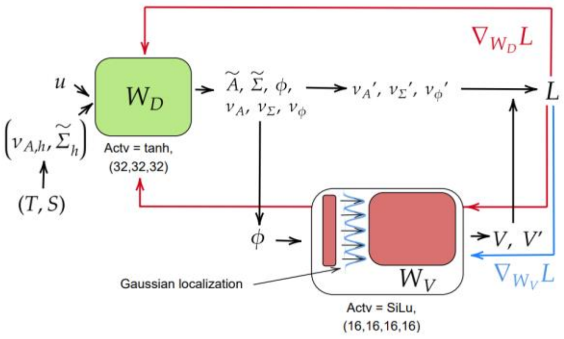
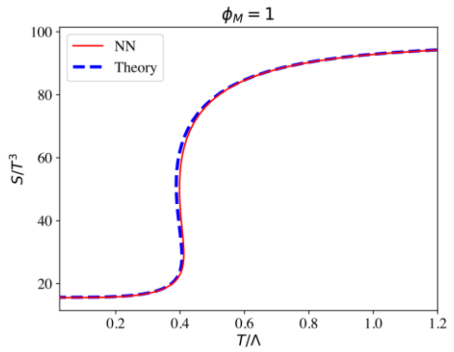
Solving an inverse problem using neural networks

Yago Bea

University of Barcelona





Based on: 2403.14763

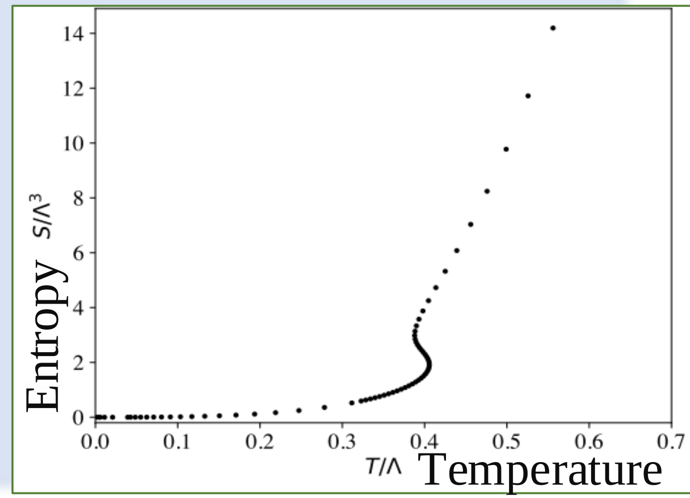
With Raul Jiménez, David Mateos, Shuheng Liu,
Pavlos Protopapas, Pedro Tarancón and Pablo Tejerina

Main idea

Einstein eqs.

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$

Direct problem



Main idea

Holography

Gravity 4+1

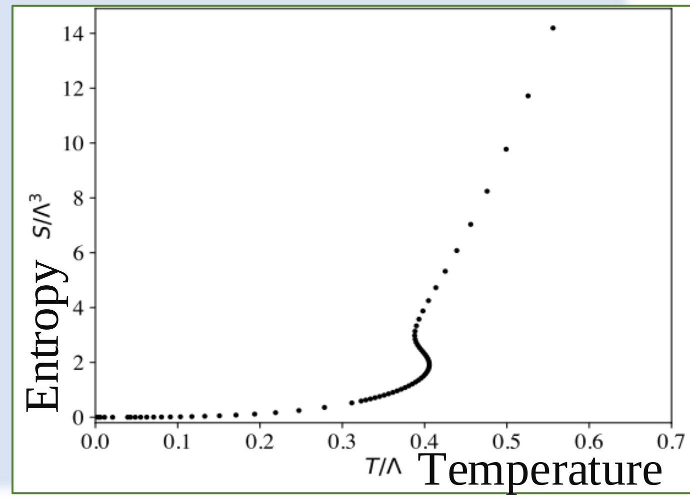
QFT 3+1

Einstein eqs.

Direct problem →

Equation of state

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$



Main idea

Holography

Gravity 4+1

QFT 3+1

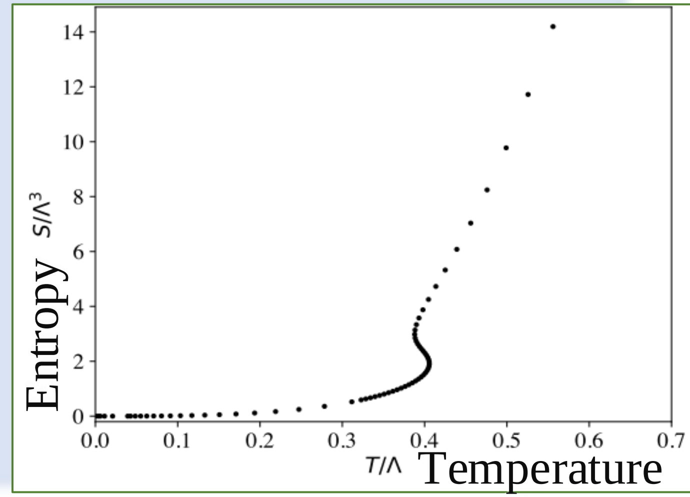
Einstein eqs.

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$

Direct problem →

← Inverse problem

Equation of state



Main idea

Holography

Gravity 4+1

QFT 3+1

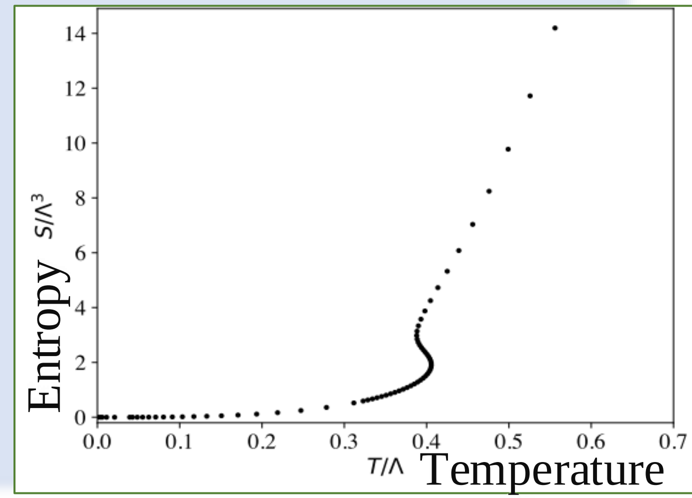
Einstein eqs.

$$S = \frac{2}{\kappa_5^2}\int dx^5\sqrt{-g}\left[\frac{1}{4}R - \frac{1}{2}(\nabla\phi)^2 - V(\phi)\right]$$

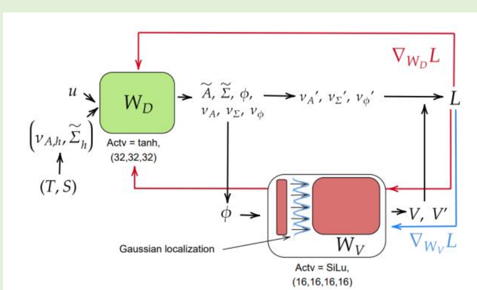
Direct problem →

← Inverse problem

Equation of state



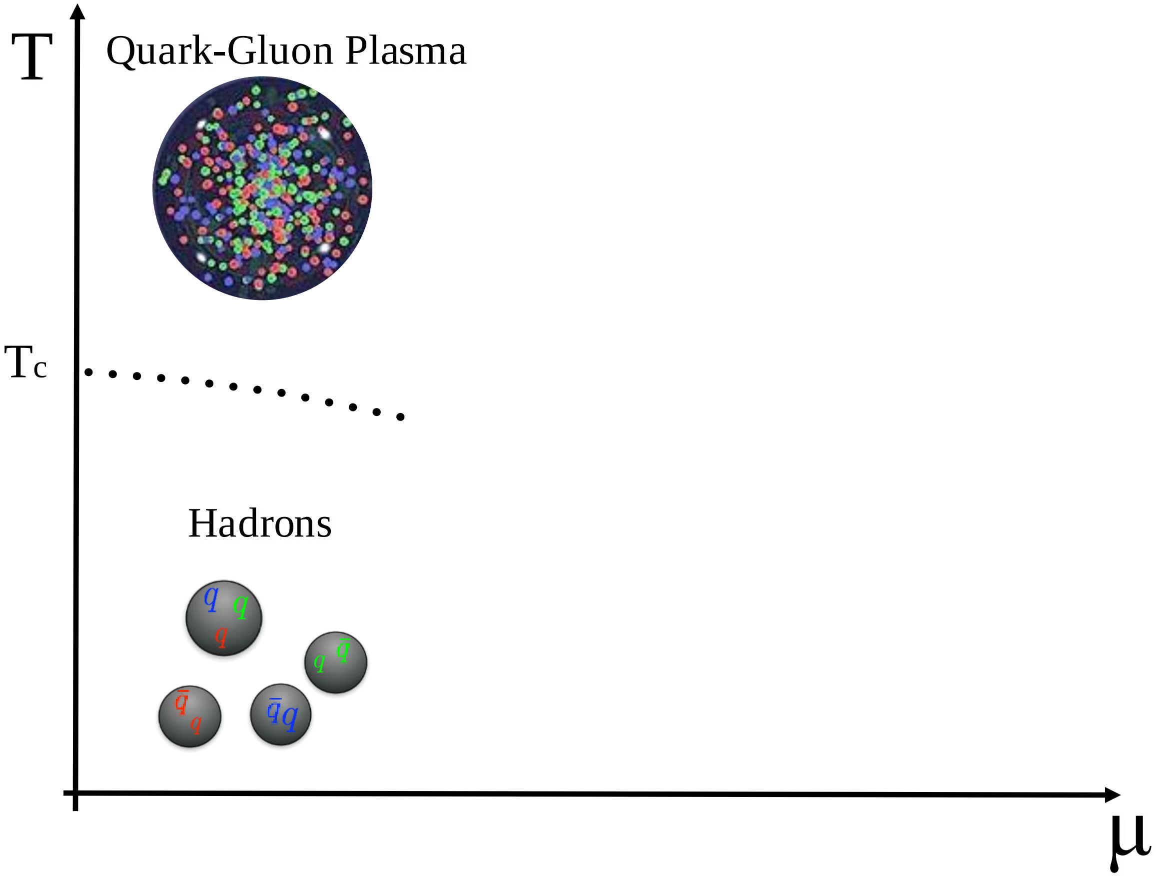
Neural Networks



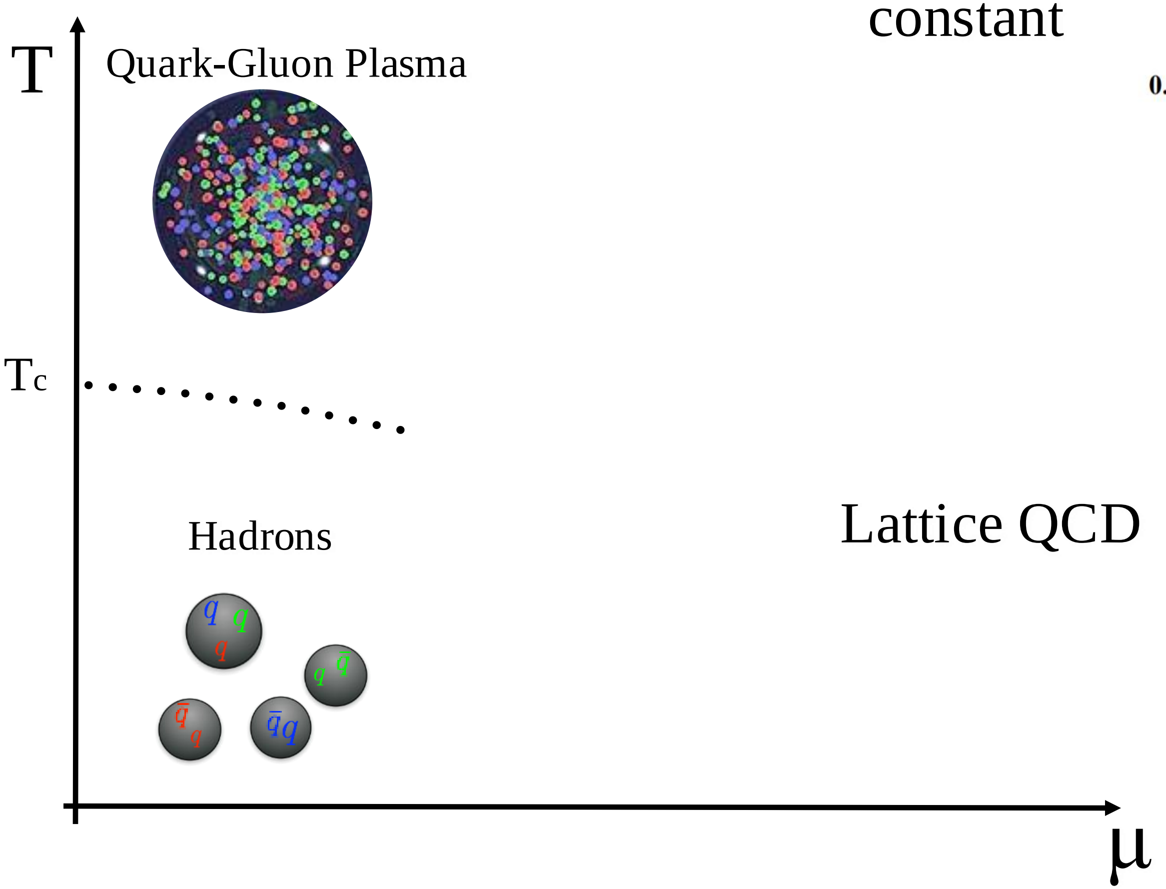
→ We use NN to solve a problem not addressed before

Holography: motivation

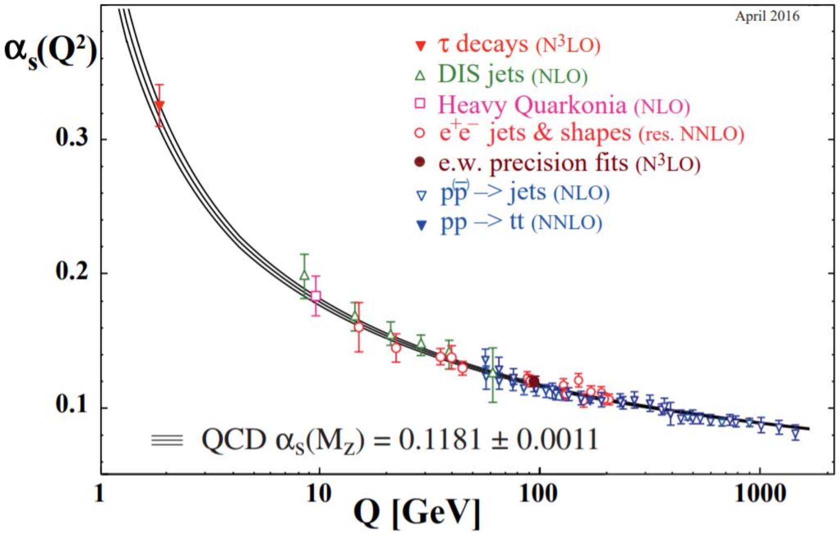
QCD & Holography



QCD & Holography

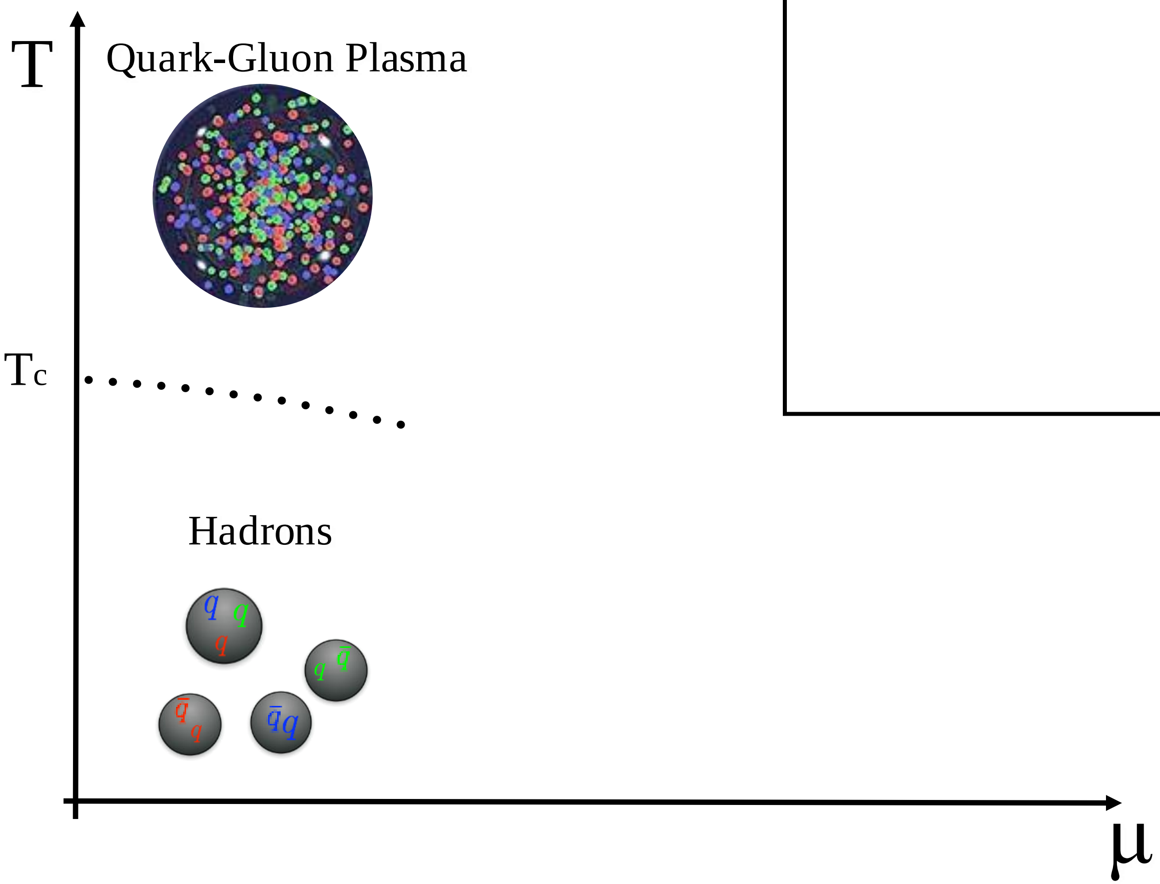


QCD coupling constant



Lattice QCD

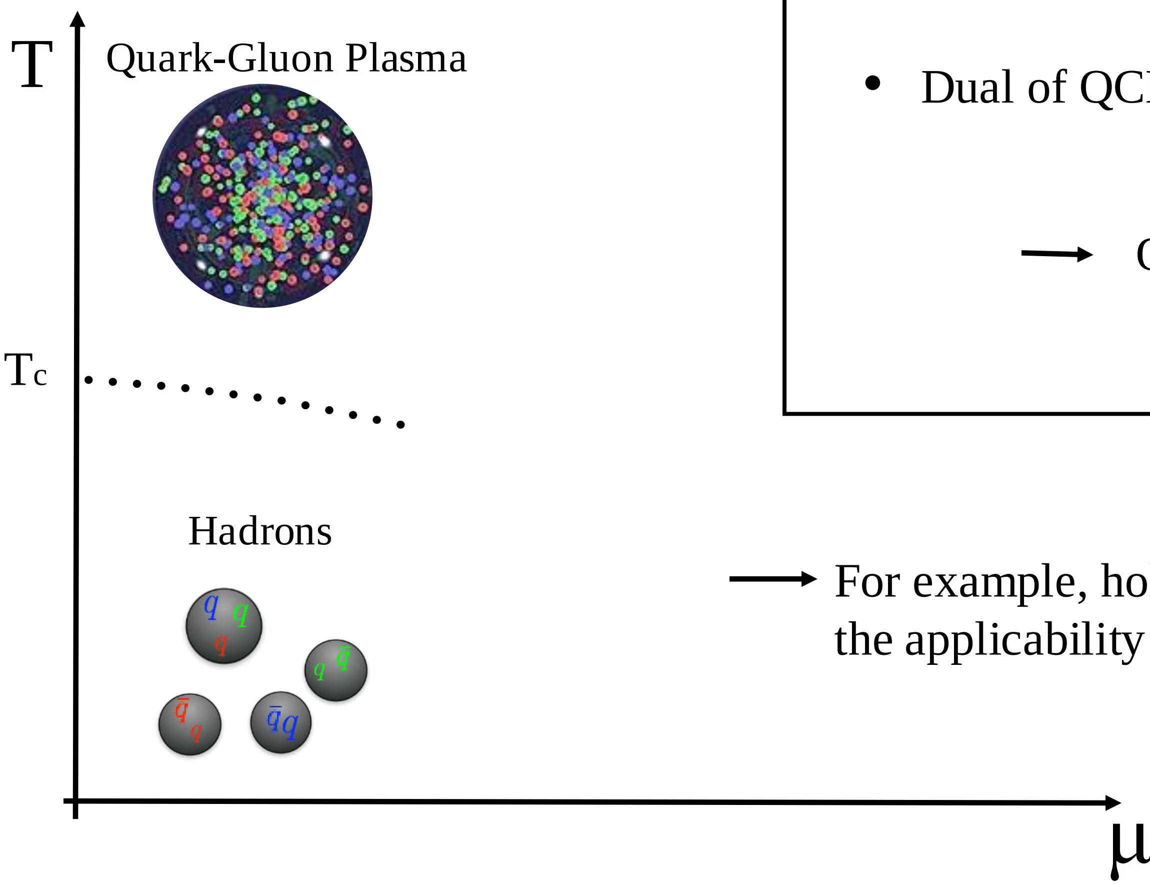
QCD & Holography



Holography

- Strongly coupled QFT
- Out of equilibrium physics

QCD & Holography



Holography

- Strongly coupled QFT
- Out of equilibrium physics
- Dual of QCD not known…

→ Qualitative insights

→ For example, holography as a laboratory to study the applicability of hydrodynamics.

