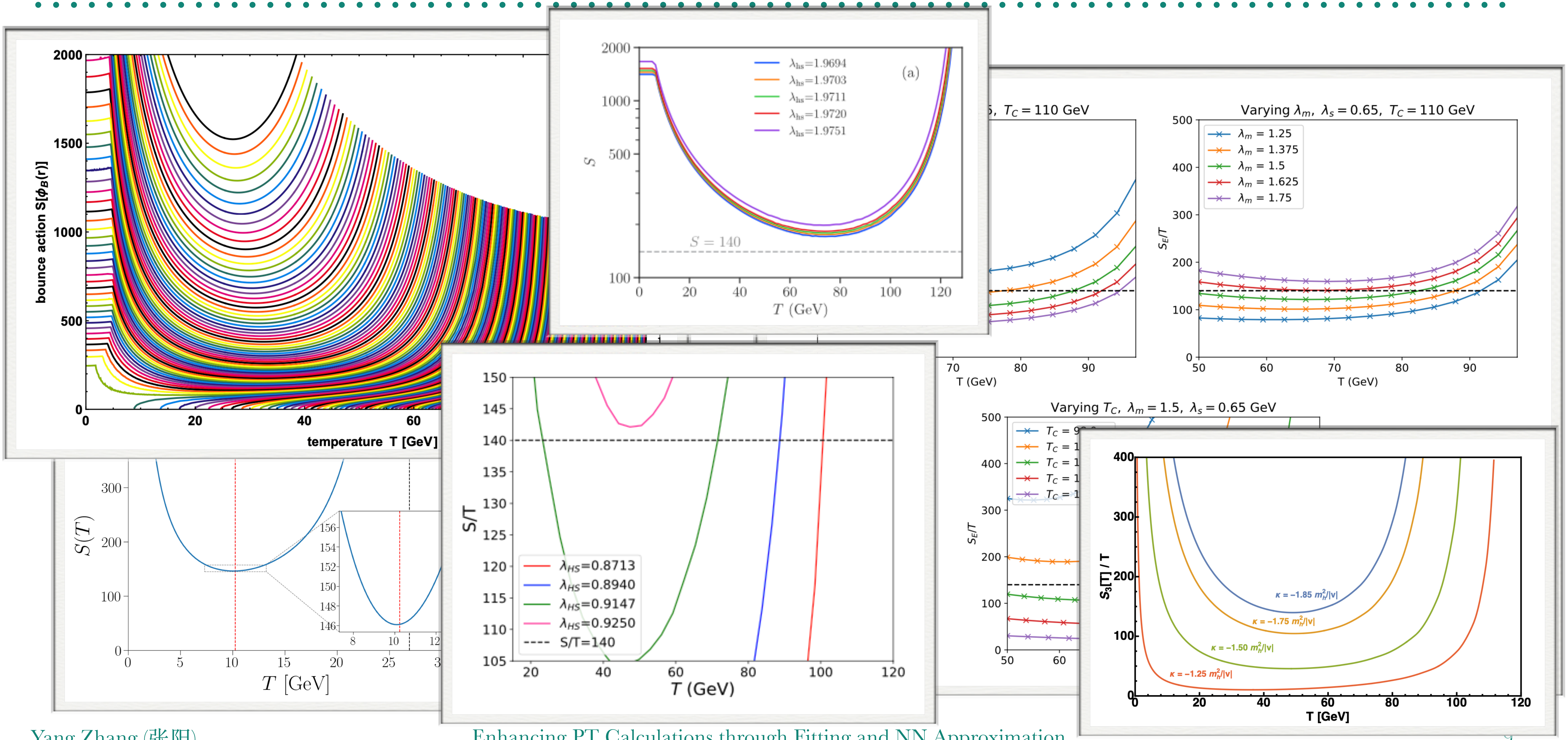
$$\beta(T) = \frac{d}{dt}\left(\frac{S_E}{T}\right) = TH(T)\frac{d}{dT}\left(\frac{S_E}{T}\right)$$

- Ratio of latent heat to radiation density

$$\alpha = \frac{D\theta}{\pi^2 g_* T_*^4/30} = \frac{1}{\pi^2 g_* T_*^4/30}\left(V(\phi) - \frac{T}{4}\frac{\partial V(\phi, T)}{\partial T}\right)\Bigg|_{\phi_t}^{\phi_f}$$

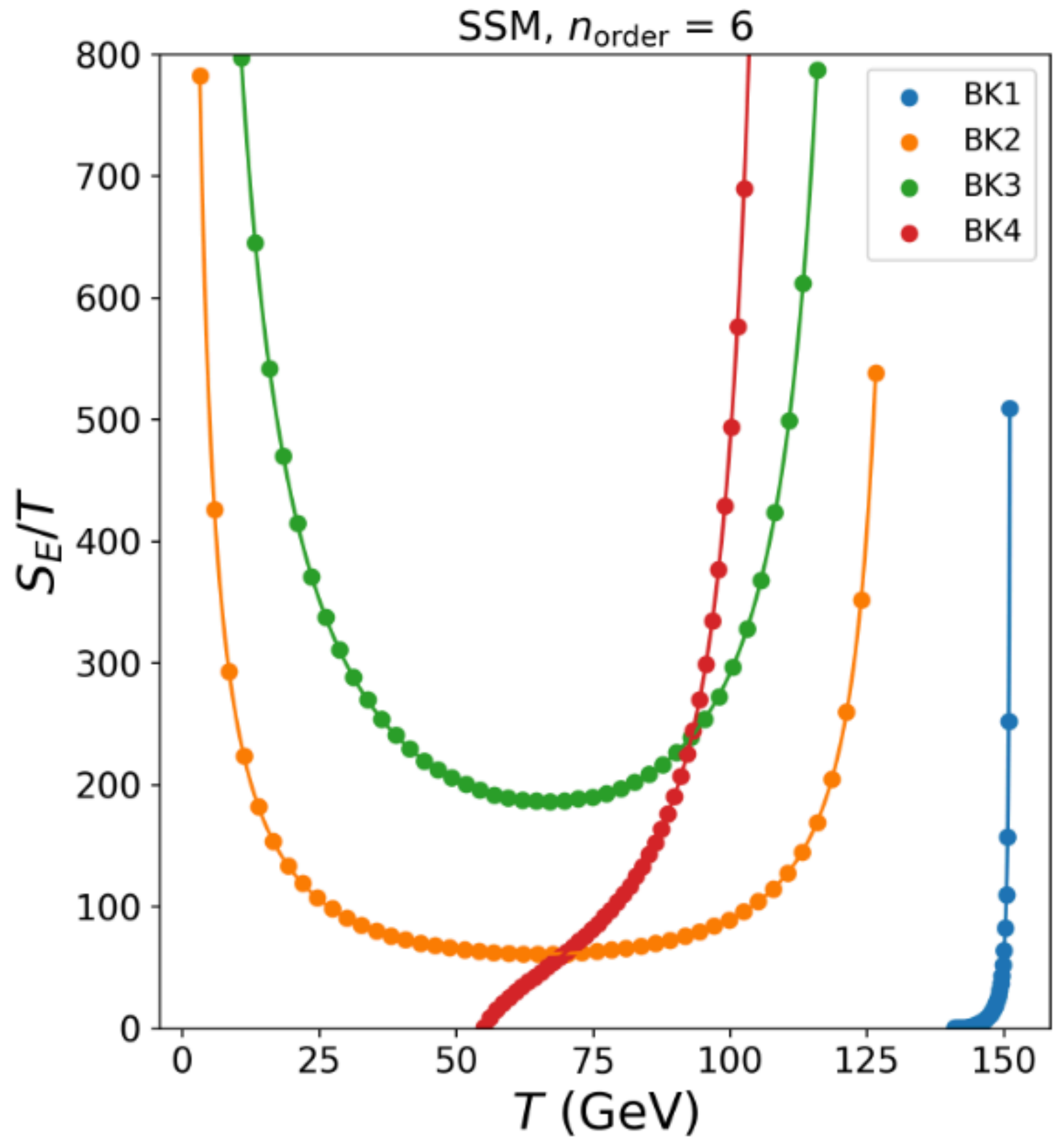


Action curve



Action curve

- Can we find a function to model the action curve?
- If achievable, this would enable us to:
  - Precisely calculate the β parameter
  - Perform rapid computations of the $T_{\rm per}$
  - … …
- Using machine learning?
- Polynomial fitting is enough.



Action curve

- In the thin-wall approximation, the potential difference $\epsilon = V_{\text{eff}}(\phi_f; T) - V_{\text{eff}}(\phi_t; T)$ is much smaller than the height of the barrier, so we neglect the viscous damping term in the bounce equation:

$$\frac{\mathrm{d}^2\phi}{\mathrm{d}r^2} + \frac{2}{r}\frac{\mathrm{d}\phi}{\mathrm{d}r} = \frac{\partial V_{\text{eff}}(\phi; T)}{\partial \phi} \quad \rightarrow \quad \frac{\mathrm{d}^2\phi}{\mathrm{d}r^2} = \frac{\partial V_{\text{eff}}(\phi; T)}{\partial \phi} \quad \rightarrow \quad \frac{d\phi}{dr} = \sqrt{2V_{\text{eff}}(\phi; T)}\,.$$

Define the surface tension of the bubble

$$\sigma = \int_0^{\infty} dr \left[\frac{1}{2}\left(\frac{d\phi}{dr}\right)^2 + V_{\text{eff}}(\phi; T)\right] = \int_{\phi_t}^{\phi_f} d\phi \sqrt{2V_{\text{eff}}(\phi; T)},$$

$$S_E = 4\pi \int_0^{+\infty} r^2 \mathrm{d}r \left[\frac{1}{2}\left(\frac{\partial\phi}{\partial r}\right)^2 + V_{\text{eff}}(\phi; T)\right] = 4\pi R^2 \sigma - \frac{4}{3}\pi R^3 \epsilon\,.$$

where R is the radius of the critical bubble and can be calculated by minimization of S_E, $R = \dfrac{2\sigma}{\epsilon}$. As

$$\epsilon = V_{\text{eff}}(\phi_f; T) - V_{\text{eff}}(\phi_t; T) = \left(\frac{\partial V_{\text{eff}}(\phi_f; T)}{\partial T}\bigg|_{T=T_c} - \frac{\partial V_{\text{eff}}(\phi_t; T)}{\partial T}\bigg|_{T=T_c}\right)(T - T_c)$$

