Can AI Understand Hamiltonian Mechanics?

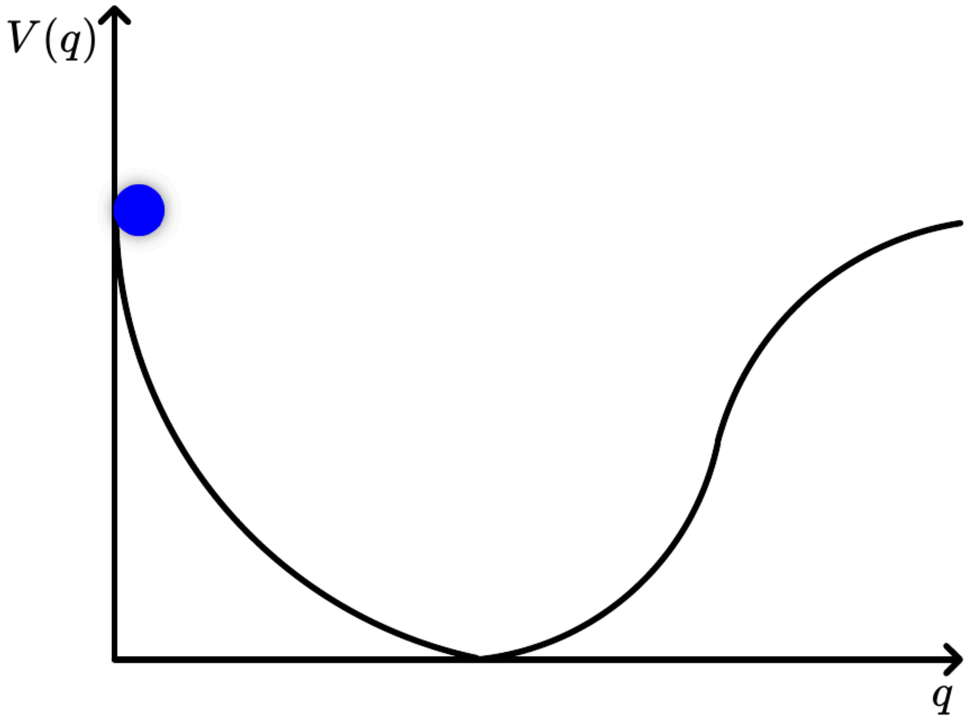
[arXiv: 2410.20951]

Tae-Geun Kim Seong Chan Park

Summer Institute 2025

2025.08.21

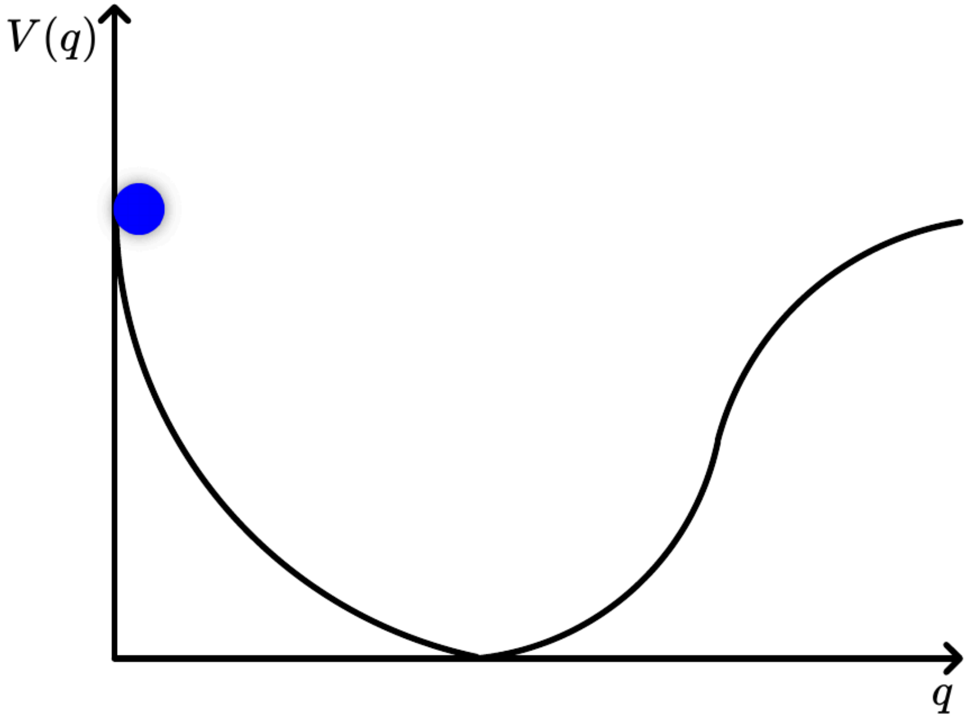
Understanding Hamiltonian Mechanics



Initial Condition

$$q(0) = 0$$
$$p(0) = 0$$

Understanding Hamiltonian Mechanics



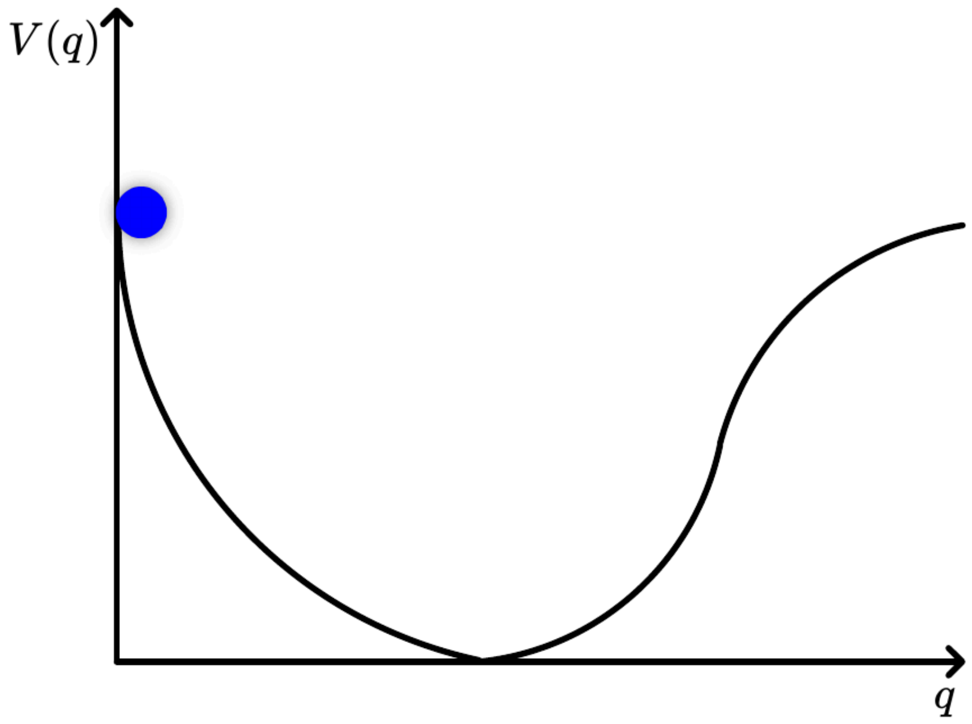
Initial Condition

$$q(0) = 0$$
$$p(0) = 0$$

Hamilton equation

$$\dot{q} = \frac{\partial H}{\partial p}$$
$$\dot{p} = -\frac{\partial H}{\partial q}$$

Understanding Hamiltonian Mechanics



Initial Condition

$$q(0) = 0$$
$$p(0) = 0$$

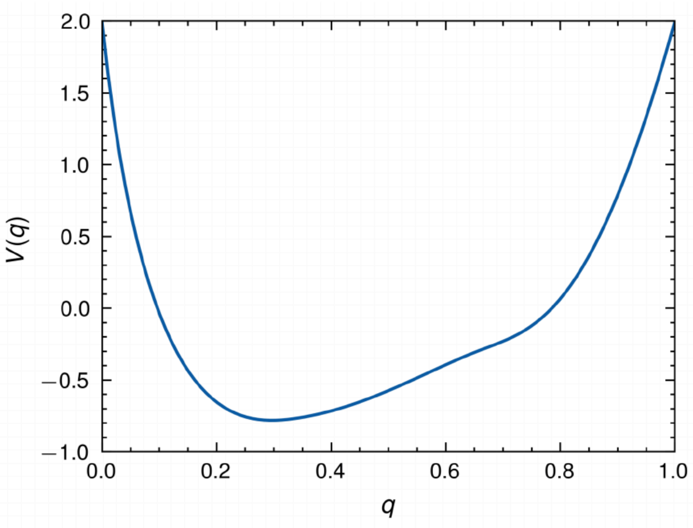
Hamilton equation

$$\dot{q} = \frac{\partial H}{\partial p}$$
$$\dot{p} = -\frac{\partial H}{\partial q}$$

Solve ODE (with ODESolver)

$$q(\Delta t) = \int_0^{\Delta t} \dot{q} \mathrm{d}t$$
$$p(\Delta t) = \int_0^{\Delta t} \dot{p} \mathrm{d}t$$

Understanding Hamiltonian Mechanics



Hamilton equation

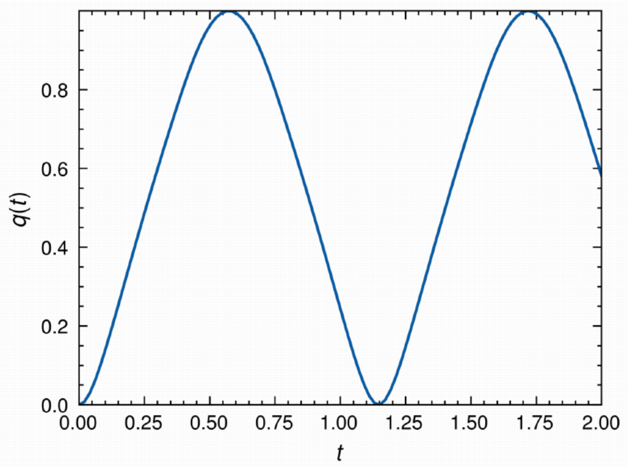
$$\dot{q} = \frac{\partial H}{\partial p}$$

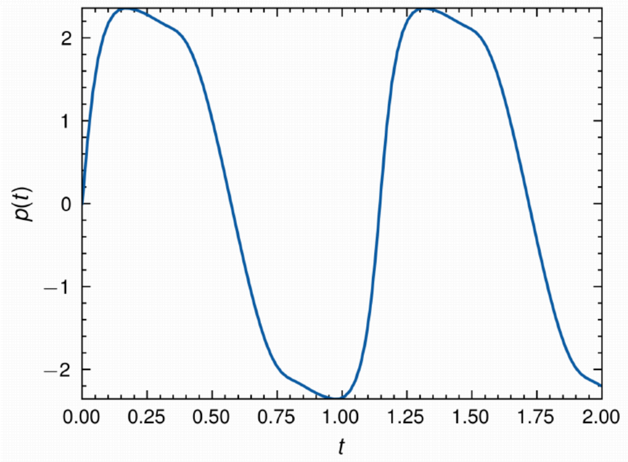
$$\dot{p} = -\frac{\partial H}{\partial q}$$

Solve ODE (with ODESolver)

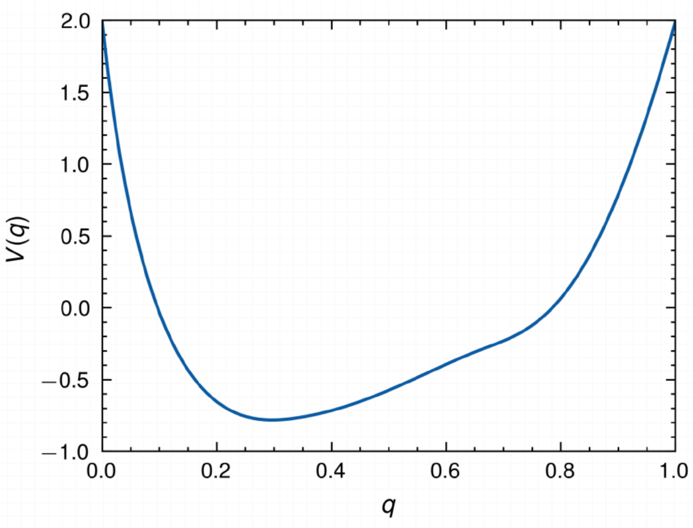
$$q(t + \Delta t) = \int_t^{t+\Delta t} \dot{q}\,\mathrm{d}t$$

$$p(t + \Delta t) = \int_t^{t+\Delta t} \dot{p}\,\mathrm{d}t$$

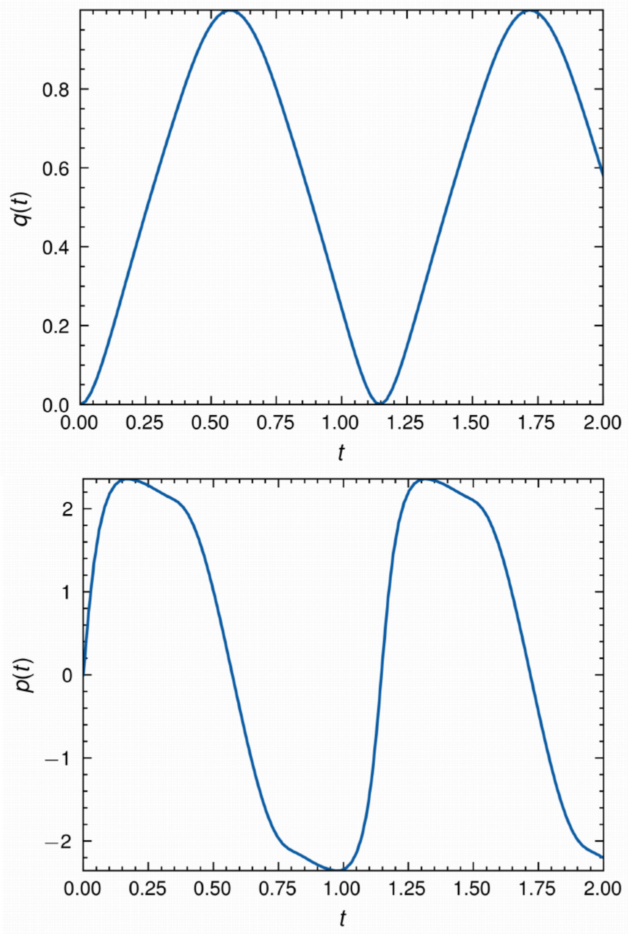




Understanding Hamiltonian Mechanics



Can AI learn this,
w/o any equations?