Holography: Our model

- Gravity in 4+1 dim :

$$S \sim \int d^{3+1}x \sqrt{-g}\,(R - 2\Lambda)$$

Holography: Our model

- CFT on Minkowski in 3+1 dim
- Decoupled sector of the stress tensor $T^{\mu\nu}$

Holography

- Gravity in 4+1 dim :

$$S \sim \int d^{3+1}x \sqrt{-g}\,(R - 2\Lambda)$$

Holography: Our model

- QFT on Minkowski in 3+1 dim
- Decoupled sector of the stress tensor $T^{\mu\nu}$ and $\mathcal{O}$

Holography

We want to make contact with phenomenology so we proceed to break conformality:

→ We introduce a scalar field

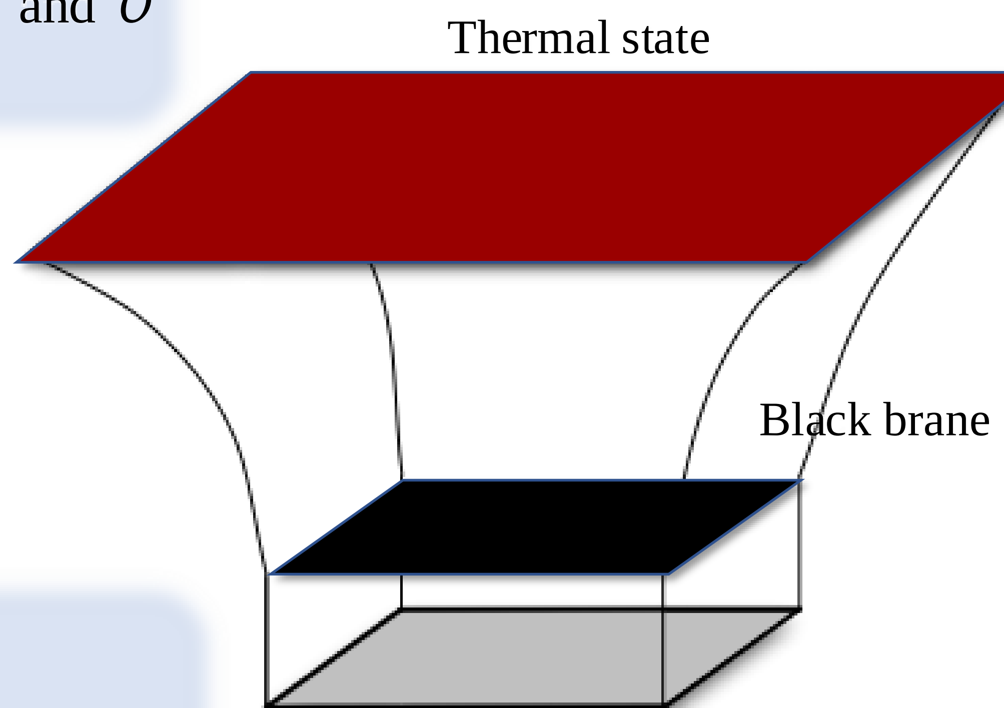
- Gravity in 4+1 dim :

$$S \sim \int d^{3+1}x\,\sqrt{-g}\,(R - \frac{1}{2}(\partial\phi)^2 - V(\phi))$$

Holography: Our model

- QFT on Minkowski in 3+1 dim
- Decoupled sector of the stress tensor $T^{\mu\nu}$ and $\mathcal{O}$

Holography



- Gravity in 4+1 dim :

$$S \sim \int d^{3+1}x\,\sqrt{-g}\,(R - \frac{1}{2}(\partial\phi)^2 - V(\phi))$$

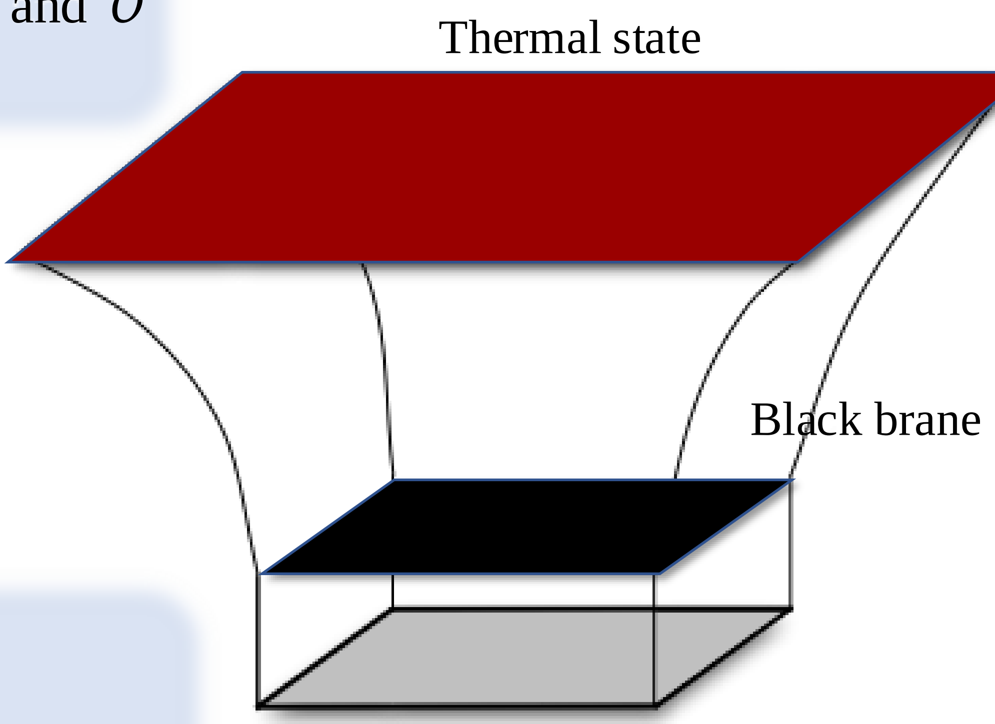
Holography: Our model

- QFT on Minkowski in 3+1 dim
- Decoupled sector of the stress tensor $T^{\mu\nu}$ and $\mathcal{O}$

Holography

- Gravity in 4+1 dim :

$$S \sim \int d^{3+1}x\sqrt{-g}\,(R - \frac{1}{2}(\partial\phi)^2 - V(\phi))$$



Bekenstein-Hawking entropy S

Hawking temperature T

Holography: Our model

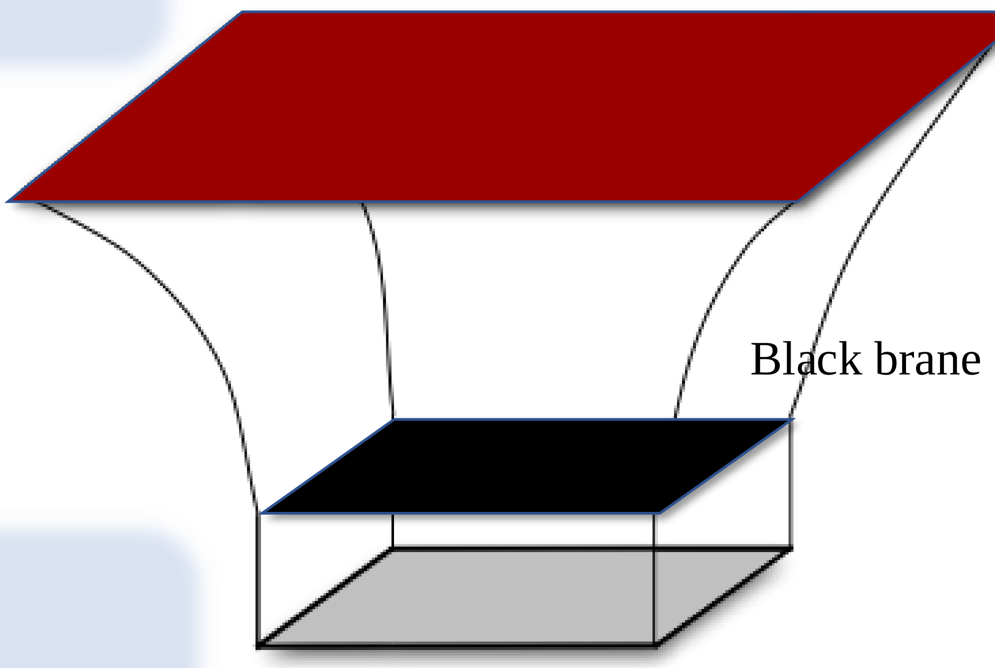
- QFT on Minkowski in 3+1 dim
- Decoupled sector of the stress tensor $T^{\mu\nu}$ and $\mathcal{O}$

Holography

- Gravity in 4+1 dim :

$$S \sim \int d^{3+1}x \sqrt{-g}\,(R - \frac{1}{2}(\partial\phi)^2 - V(\phi))$$

By constructing all black branes, we reconstruct the equation of state S(T)



Bekenstein-Hawking entropy S
Hawking temperature T

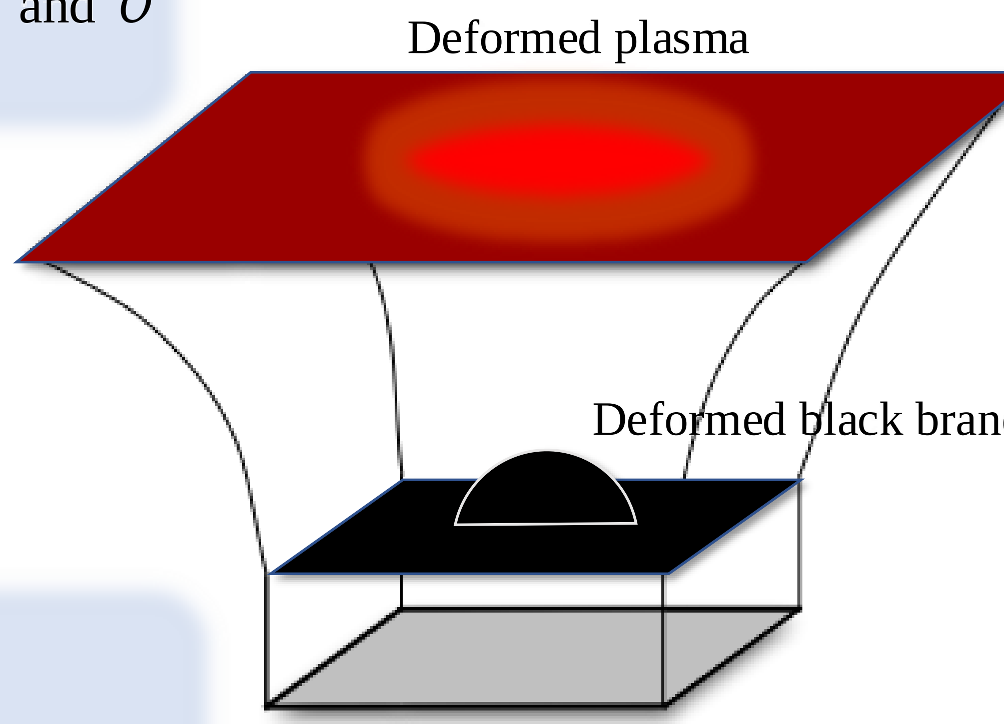
Holography: Our model

- QFT on Minkowski in 3+1 dim
- Decoupled sector of the stress tensor $T^{\mu\nu}$ and $\mathcal{O}$

Holography

- Gravity in 4+1 dim :

$$S \sim \int d^{3+1}x \sqrt{-g}\,(R - \frac{1}{2}(\partial\phi)^2 - V(\phi))$$



Holography: Our model

- QFT on Minkowski in 3+1 dim
- Decoupled sector of the stress tensor $T^{\mu\nu}$ and $\mathcal{O}$

Real-time quantum dynamics

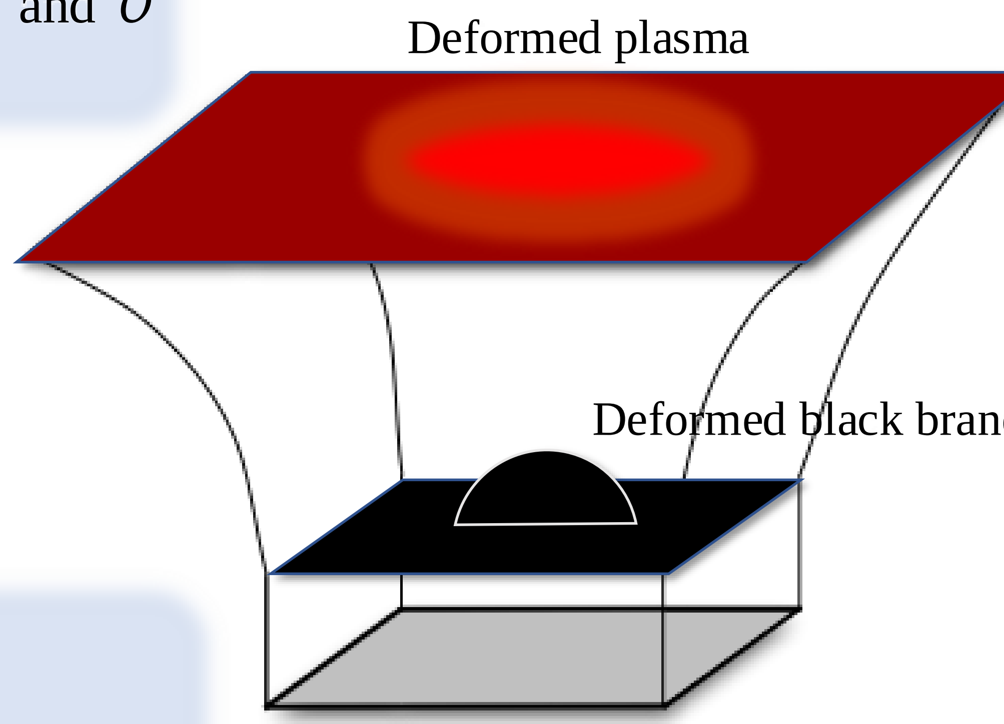
Holography

Numerical Relativity

Dynamical classical gravity

- Gravity in 4+1 dim :

$$S \sim \int d^{3+1}x\sqrt{-g}\,(R - \frac{1}{2}(\partial\phi)^2 - V(\phi))$$



Holography: Our model

- QFT on Minkowski in 3+1 dim
- Decoupl

Numerica

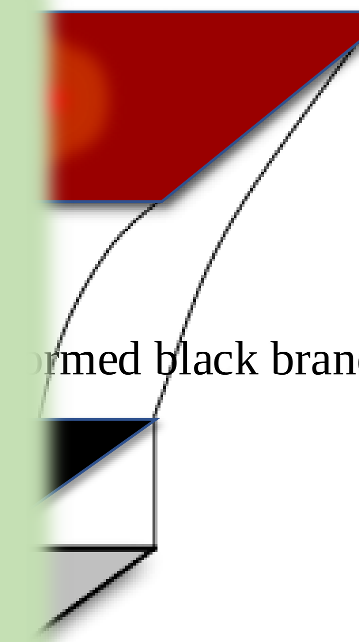
Relativity

- Gravity

S

- We have explored the real time dynamics in different context by using holography
- Applications in QGP, phase transitions, cosmology

sma



Holography: Our model

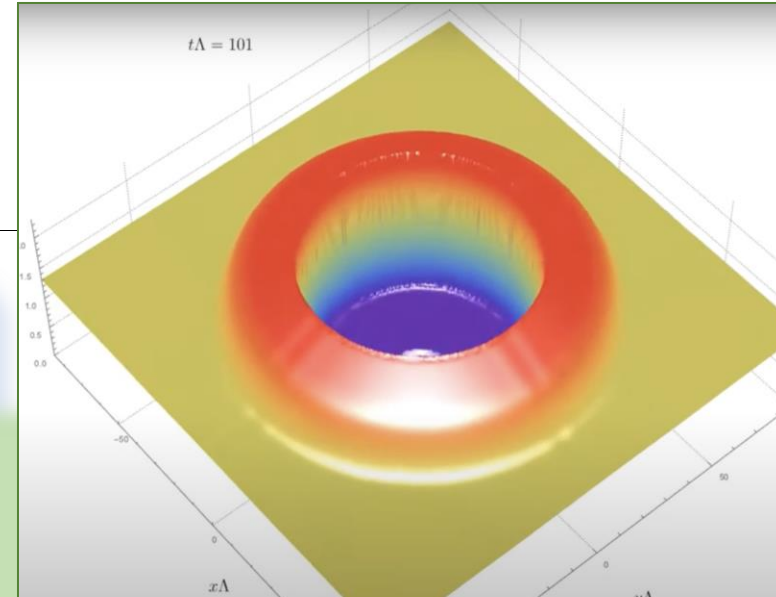
- QFT on Minkowski in 3+1 dim
- Decoupl

Numerical

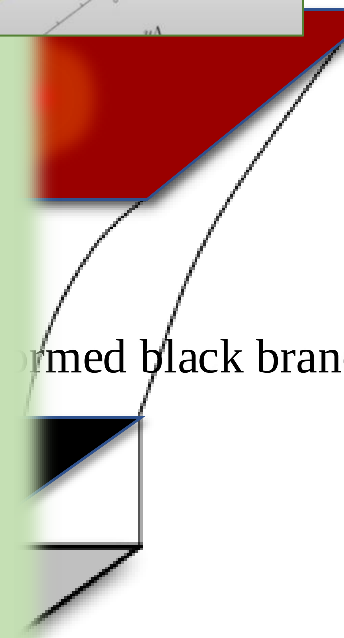
Relativity

- Gravity

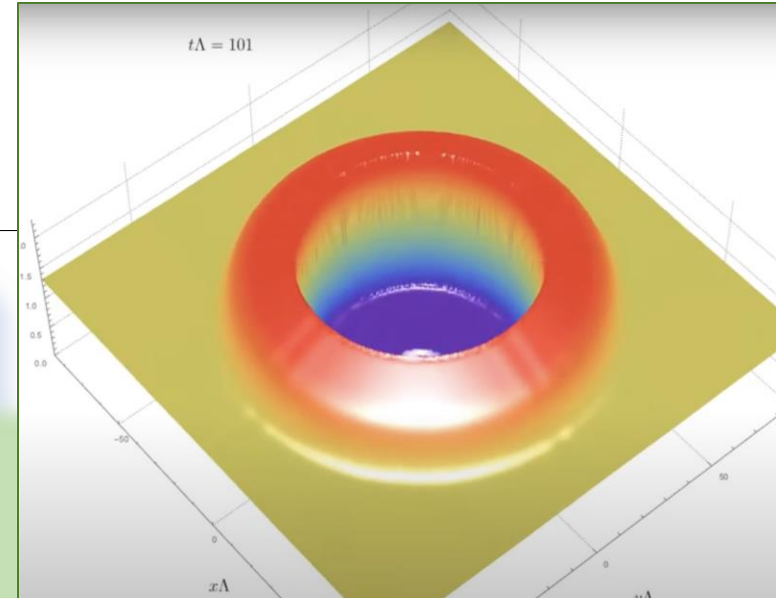
S



- We have explored the real time dynamics in different context by using holography
- Applications in QGP, phase transitions, cosmology