$$S_E = \frac{16\pi\sigma^3}{3\epsilon^2} \propto \frac{1}{(T - T_C)^2}$$

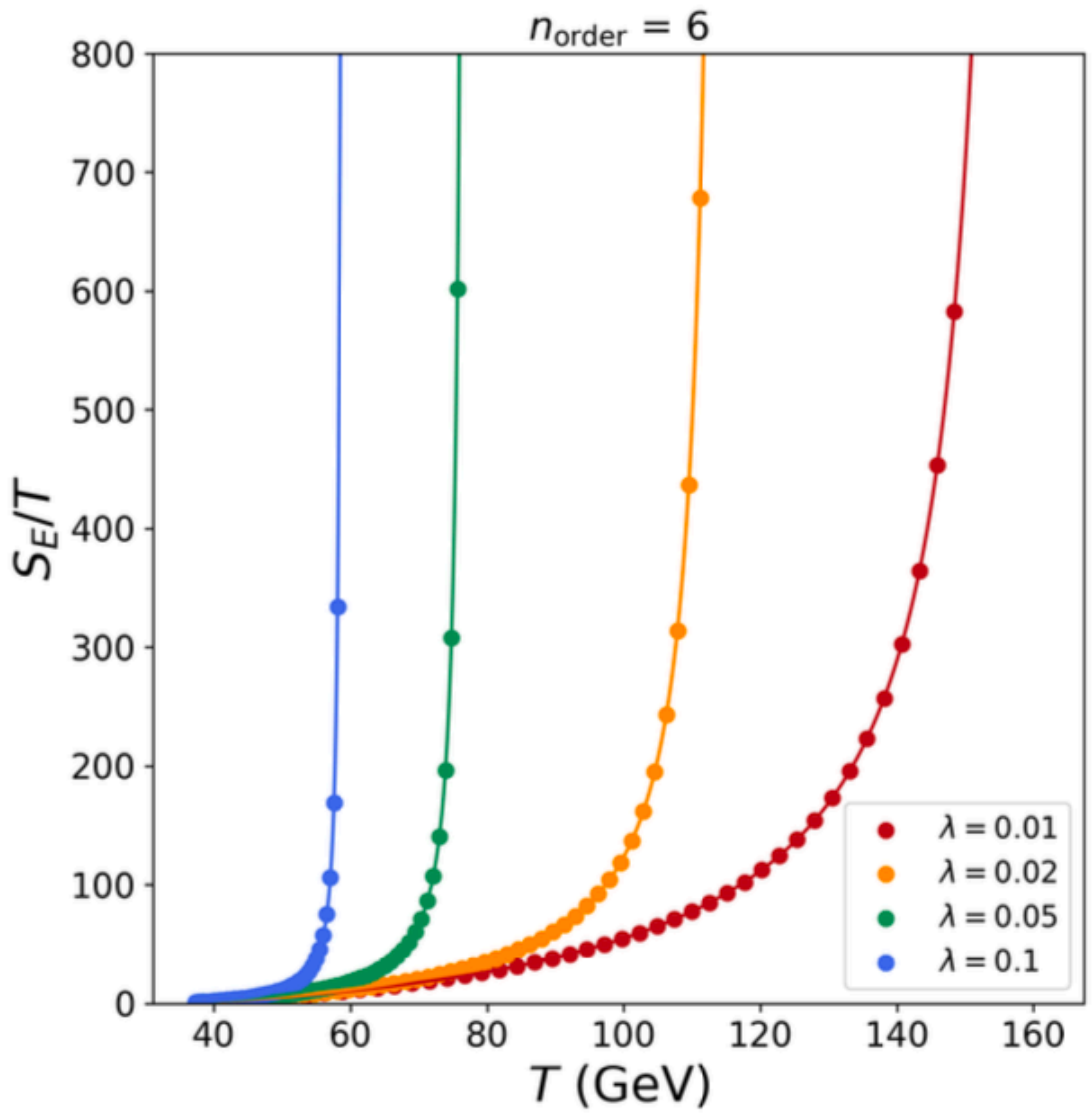
Action curve

- Therefore, it is reasonable to use the polynomial fitting formula

$$S_E = \frac{\sum_{i=0}^{n_{\text{order}}} q_i T^i}{(T - T_c)^2}$$

- And one can get the expression for the inverse phase transition duration time

$$\frac{\beta}{H} = \frac{\sum_{i=1}^{n_{\text{order}}} q_i i T^i (T - T_c) - \sum_{i=0}^{n_{\text{order}}} q_i T^i (3T - T_c)}{T(T - T_c)^3}$$

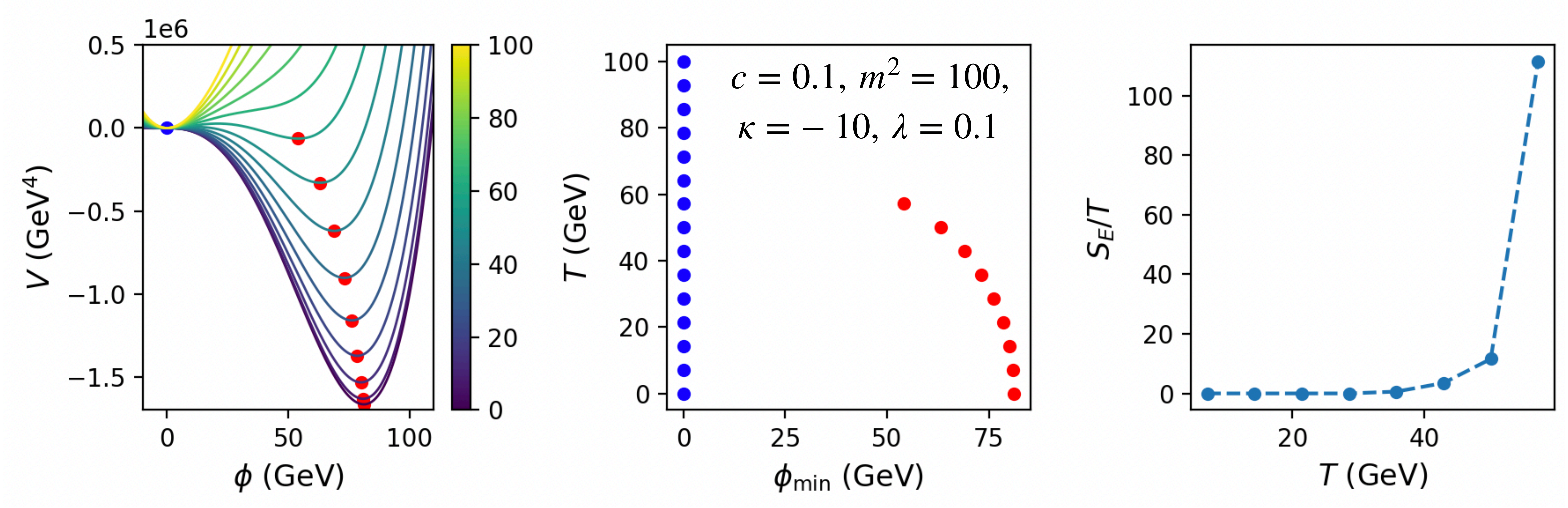


Action curve for one-dimensional toy model

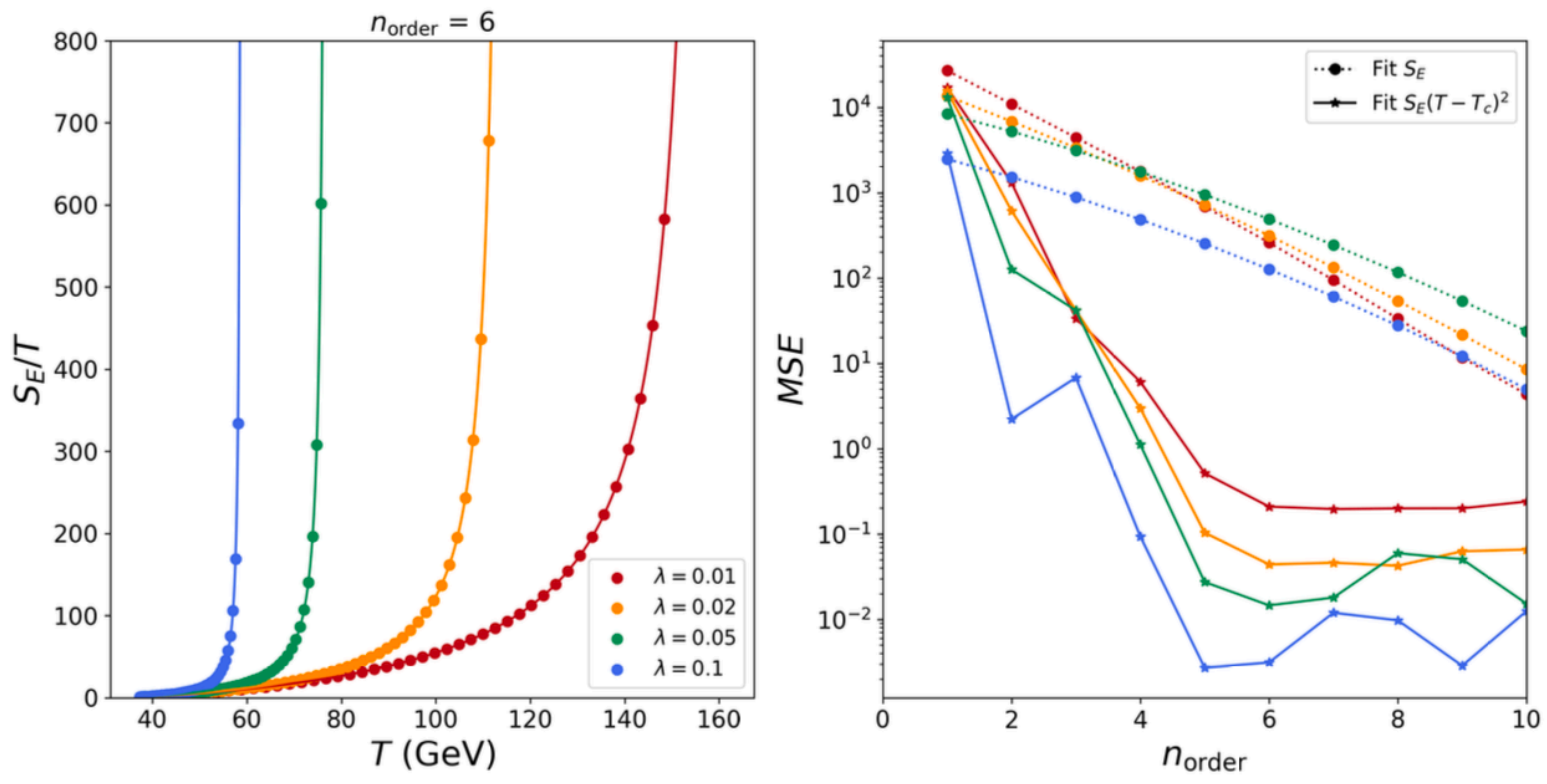
- To validate the polynomial fitting approach, we utilize a 1D toy model in which the action can be accurately computed.

$$V_{\rm eff}(\phi;T) = (cT^2 - m^2)\phi^2 + \kappa\phi^3 + \lambda\phi^4$$

- It mimics a simple model that includes high- temperature corrections.