Formulation

Is this operator well-defined?

- **Goal**: From potential function $V(q)$, obtain $q(t)$ and $p(t)$ without any equations & solvers.

Formulation

Is this operator well-defined?

- **Goal**: From potential function $V(q)$, obtain $q(t)$ and $p(t)$ without any equations & solvers.

More specifically,

Find $G : \mathcal{V} \to \mathcal{T}$ such that $G(V)(t) = [q(t), p(t)]$

  - $\mathcal{V}$: The space of potential functions ($V : \mathbb{R} \to \mathbb{R}$)
  - $\mathcal{T}$: The space of trajectory functions ($T : \mathbb{R}_{\geq 0} \to \mathbb{R}^2$)
  - G: The Hamilton "Operator" which maps potential functions to trajectory functions.

Formulation

Is this operator well-defined?

- **Goal**: From potential function $V(q)$, obtain $q(t)$ and $p(t)$ without any equations & solvers.

More specifically,

Find $G : \mathcal{V} \to \mathcal{T}$ such that $G(V)(t) = [q(t), p(t)]$

- $\mathcal{V}$: The space of potential functions ($V : \mathbb{R} \to \mathbb{R}$)
- $\mathcal{T}$: The space of trajectory functions ($T : \mathbb{R}_{\geq 0} \to \mathbb{R}^2$)
- G: The Hamilton "Operator" which maps potential functions to trajectory functions.

Our goal is to learn the operator G!

Can AI learn an Operator?

Q. How can we ensure that AI learns?

Can AI learn an Operator?

Q. How can we ensure that AI learns?

[Lu et al., NeurIPS (2017), G. Cybenko, MCSS (1989)]

Theorem 1 (Universal Approximation Theorem for ReLU Networks)

For any Lebesgue-integrable function $f : \mathbb{R}^n \to \mathbb{R}$ *and any* $\varepsilon > 0$*, there exists a fully-connected ReLU network* $\mathcal{A}$ *with width* $\mathrm{d}_m \leq n + 4$*, such that the function* $F_{\mathcal{A}}$ *represented by this network satisfies*

$$\int_{\mathbb{R}^n} |f(x) - F_{\mathcal{A}}(x)| \mathrm{d}x < \varepsilon.$$

More details, see KC Kong's first lecture

Can AI learn an Operator?

Q. How can we ensure that AI learns?

[Lu et al., Nat. Mach. Intell. (2021), Chen & Chen, IEEE Trans. Neural Netw. (1995)]

Theorem 2 (Universal Approximation Theorem for Operator)

Suppose that X is a Banach space, $K_1 \subset X, K_2 \subset \mathbb{R}^d$ are two ***compact*** *sets in X and $\mathbb{R}^d$, respectively, V is a* ***compact*** *set in $C(K_1)$. Assume that $G : V \to C(K_2)$ is a nonlinear continuous operator. Then, for any $\varepsilon > 0$, there exist positive integers m, p, continuous vector functions $\boldsymbol{g} : \mathbb{R}^m \to \mathbb{R}^p$, $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^p$, and $x_1, x_2, \cdots, x_m \in K_1$, such that*

$$\left| G(u)(y) - \langle \underbrace{\boldsymbol{g}(u(x_1), u(x_2), \cdots, u(x_m))}_{\text{branch}}, \underbrace{\boldsymbol{f}(y)}_{\text{trunk}} \rangle \right| < \varepsilon$$

holds for all $u \in V$ and $y \in K_2$. Furthermore, the functions $\boldsymbol{g}$ and $\boldsymbol{f}$ can be chosen as diverse classes of ***neural networks****, which satisfy the classical universal approximation theorem of functions, for example, (stacked/unstacked) fully connected neural networks, residual neural networks and convolutional neural networks.*

How to learn an Operator?


ARTICLES

https://doi.org/10.1038/s42256-021-00302-5

nature machine intelligence

Check for updates

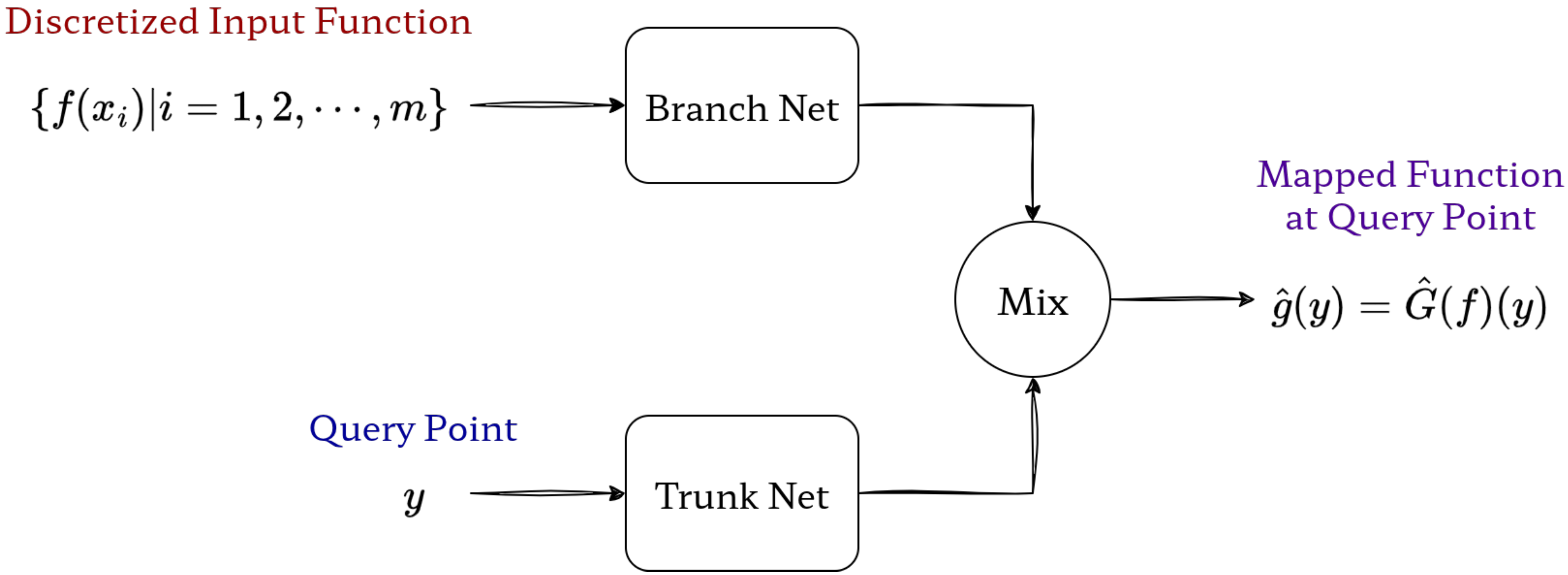
Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators

Lu Lu[1], Pengzhan Jin[2,3], Guofei Pang[2], Zhongqiang Zhang[4] and George Em Karniadakis[2]

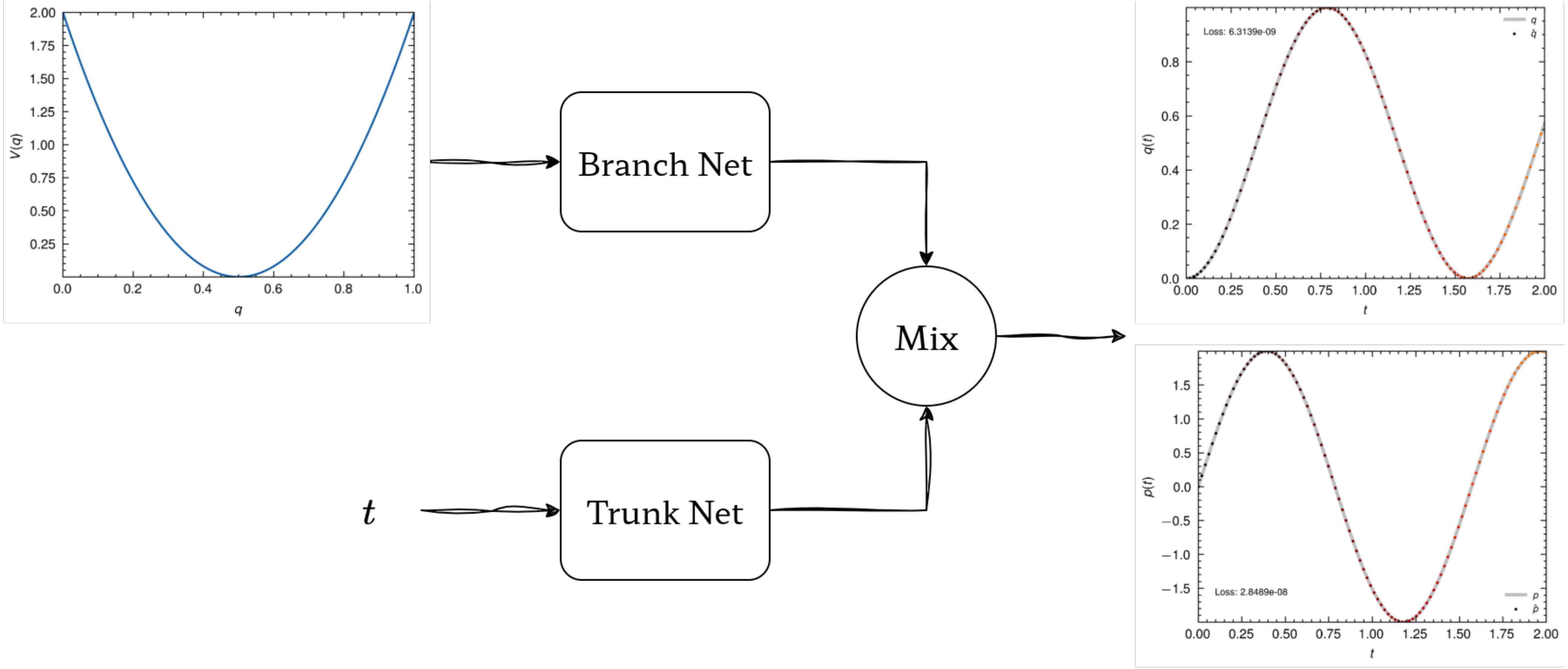
Figure 1: Lu et al., Nat. Mach. Intell. (2021) [2,934 Citations]

How to learn an Operator?

- Consider an operator $G : \mathcal{F} \to \mathcal{G}$, where $f(x) \in \mathcal{F}$ and $g(y) \in \mathcal{G}$ are functions.

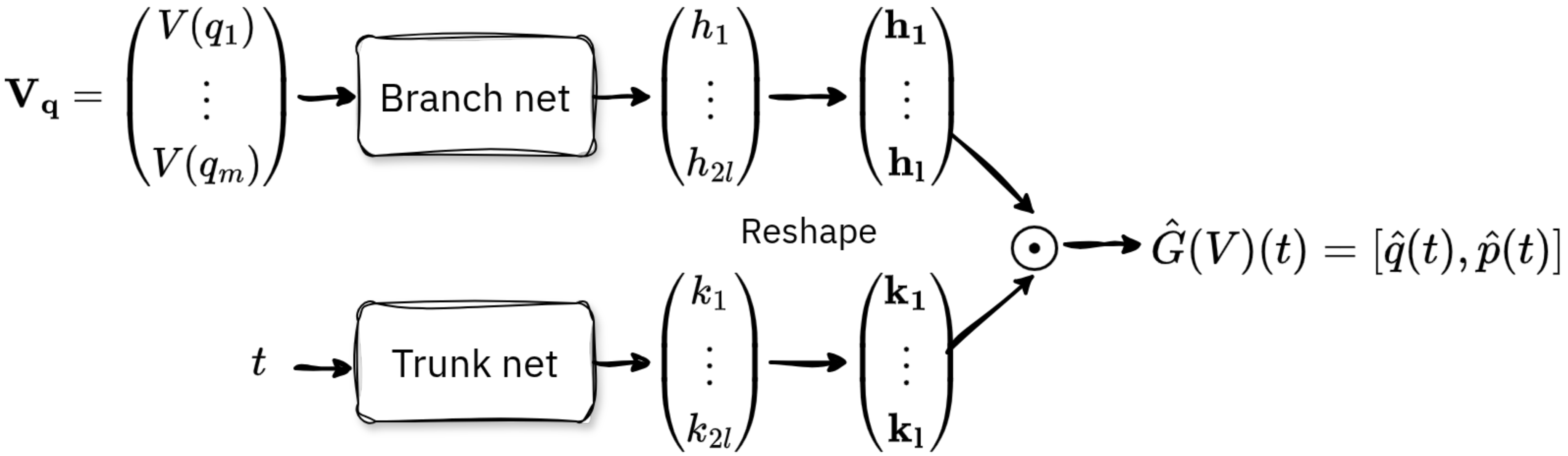


How to learn an Operator?



How to learn an Operator?