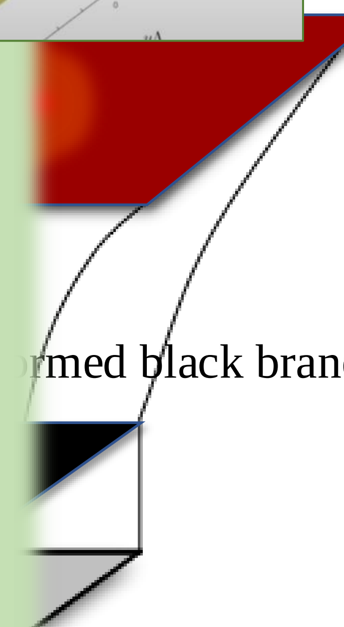
Holography: Our model



- QFT on Minkowski in 3+1 dim
- Decoupl

- We have explored the real time dynamics in different context by using holography
- Applications in QGP, phase transitions, cosmology

- We would like to perform evolutions in a holographic theory with a given eq. of state



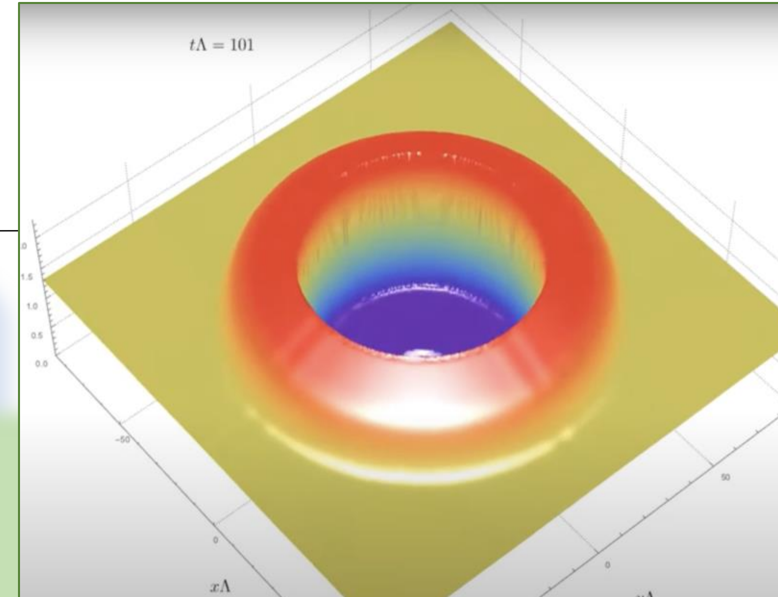
Numerica

Relativity

- Gravity

S

Holography: Our model



- QFT on Minkowski in 3+1 dim
- Decoupl

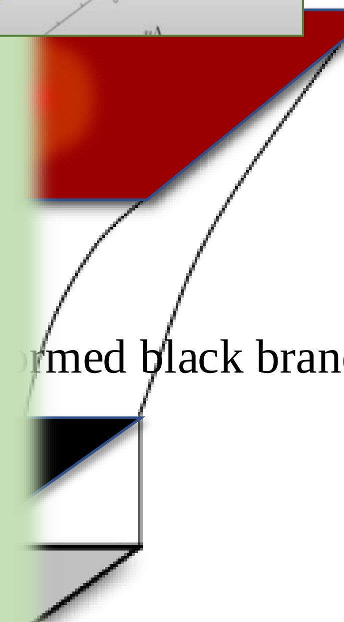
Numerica

Relativity

- Gravity

S

- We have explored the real time dynamics in different context by using holography
- Applications in QGP, phase transitions, cosmology



- We would like to perform evolutions in a holographic theory with a given eq. of state
- This is why we want to find the potential that gives rise to a given eq. of state $\longrightarrow$ inverse problem

Holography: A simple model

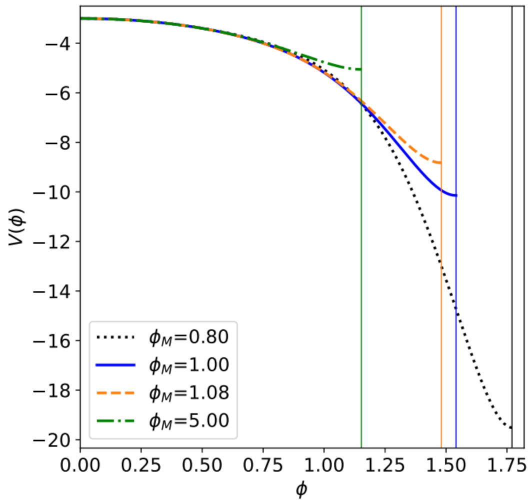
$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$

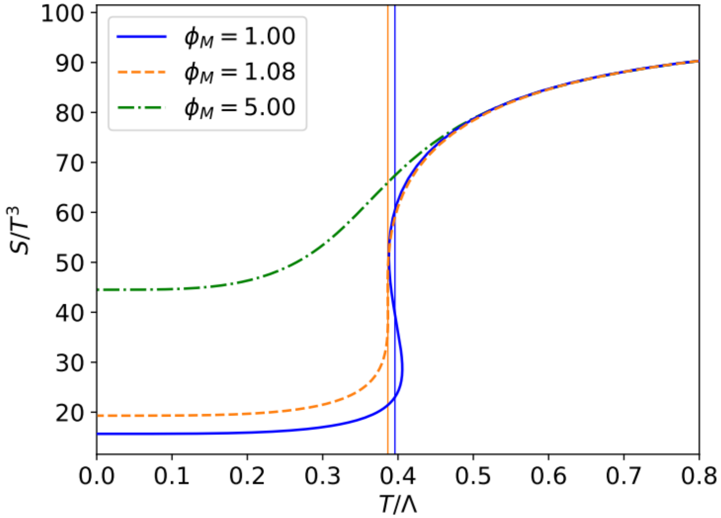
$$V_{\text{theory}}(\phi) = -\frac{4}{3} \mathcal{W}(\phi)^2 + \frac{1}{2} \mathcal{W}'(\phi)^2 \,,$$

$$\mathcal{W}(\phi) = -\frac{3}{2} - \frac{\phi^2}{2} - \frac{\phi^4}{4\phi_M^2} + \frac{\phi^6}{10} \,.$$

Bea, Mateos ‘18

- Simple, smooth at IR, captures first order/2nd order/crossover
- RG flow from CFT at the UV to a CFT at the IR





Holography: A simple model

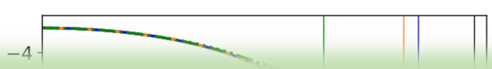
$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$

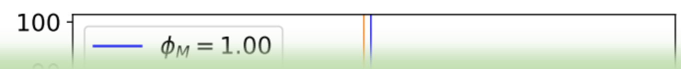
$$V_{\text{theory}}(\phi) = -\frac{4}{3} \mathcal{W}(\phi)^2 + \frac{1}{2} \mathcal{W}'(\phi)^2 \,,$$

$$\mathcal{W}(\phi) = -\frac{3}{2} - \frac{\phi^2}{2} - \frac{\phi^4}{4\phi_M^2} + \frac{\phi^6}{10} \,.$$

Bea, Mateos ‘18

- Simple, smooth at IR, captures first order/2nd order/crossover
- RG flow from CFT at the UV to a CFT at the IR





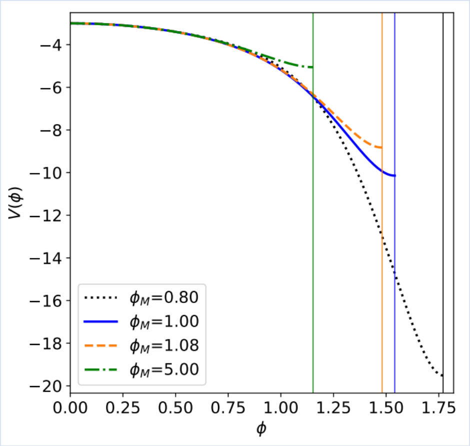
- This model has been used in several contexts in holography:
  ⟶ cosmology, dynamics of phase transitions...

Casalderrey-Solana, Mateos, Serantes ‘23
Ares, Henriksson, Hindmarsh, Hoyos, Jokela ‘21
Bea, Casalderrey-Solana, Gianakopoulos, Mateos, Sanchez-Garitaonandia, Zilhao ‘21
.....

The inverse problem

Einstein eqs.

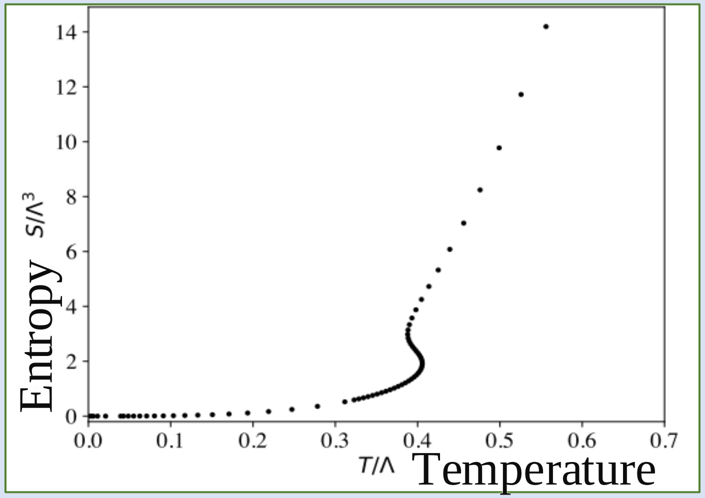
$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$



Direct problem

Inverse problem

Equation of State



→ We address the inverse problem by using **Neural Networks**

Neural Networks

Neural Networks

I will present neural networks from the perspective of our problem

Neural Networks

I will present neural networks from the perspective of our problem

We use **P**hysics **I**nformed **N**eural **N**etwoks, **PINNs**

Neural Networks

I will present neural networks from the perspective of our problem

We use **P**hysics **I**nformed **N**eural **N**etwoks, **PINNs**

→ The Neural Network is informed by the physical equations

→ In our case, Einstein's equations

Neural Networks

I will present neural networks from the perspective of our problem

We use **P**hysics **I**nformed **N**eural **N**etwoks, **PINNs**

→ The Neural Network is informed by the physical equations

→ In our case, Einstein's equations

Other uses: training on known solutions ("supervised training"; "recognizing faces")
....but not in our work

Neural Networks

For our purposes, we can understand a NN as a **function**:

Neural Networks

For our purposes, we can understand a NN as a **function**:

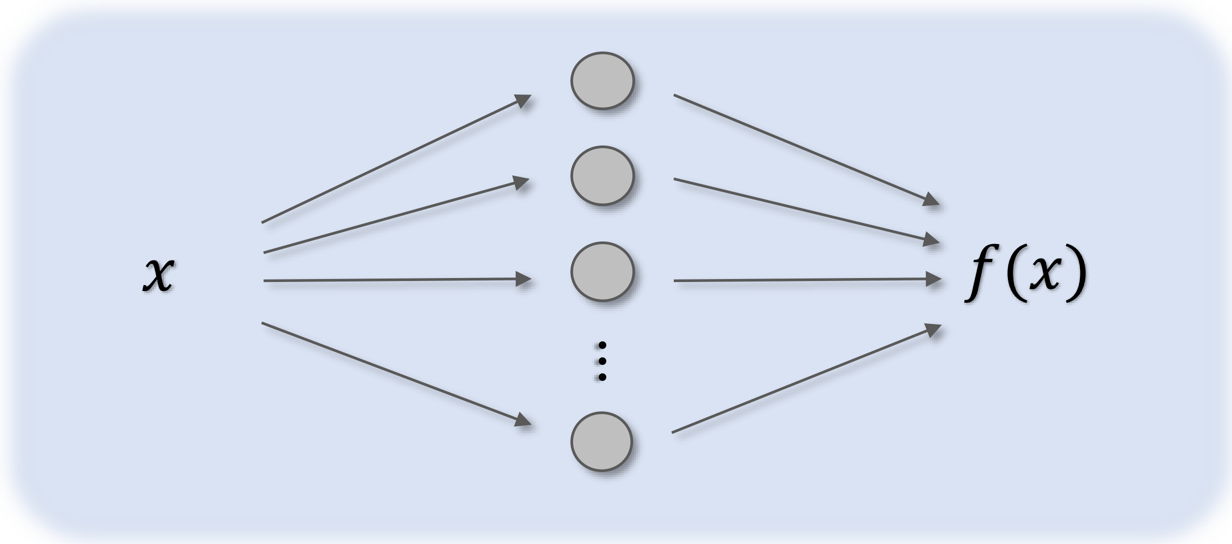
→ Cos(x)

→ Exp(x)

→ A solution to our equations

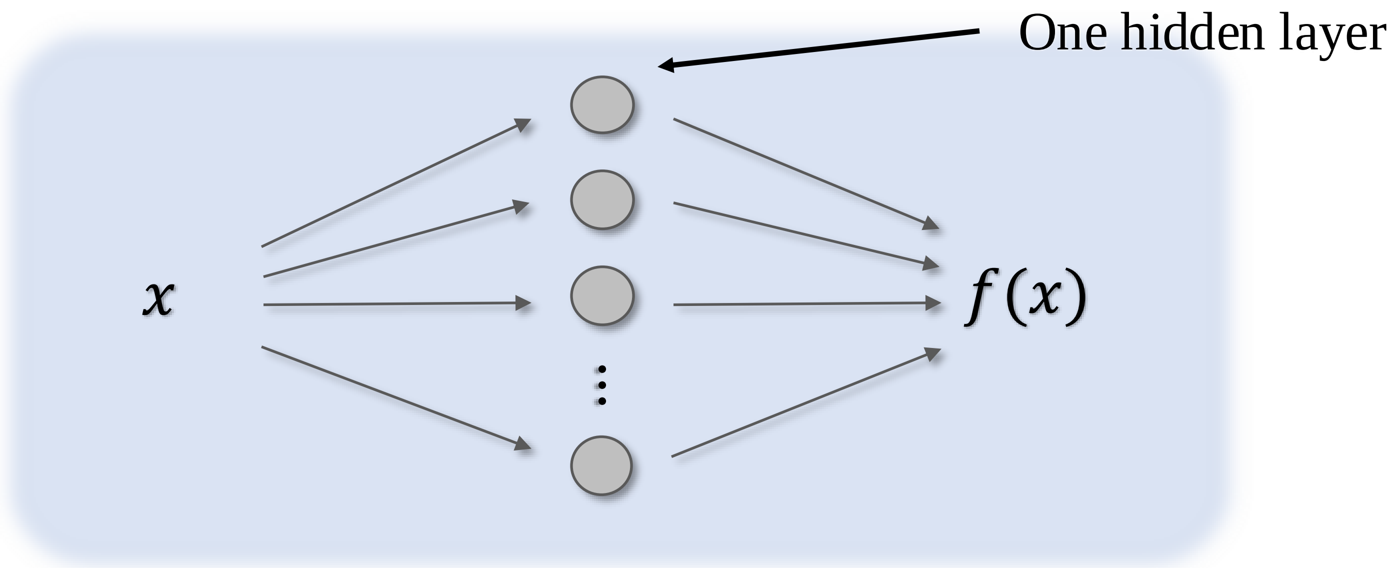
Neural Networks

For our purposes, we can understand a NN as a **function**:



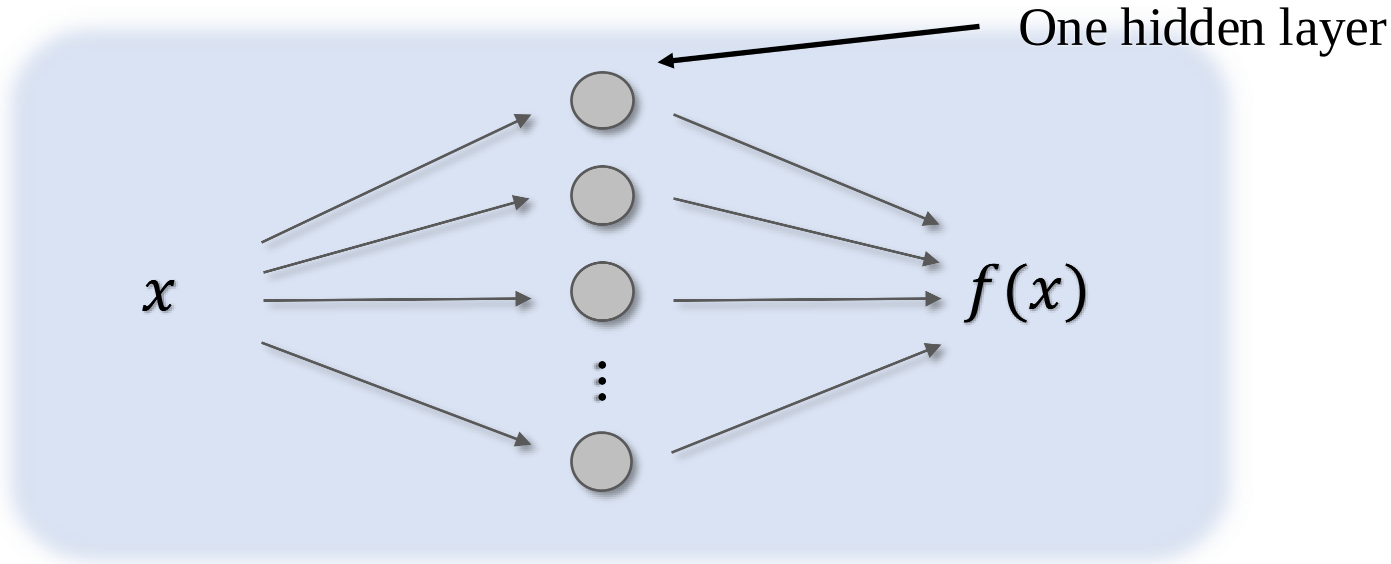
Neural Networks

For our purposes, we can understand a NN as a **function**:



Neural Networks

For our purposes, we can understand a NN as a **function**:

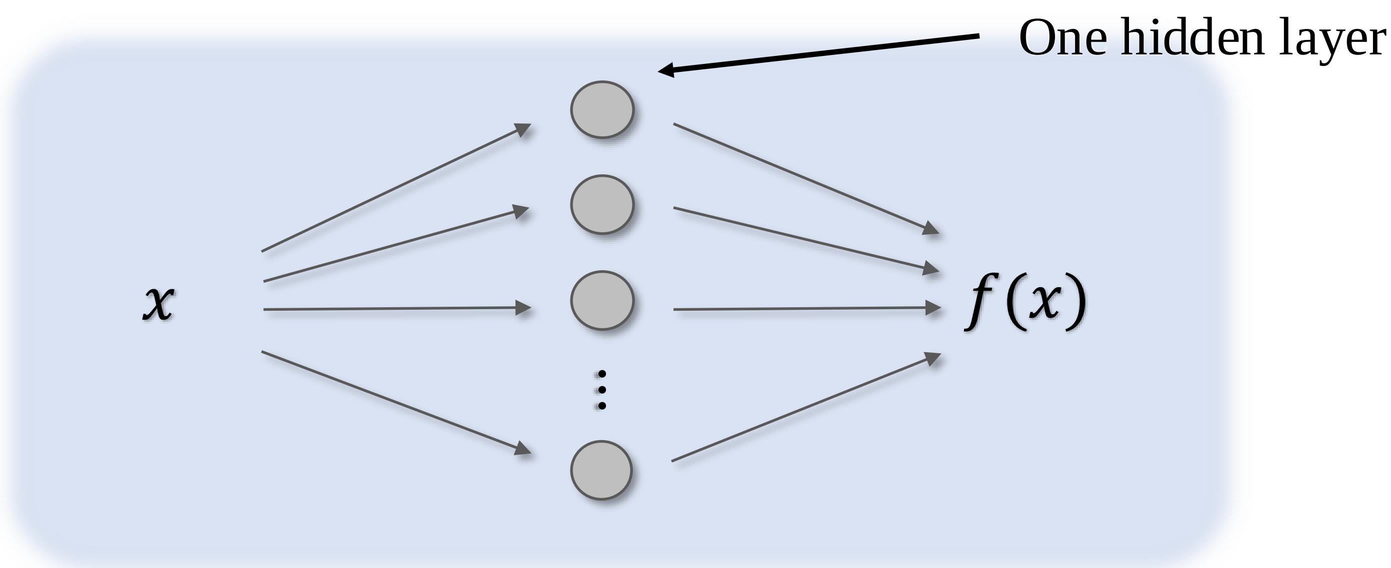


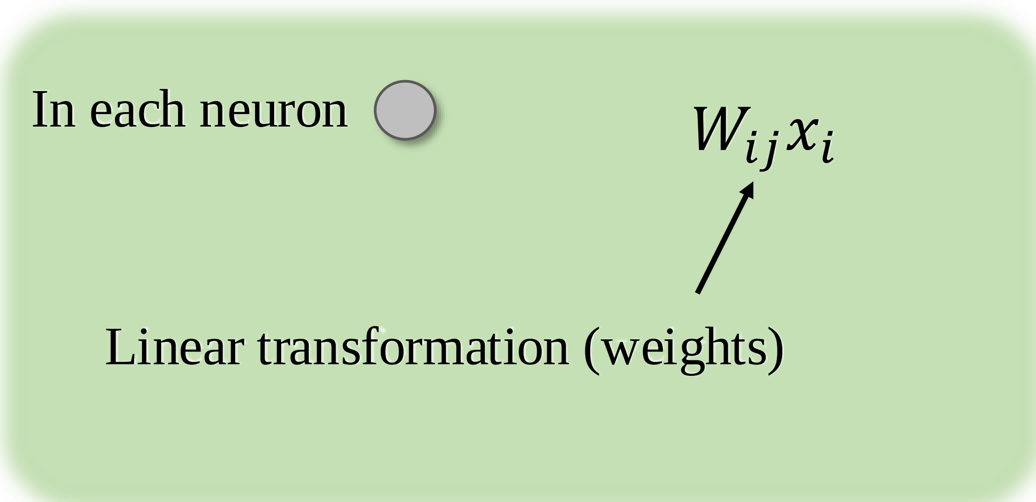
In each neuron

x_i

Neural Networks

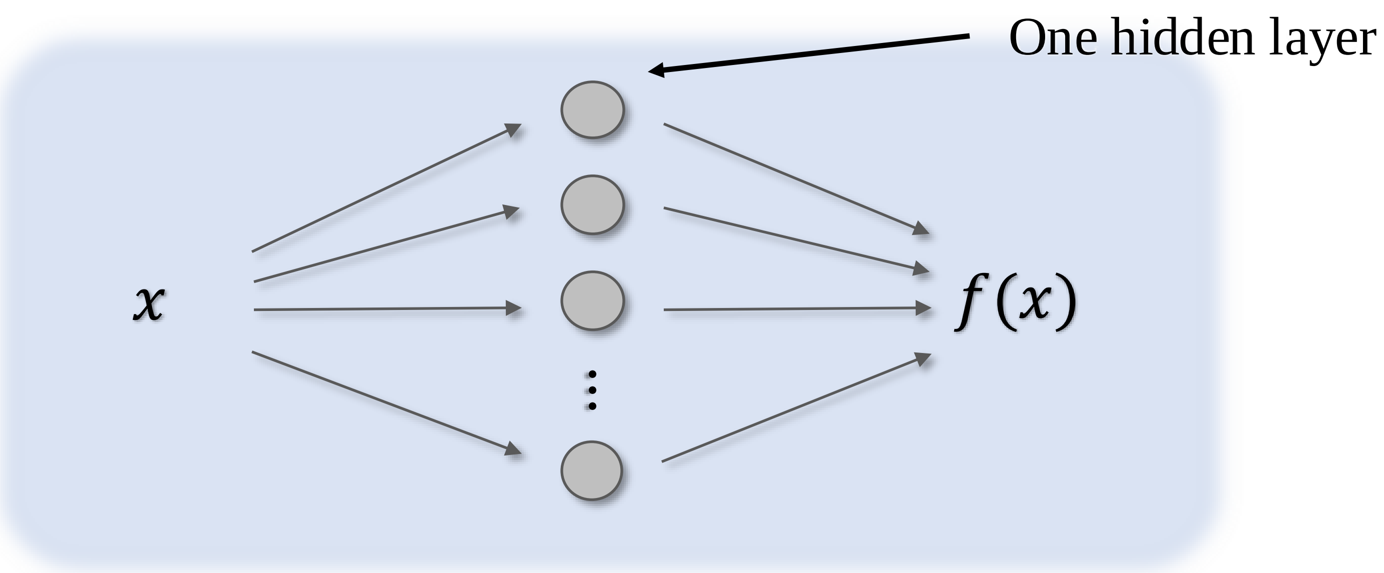
For our purposes, we can understand a NN as a **function**:

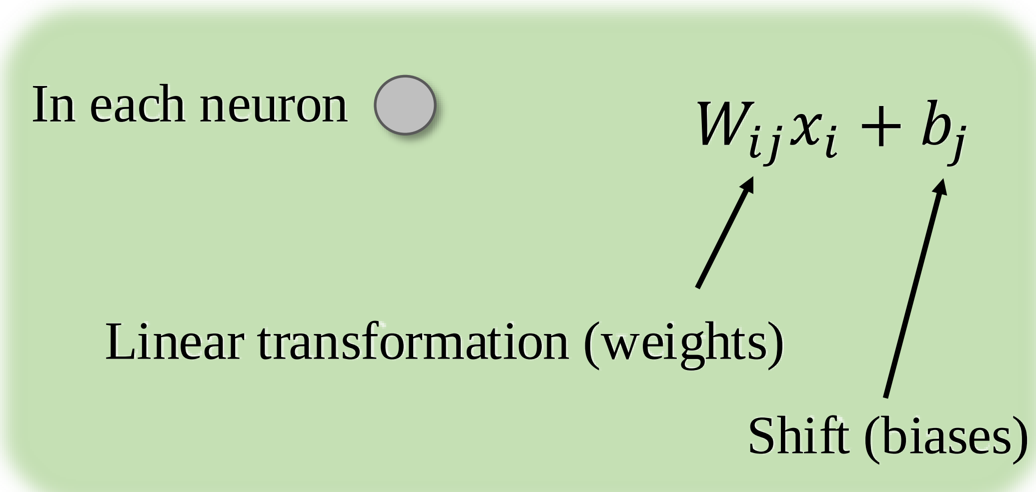




Neural Networks

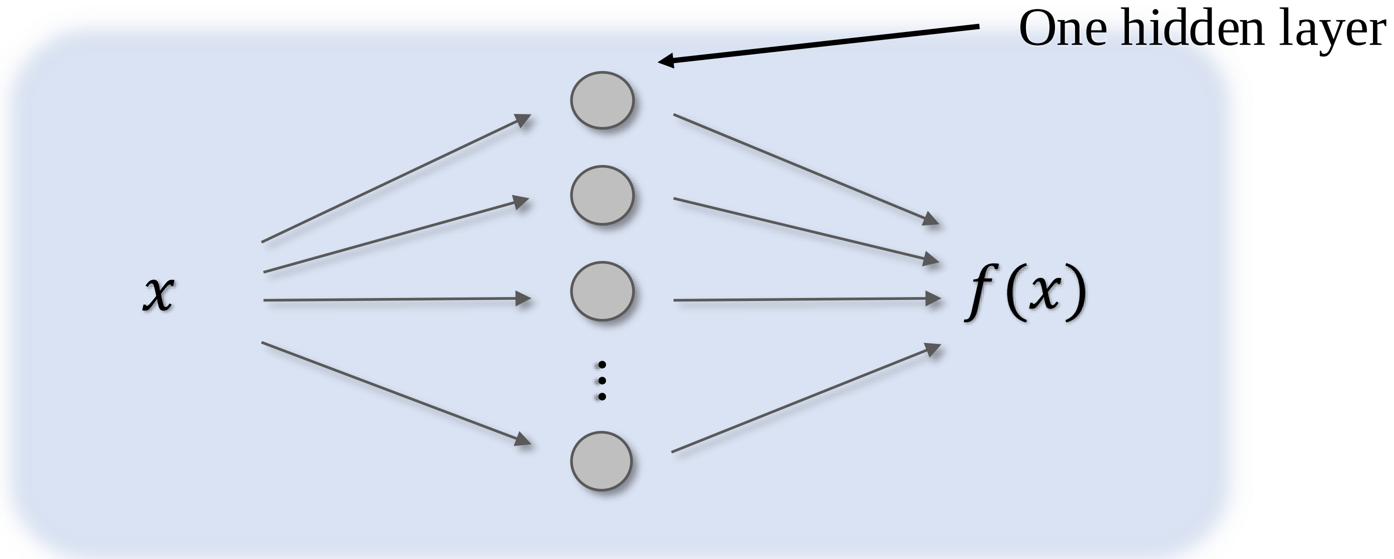
For our purposes, we can understand a NN as a **function**:





Neural Networks

For our purposes, we can understand a NN as a **function**:



In each neuron

$$Tanh(W_{ij}x_i + b_j)$$

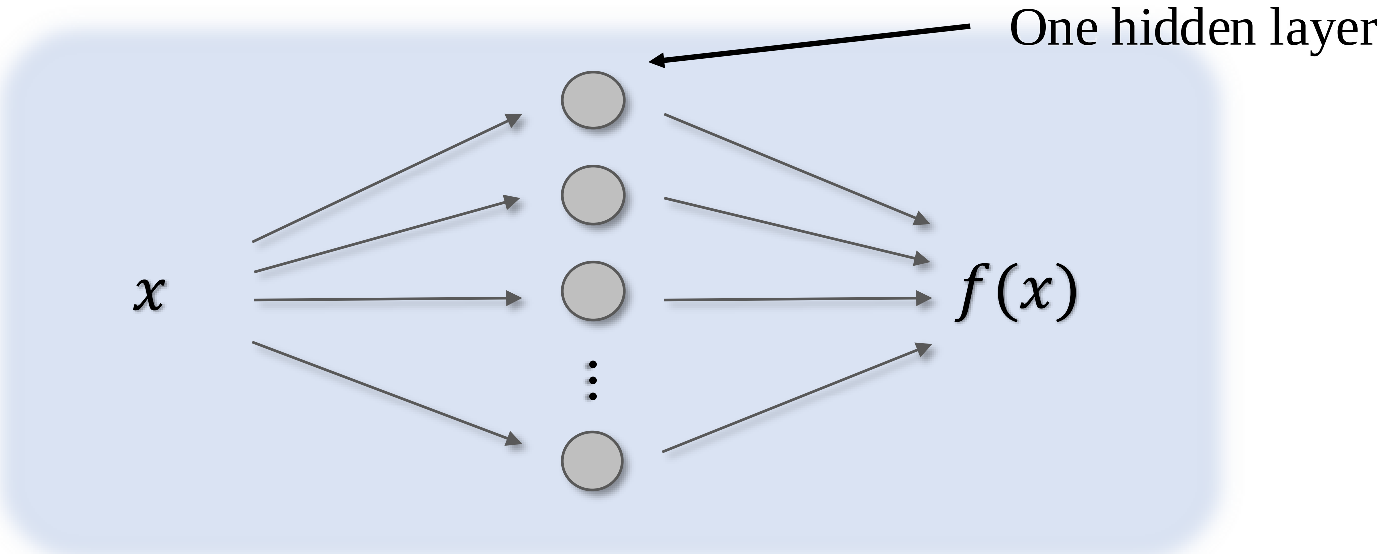
Activation function

Linear transformation (weights)

Shift (biases)

Neural Networks

For our purposes, we can understand a NN as a **function**:



In each neuron

$$Tanh(W_{ij}x_i + b_j)$$

Activation function

Linear transformation (weights)

Shift (biases)

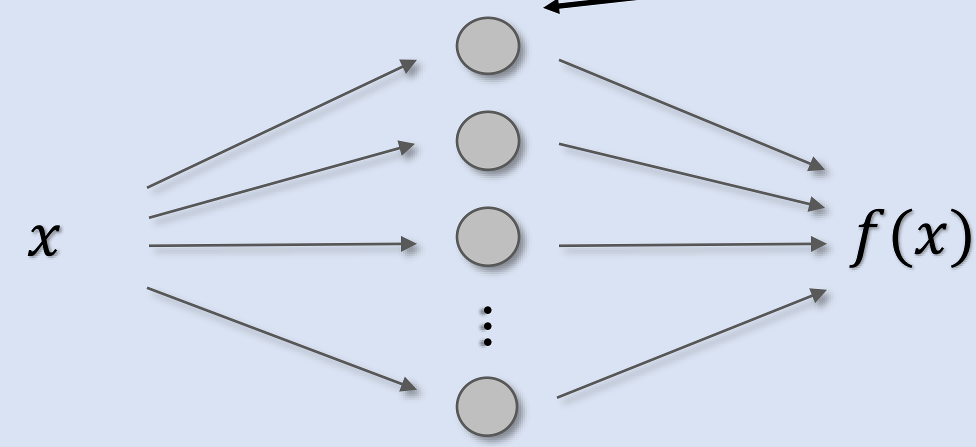
Internally is **analytical**

Just a combination of tanh's...

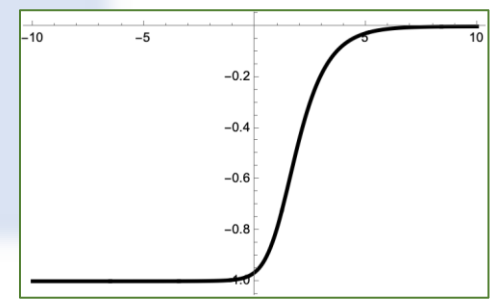
Neural Networks

For our purposes, we can understand a NN as a **function**:

One hidden layer



Tanh[2Tanh[0.5x]-2]



In each neuron

$$Tanh(W_{ij}x_i + b_j)$$

Activation function

Linear transformation (weights)

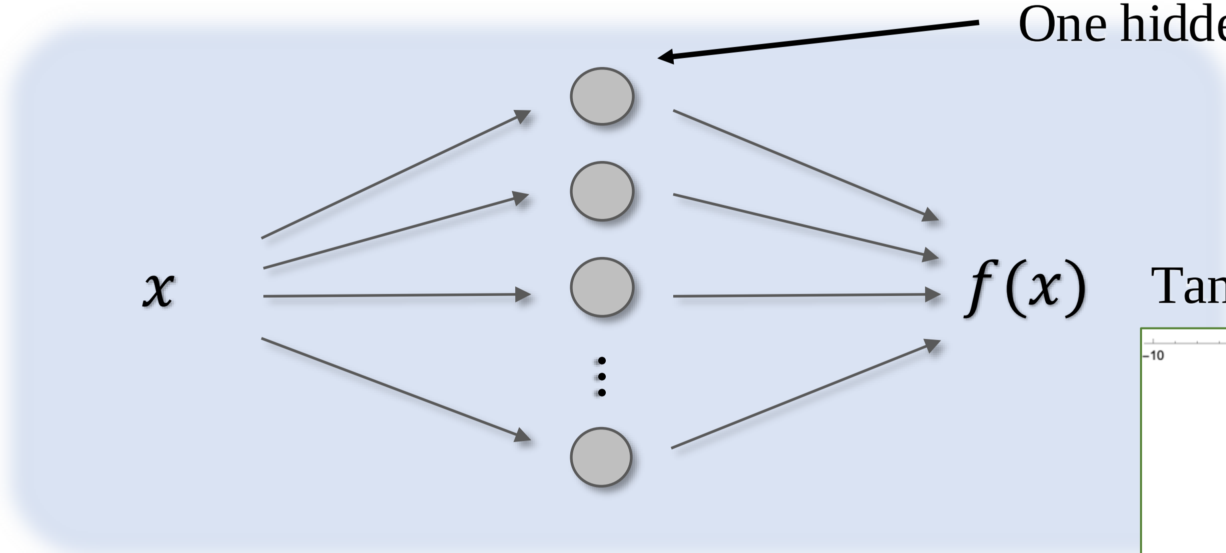
Shift (biases)

Internally is **analytical**

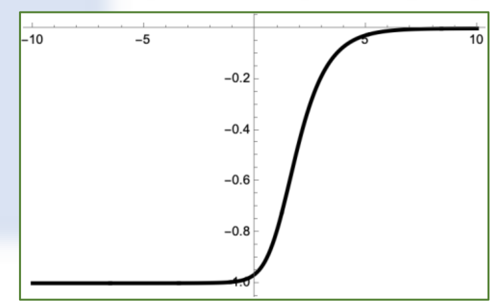
Just a combination of tanh's...

Neural Networks

For our purposes, we can understand a NN as a **function**:



Tanh[2Tanh[0.5x]-2]



In each neuron $Tanh(W_{ij}x_i + b_j)$

Activation function

Linear transformation (weights)

Shift (biases)

Internally is **analytical**

Just a combination of tanh's...

Finite but large number of parameters
(in our case 15.000)

Neural Networks

For our purposes, we can understand a NN as a **function**:

One hidden layer

But.......why is this useful?

In each n

Activati

nalytical

on of tanh's...

Linear transformation (weights)

Shift (biases)

Finite but large number of parameters
(in our case 15.000)

Neural Networks

For our purposes, we can understand a NN as a **function**:

One hidden layer

But.......why is this useful?

Two main pilars of NN:

In each n

Activati

Linear transformation (weights)

Shift (biases)

nalytical

on of tanh's...

Finite but large number of parameters
(in our case 15.000)

Neural Networks

For our purposes, we can understand a NN as a **function**:

One hidden layer

But.......why is this useful?

Two main pilars of NN:

- **a NN can approximate any continuous function**

In each n

Activati

Linear transformation (weights)

Shift (biases)

nalytical

on of tanh's...

Finite but large number of parameters
(in our case 15.000)

Neural Networks

For our purposes, we can understand a NN as a **function**:

One hidden layer

But.......why is this useful?

Two main pilars of NN:

- **a NN can approximate any continuous function**
- **NN can learn**

In each n

nalytical

on of tanh's...