Action curve for one-dimensional toy model

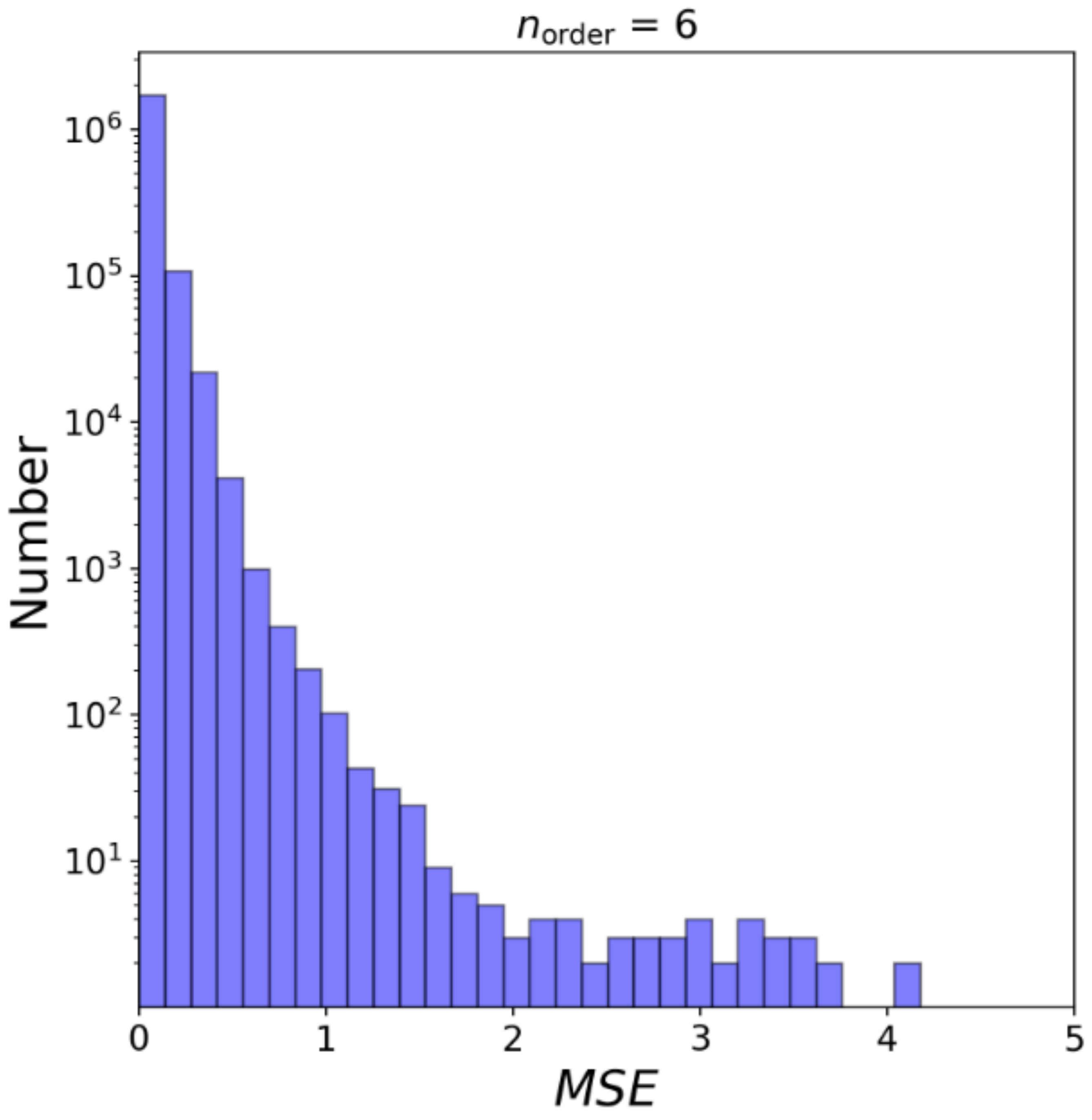


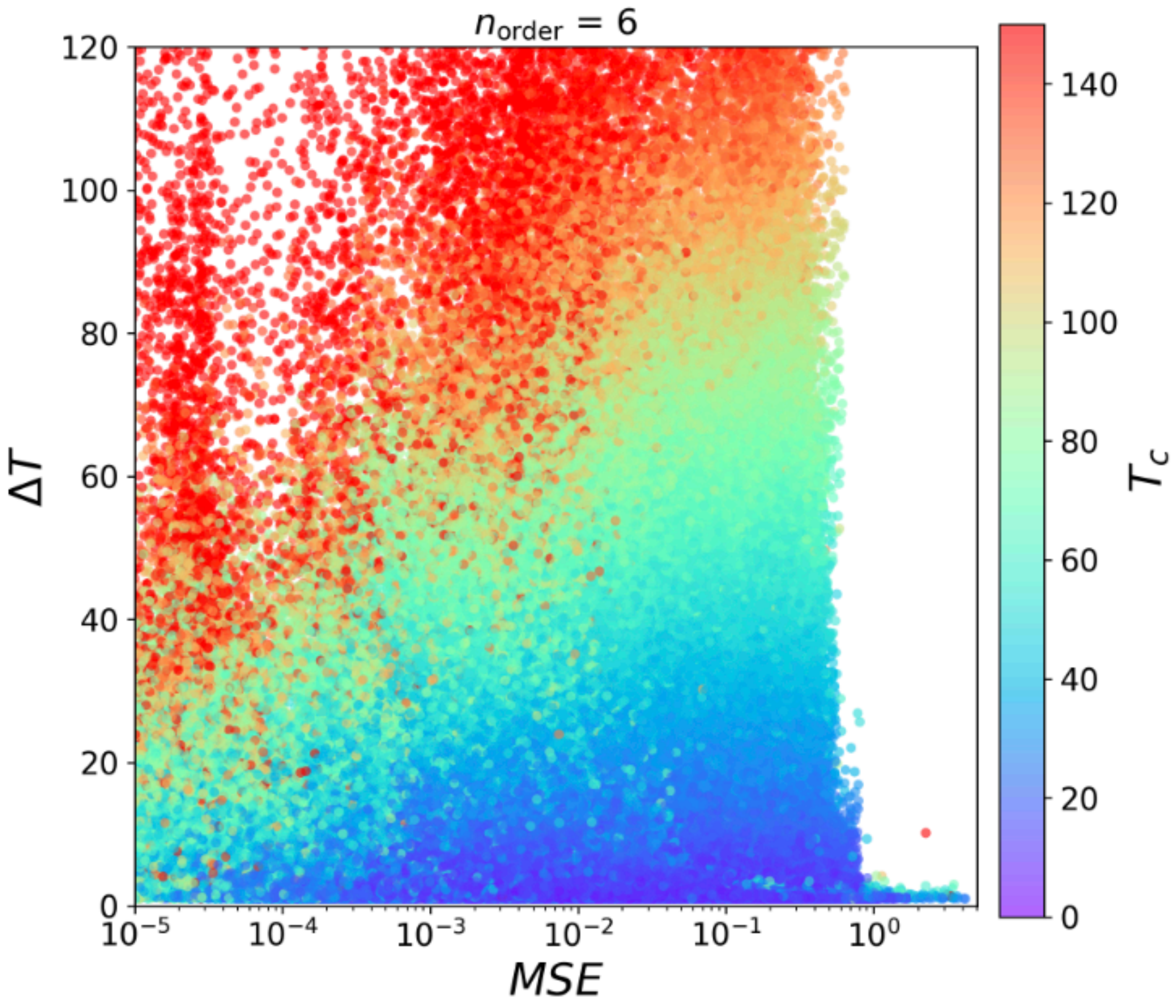
- The fitting results align very well with the raw data.
- We utilize the mean square error (MSE) to quantify the degree of agreement,

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{S_E(T_i)}{T_i} - \frac{\hat{S}_E(T_i)}{T_i}\right)^2$$

- With the factor of $(T - T_C)^2$, the MSE drops quickly with the increasing of $n_{\rm order}$.

Action curve for one-dimensional toy model





- A random scan in

$c \in [0,2]$, $m^2 \in [0,200]$,

$\lambda \in [0,2]$, $\kappa \in [-30,0]$.

- The majority of samples exhibit an MSE below 1, with a maximum value of 4.2.

$$S_E/T_{\text{nuc}} \simeq 140$$

- The MSE exceeds 1 only for $\Delta T < 2$ GeV

Action curve for SSM

- Now we turn to a physical model, the singlet scalar extensions of the Standard Model, which is wildly used in instructional studies of phase transition:

$$V_0(h,s) = -\frac{\mu_H^2}{2}h^2 + \frac{\lambda_H}{4}h^4 - \frac{\mu_S^2}{2}s^2 + \frac{\lambda_S}{4}s^4 + \frac{\lambda_{HS}}{4}h^2s^2$$

$$V_{\rm eff}(h,s;T) = V_0(h,s) + V_{\rm CW}(h,s) + V_{\rm CT}(h,s) + V_{\rm 1T}(h,s;T) + V_{\rm ring}(h,s;T)$$

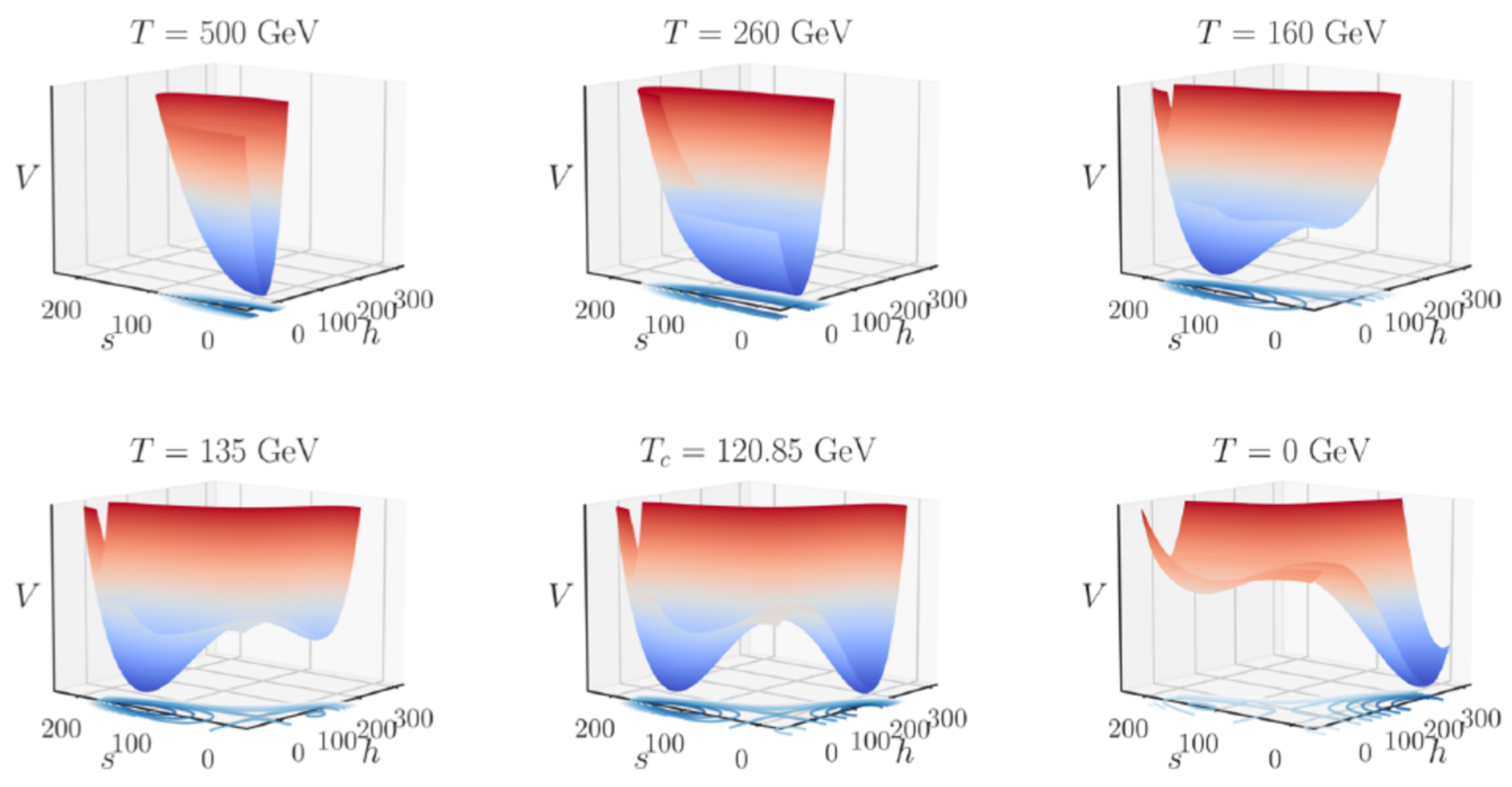
We choose the OS-like scheme, the Landau gauge and the Parwani method:

$$V_{\rm CW}(h,s) + V_{\rm CT}(h,s) = \sum_i (-1)^{s_i}\frac{g_i}{64\pi^2}\left\{m_i^4(h,s)\left[\log\frac{m_i^2(h,s)}{m_i^2(v_h,v_s)} - \frac{3}{2}\right] + 2m_i^2(h,s)m_i^2(v_h,v_s)\right\}$$

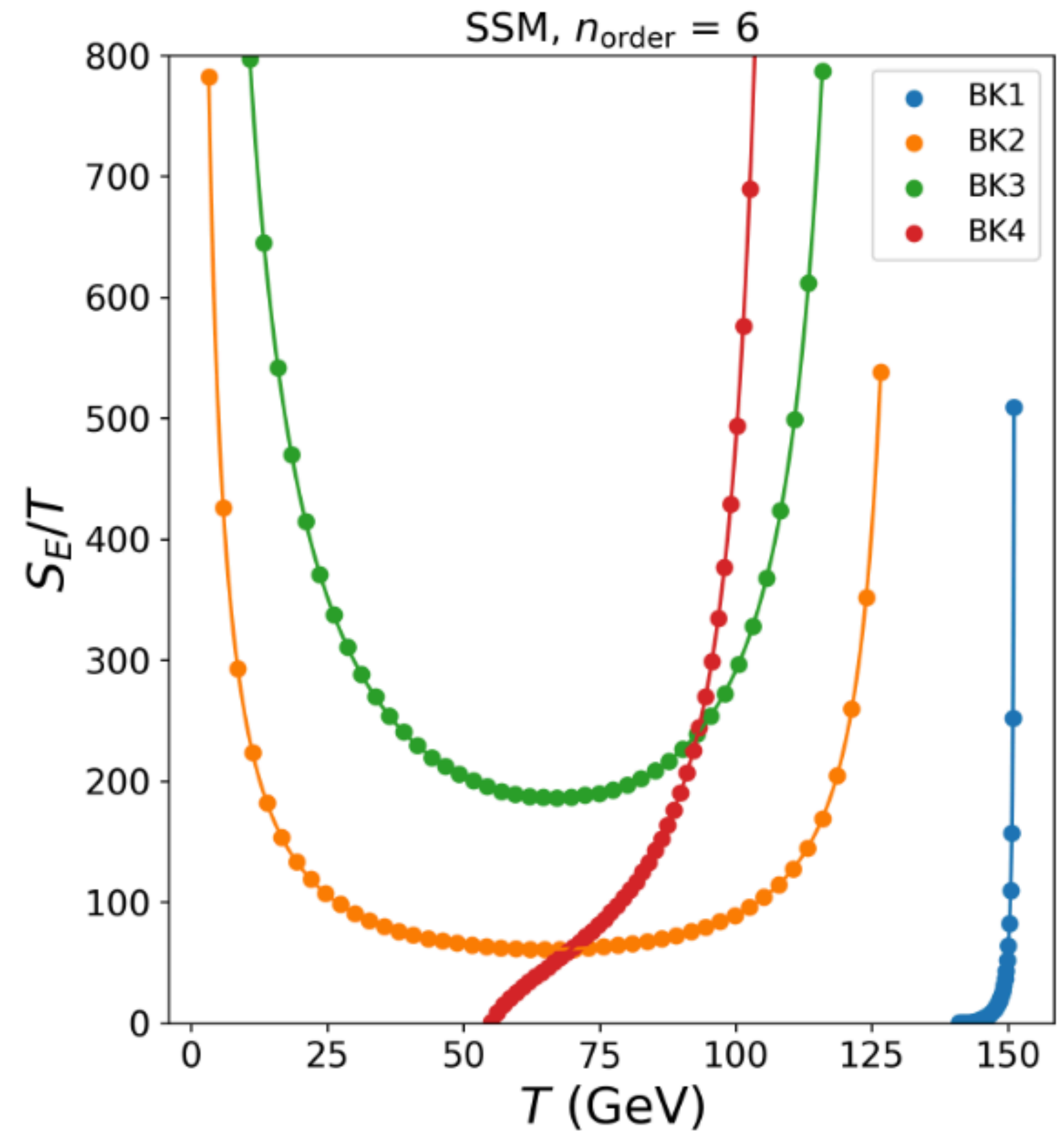
$$V_{\rm 1T}(h,s) = \frac{T^4}{2\pi^2}\left[\sum_B g_B J_B\left(\frac{m_B(h,s)}{T}\right) + \sum_F g_F J_F\left(\frac{m_F(h,s)}{T}\right)\right]$$