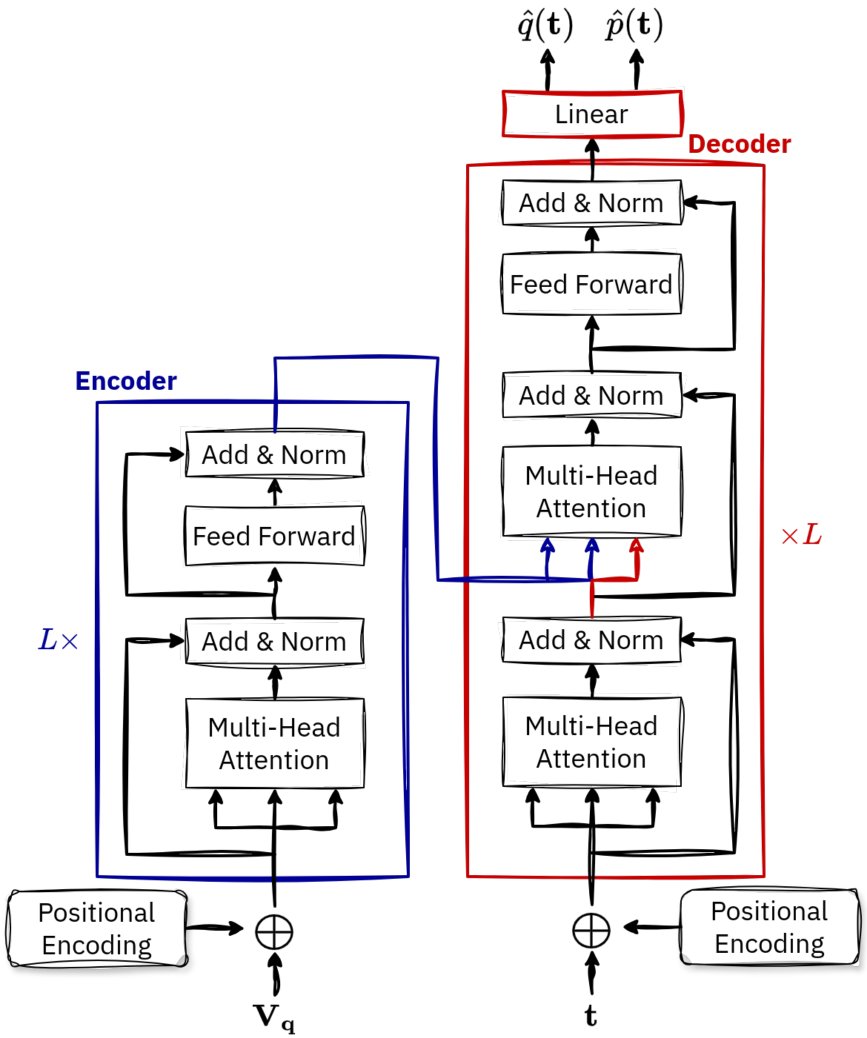
DeepONet (Deep Operator Network)



How to learn an Operator?

TraONet (Transformer Operator Network)

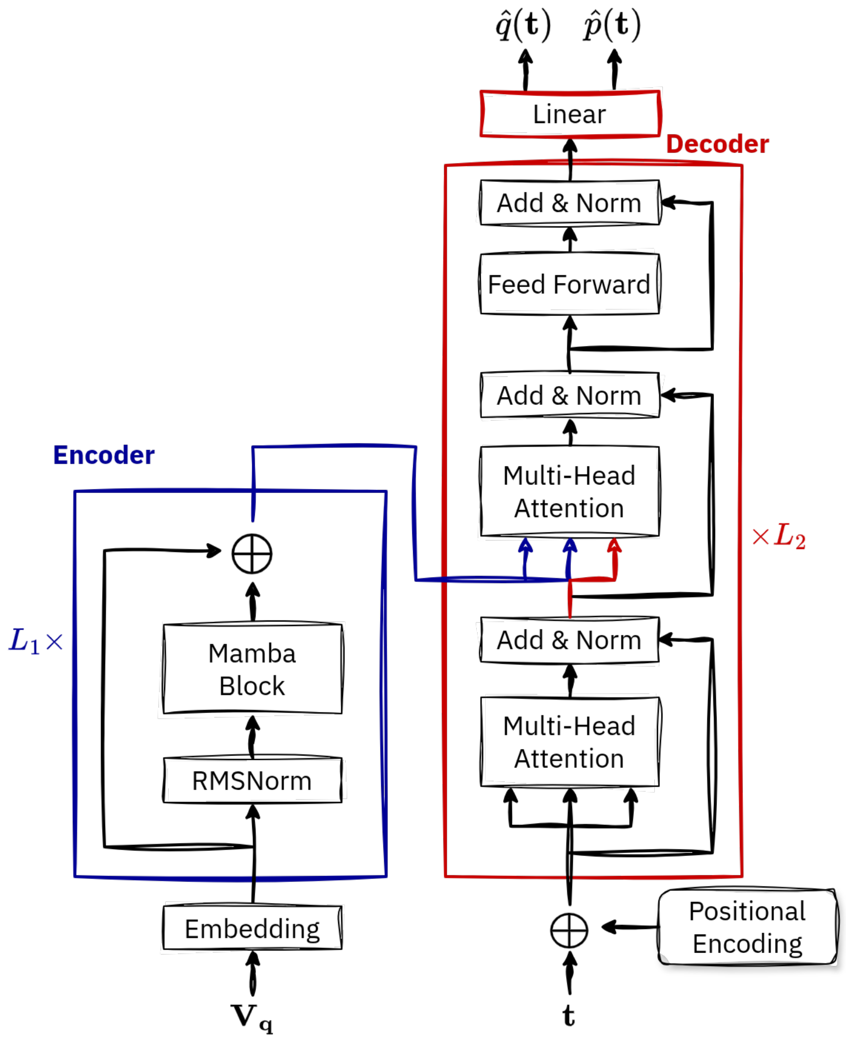
: Transformer Encoder + Transformer Decoder



How to learn an Operator?

MambONet (Mamba Operator Network)

: Mamba Encoder + Transformer Decoder



How to learn an Operator?

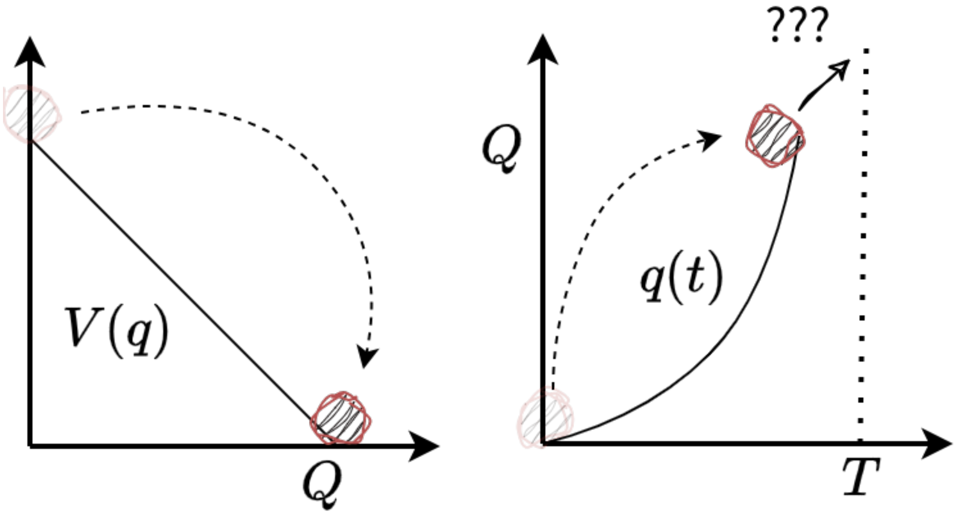
- **DeepONet**
  - Simple architecture + Fast evaluation
  - Inner product is not good for capturing local features
- **TraONet**
  - Attention mechanism enables focusing on local features

    (= for each t, which part of $V(q)$ is important?)
  - Attention mechanism requires quadratic complexity in the number of points in $V(q)$

    ($\Rightarrow$ may not be suitable for memorizing all $V(q)$)
- **MambONet**
  - Mamba architecture requires only linear complexity in the number of points in $V(q)$
  - With hybrid architecture, it can capture both local and global features

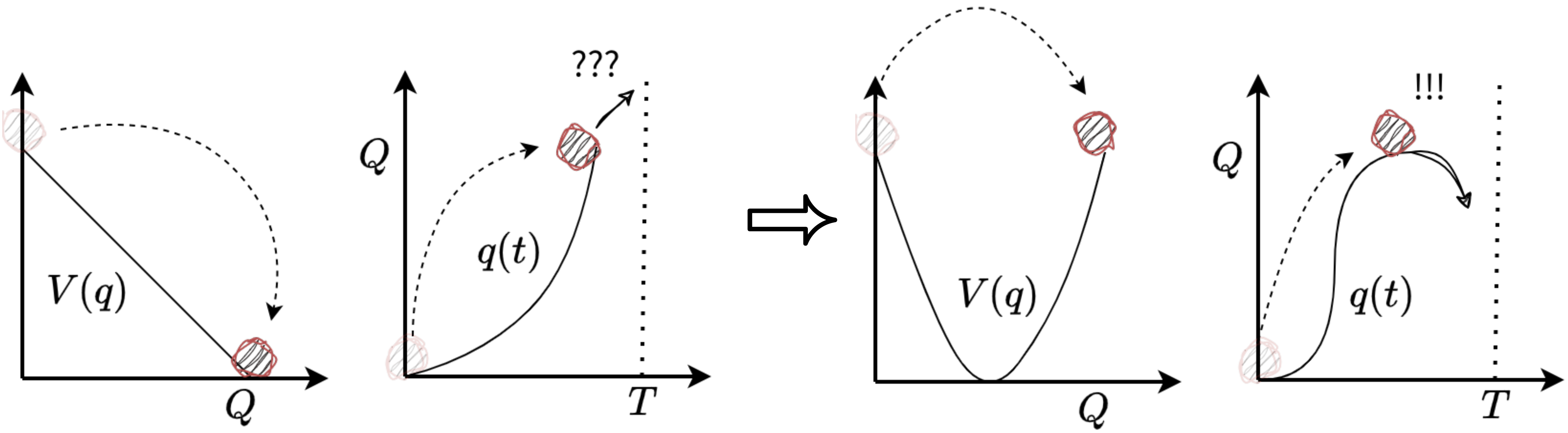
Compact Potential

- From the UAT for operator, domain & range of V and q, p should be compact



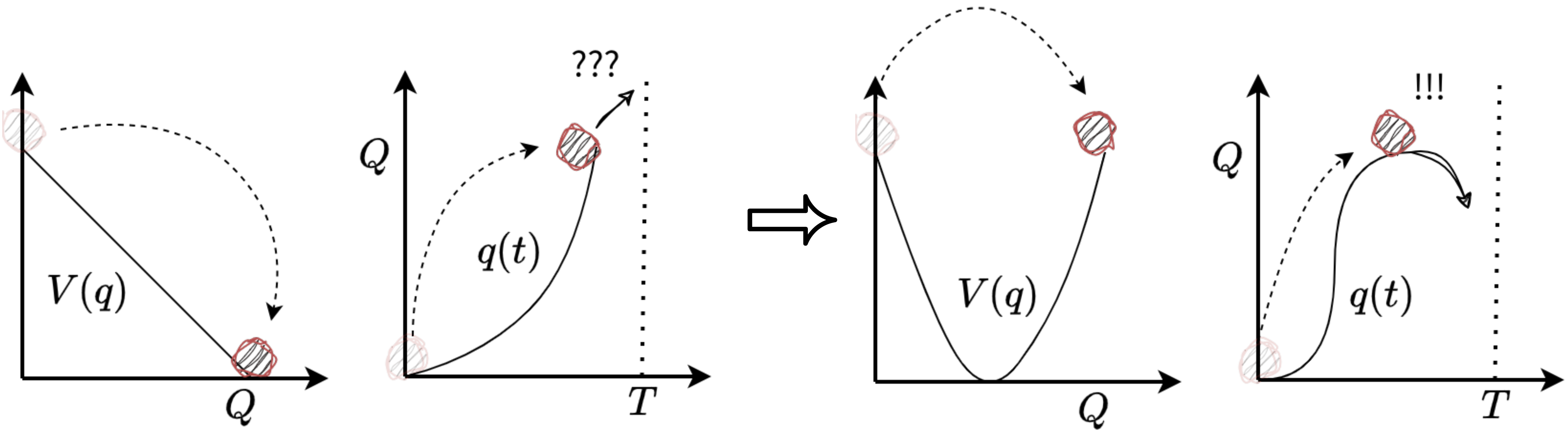
Compact Potential

- From the UAT for operator, domain & range of V and q, p should be compact



Compact Potential

- From the UAT for operator, domain & range of V and q, p should be compact



We need ***twice continuously differentiable & bounded*** potential functions

How to prepare data?

Constraints for Potential

1. C^2 continuity to guarantee the local existence & uniqueness of the solution
2. Boundedness for global existence & uniqueness and well-defined compactness

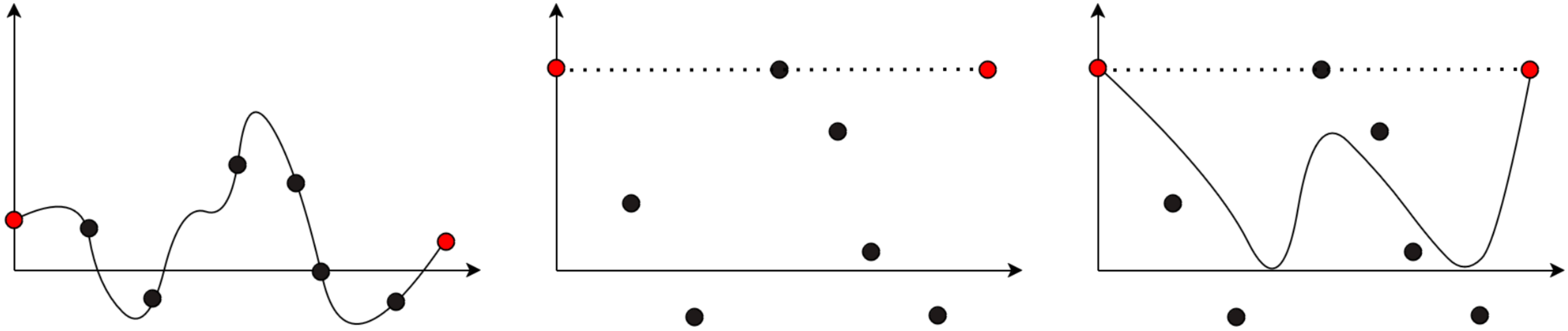
How to prepare data?