Activati

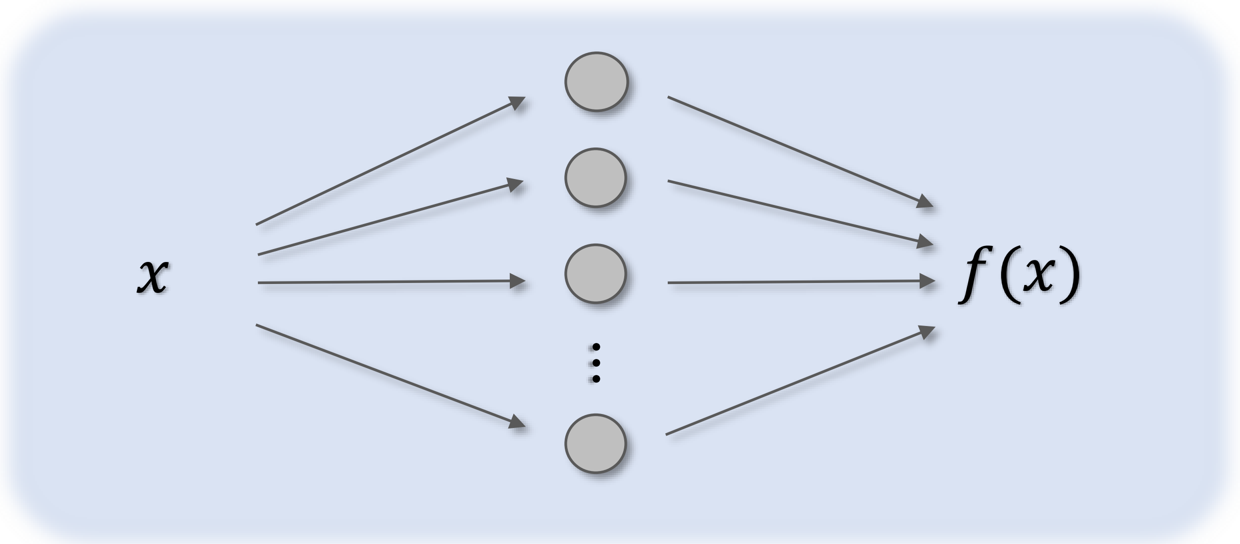
Linear transformation (weights)

Shift (biases)

Finite but large number of parameters
(in our case 15.000)

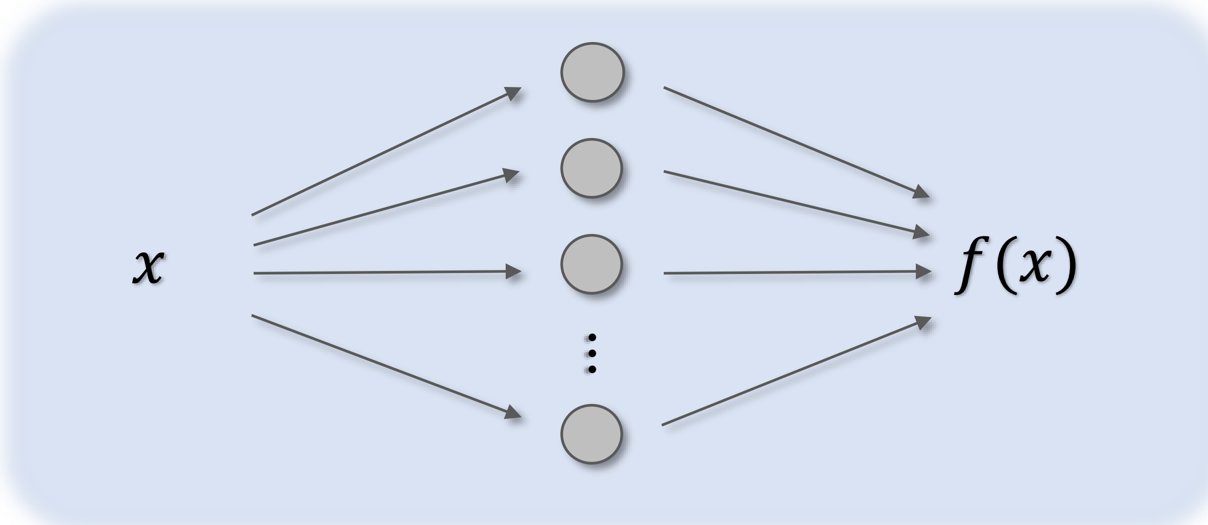
Universal approximation theorems

NN are supported on mathematical theorems



Universal approximation theorems

NN are supported on mathematical theorems

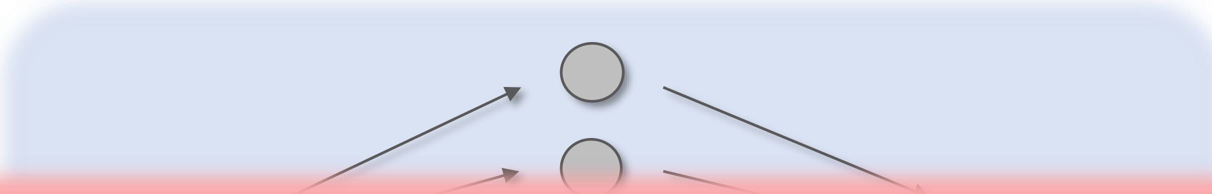


Universal approximation theorem

"With enough neurons, we can recover a given function to a certain accuracy"

Universal approximation theorems

NN are supported on mathematical theorems

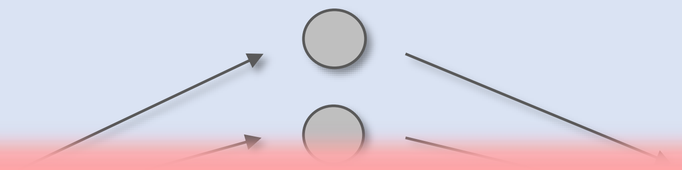


This means, that, in particular, that a NN can represent a solution to our equations

“With enough neurons, we can recover a given function to a certain accuracy”

Universal approximation theorems

NN are supported on mathematical theorems



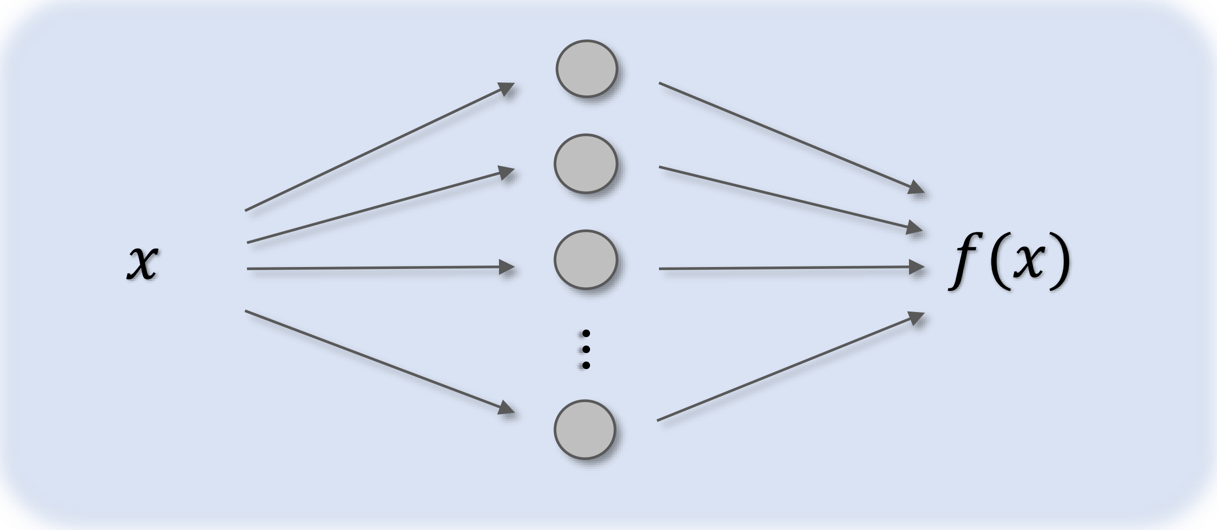
This means, that, in particular, that a NN can represent a solution to our equations

But.... how do we get to that NN?

“With enough neurons, we can recover a given function to a certain accuracy”

Definition of Loss

We now want this NN to be a solution to our equations



Loss function   L := (residual of Einstein's equations)^2

We want to minimize the loss function

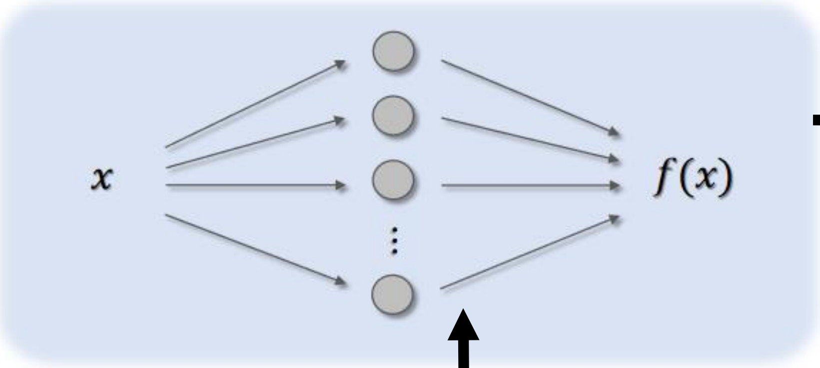
This is our definition of PINNs

Neural Networks: learning process

But..... how does it learn?

Neural Networks: learning process

But..... how does it learn?



Loss=(Eqs residuals)^2

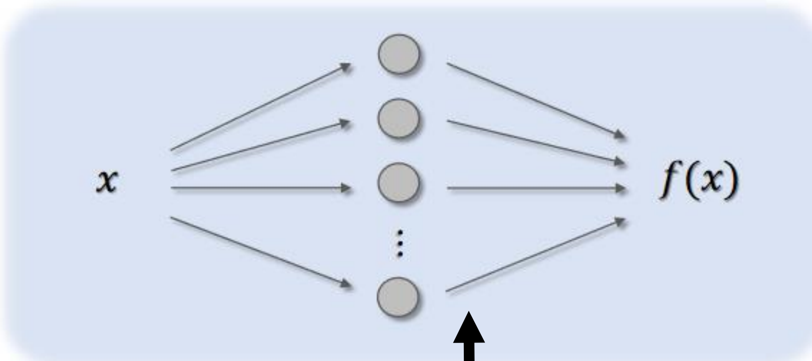
Analytical expression with params W, b

“$Tanh(Wx + b)$”

Gradient $\nabla_{W,b} L$

Neural Networks: learning process

But….. how does it learn?



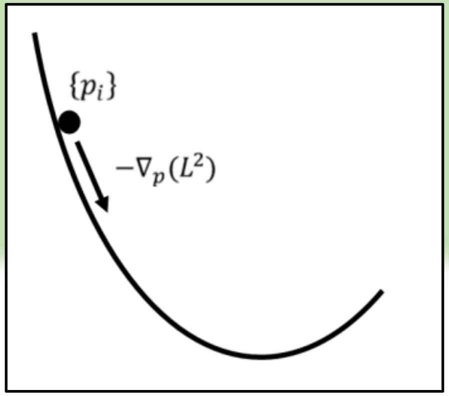
Loss=(Eqs residuals)^2

Analytical expression with params W, b

"$Tanh(Wx + b)$"

Gradient $\nabla_{W,b} L$

Gradient descent



Our set up

Einstein-Klein Gordon

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$

Our set up

Einstein-Klein Gordon

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$

Metric ansatz

$$ds^2 = -A dt^2 + \Sigma^2 (dx^2 + dy^2 + dz^2) - \frac{2}{u^2} dt du$$

$A(u), \Sigma(u), \phi(u) \longrightarrow$ Unknown functions

Our set up

Einstein-Klein Gordon

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$

Metric ansatz

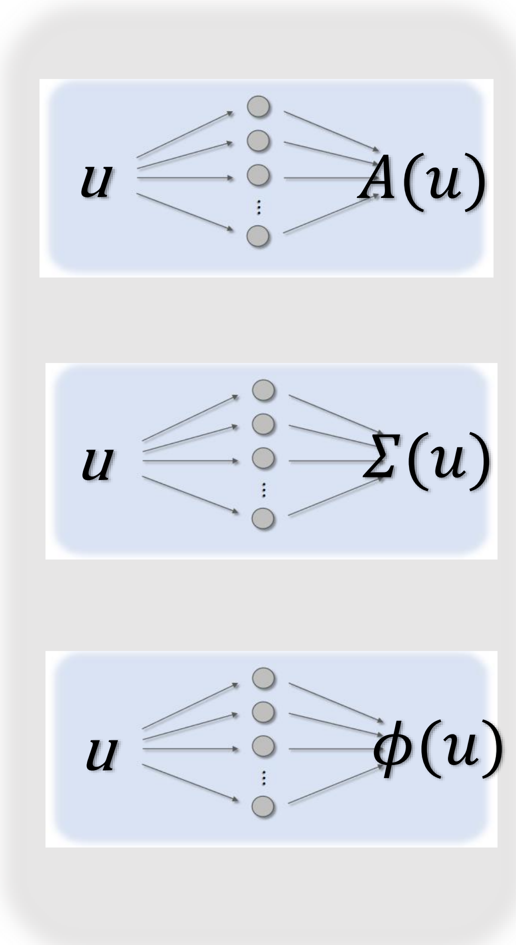
$$ds^2 = -A dt^2 + \Sigma^2 (dx^2 + dy^2 + dz^2) - \frac{2}{u^2} dt du$$

$A(u), \Sigma(u), \phi(u) \longrightarrow$ Unknown functions

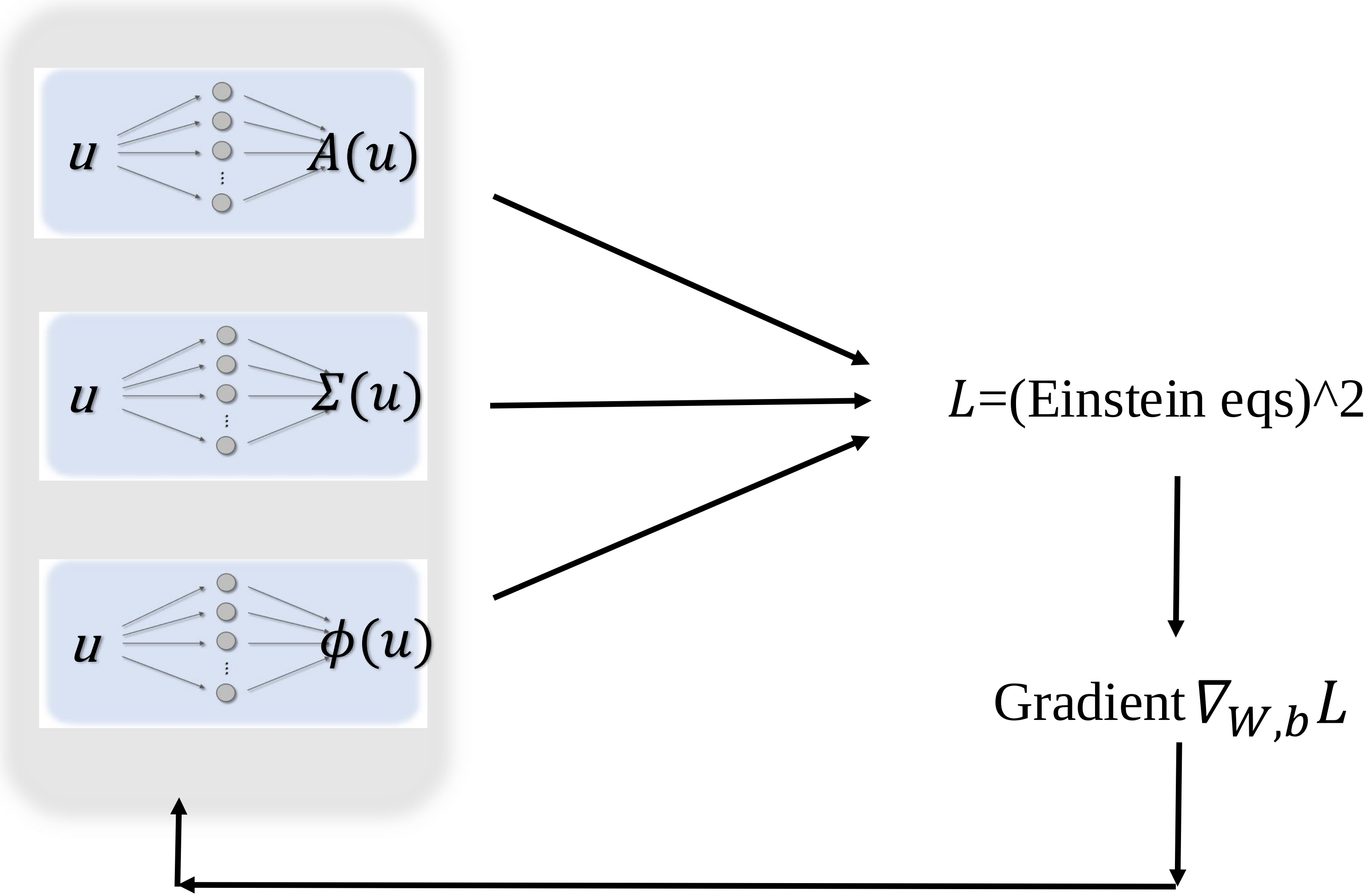
$u \longrightarrow$ Holographic variable

$u \in [0,1]$

The direct problem



The direct problem



The direct problem

we solve the direct problem as a **test**

u

u s)^2

u $_b L$

The direct problem

we solve the direct problem as a **test**

-We fix the potential V(φ)

u

u s)^2

u $_b L$

The direct problem

we solve the direct problem as a **test**

-We fix the potential V(φ)

- Run for ~million iterations

u

u s)^2

u $_b L$

The direct problem

u

u … s)^2

u … $_b L$

we solve the direct problem as a **test**

-We fix the potential V(φ)

- Run for ~million iterations

- Compare with traditional methods

The direct problem

u

u

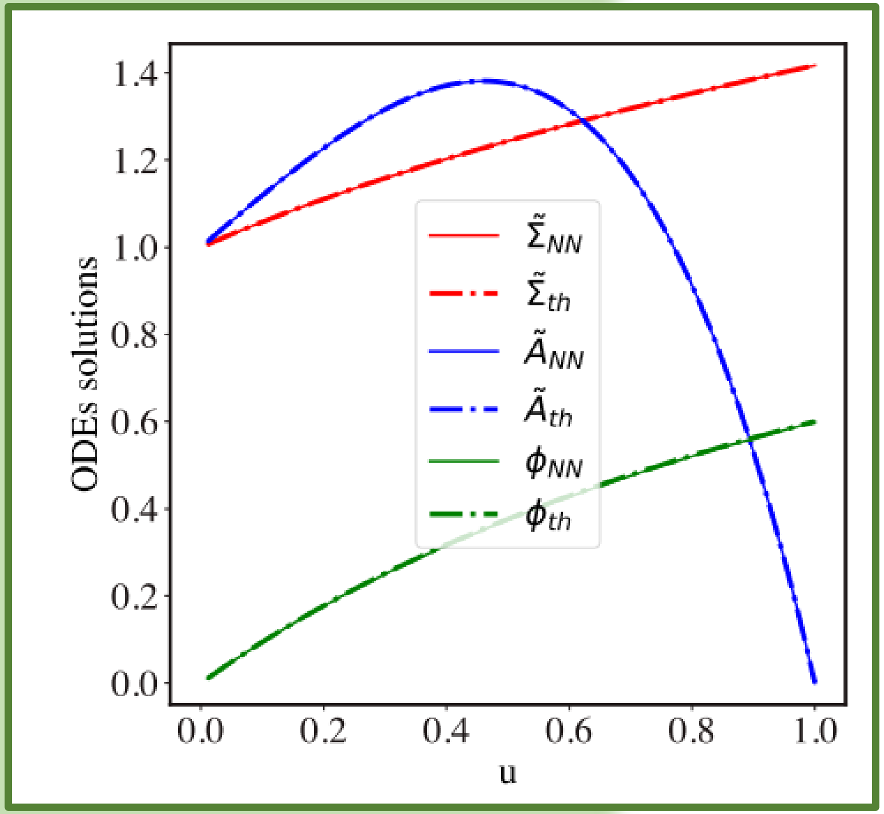
u

we solve the direct problem as a **test**

-We fix the potential V(φ)

- Run for ~million iterations

- Compare with traditional methods



The direct problem

we solve the direct problem as a **test**

u

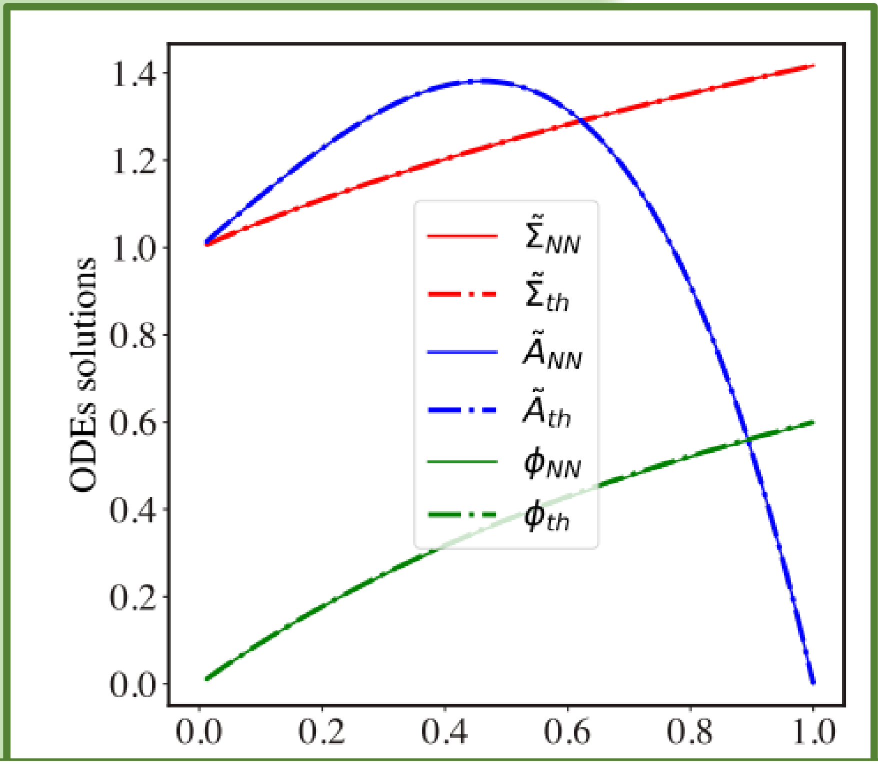
u

u

-We fix the potential V(φ)

- Run for ~million iterations

- Compare with traditional methods



Direct problem works very well!