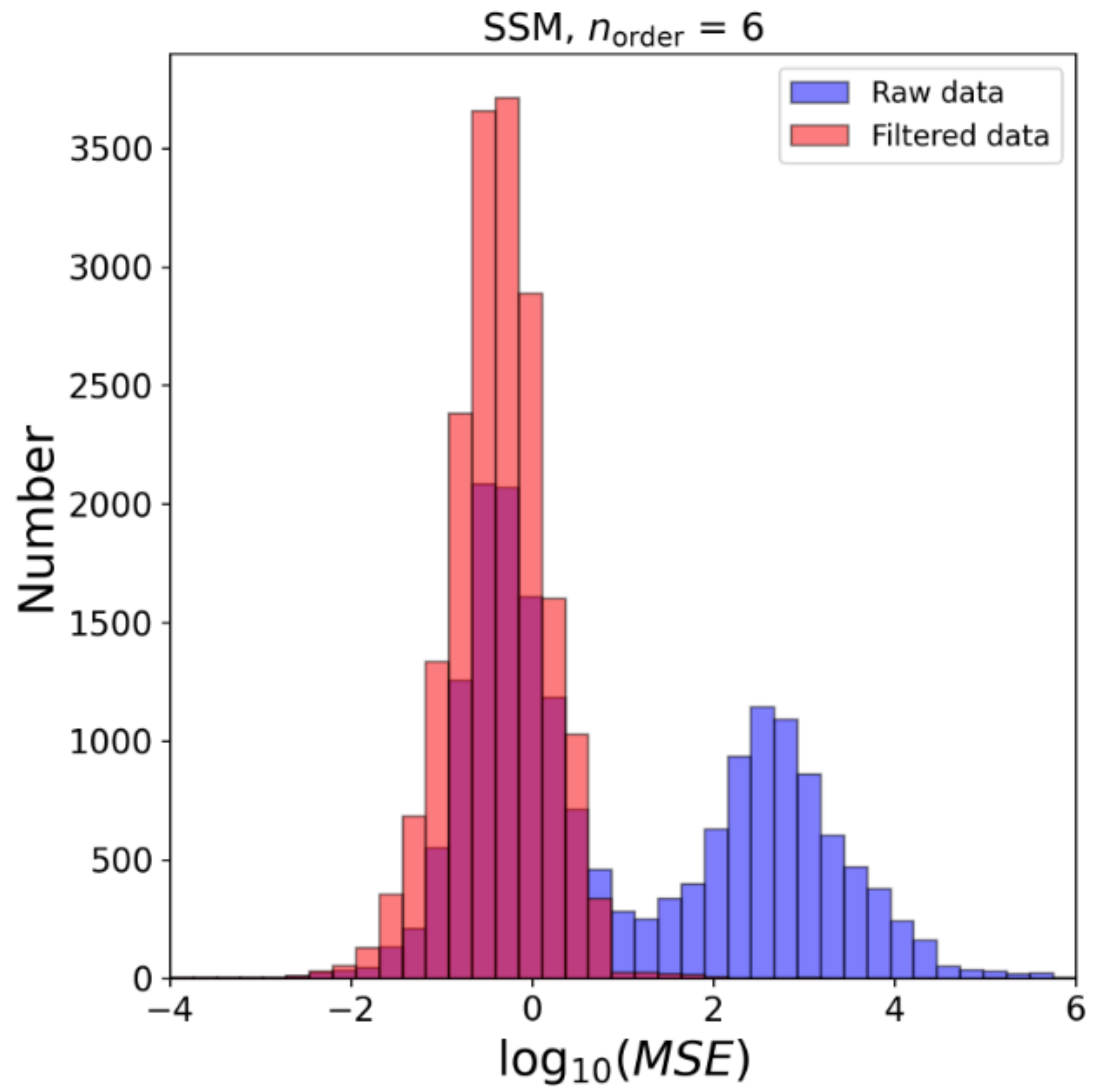
Action curve for SSM

➤ Now we turn to a physical model, the singlet scalar extensions of the Standard Model, which is wildly used in instructional studies of phase transition:



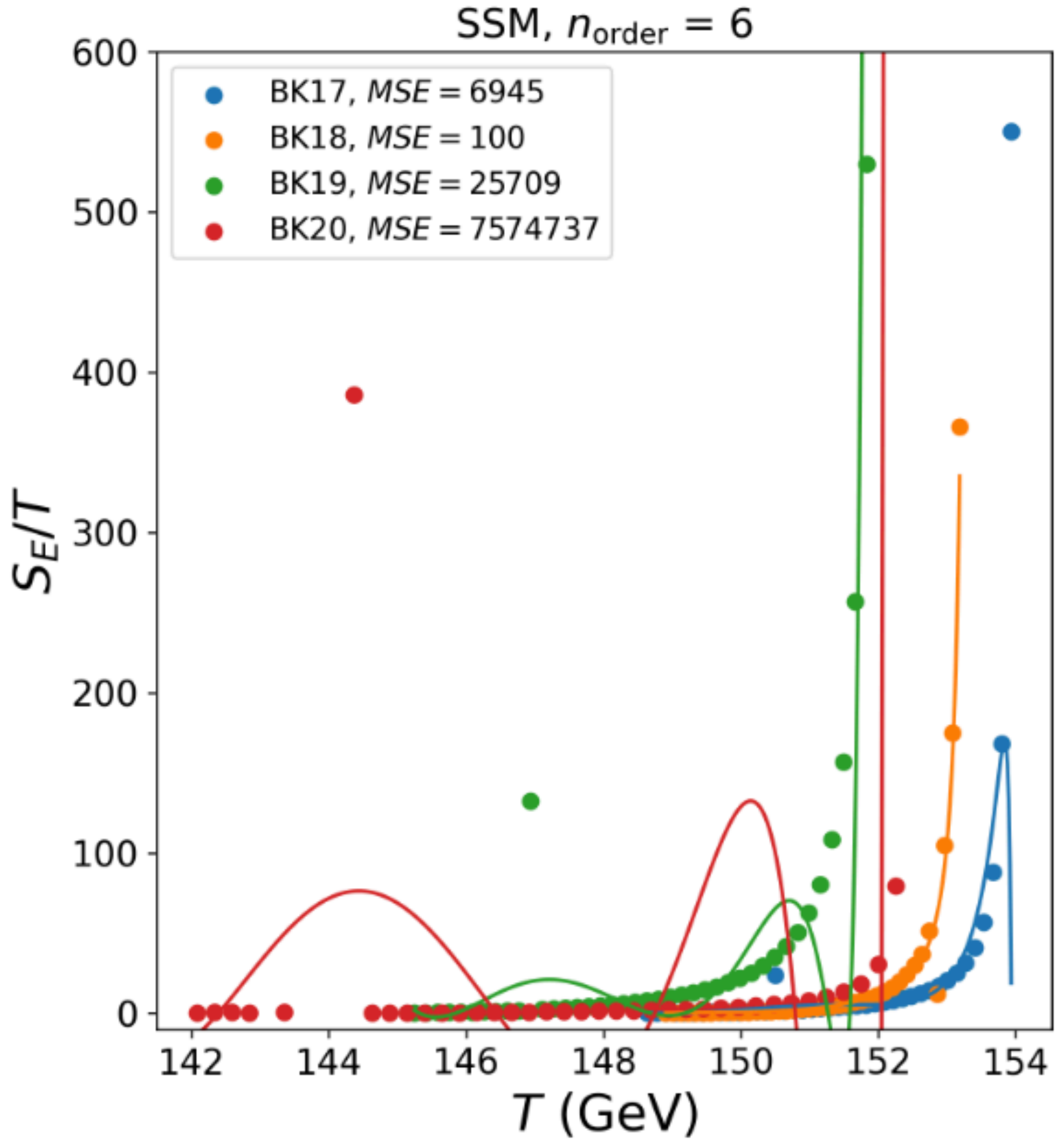
Action curve for SSM

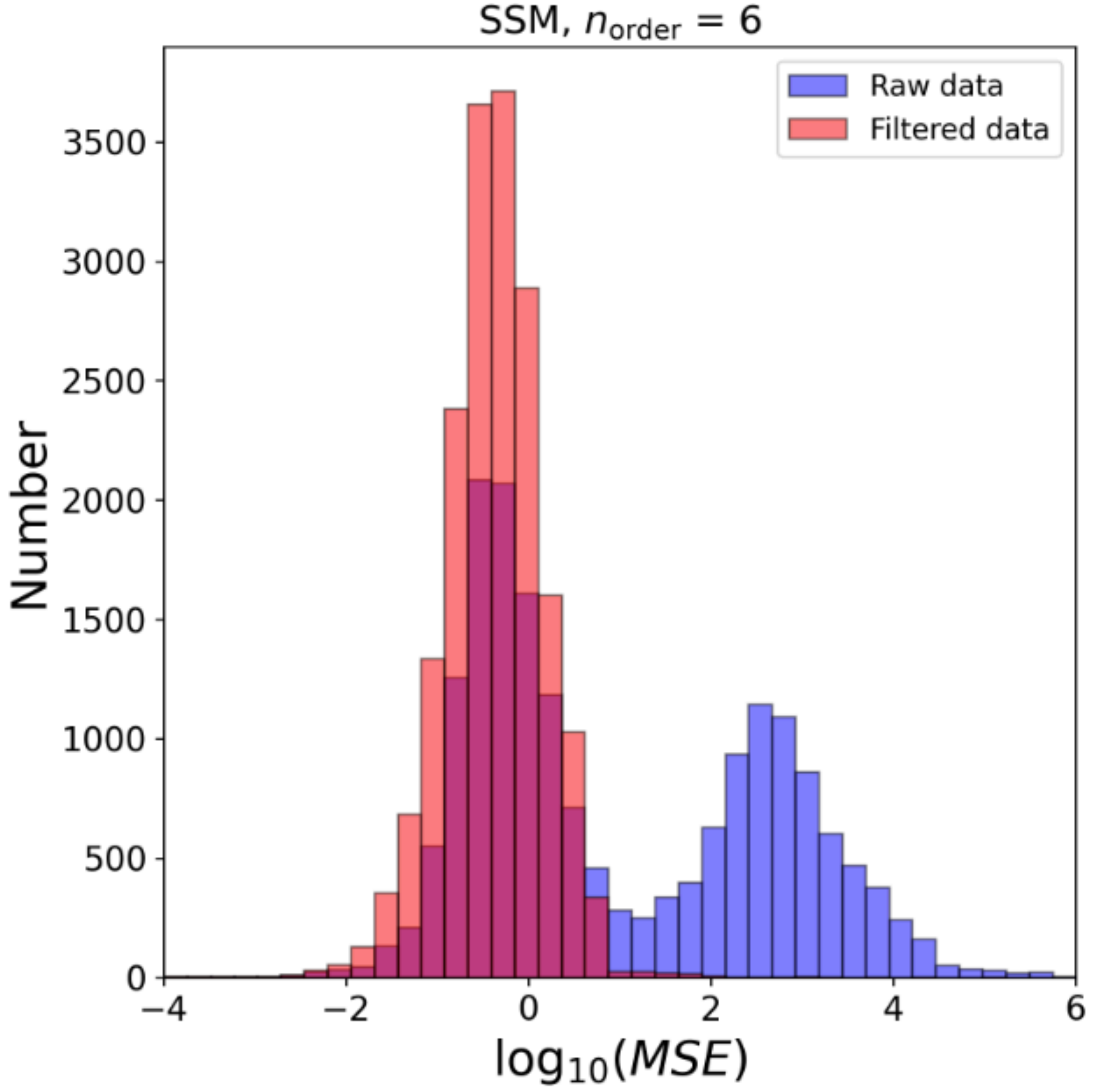




- The fitting results also align very well with the raw data for most of the samples.
- The samples of large MSE are attributed to incorrect S_E.
- After excluding these erroneous S_E, we can achieve good agreement.
- As this serves as an auxiliary approach, users retain the flexibility to revert to traditional methods when necessary.