How to effectively cover the potential space?

Constraints for Potential

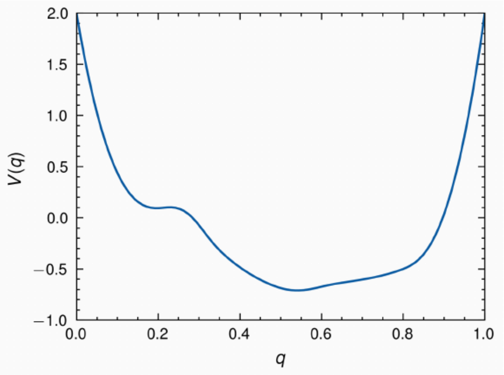
1. C^2 continuity to guarantee the local existence & uniqueness of the solution
2. Boundedness for global existence & uniqueness and well-defined compactness

Use ***Gaussian Random Field + Cubic B-Spline*** to generate potential functions

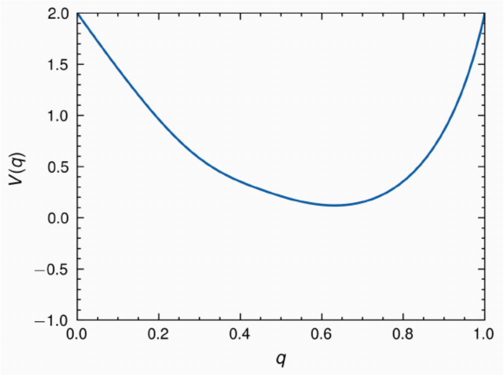


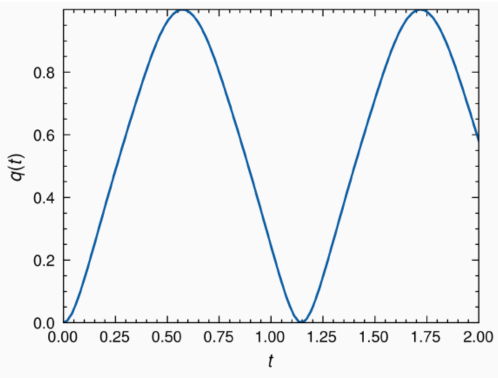
Generate GRF | Normalize | Cubic B-Spline

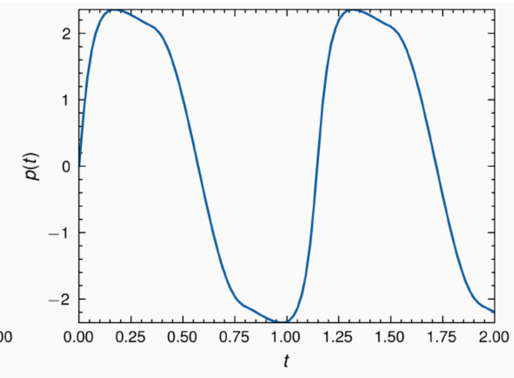
How to prepare data?

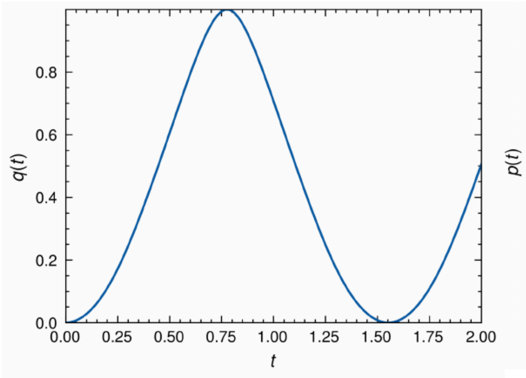


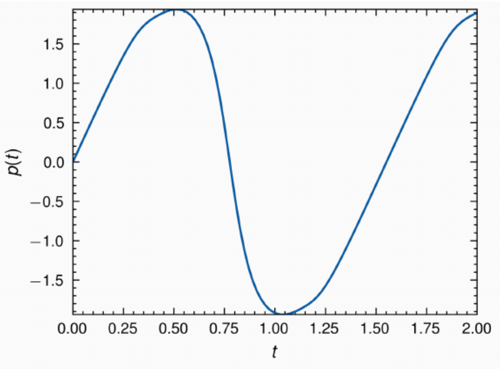
- Generate 100k potentials
- Use Yoshida integrator
- $q \in [0, 1]$ (100 nodes)
- $t \in [0, 2]$ (100 nodes)
- $V(0) = V(1) = 2$











Evaluations

Test Dataset

- Generate and sample 80k potentials with different random seed

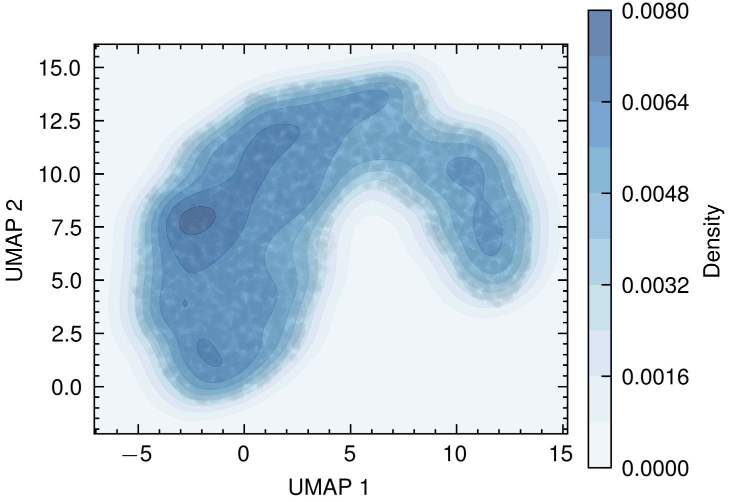


Figure 2: UMAP projection of the sampled 80k dataset

Physically Relevant Potentials

Potential	Formula ($V(q)$)	Description
SHO	$8(q-0.5)^2$	Analytic solution availalbe
Double Well	$\frac{625}{8}(q-0.2)^2(q-0.8)^2$	Common in quantum mechanics
Morse	$D_e\left(1-e^{-a(q-1/3)}\right)^2$	Models molecular bonds
ATW	$2-2\left[\frac{q}{\lambda}\right]_{q<\lambda}-2\left[\frac{1-q}{1-\lambda}\right]_{q\geq\lambda}$	Non-differentiable at $q=\lambda$
STW	$4\|q-0.5\|$	Non-differentiable at $q=0.5$
SSTW	$\frac{4}{\coth(\alpha/2)}\left(q-\frac{1}{2}\right)\coth\left(\alpha\left(q-\frac{1}{2}\right)\right)$	Smooth version of the STW

Table 1: List of potential functions used for testing the models.

Physically Relevant Potentials

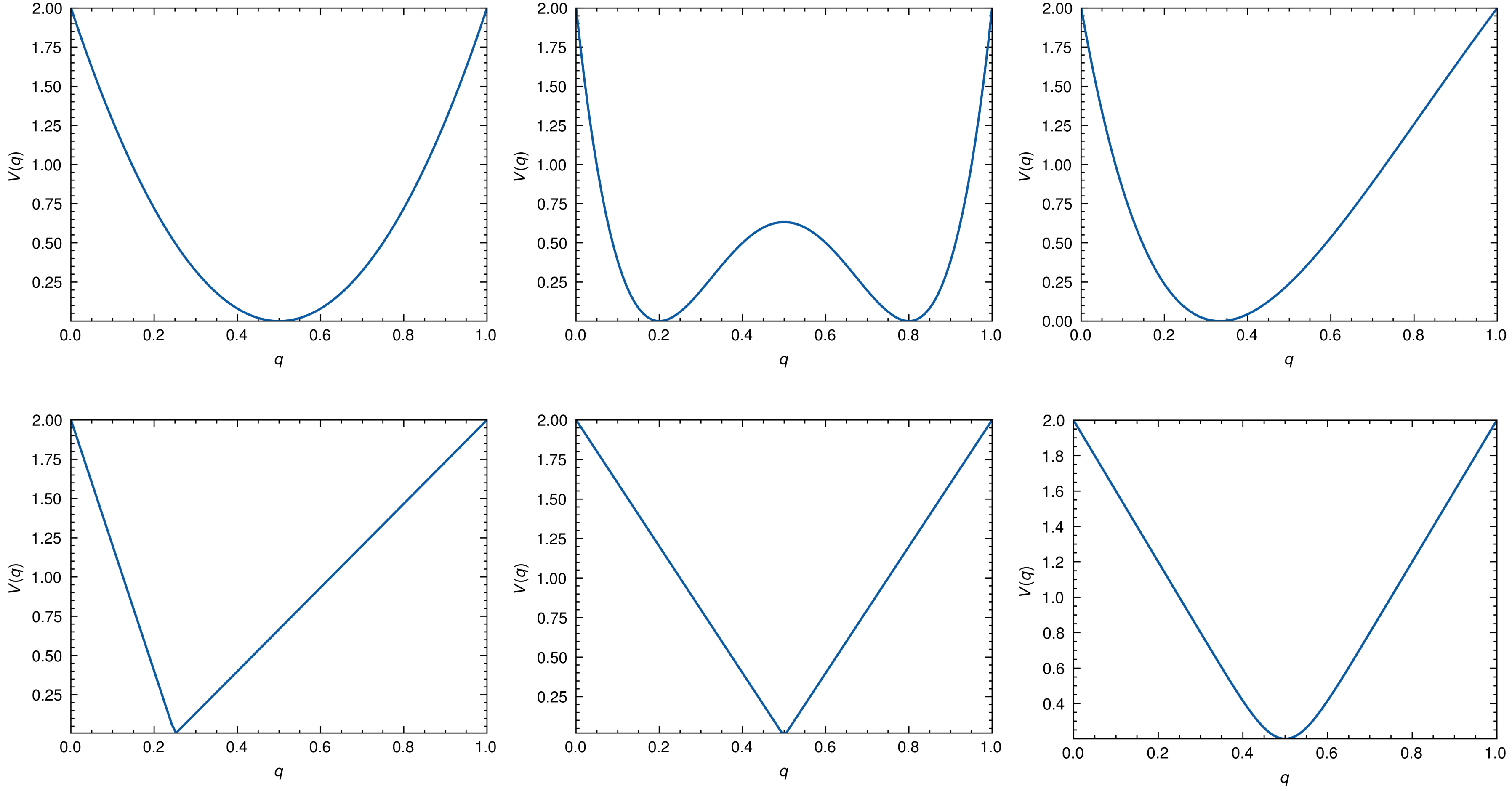


Figure 3: (Top) SHO, Double Well, Morse, (Bottom) ATW, STW and SSTW potential functions

Results (Test Dataset)

- Test for 8,000 potentials
- Generate labels with Kahan-Li 8th order symplectic integrator (**KL8**) ($\Delta t = 10^{-4}$)
- Use $\Delta t = 2 \times 10^{-2}$ for all models
- Compare with numerical solvers
  - **Y4**: Yoshida 4th order symplectic integrator (Symplectic)
  - **RK4**: Runge-Kutta 4th order integrator
  - **GL4**: Gauss-Legendre 4th order integrator (Implicit)

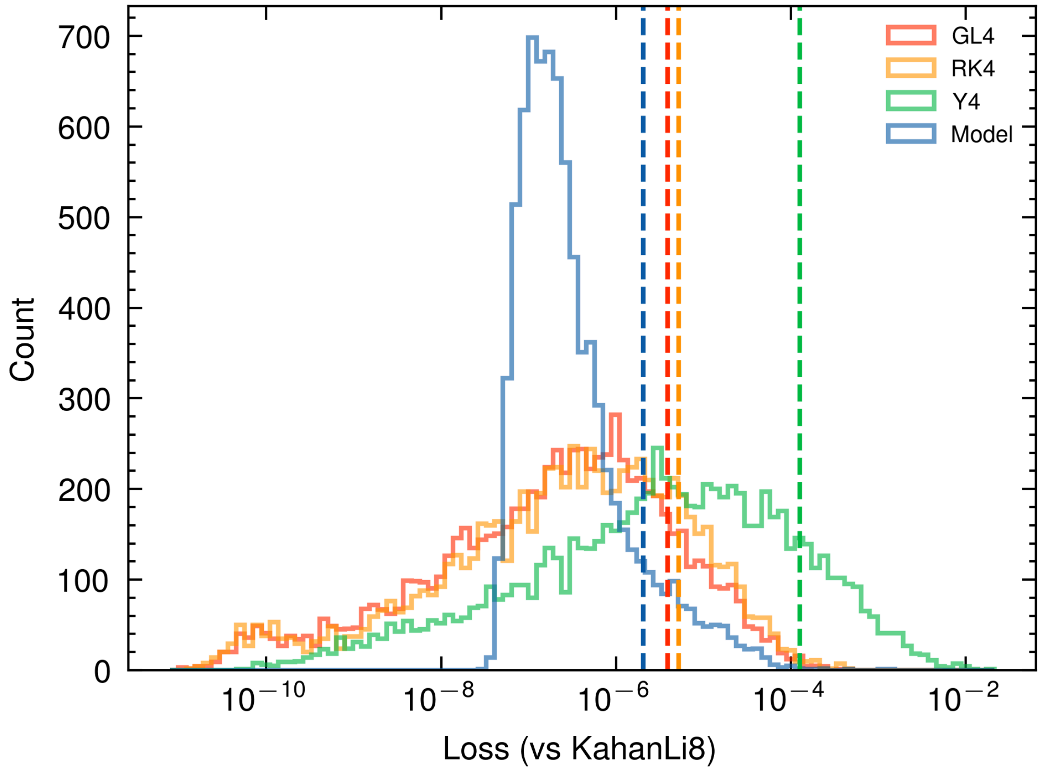


Figure 4: Loss histogram for the test dataset with MambONet

Results (Test Dataset)