The direct problem

we solve the direct problem as a **test**

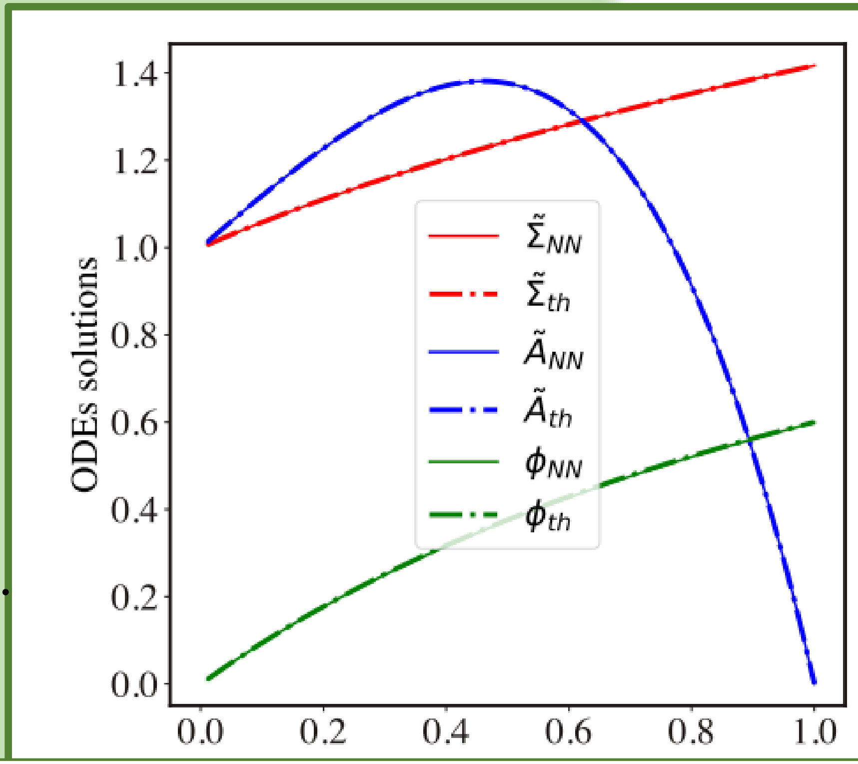
-We fix the potential V(φ)

- Run for ~million iterations

- Compare with traditional methods

-This corresponds to one black brane solution, i.e., one (T,S) point.

-We have control on the solution: more iteration, reduce the loss



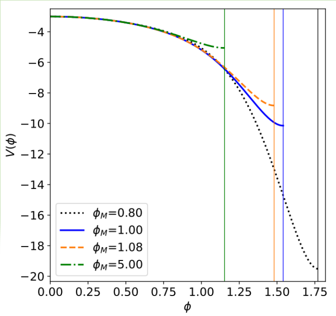
Direct problem works very well!

The inverse problem

We are given the potential V(φ)
We construct the thermodynamics S(T)

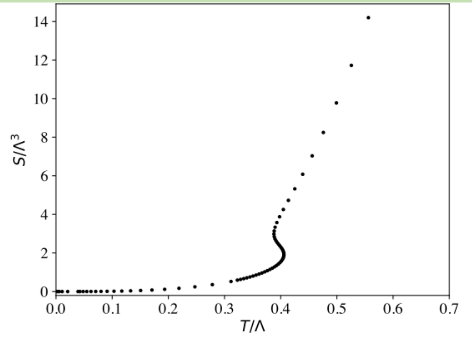
Einstein eqs.

$$S = \frac{2}{\kappa_5^2}\int dx^5 \sqrt{-g}\left[\frac{1}{4}R - \frac{1}{2}(\nabla\phi)^2 - V(\phi)\right]$$



Direct problem

Equation of state

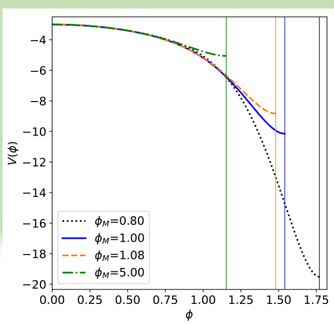


The inverse problem

We are given the potential V(φ)
We construct the thermodynamics S(T)

Einstein eqs.

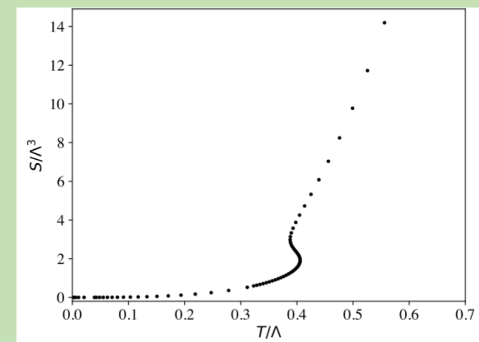
$$S = \frac{2}{\kappa_5^2}\int dx^5\sqrt{-g}\left[\frac{1}{4}R - \frac{1}{2}(\nabla\phi)^2 - V(\phi)\right]$$



Direct problem

Inverse problem

Equation of state



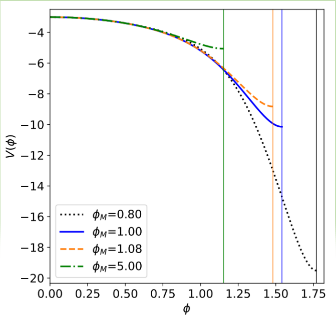
We are given an equation of state S(T)
and we want to reconstruct V(φ)

→ This corresponds to reconstructing the theory

The inverse problem

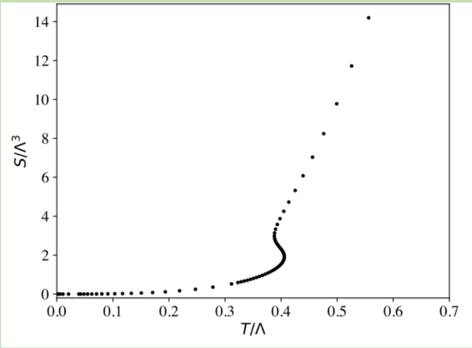
Einstein eqs.

$$S = \frac{2}{\kappa_5^2}\int dx^5\sqrt{-g}\left[\frac{1}{4}R - \frac{1}{2}(\nabla\phi)^2 - V(\phi)\right]$$



Inverse problem

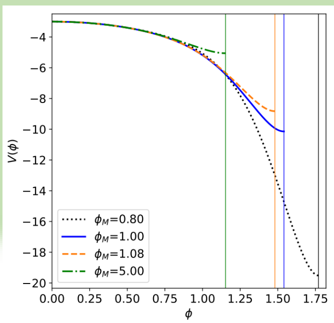
Equation of state



The inverse problem

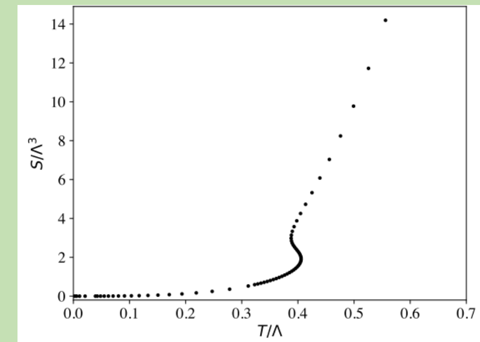
Einstein eqs.

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$



Inverse problem

Equation of state

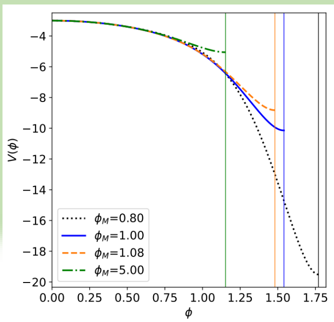


-Sampling of the eq. of state S(T), ~70 points

The inverse problem

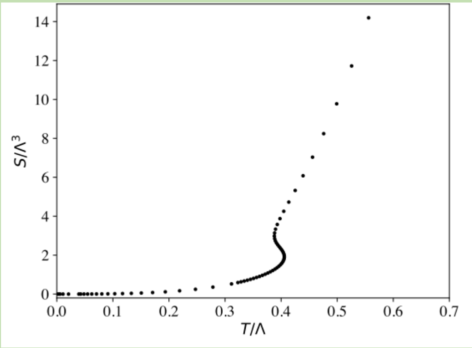
Einstein eqs.

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla\phi)^2 - V(\phi) \right]$$



Inverse problem

Equation of state



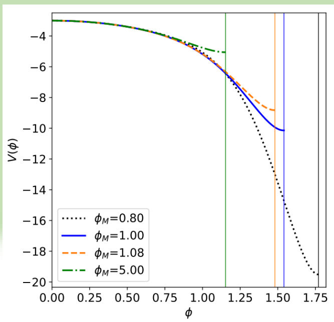
-Sampling of the eq. of state S(T), ~70 points

-Each point in the sampling corresponds to a black brane

The inverse problem

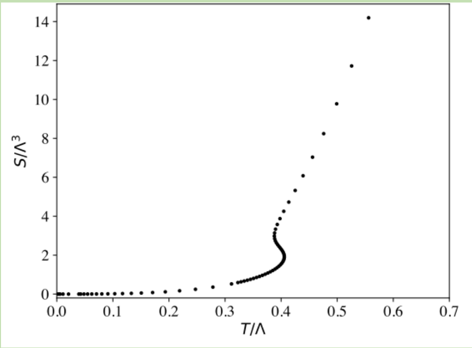
Einstein eqs.

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$



Inverse problem

Equation of state



-Sampling of the eq. of state S(T), ~70 points

-Each point in the sampling corresponds to a black brane

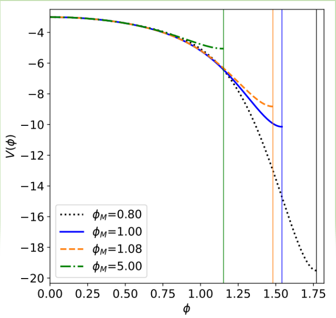
-All these black branes are solutions of the Einstein eqs with the same potential

→ boundle solution

The inverse problem

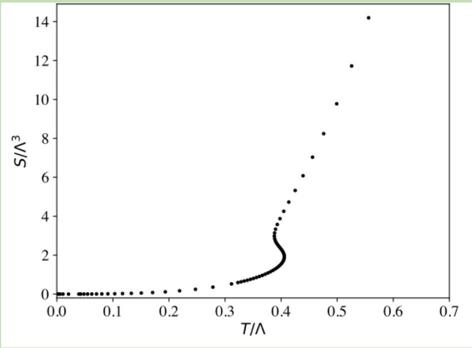
Einstein eqs.

$$S = \frac{2}{\kappa_5^2}\int dx^5 \sqrt{-g}\left[\frac{1}{4}R - \frac{1}{2}(\nabla\phi)^2 - V(\phi)\right]$$



Inverse problem

Equation of state



-Sampling of the eq. of state S(T), ~70 points

-Each point in the sampling corresponds to a black brane

-All these black branes are solutions of the Einstein eqs with the same potential

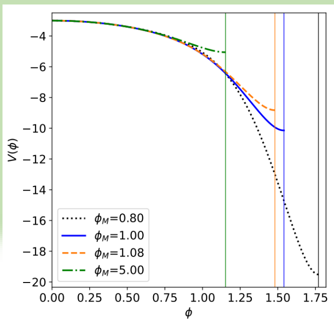
→ boundle solution

-We introduce an additional NN for V(φ) (now it is an unknown)

The inverse problem

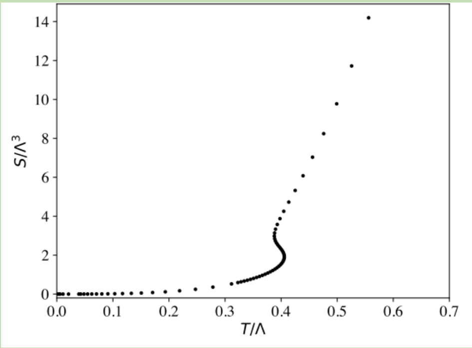
Einstein eqs.

$$S = \frac{2}{\kappa_5^2}\int dx^5 \sqrt{-g}\left[\frac{1}{4}R - \frac{1}{2}(\nabla\phi)^2 - V(\phi)\right]$$



Inverse problem

Equation of state

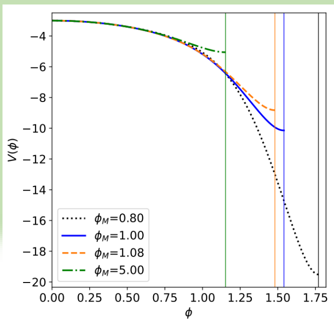


-Loss= Σ(residuals Einstein eqs)^2 (sum over branes)

The inverse problem

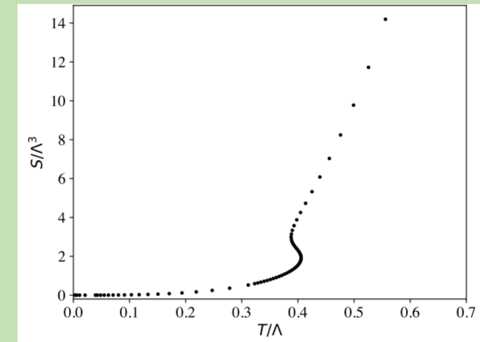
Einstein eqs.

$$S = \frac{2}{\kappa_5^2}\int dx^5 \sqrt{-g}\left[\frac{1}{4}R - \frac{1}{2}(\nabla\phi)^2 - V(\phi)\right]$$



Inverse problem

Equation of state



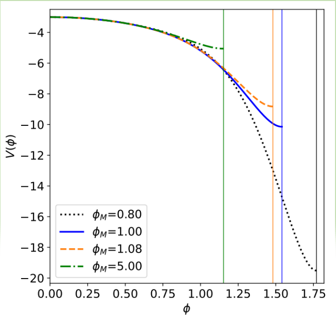
-Loss= Σ(residuals Einstein eqs)^2 (sum over branes)

-For each black brane, S and T enter as boundary conditions at the horizon

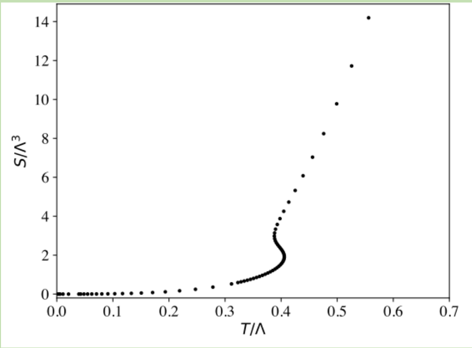
The inverse problem

Einstein eqs.

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$



Equation of state



Inverse problem

-Loss= Σ(residuals Einstein eqs)^2 (sum over branes)

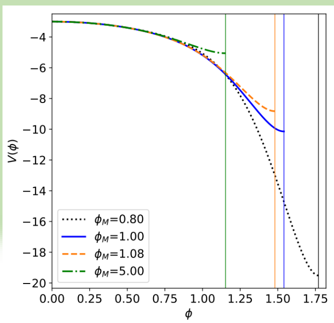
-For each black brane, S and T enter as boundary conditions at the horizon

-No assumptions of the functional form of V(φ)

The inverse problem

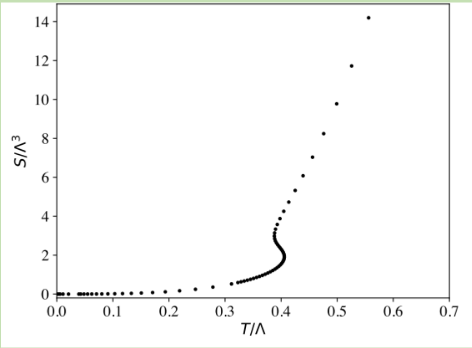
Einstein eqs.

$$S = \frac{2}{\kappa_5^2}\int dx^5\sqrt{-g}\left[\frac{1}{4}R - \frac{1}{2}(\nabla\phi)^2 - V(\phi)\right]$$



Inverse problem

Equation of state



-Loss= Σ(residuals Einstein eqs)^2 (sum over branes)

-For each black brane, S and T enter as boundary conditions at the horizon

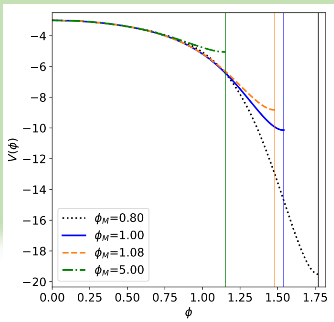
-No assumptions of the functional form of V(φ)

-We do not provide any initial guess for V(φ), it is random.

The inverse problem

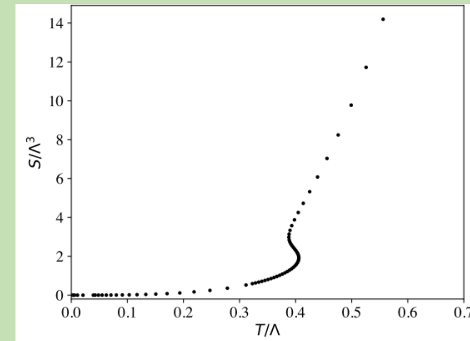
Einstein eqs.