Action curve for SSM

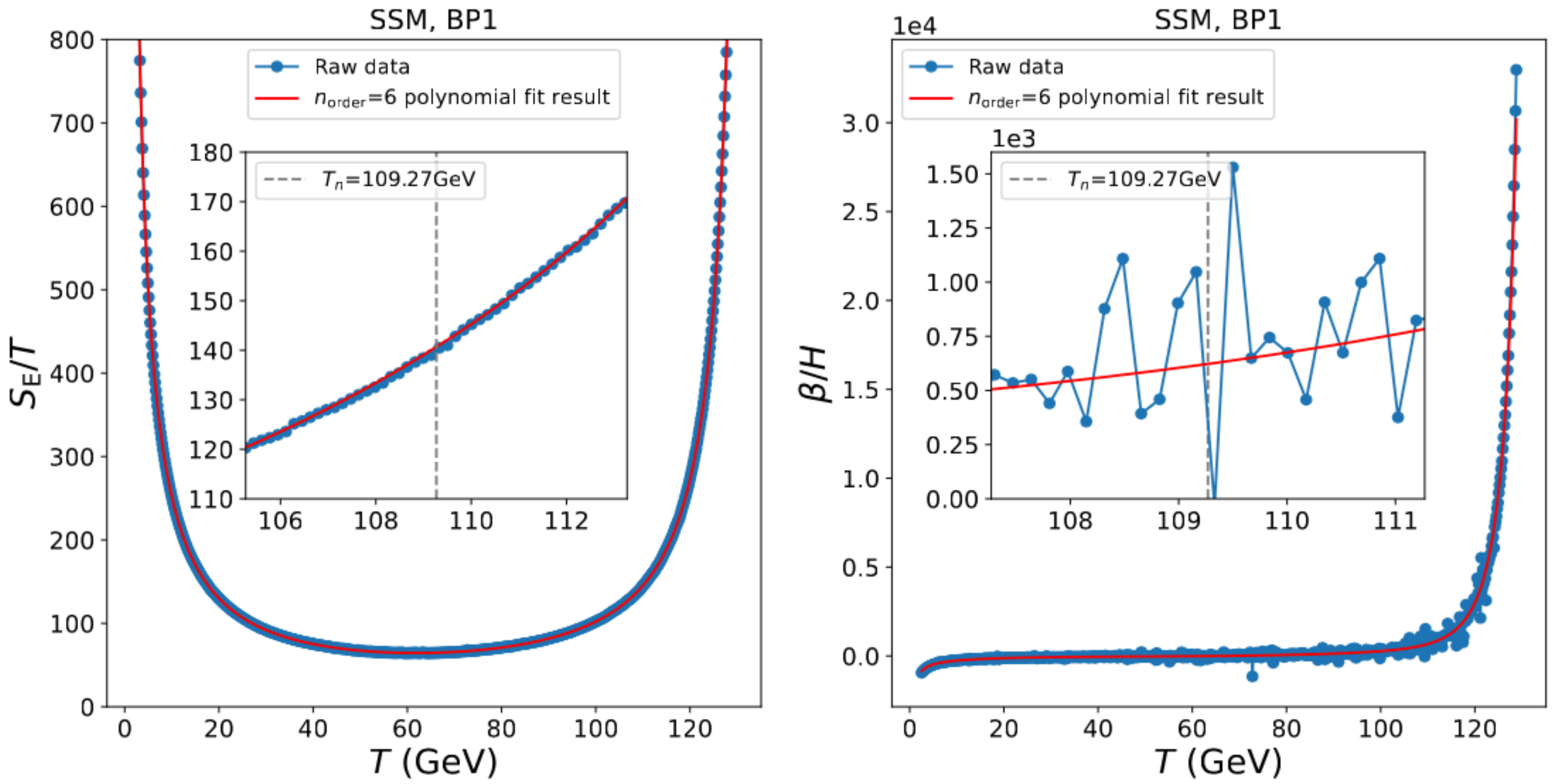




- The fitting results also align very well with the raw data for most of the samples.
- The samples of large MSE are attributed to incorrect S_E.
- After excluding these erroneous S_E, we can achieve good agreement.
- As this serves as an auxiliary approach, users retain the flexibility to revert to traditional methods when necessary.

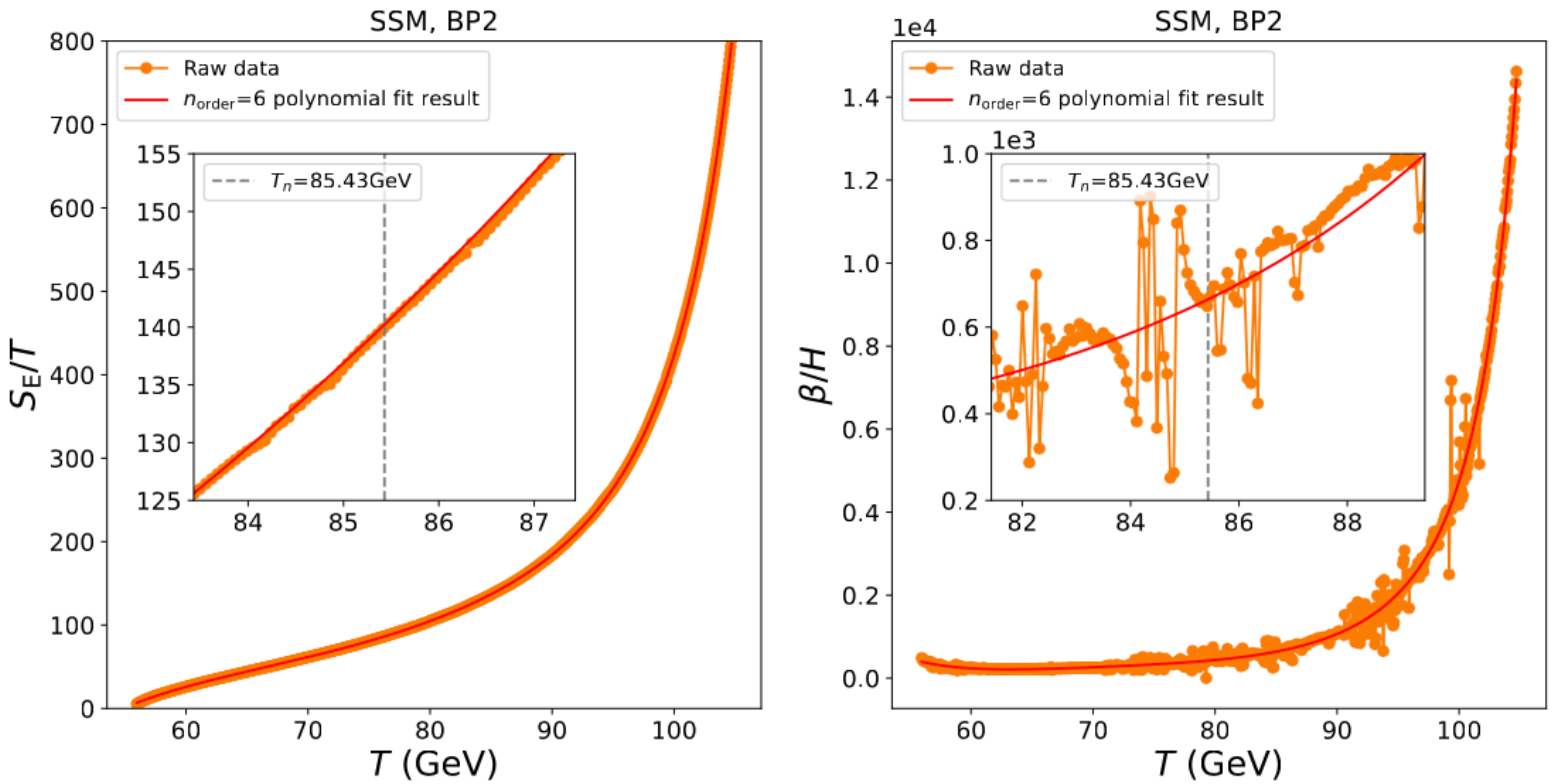
$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \frac{f(x+h) - f(x-h)}{2h}$$

Action curve for SSM



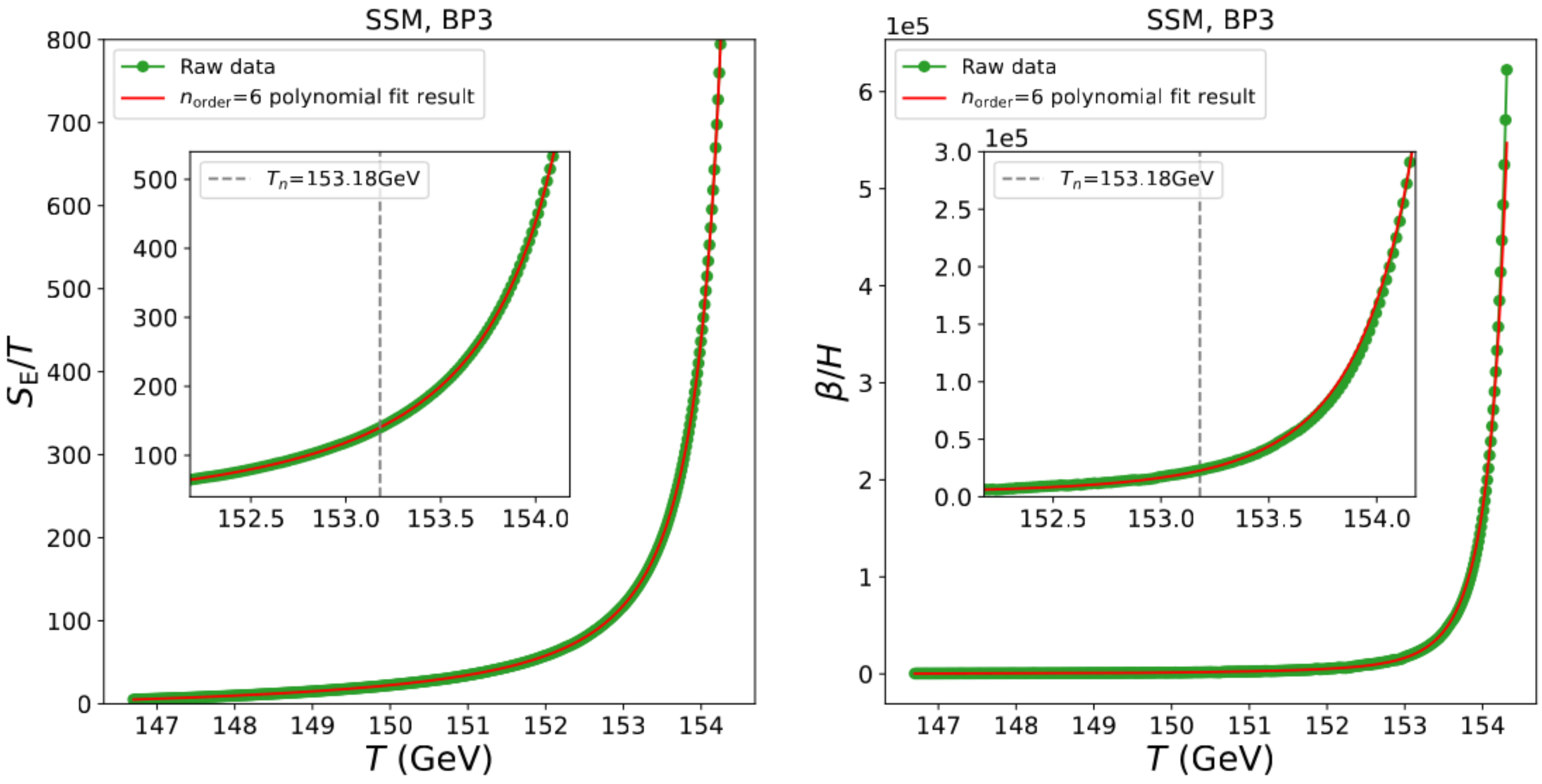
$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \frac{f(x+h) - f(x-h)}{2h}$$