- Test for 8,000 potentials
- Generate labels with Kahan-Li 8th order symplectic integrator (**KL8**) ($\Delta t = 10^{-4}$)
- Use $\Delta t = 2 \times 10^{-2}$ for all models
- Compare with numerical solvers
  - **Y4**: Yoshida 4th order symplectic integrator (Symplectic)
  - **RK4**: Runge-Kutta 4th order integrator
  - **GL4**: Gauss-Legendre 4th order integrator (Implicit)

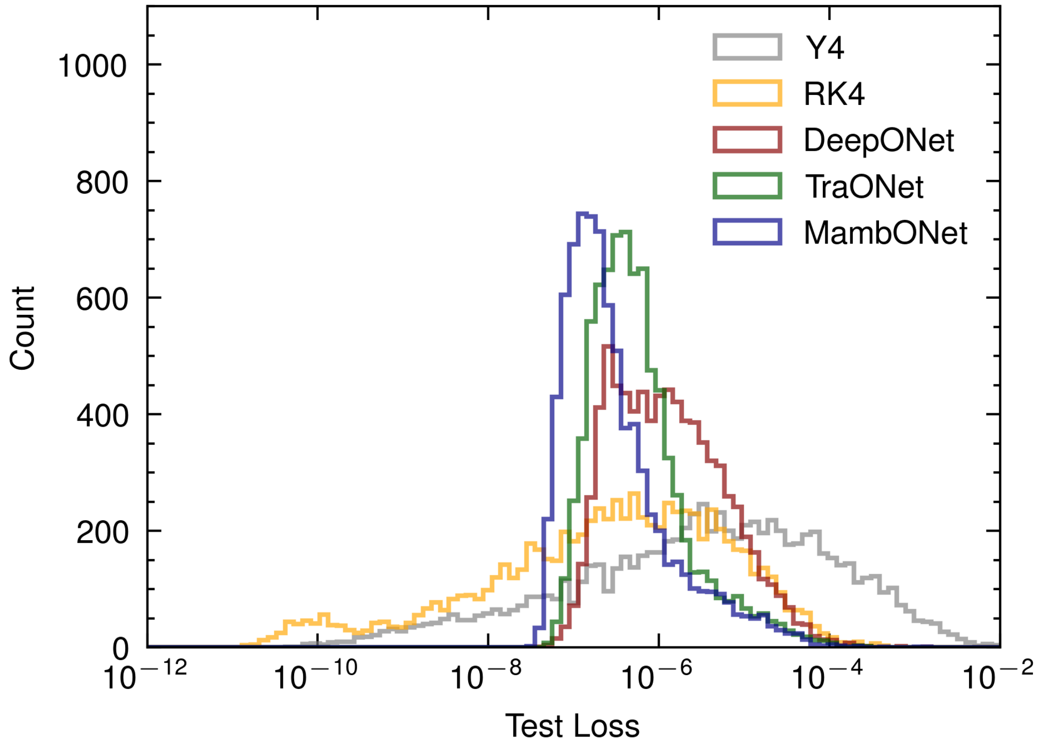


Figure 4: Loss histogram for the test dataset

Results (SHO)

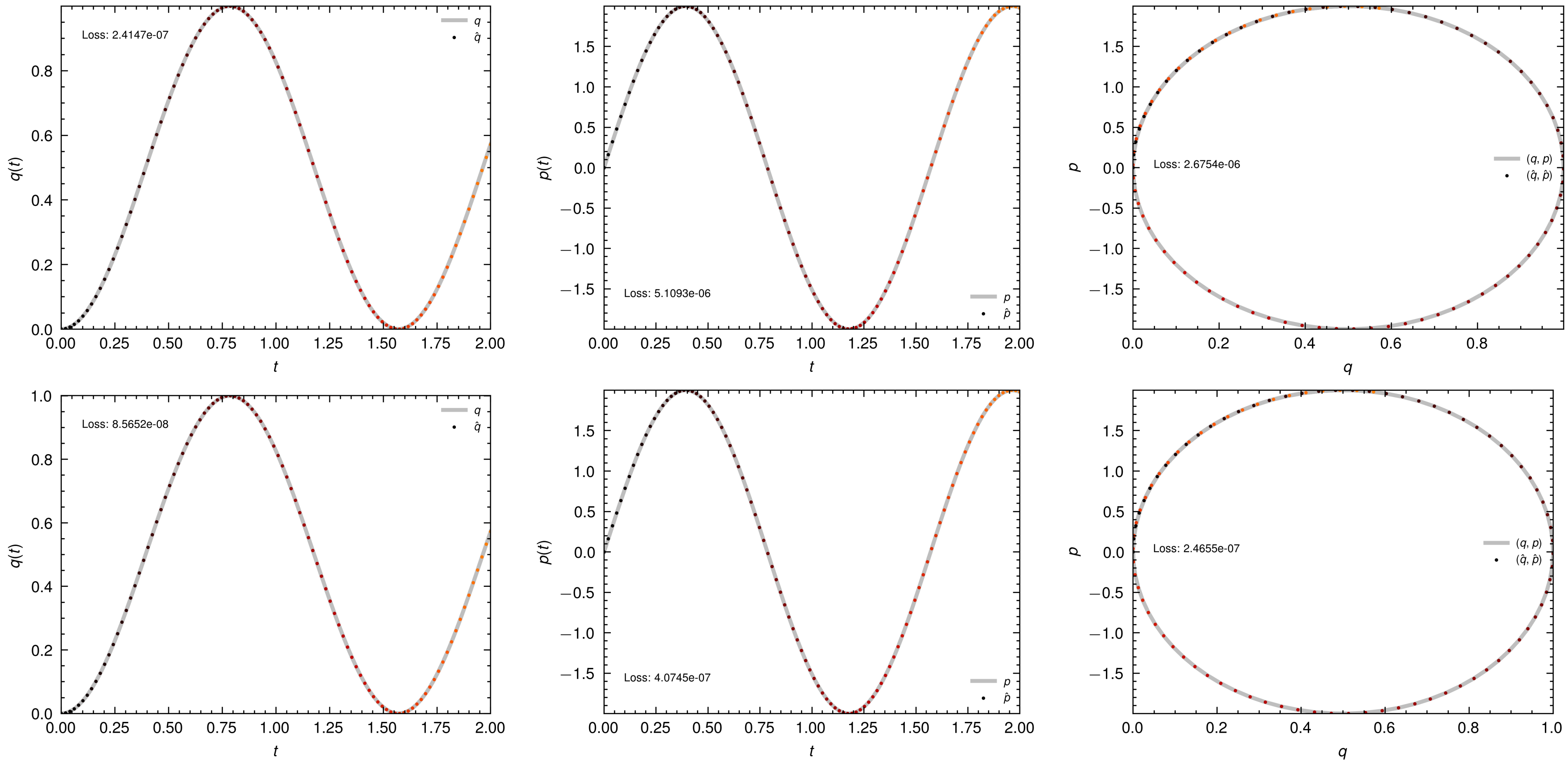


Figure 5: Comparison of the predicted trajectory of the SHO potential function by DeepONet (Top) and MambONet (Bottom)

Results (Double Well)

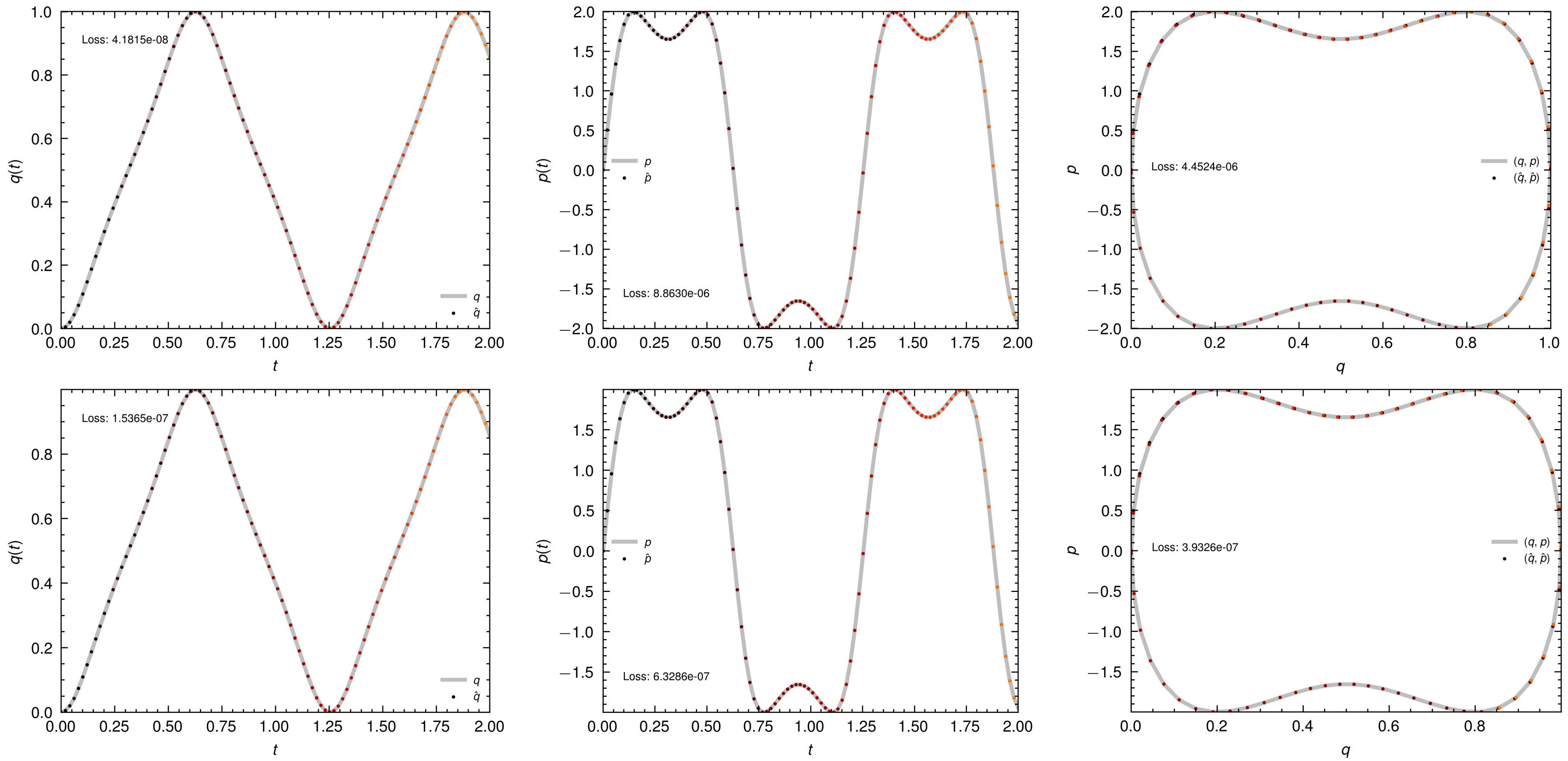


Figure 6: Comparison of the trajectory of the double well potential function by DeepONet (Top) and MambONet (Bottom)

Results (ATW)

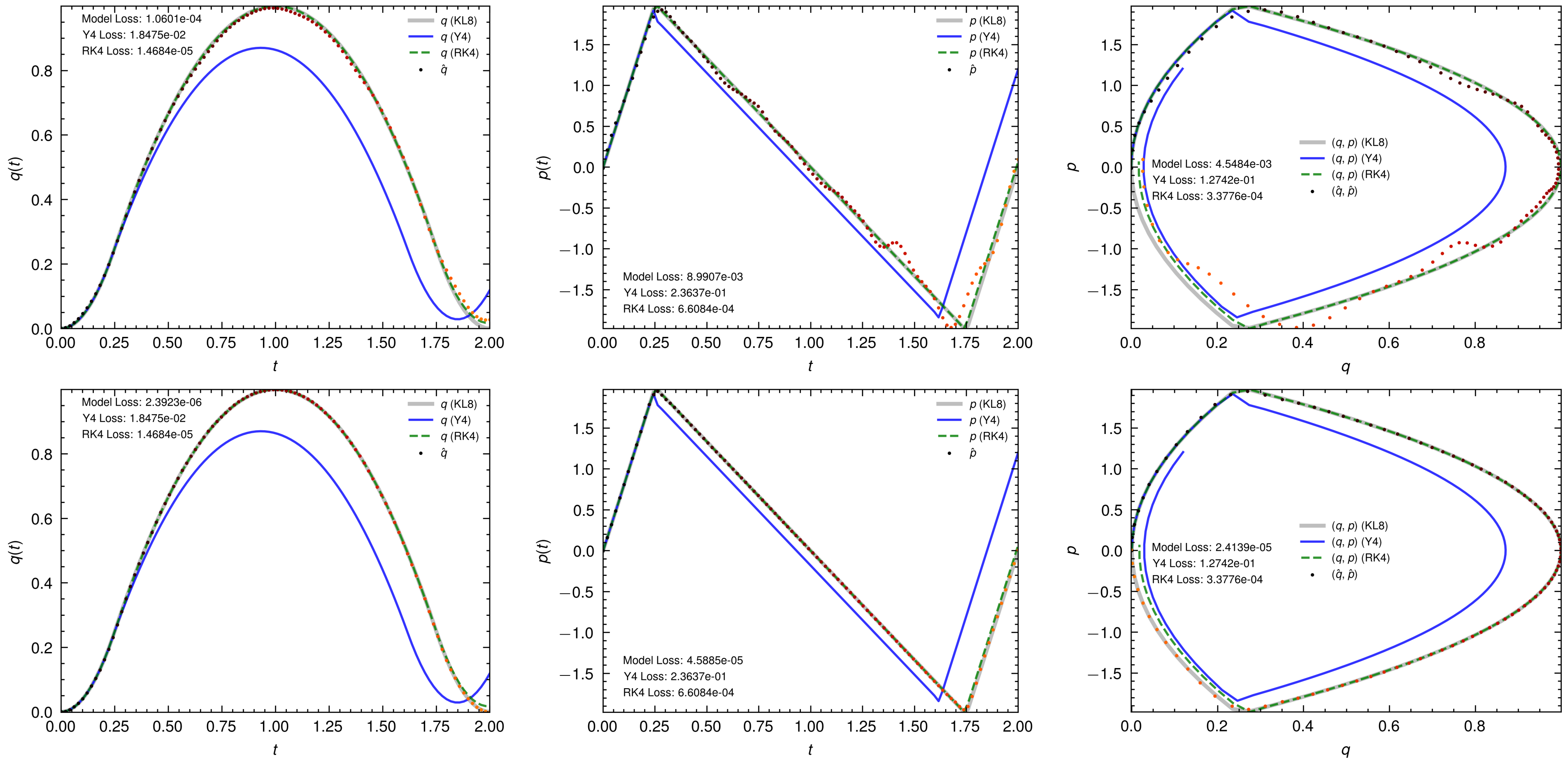


Figure 7: Comparison of the trajectory of the ATW potential function by DeepONet (Top) and MambONet (Bottom)

Where can we apply this?