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$



Inverse problem

Equation of state



-Loss= Σ(residuals Einstein eqs)^2 (sum over branes)

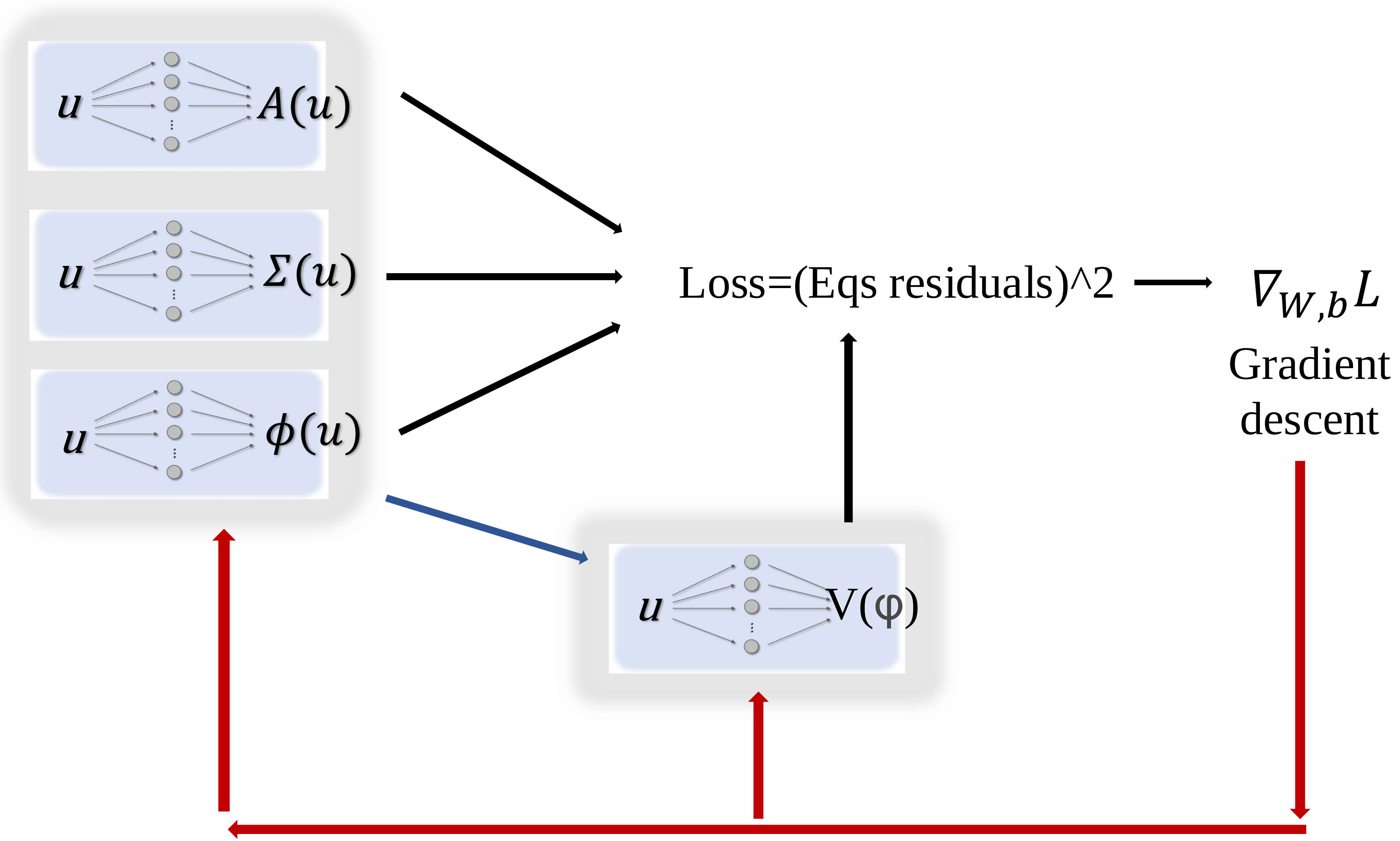
-For each black brane, S and T enter as boundary conditions at the horizon

-No assumptions of the functional form of V(φ)

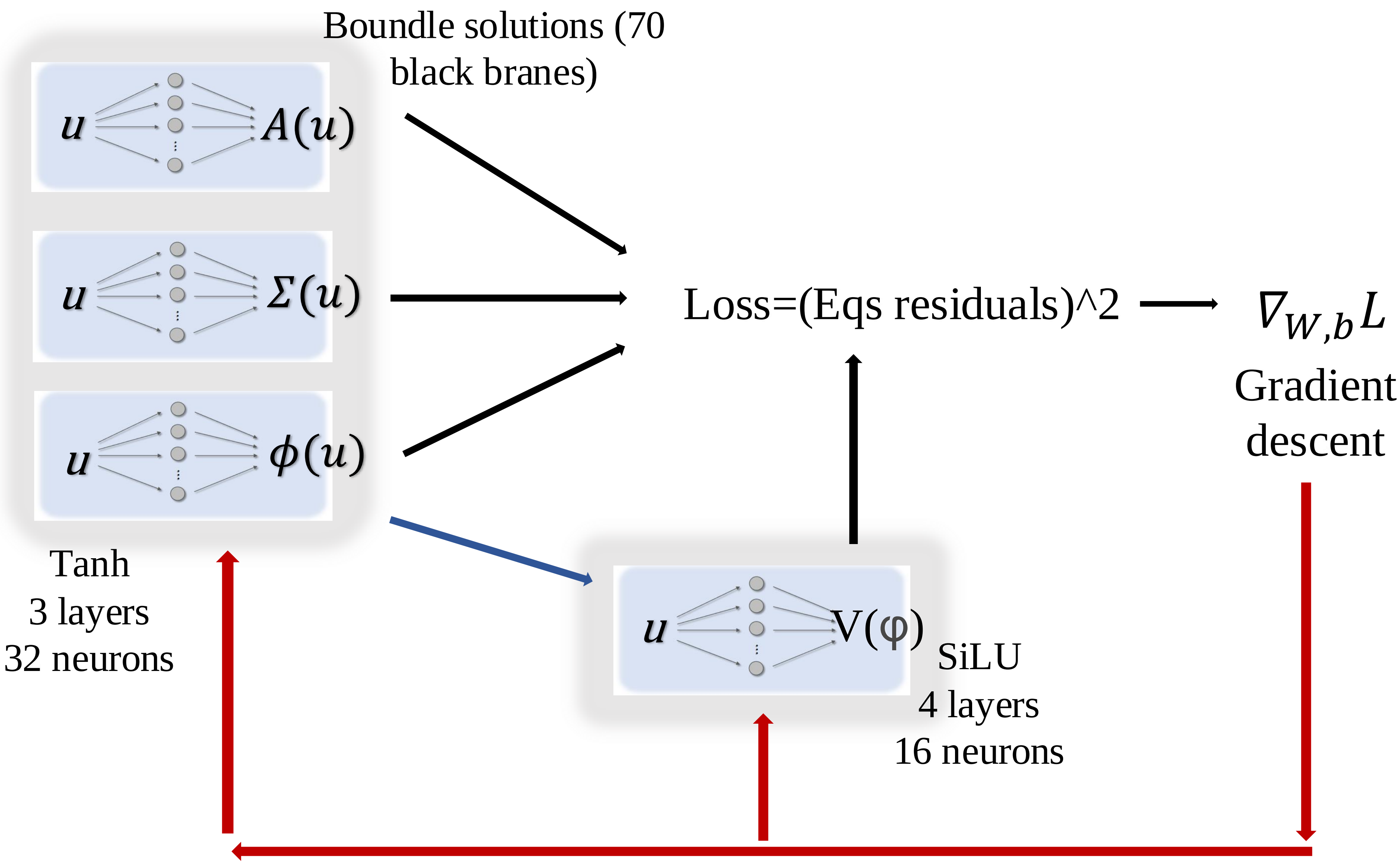
-We do not provide any initial guess for V(φ), it is random.

-The range of phi's is found by the NN

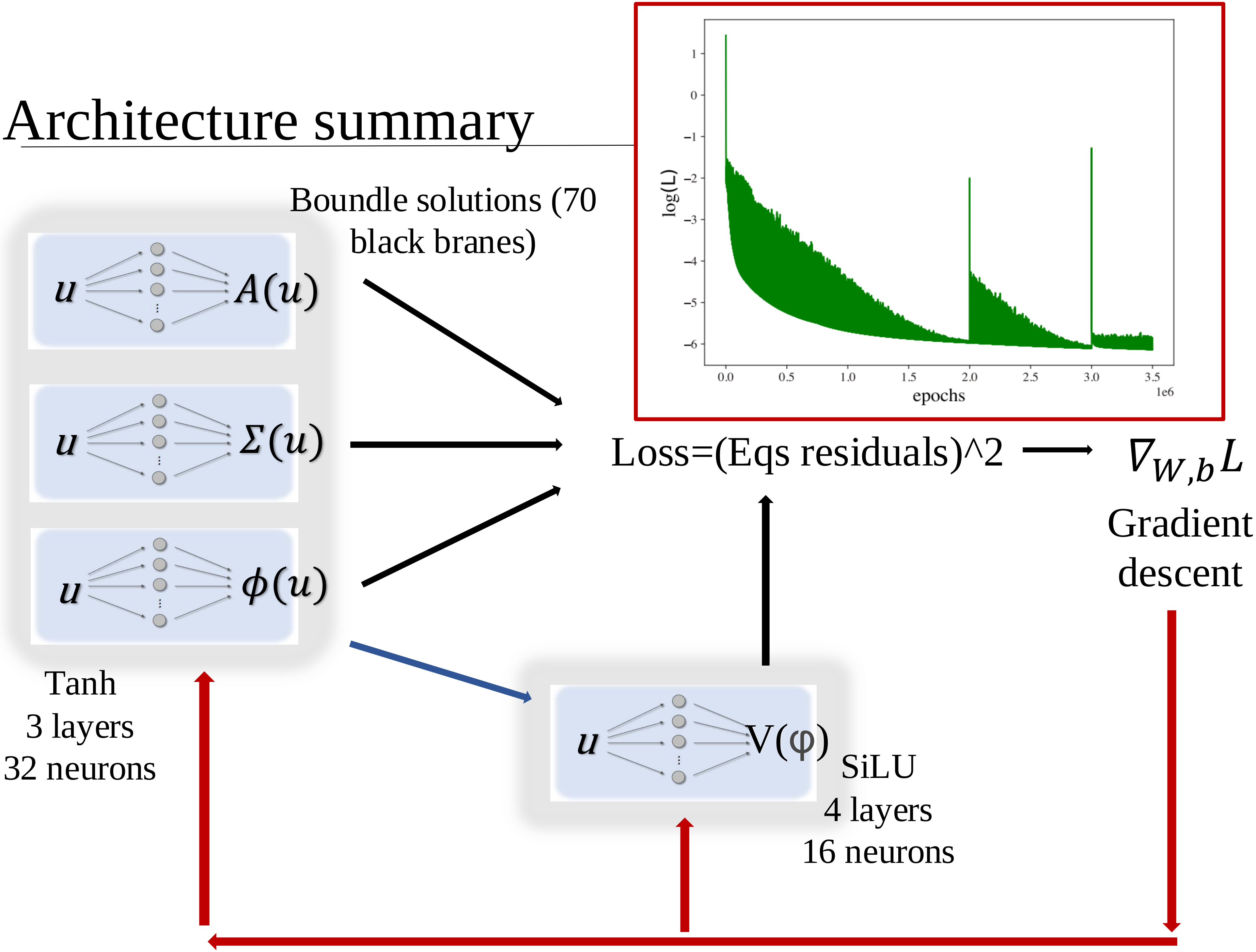
Architecture summary



Architecture summary



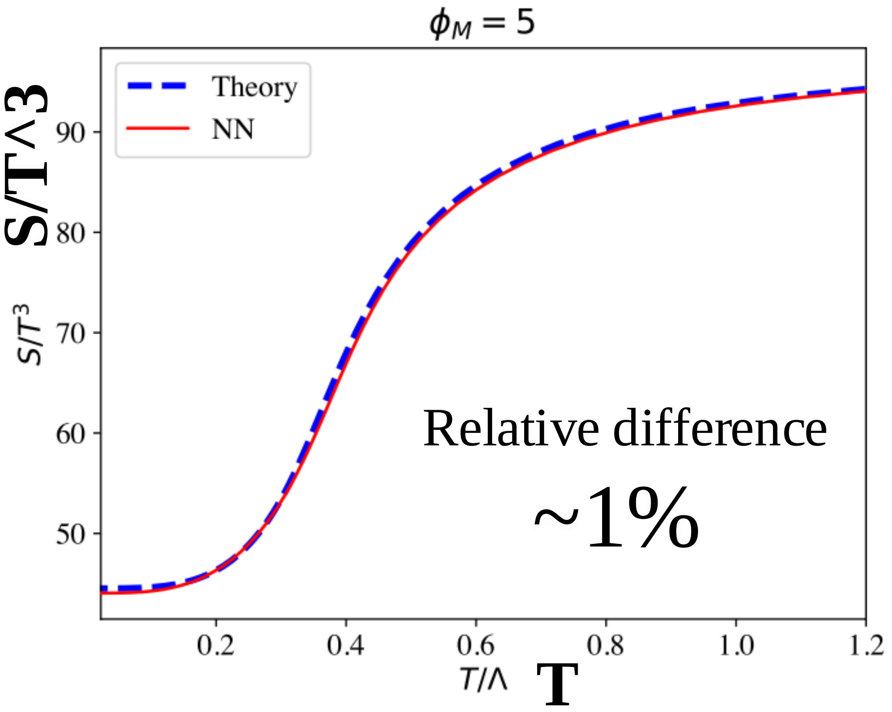
Architecture summary



Main results

Solving the inverse problem

Crossover second order first order



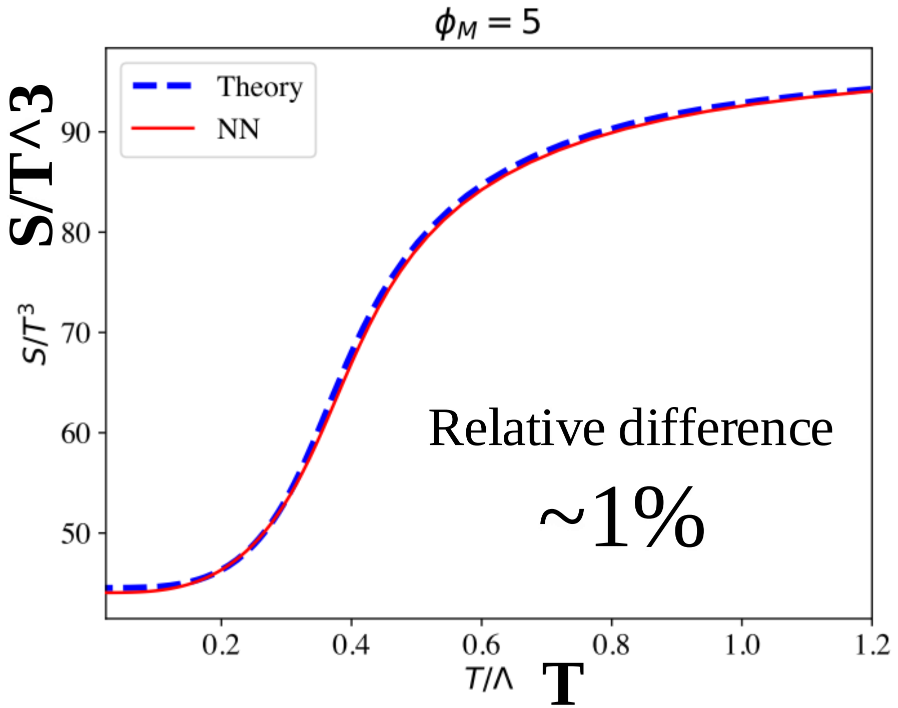
Main conclusion

We have solved the inverse problem using NN within ~1%, not solved before with other methods

Solving the inverse problem

Crossover second order first order

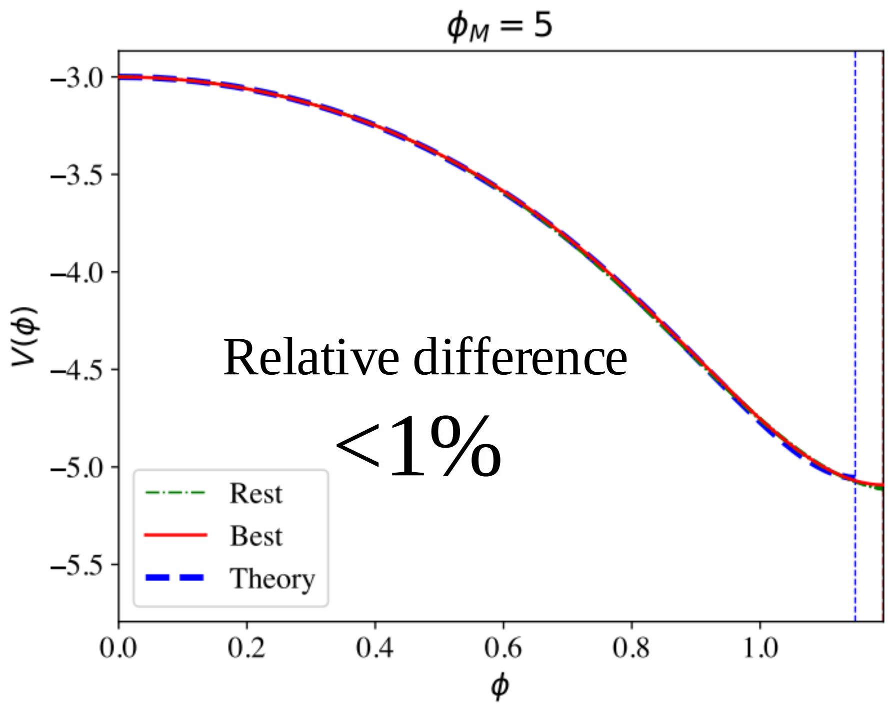
As we known the original potential,
we can perform a further check:

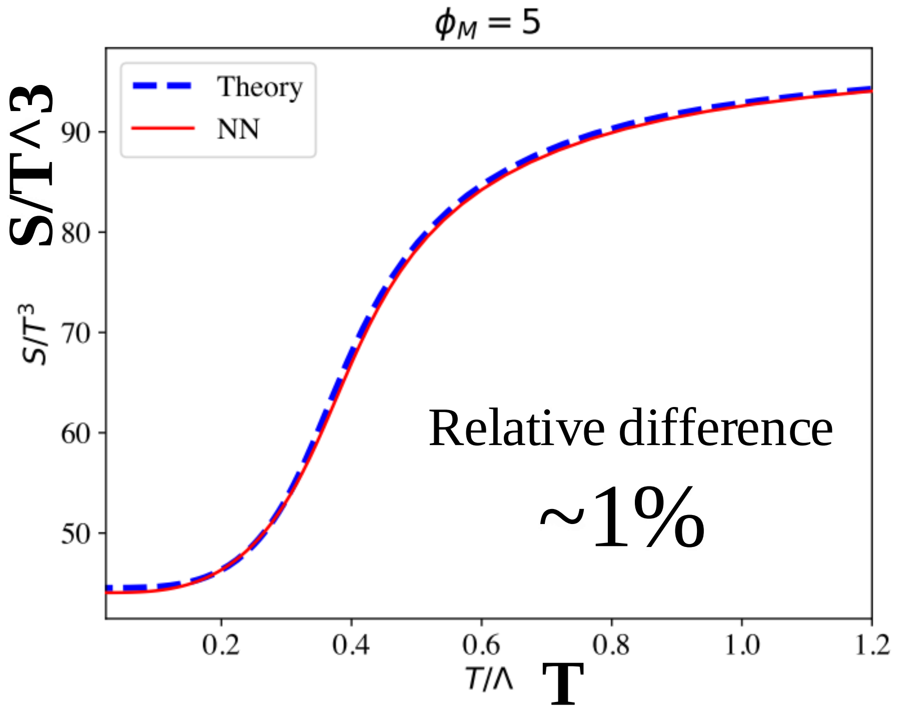


Main conclusion
We have solved the inverse problem using NN within ~1%,
not solved before with other methods

Solving the inverse problem

Crossover second order first order



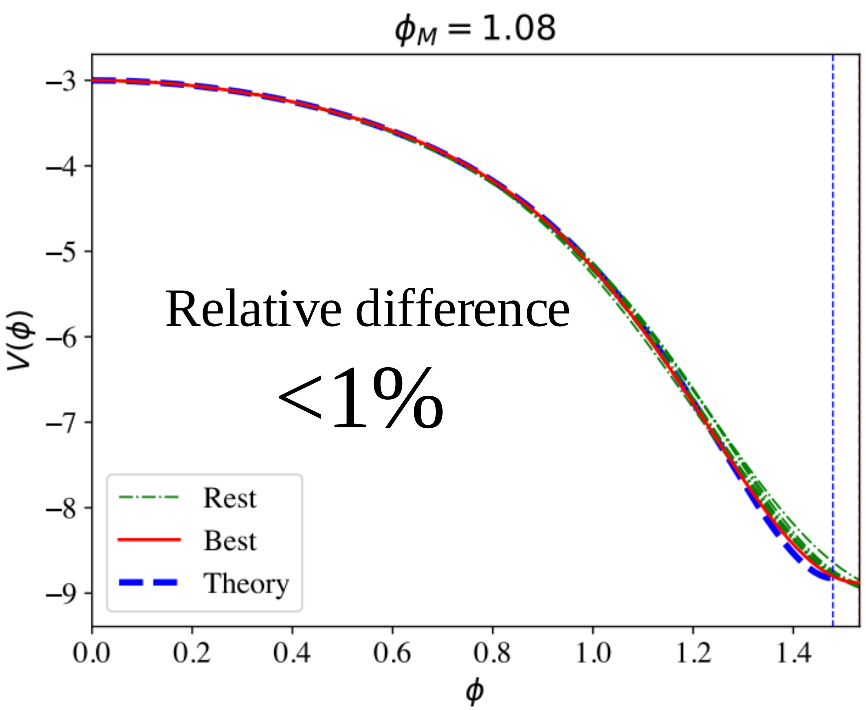


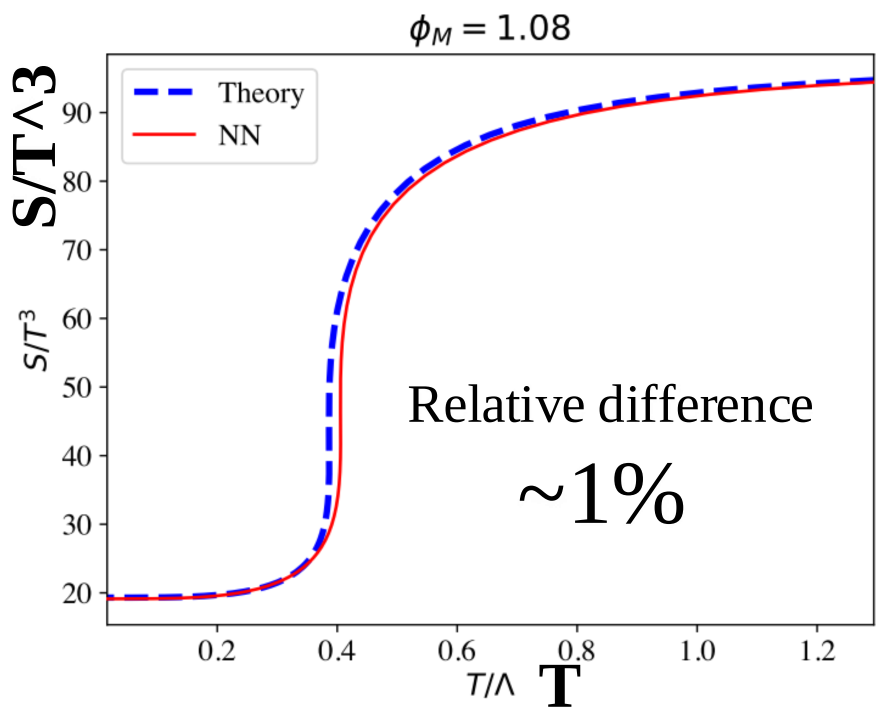
Main conclusion

We have solved the inverse problem using NN within ~1%, not solved before with other methods