Action curve for SSM



$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \frac{f(x+h) - f(x-h)}{2h}$$

Action curve for SSM



Action curve for SSM

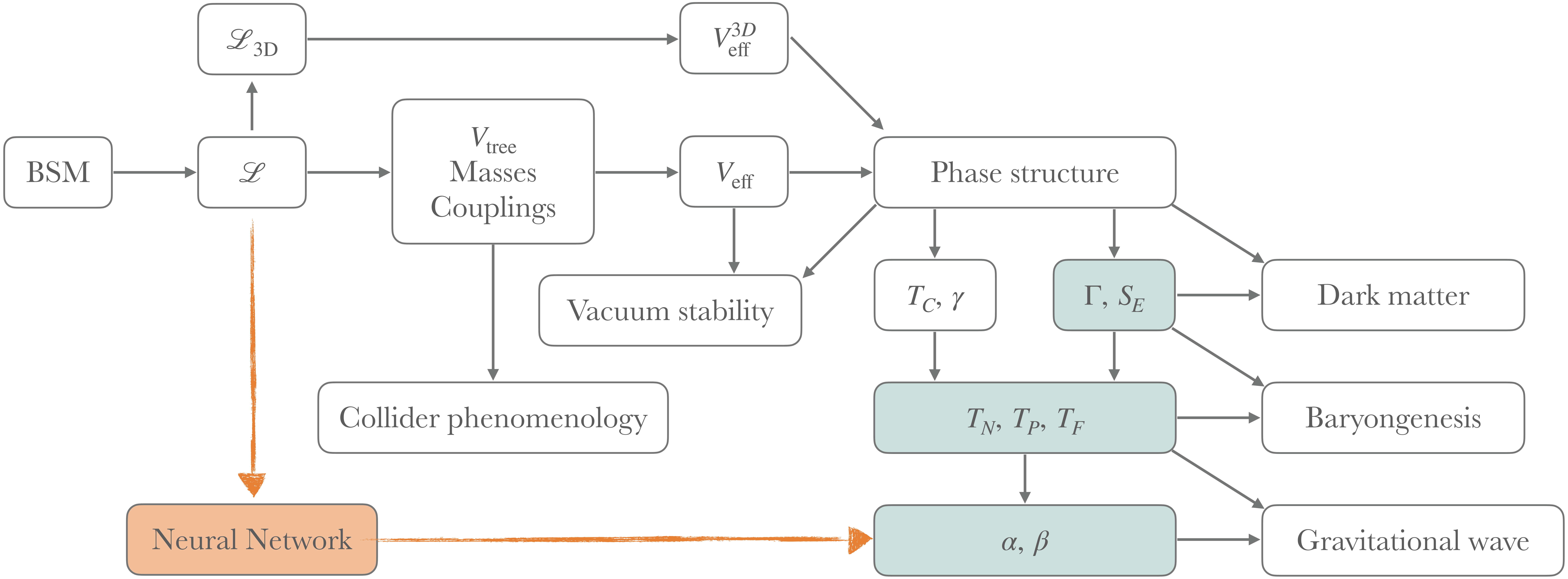
Calculate one β/H for 10 times with different h

	$h = 1$	$h = 0.1$	$h = 0.01$	$h = 0.001$	Fititng
1	1621.95	1573.90	1530.38	1461.57	1500.25
2	1622.96	1534.27	1915.43	901.45	1500.24
3	1622.34	1533.25	1505.82	991.253	1499.92
4	1618.56	1503.12	1278.73	-1502.6	1500.33
5	1621.08	1466.95	1202.39	5570.78	1500.12
6	1579.43	1503.90	1635.98	845.765	1500.00
7	1622.35	1506.99	1691.52	1728.72	1499.98
8	1623.08	1517.53	1077.92	2533.38	1500.44
9	1620.85	1503.17	1171.85	-380.135	1500.08
10	1622.14	1523.33	1812.96	1863.13	1500.13
Mean	1617.47	1516.64	1482.30	1401.33	1500.15
Uncertainty	12.74	26.46	273.30	1769.83	0.15

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \frac{f(x+h) - f(x-h)}{2h}$$

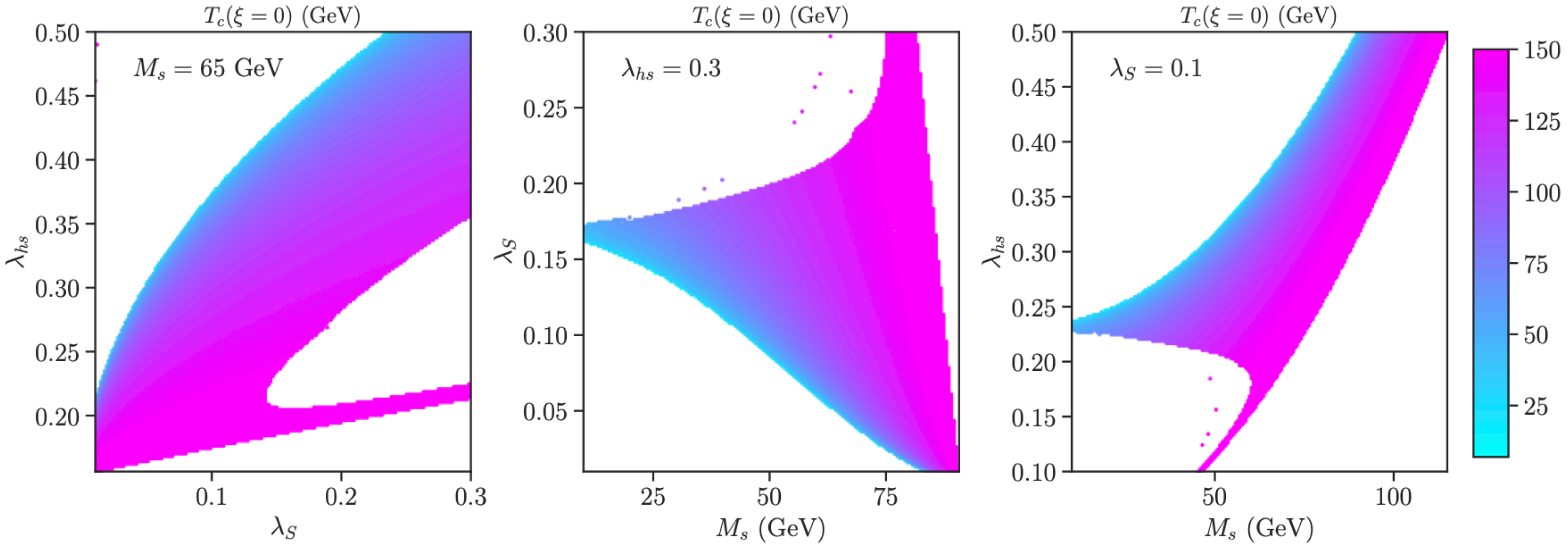
- It is doable to model the action curve using polynomial fitting.
- For one benchmark point, we only need to calculate action about 30 times, then we can
  - Precisely calculate the β
  - Rapidly calculate the T_{nuc} and T_{per} .
  - And one more thing.

Neural Network Approximation



Neural Network Approximation

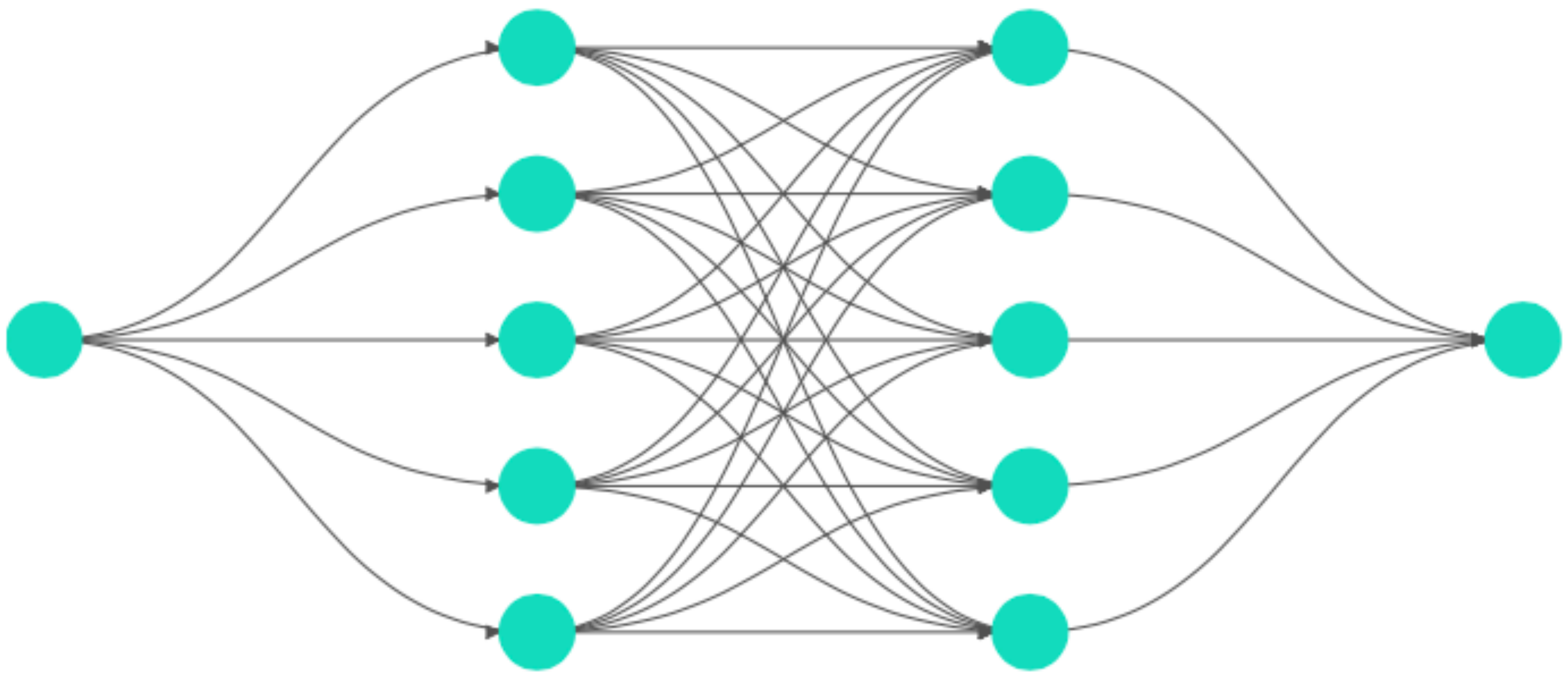
- The transition parameters changes smoothly with the Lagrangian Parameters.



- This allows them to be predicted via neural network approximation.