Operator Learning for Primordial Black Holes

Tae-Geun Kim   Hyunjoo Jung   Jeonghwan Park   **Min Gi Park** [†]   Seong Chan Park[‡]   Yeji Park

Department of Physics Yonsei University Seoul 03722, Republic of Korea

[†]**Speaker**   [‡]Advisor

YONSEI HEP-COSMO
High Energy Physics and Cosmology

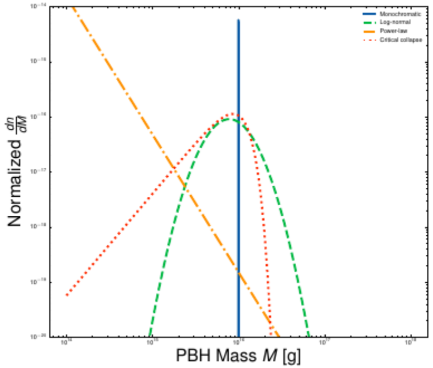
Abstract

We construct both the Hawking forward operator $\mathcal{H}$ and its inverse $\mathcal{H}^{-1}$ using machine learning. By employing operator learning (DeepONet and its variants), we map PBH mass functions $\psi(M)$ to composite secondary spectra $\Phi(E)$ and train the inverse mapping from $\Phi(E)$ back to $\psi(M)$. This ML-based framework enables fast and accurate forward predictions, stable inversions, and naturally supports extended (non-monochromatic) PBH mass distributions.

Primordial Black Hole

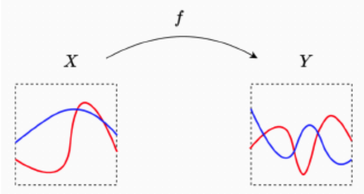
Primordial black holes (PBHs) may form from large density fluctuations in the early Universe, producing diverse mass distributions dn/dM depending on the formation mechanism.

- **Log-normal** :
  $$\frac{dn}{dM} \propto \frac{1}{\sqrt{2\pi}\sigma M^2} \exp\left(-\frac{\ln^2(M/M_c)}{2\sigma^2}\right)$$
- **Power-law** :
  $$\frac{dn}{dM} \propto M^{\gamma-2}, \quad \left(\gamma = \frac{-2w}{1+w}\right)$$
- **Critical collapse** :
  $$\frac{dn}{dM} \propto M^{1.85} \exp\left(-\left(\frac{M}{M_f}\right)^{2.85}\right)$$



Operator Learning

Operator learning aims to approximate a mapping between function spaces, unlike traditional machine learning. The following theorem guarantees that continuous nonlinear operators can be approximated by a neural network of a specific form.



- **Universal Approximation Theorem for Operators**:
  Let X be a Banach space, and let $K_1 \subset X$, $K_2 \subset \mathbb{R}^d$ be compact sets. Let V be a compact subset of $C(K_1)$, and let $G : V \to C(K_2)$ be a continuous (possibly nonlinear) operator. Then, for any $\epsilon > 0$, there exist integers m, p, continuous functions $g : \mathbb{R}^m \to \mathbb{R}^p$ and $f : \mathbb{R}^d \to \mathbb{R}^p$, and sample points $x_1, \ldots, x_m \in K_1$ such that the approximation
  $$|G(u)(y) - \langle g(u(x_1), \ldots, u(x_m)),\ f(y)\rangle| < \epsilon$$
  holds for all $u \in V$ and $y \in K_2$.

Where can we apply this? (Hawking Operator)

- Total photon flux is defined by convolution of the single secondary photon flux and the mass function:

$$\left(\frac{\mathrm{d}^2 N_\gamma^{\mathrm{tot}}}{\mathrm{d}E\mathrm{d}t}\right)_\psi = \int_{M_{\min}}^{M_{\max}} \frac{\mathrm{d}^2 N_\gamma^{\mathrm{sec}}}{\mathrm{d}E\mathrm{d}t} \psi(M)\mathrm{d}M$$

$$\int_{M_{\min}}^{M_{\max}} \psi(M)\mathrm{d}M = 1$$

- If we fix $M_{\min}$ and $M_{\max}$, then this can be expresses as the liner operator:

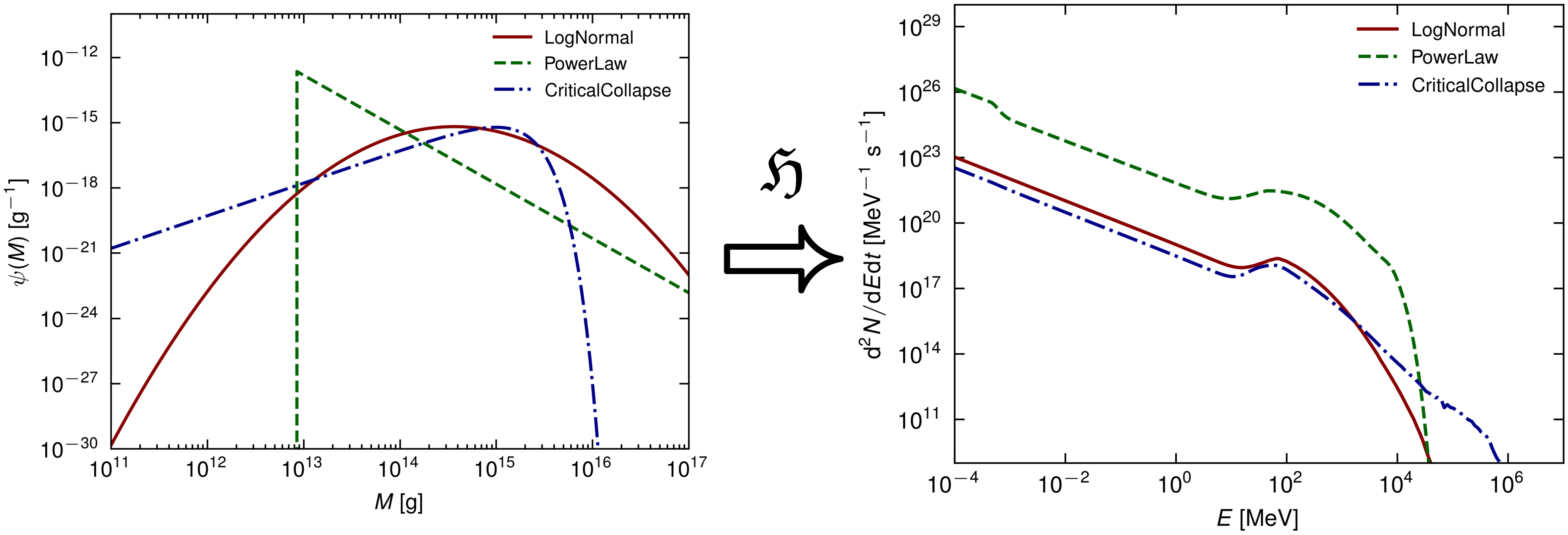
$$\mathfrak{H} : \psi(M) \xrightarrow{\mathfrak{H}} \left(\frac{\mathrm{d}^2 N_\gamma^{\mathrm{tot}}}{\mathrm{d}E\mathrm{d}t}\right)_\psi$$

We call this operator the **Hawking Operator**.

Where can we apply this? (Hawking Operator)

$$\psi(M)$$

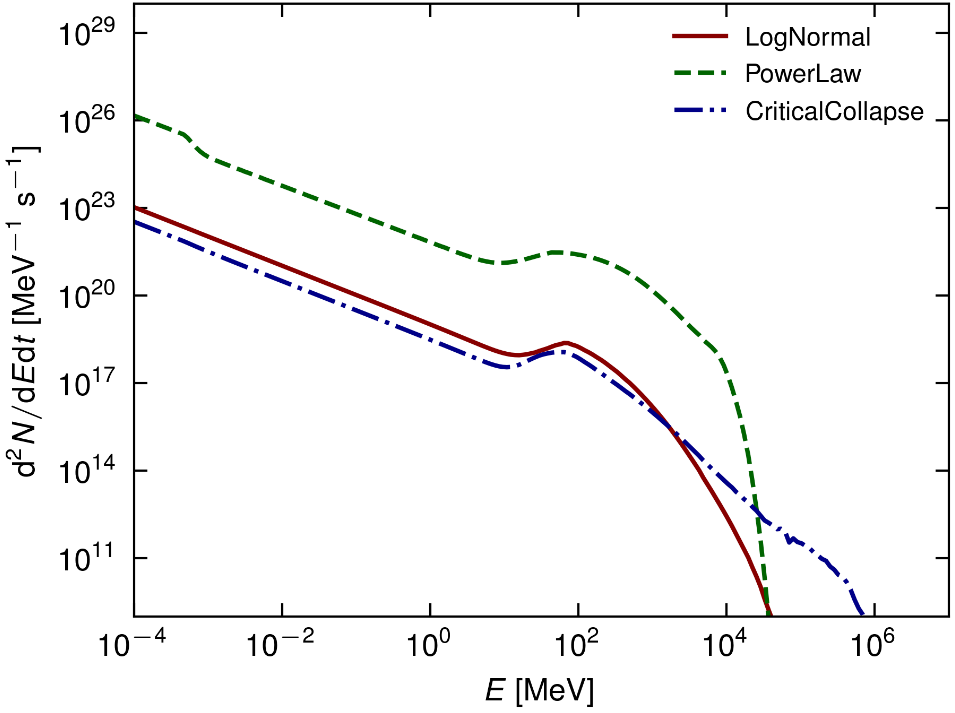
$$\frac{\mathrm{d}^2 N_\gamma^{\mathrm{total}}}{\mathrm{d}E_\gamma \mathrm{d}t}$$



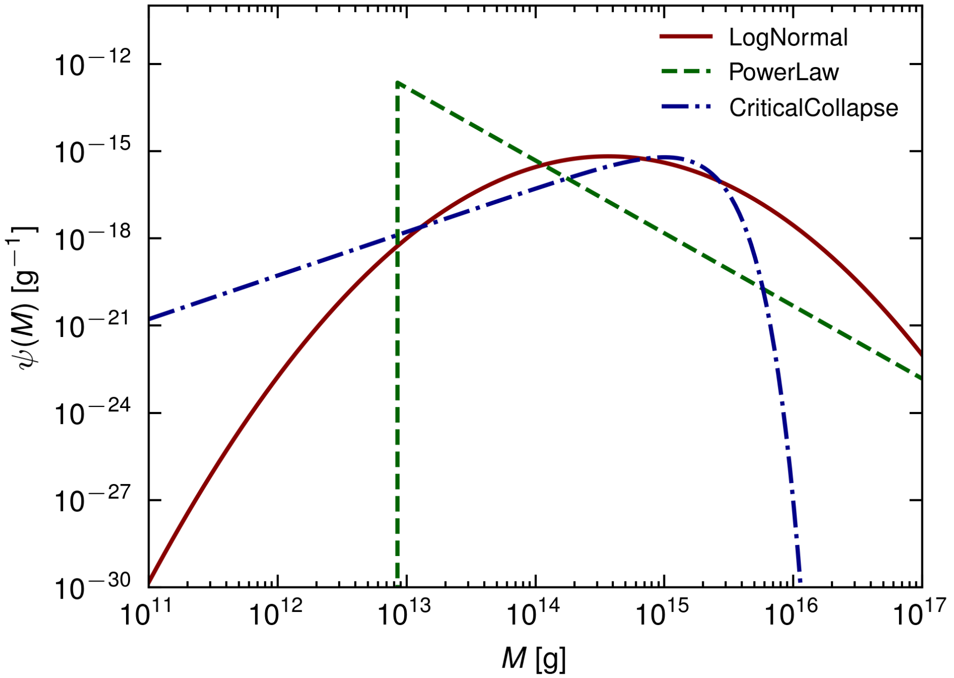
Where can we apply this? (Hawking Operator)

$$\frac{\mathrm{d}^2 N_\gamma^{\mathrm{total}}}{\mathrm{d}E_\gamma \mathrm{d}t}$$

$$\psi(M)$$



$$\mathfrak{H}^{-1}$$



Summary & Conclusion

- We show that (a class of) Hamiltonian Mechanics can be formulated by an operator.
- Using operator learning, AI can learn this operator.
  - Introduce new architectures: **TraONet & MambONet**
  - Develop new data generation algorithm: **GRF + Cubic B-Spline**
- We expect that operator learning can be applied to various physics problems
  - e.g. *Photon spectrum from Primordial Black Holes and vice-versa*

Summary & Conclusion

- We show that (a class of) Hamiltonian Mechanics can be formulated by an operator.
- Using operator learning, AI can learn this operator.
  - Introduce new architectures: **TraONet & MambONet**
  - Develop new data generation algorithm: **GRF + Cubic B-Spline**
- We expect that operator learning can be applied to various physics problems
  - e.g. *Photon spectrum from Primordial Black Holes and vice-versa*

AI can learn (a class of) Hamiltonian Mechanics!