Solving the inverse problem

Crossover **second order** first order



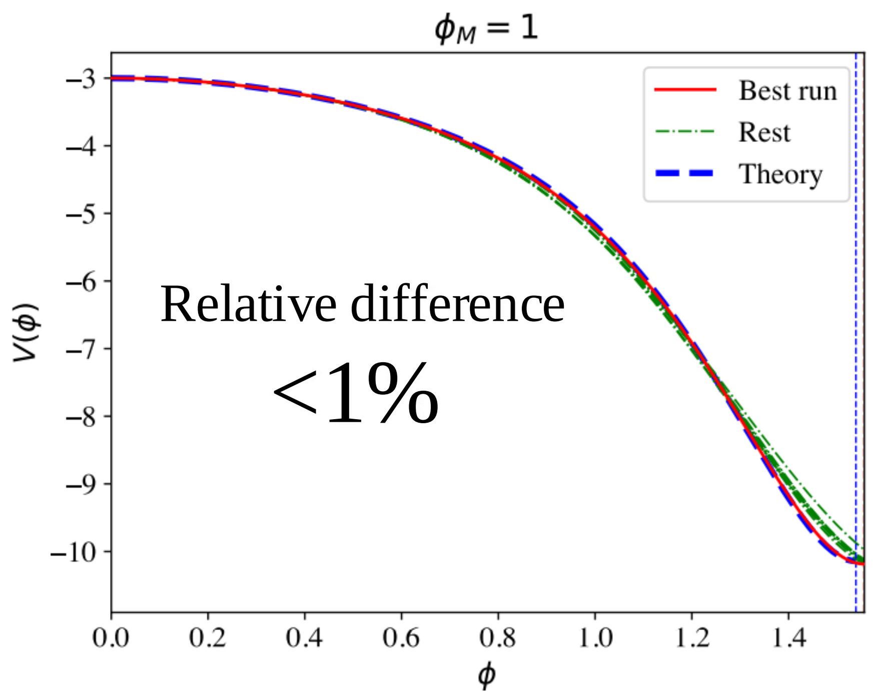


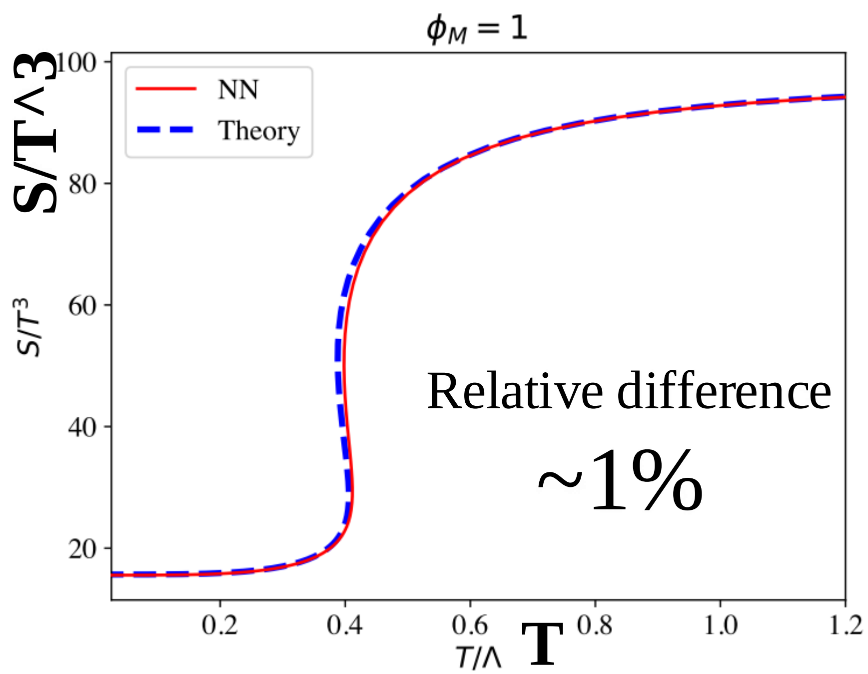
Main conclusion

We have solved the inverse problem using NN within ~1%, not solved before with other methods

Solving the inverse problem

Crossover　　second order　　**first order**





Main conclusion

We have solved the inverse problem using NN within ~1%,
not solved before with other methods

Additional comments

Additional comments

→ Solving with NN 'is an art' ...we have some guide, but there is also some part of trial and error

Additional comments

→ Solving with NN 'is an art' ...we have some guide, but there is also some part of trial and error

- Choose specific architecture, number of neurons, layers, etc
- Sampling in u (holographic variable)
- Sampling in the S(T) curve

Additional comments

→ <u>Solving with NN 'is an art'</u> ...we have some guide, but there is also some part of trial and error

- Choose specific architecture, number of neurons, layers, etc
- Sampling in u (holographic variable)
- Sampling in the S(T) curve

→ There is some stochasticity:

- Random initial data (10 runs)
- Stochastic gradient descent

Additional comments

→ Solving with NN 'is an art' ...we have some guide, but there is also some part of trial and error

- Choose specific architecture, number of neurons, layers, etc
- Sampling in u (holographic variable)
- Sampling in the S(T) curve

→ There is some stochasticity:
- Random initial data (10 runs)
- Stochastic gradient descent

→ Our method is general and applicable in generic inverse problems

Additional comments

→ Solving with NN 'is an art' ...we have some guide, but there is also some part of trial and error

- Choose s
- Samplin
- Samplin

→ The

Thank you!

→ Our method is general and applicable in generic inverse problems

Backup Slides

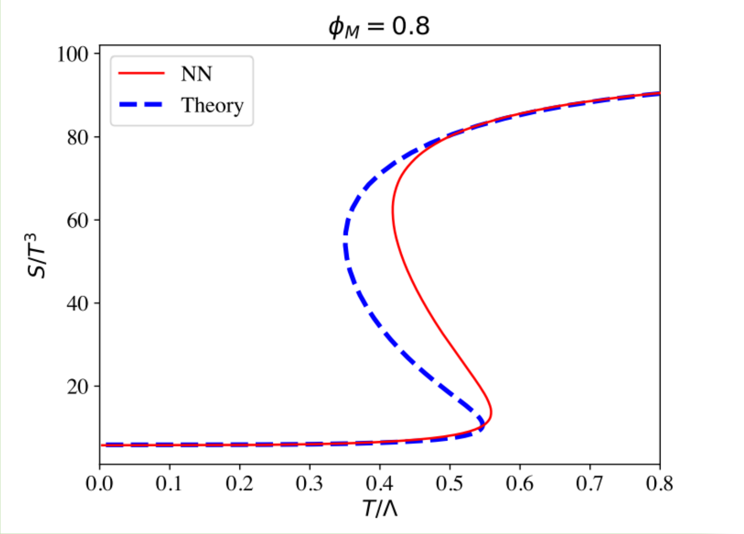
Future directions

This is just a first paper, a lot of room for improvement and extensions

→ Next step: improve accuracy

We are currently increasing the number of neurons and layers

We want to improve the resolution in problems with large separation of scales



→ We are also exploring transfer learning

Our potential

$$S = \frac{2}{\kappa_5^2} \int dx^5 \sqrt{-g} \left[\frac{1}{4} R - \frac{1}{2} (\nabla \phi)^2 - V(\phi) \right]$$

$$V_{\text{theory}}(\phi) = -\frac{4}{3} \mathcal{W}(\phi)^2 + \frac{1}{2} \mathcal{W}'(\phi)^2 \,,$$

$$\mathcal{W}(\phi) = -\frac{3}{2} - \frac{\phi^2}{2} - \frac{\phi^4}{4\phi_M^2} + \frac{\phi^6}{10} \,.$$

Einstein equations

$$E_1 = \nu_\Sigma - \tilde{\Sigma}' \,, \tag{4.3a}$$

$$E_2 = \nu_A - \tilde{A}' \,, \tag{4.3b}$$

$$E_3 = \nu_\phi - \phi' \,, \tag{4.3c}$$

$$E_4 = \nu_\Sigma' + \frac{2}{3}\tilde{\Sigma}\,\nu_\phi^2 \,, \tag{4.3d}$$

$$E_5 = u^2\,\tilde{\Sigma}\,\nu_A' + \frac{8}{3}V(\phi)\,\tilde{\Sigma} + \nu_A\,\left(3u^2\,\nu_\Sigma - 5u\,\tilde{\Sigma}\right) + \tilde{A}\left(8\tilde{\Sigma} - 6u\,\nu_\Sigma\right)\,, \tag{4.3e}$$

$$E_6 = u^2\,\tilde{\Sigma}\,\tilde{A}\,\nu_\phi' - \tilde{\Sigma}\,\frac{dV}{d\phi} + \nu_\phi\left(-3u\,\tilde{A}\,\tilde{\Sigma} + u^2\,\tilde{\Sigma}\,\nu_A + 3u^2\,\nu_\Sigma\,\tilde{A}\right)\,, \tag{4.3f}$$

$$E_7 = \left(u\,\nu_\Sigma - \tilde{\Sigma}\right)\left(u^2\,\tilde{\Sigma}\,\nu_A + 2u^2\,\tilde{A}\,\nu_\Sigma - 4u\tilde{A}\tilde{\Sigma}\right) - \frac{2}{3}u\,\tilde{\Sigma}^2\left(u^2\tilde{A}\,\nu_\phi^2 - 2V(\phi)\right)\,. \tag{4.3g}$$

Boundary conditions

$$\tilde{A}|_{u=0} = 1 \ ,$$

$$\tilde{\Sigma}|_{u=0} = 1 \ ,$$

$$\phi|_{u=0} = 0 \ ,$$

$$\nu_\phi|_{u=0} = 1 \ ,$$

$$\tilde{A}|_{u=1} = 0 \ ,$$

$$\nu_A|_{u=1} = -4\pi T \ ,$$

$$\tilde{\Sigma}_{u=1} = (S/\pi)^{1/3} \ .$$

Libraries

We have chosen the `Adam` optimizer[2] for the optimization process, which uses stochastic gradient descent with information about higher-order momenta (see [49] for details).

The described NN system has been implemented in the `Python` language through the open source `neurodiffeq` library [50], built on `PyTorch`.

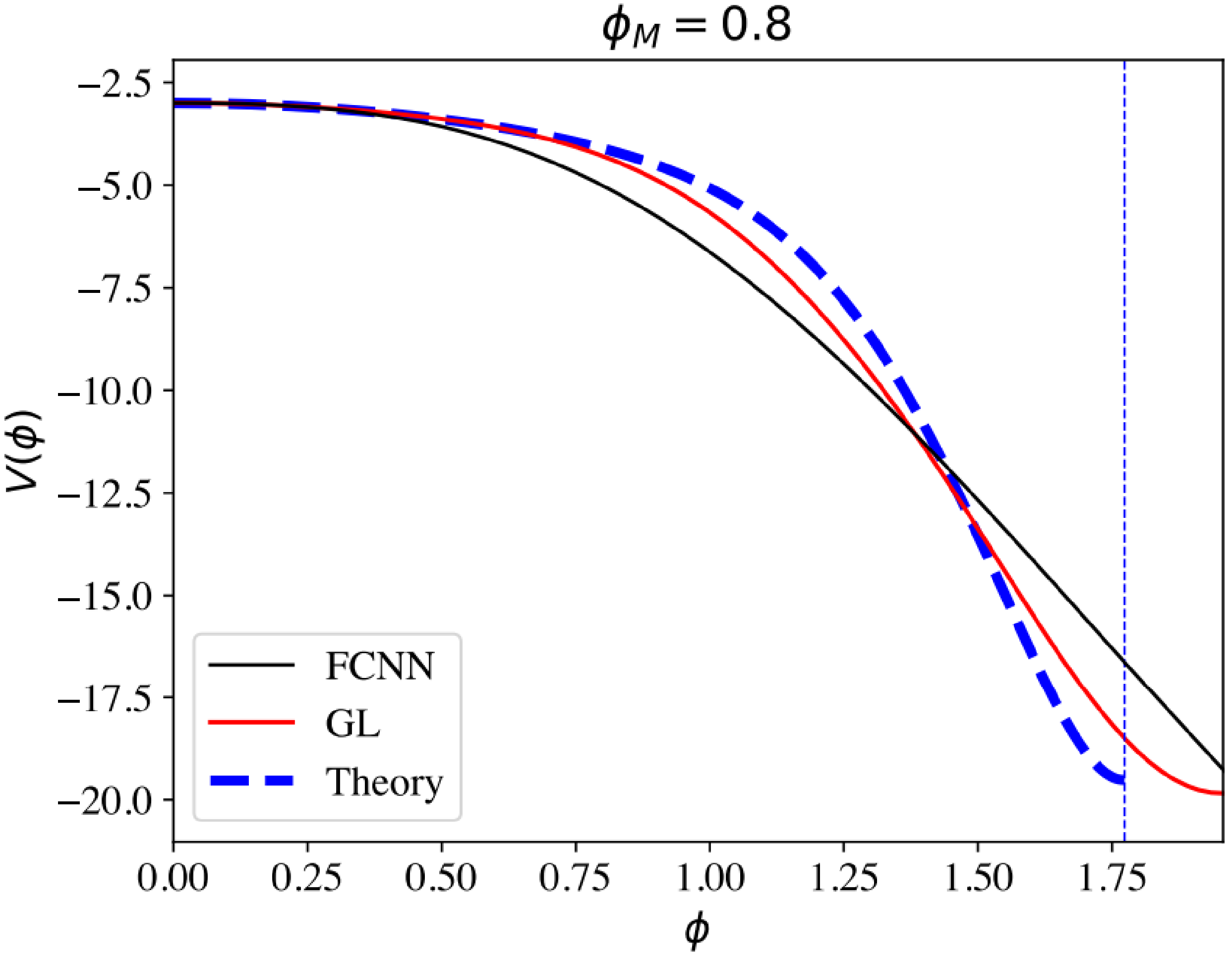
[49] D.P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017.

[50] F. Chen, D. Sondak, P. Protopapas, M. Mattheakis, S. Liu, D. Agarwal et al., *Neurodiffeq: A python package for solving differential equations with neural networks*, *Journal of Open Source Software* **5** (2020) 1931.

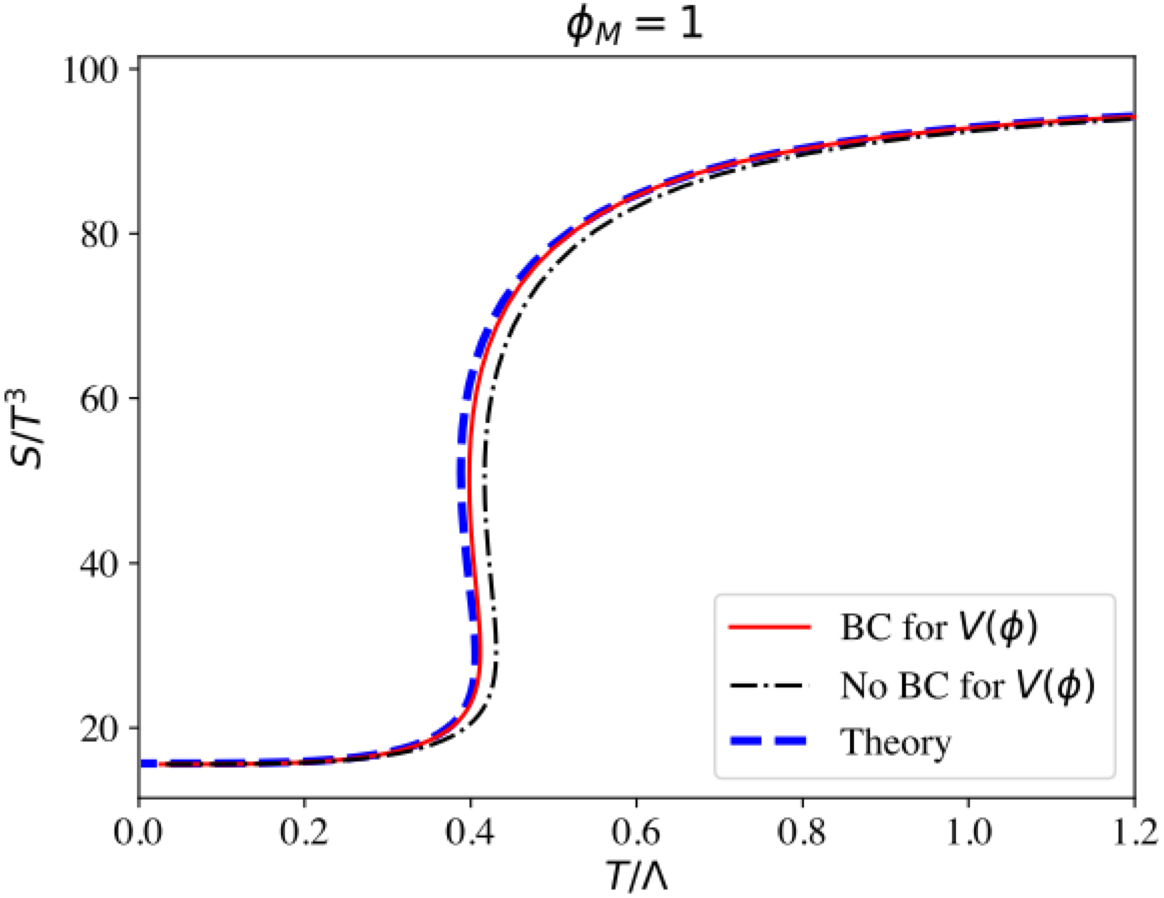
Computational considerations

Training and discovery of a solution with sub-per cent precision in the potential and per cent precision in the recovered equation of state requires about 8-16 hours and a couple of million epochs in a dedicated NVidia A40 GPU. As explained above, we performed ten

Gaussian localization vs fully connected



Additional loss



References NN in holography

The only papers that reconstruct the theory:

[8] K. Hashimoto, K. Ohashi and T. Sumimoto, *Deriving the dilaton potential in improved holographic QCD from the meson spectrum*, *Phys. Rev. D* **105** (2022) 106008 [2108.08091].

[9] K. Hashimoto, K. Ohashi and T. Sumimoto, *Deriving the dilaton potential in improved holographic QCD from the chiral condensate*, *PTEP* **2023** (2023) 033B01 [2209.04638].

[10] X. Chen and M. Huang, *Machine learning holographic black hole from lattice QCD equation of state*, 2401.06417.

Other papers (reconstruct solutions):

[12] Y.-Z. You, Z. Yang and X.-L. Qi, *Machine Learning Spatial Geometry from Entanglement Features*, *Phys. Rev. B* **97** (2018) 045153 [1709.01223].

[13] K. Hashimoto, S. Sugishita, A. Tanaka and A. Tomiya, *Deep learning and the AdS/CFT correspondence*, *Phys. Rev. D* **98** (2018) 046019 [1802.08313].

[14] K. Hashimoto, S. Sugishita, A. Tanaka and A. Tomiya, *Deep Learning and Holographic QCD*, *Phys. Rev. D* **98** (2018) 106014 [1809.10536].

[15] H.-Y. Hu, S.-H. Li, L. Wang and Y.-Z. You, *Machine Learning Holographic Mapping by Neural Network Renormalization Group*, *Phys. Rev. Res.* **2** (2020) 023369 [1903.00804].

[16] K. Hashimoto, *AdS/CFT correspondence as a deep Boltzmann machine*, *Phys. Rev. D* **99** (2019) 106017 [1903.04951].

......+ 15 references