Neural Network Approximation

Lagrangian Parameters



$T_{\rm cri}$, $T_{\rm nuc}$, $T_{\rm per}$

S_E at a fixed T

α , β

Action Curve

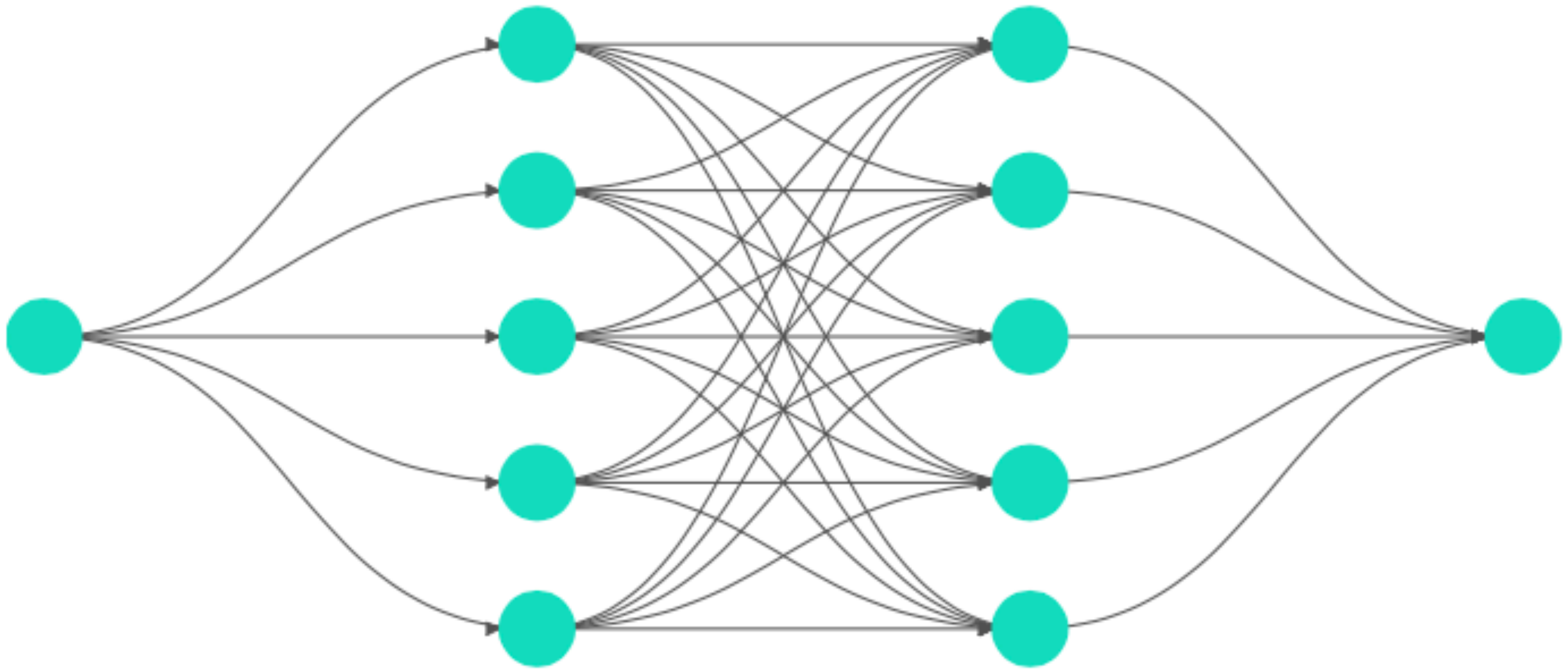
$T_{\rm cri}$, $T_{\rm min}$

$T_{\rm nuc}$, $T_{\rm per}$

α , β

Neural Network Approximation

Lagrangian Parameters

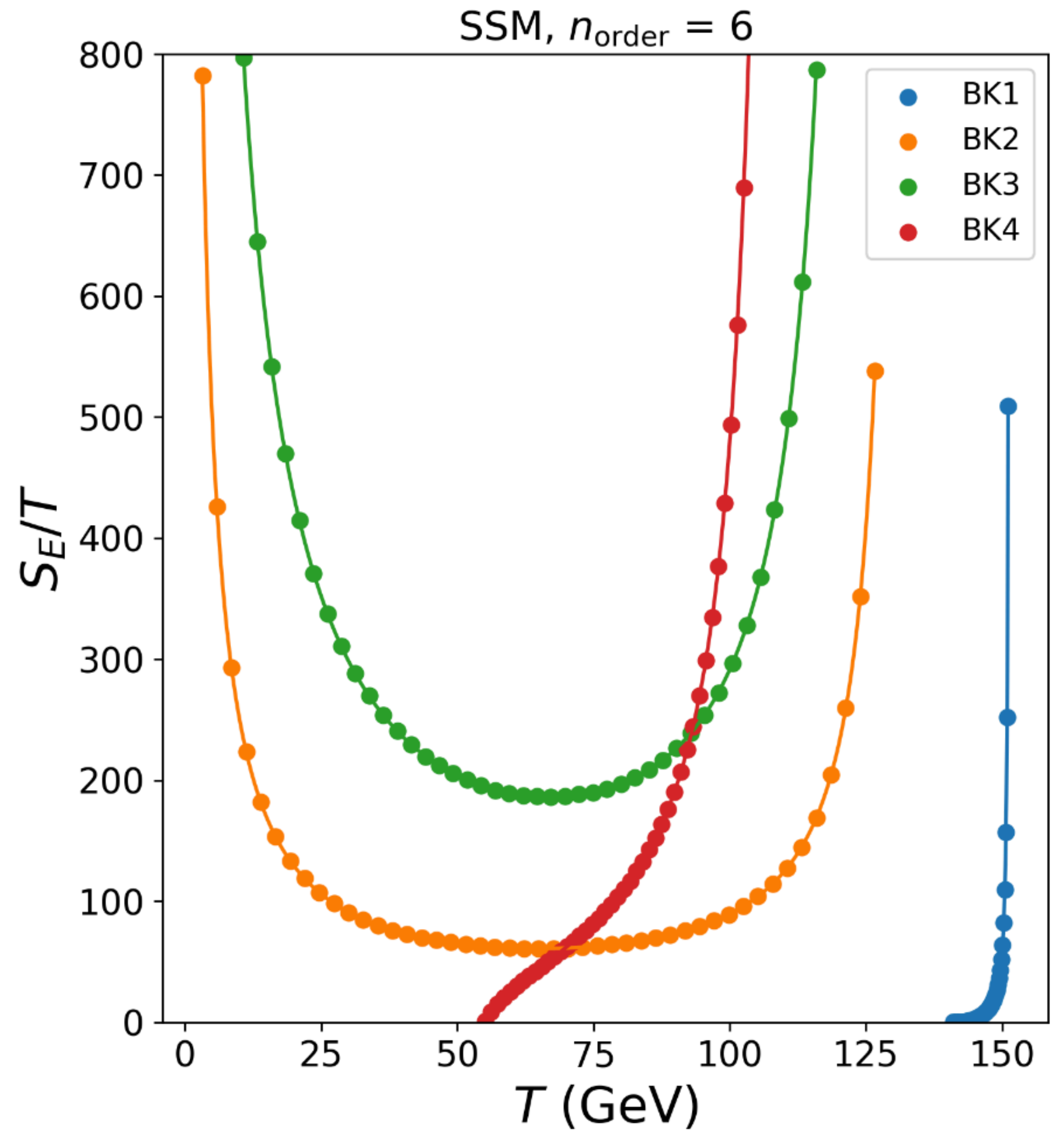


Action Curve

T_{cri} , T_{min}

T_{nuc} , T_{per}

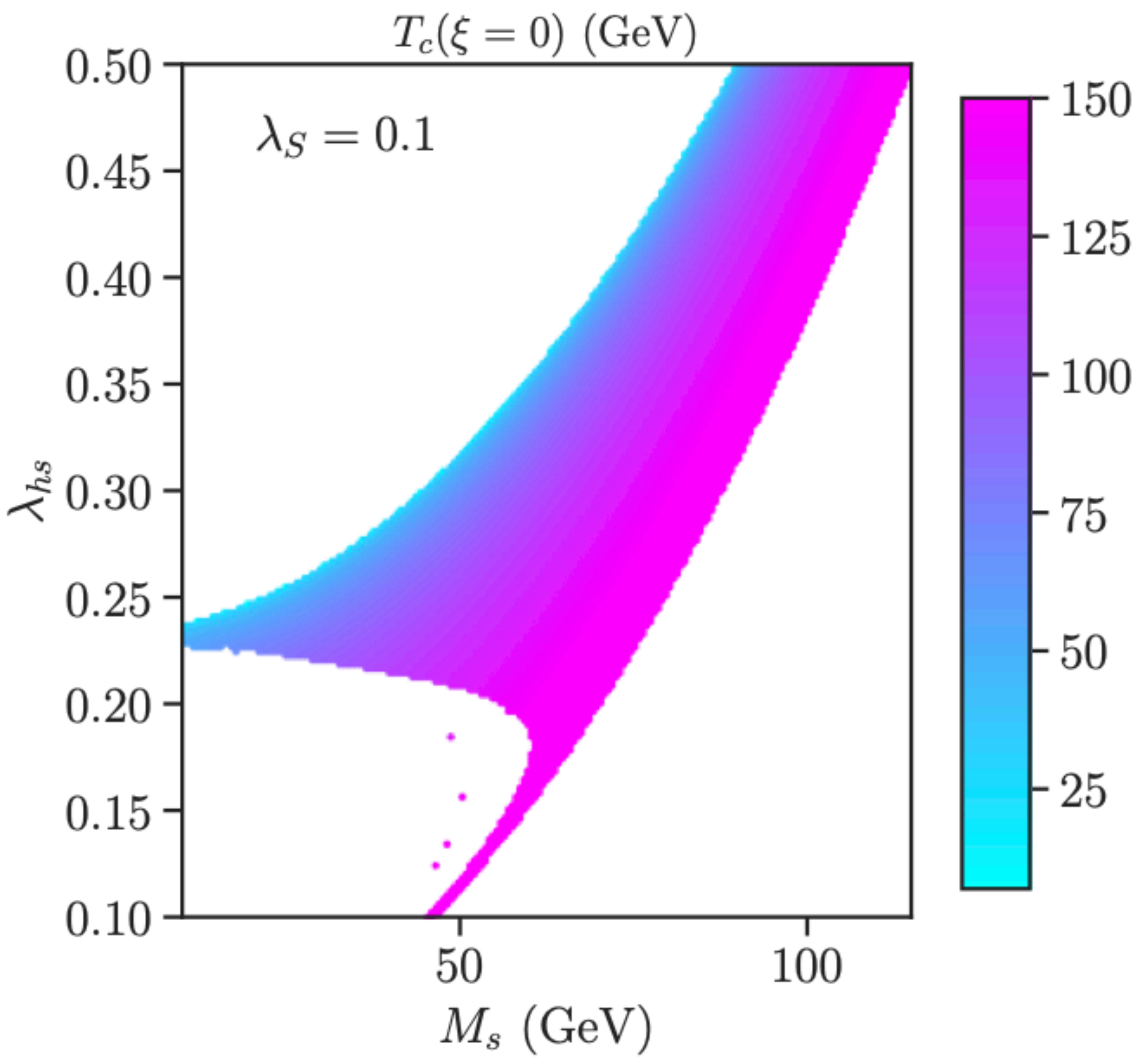
α , β

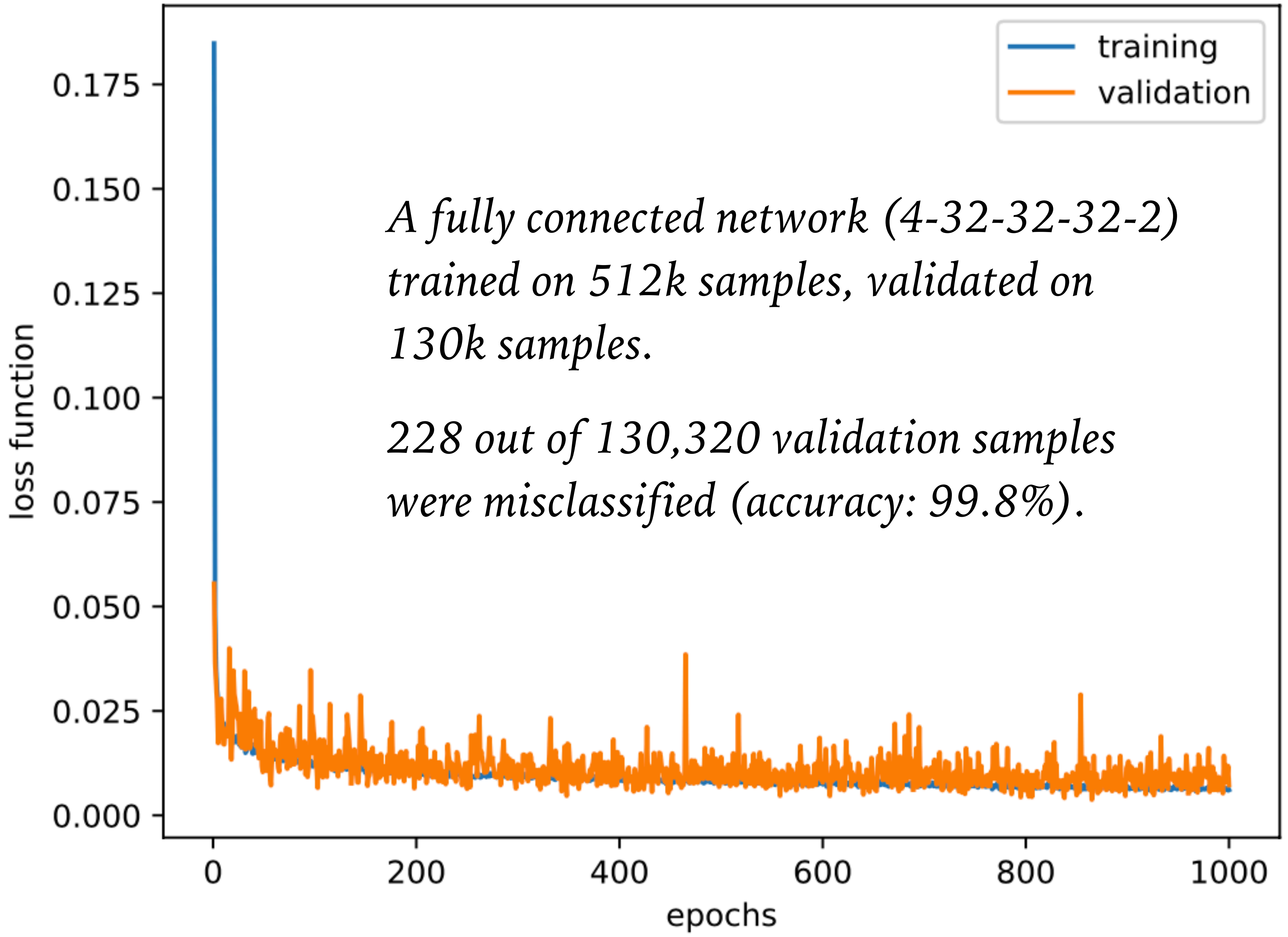


- ➤ We can verify the action curve by sampling several points along the line.
- ➤ This ensures that the predicted result is 100% accurate.

Neural Network Approximation

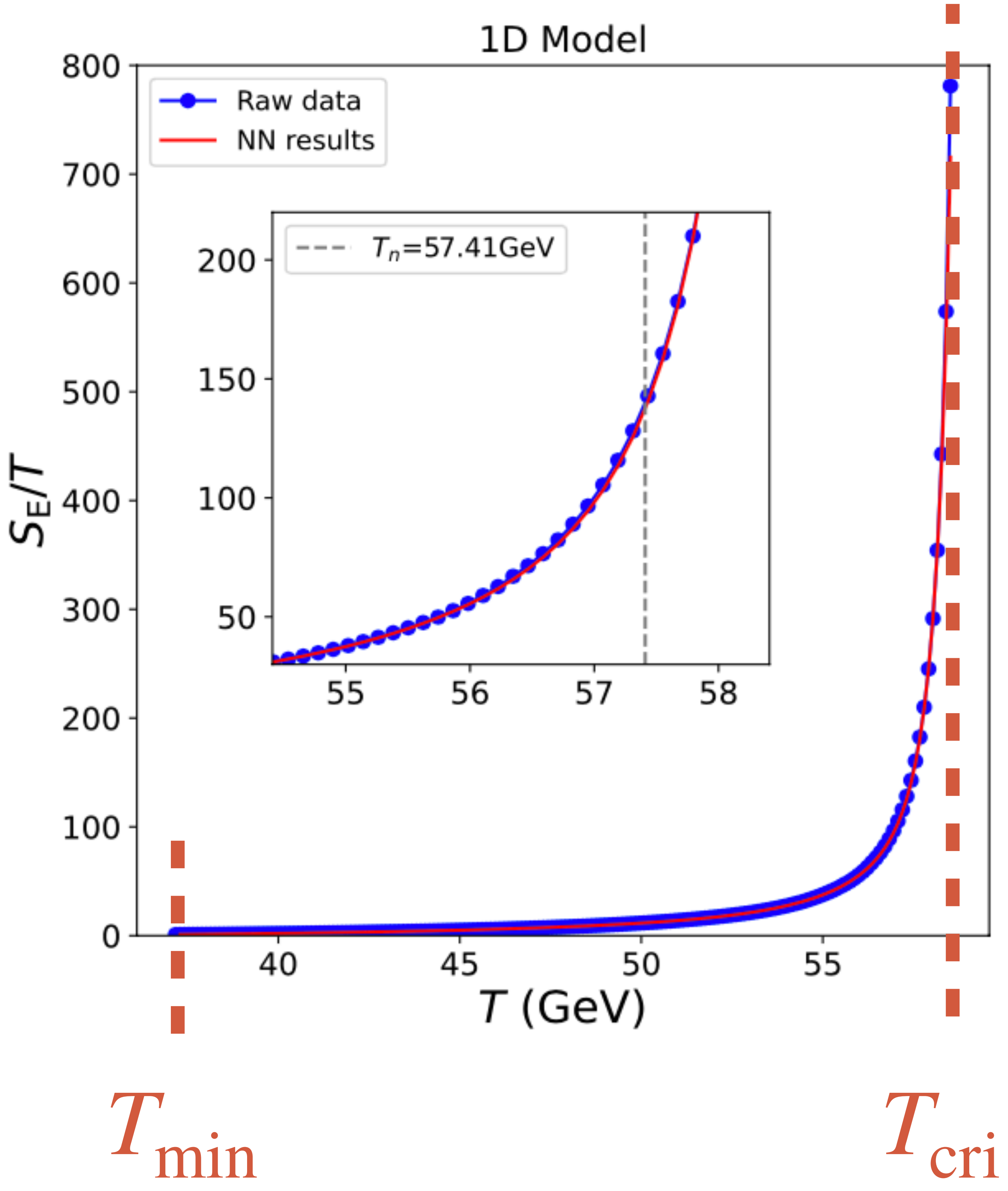
➤ Step 1: Distinguish the parameter space that has valid action curve

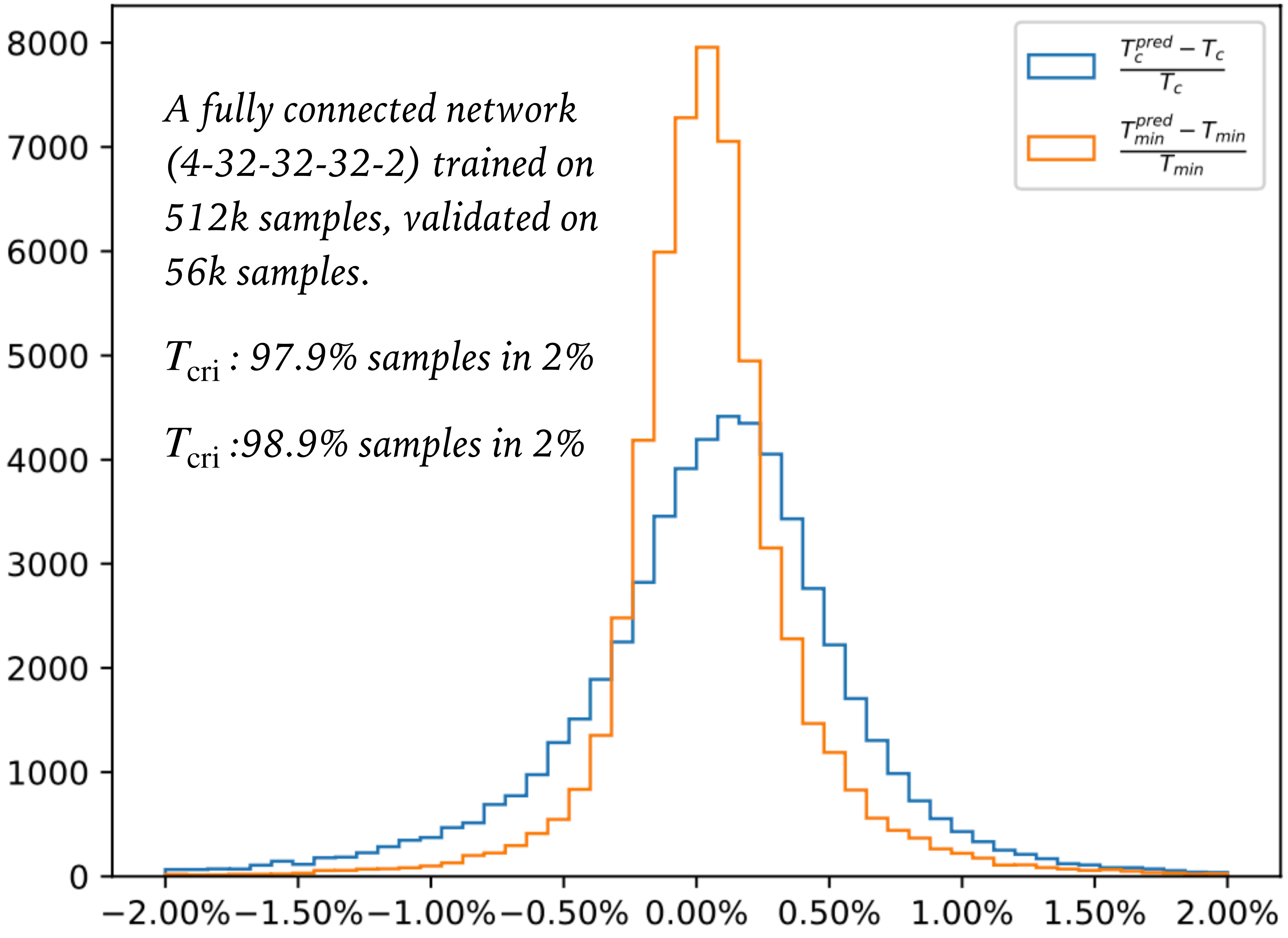




Neural Network Approximation

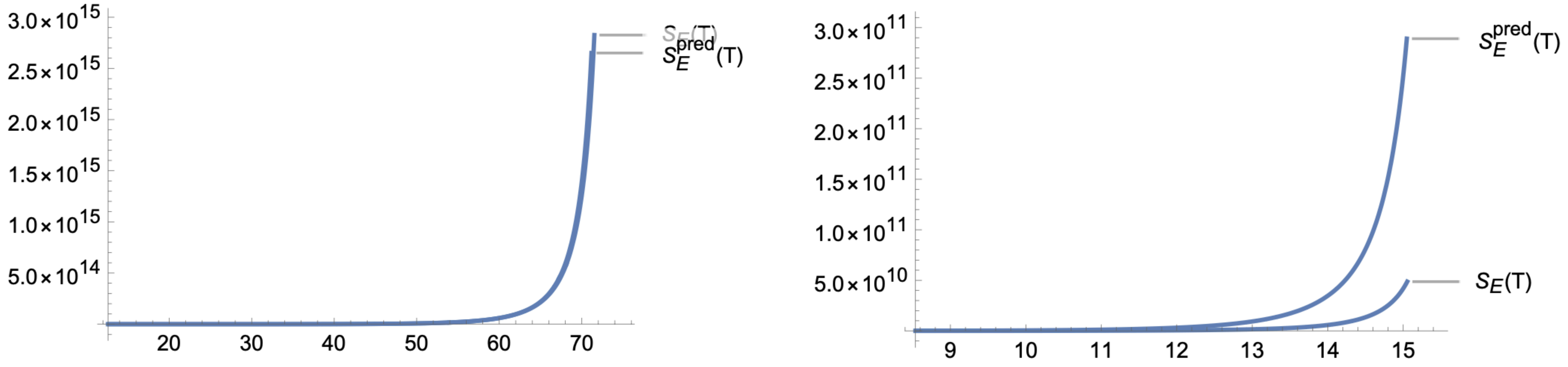
➤ Step 2: Predict $T_{\rm cri}$ and $T_{\rm min}$





Neural Network Approximation

- Step 3: Predict the polynomial coefficients
- A fast KAN network with architecture 13−64−64−64−64−64−1, trained on 350k samples and validated on 51k samples.



- Approximately 71.48% of the cases have a relative error of less than 30%.

Summary

- We find that the sixth-order polynomial fitting can provide an excellent approximation for the groomed action curves. This approach offers advantages for accurately calculating transition temperatures and gravitational wave spectra.
- We can use neural networks to predict action curves directly from model Lagrangian parameters. This method enables efficient validation of estimation accuracy
- The paper is in preparation, but the codes of our polynomial fitting for action curves in CosmoTransitions and PhaseTracer2 are almost ready.

Thank you!

Yang Zhang，Henan Normal University