Supplements

Operator formulation of Hamilton's equations

- Let denote $x(t) = [q(t), p(t)]^T$ then we can rewrite the Hamilton's equation as

$$\begin{cases} \dot{q} = \frac{\partial H}{\partial p} \\ \dot{p} = -\frac{\partial H}{\partial q} \end{cases} \implies \dot{x} = J\nabla H(x) \equiv F(x) \quad \text{where } H = \frac{p^2}{2m} + V(q),\ J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

- If F is *locally Lipschitz*, then a local unique solution exists

$\Rightarrow H$ is $C^2(\mathbb{R}^{2n}, \mathbb{R})$

$$x(\Delta t) = x(0) + \int_0^{\Delta t} F(x(\tau))\mathrm{d}\tau$$

- If $x(t)$ is *bounded*, then a global unique solution exists

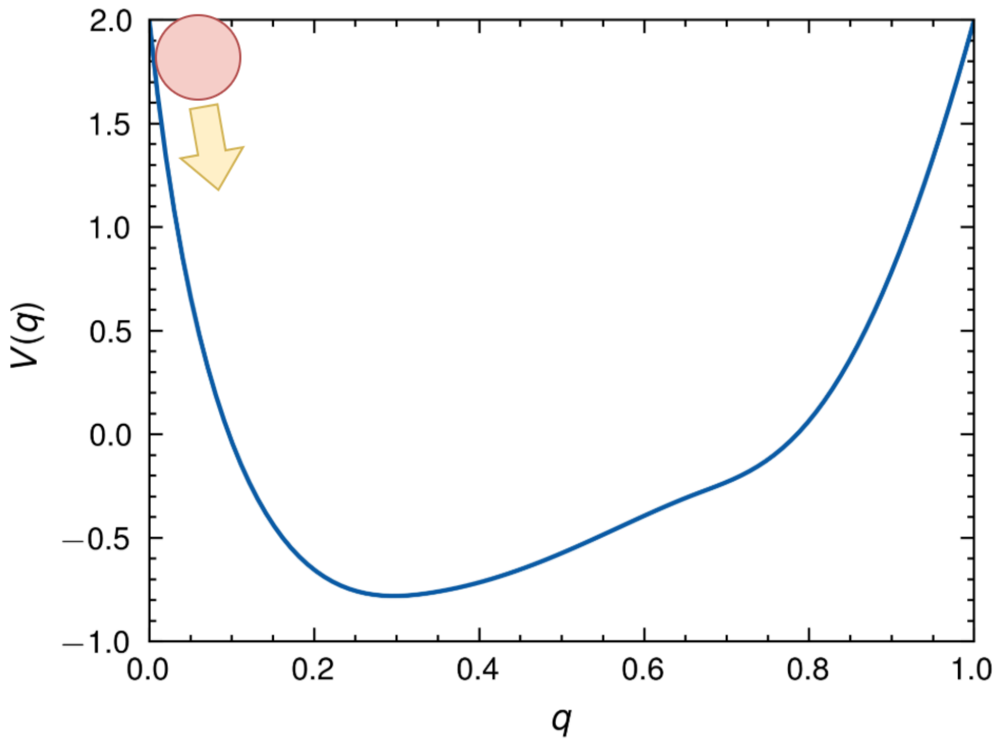
$\Rightarrow V$ is *coercive* or $\{q \in \mathbb{R}^n \mid V(q) \leq E_0\}$ is bounded

$$x(t) = x(0) + \int_0^{t} F(x(\tau))\mathrm{d}\tau$$

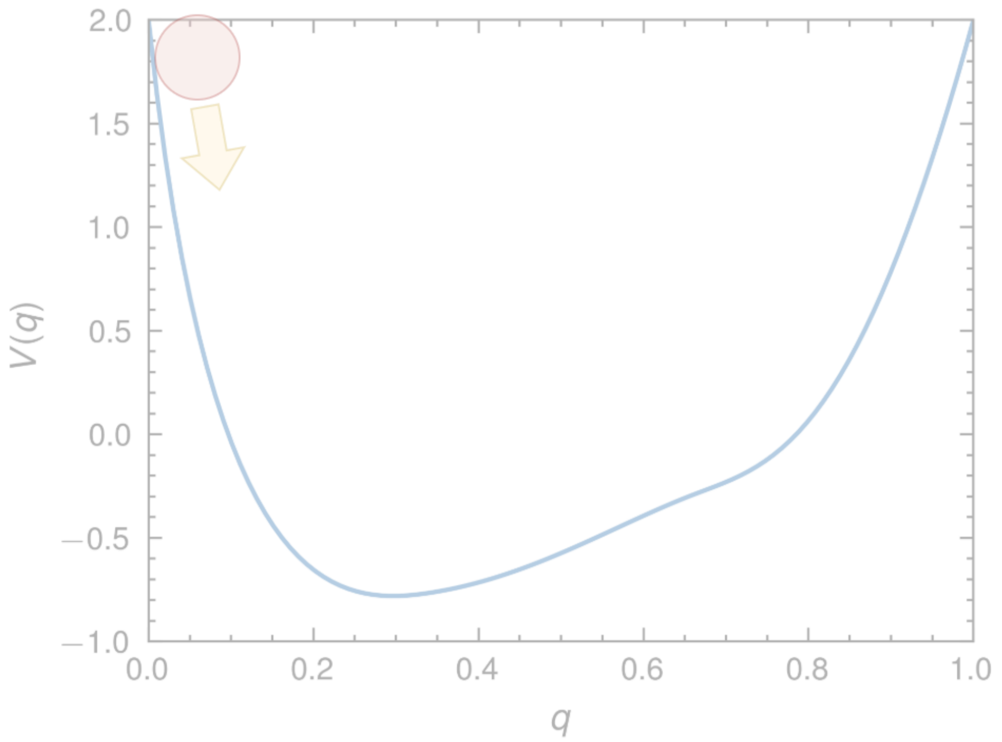
- This can be described as an operator $G : V(q) \mapsto (q(t), p(t))$

$$G(V)(t) = x(t) = \begin{pmatrix} q(t) \\ p(t) \end{pmatrix}$$

Differences with HNN

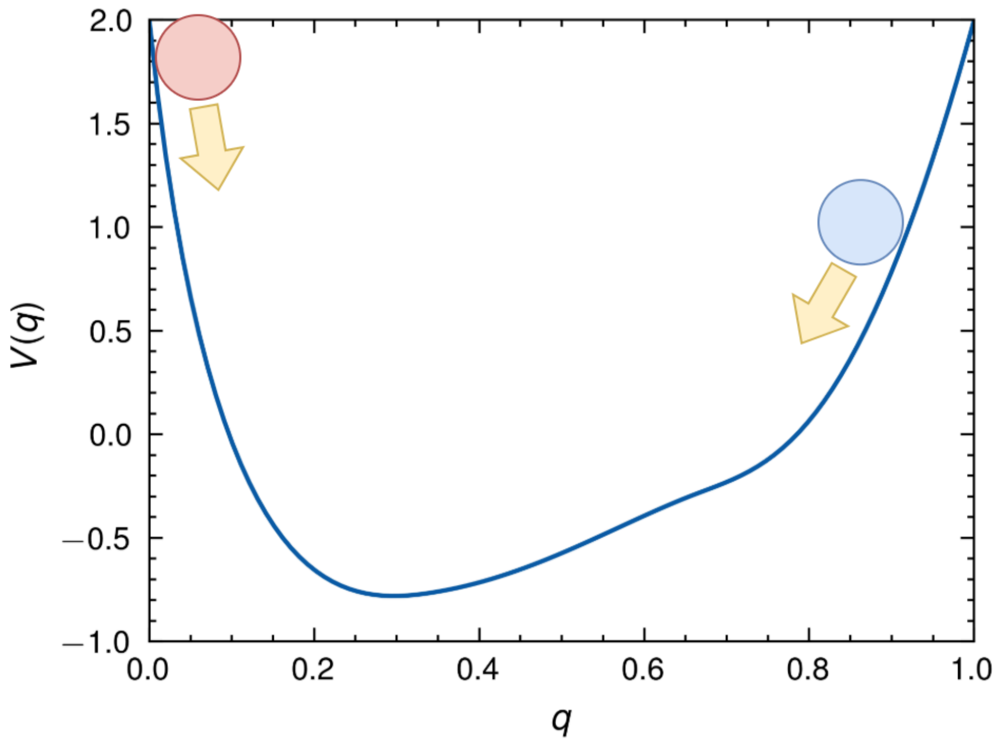


HNN

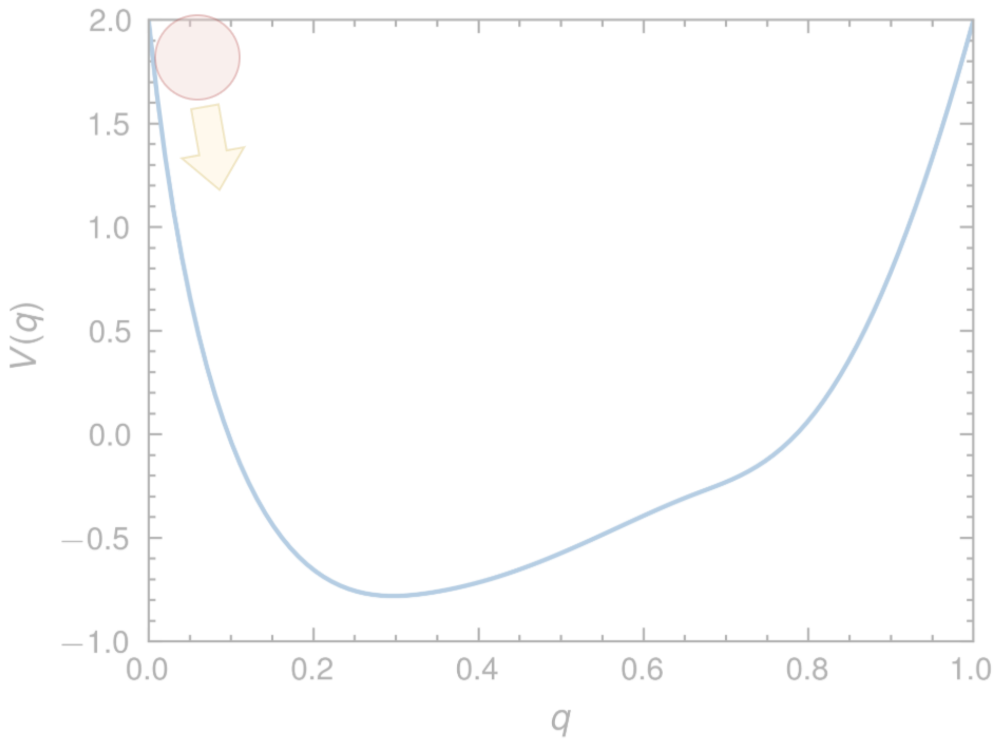


Neural Hamilton

Differences with HNN

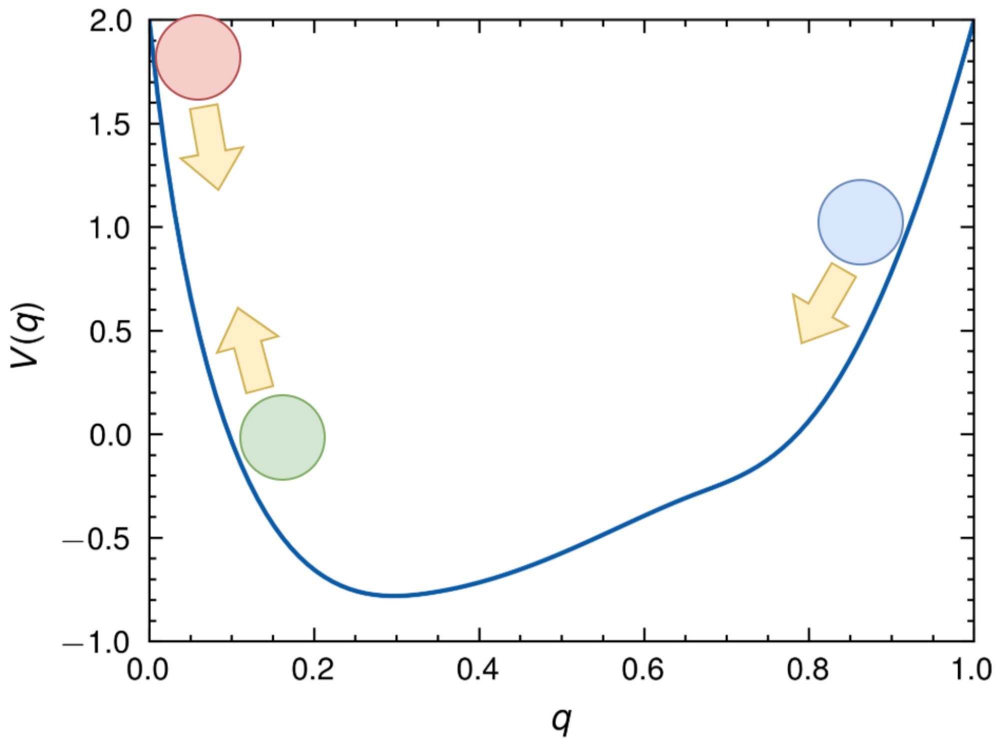


HNN

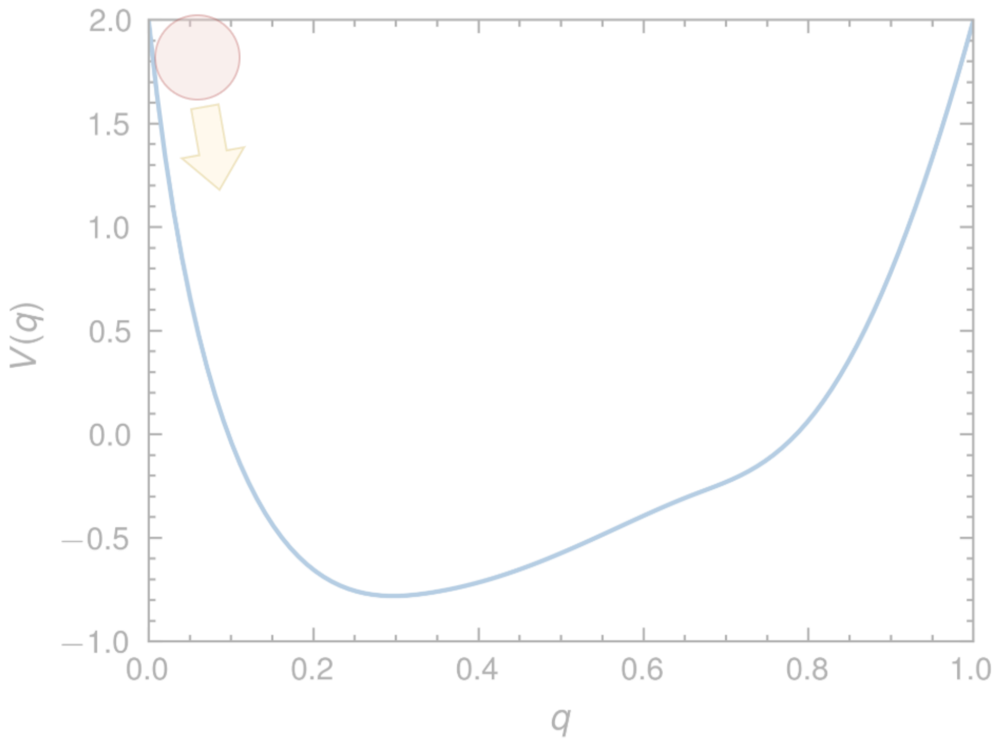


Neural Hamilton

Differences with HNN

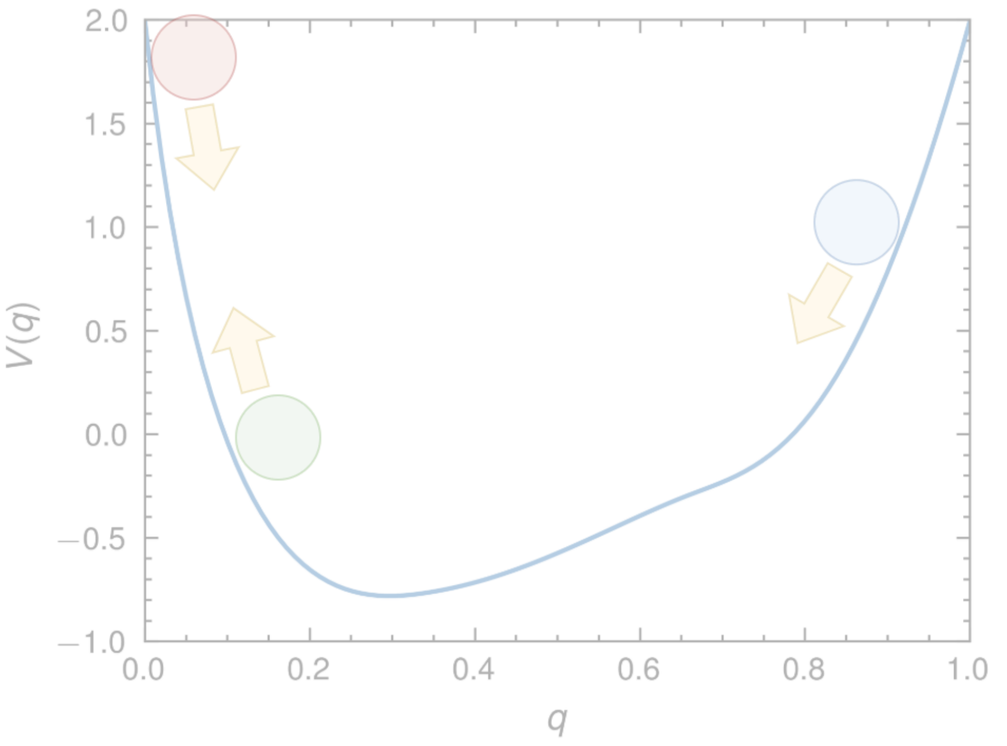


HNN

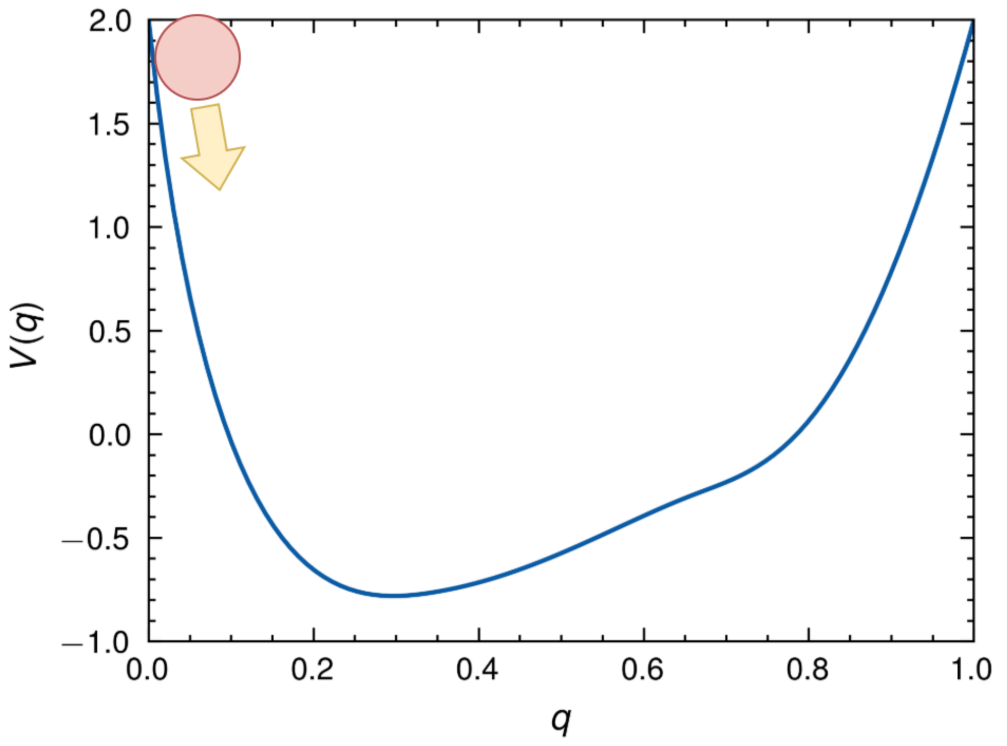


Neural Hamilton

Differences with HNN

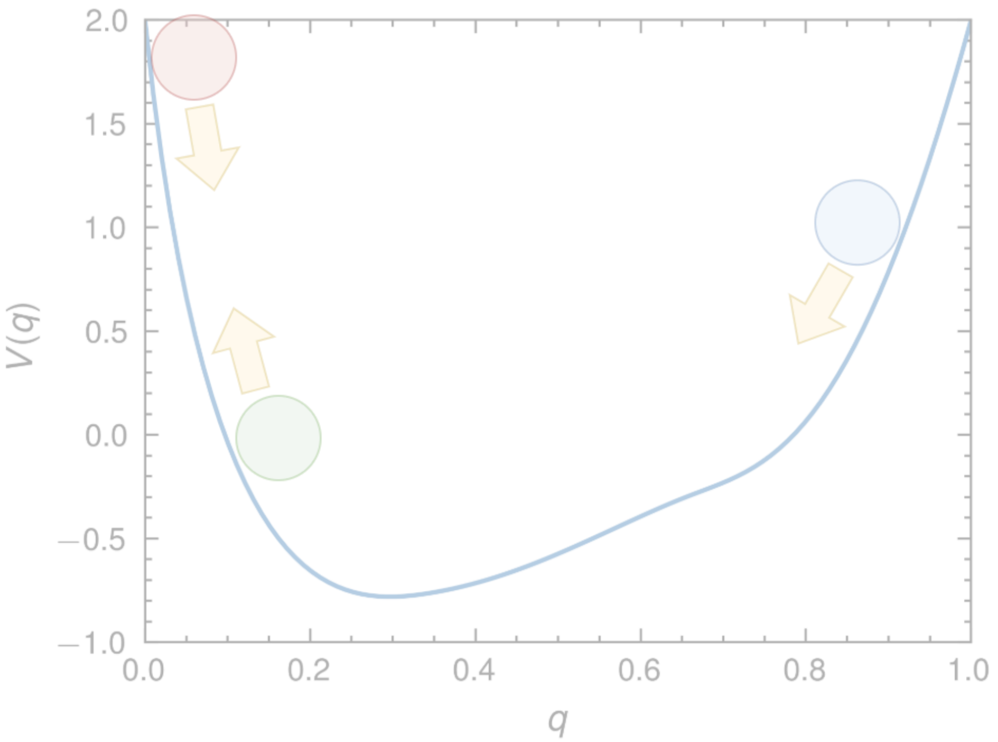


HNN

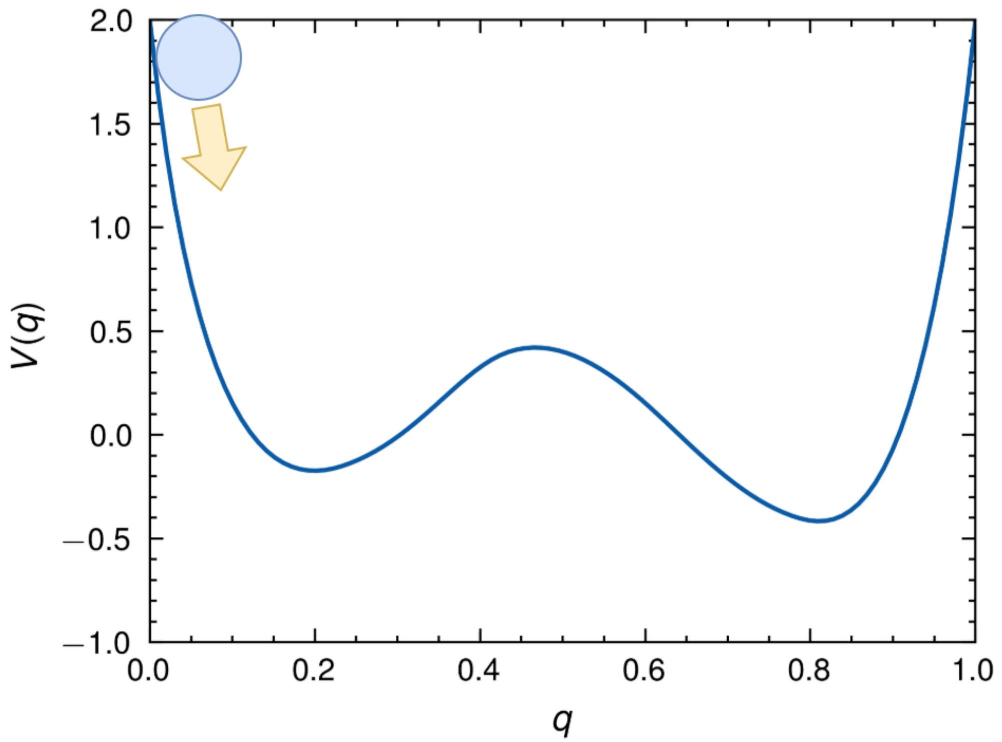


Neural Hamilton

Differences with HNN

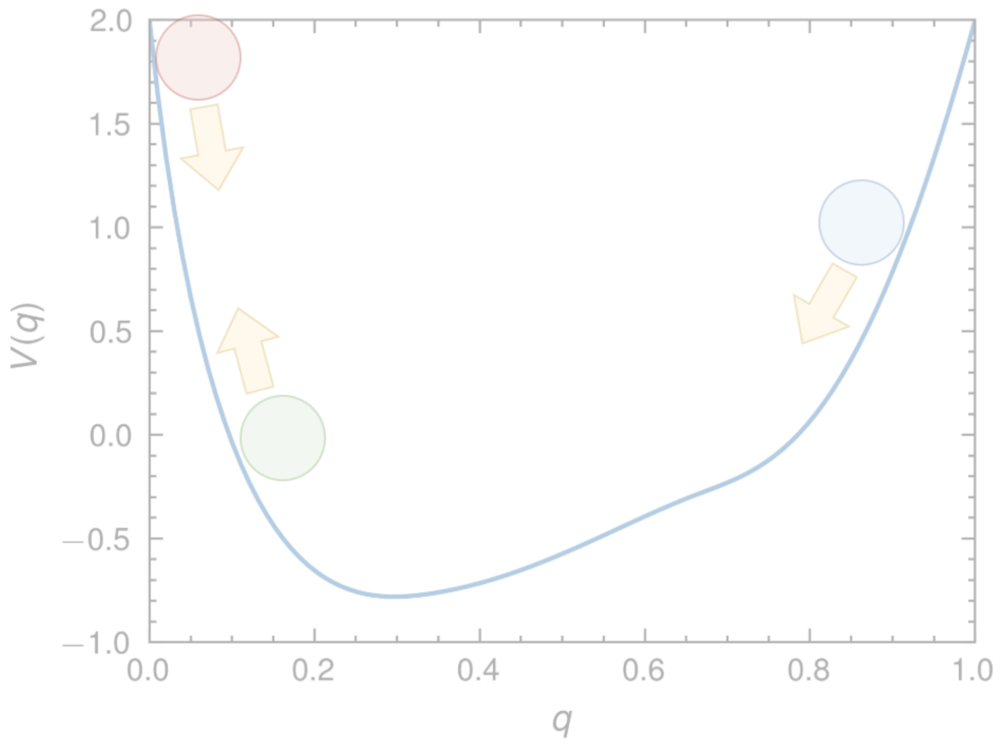


HNN

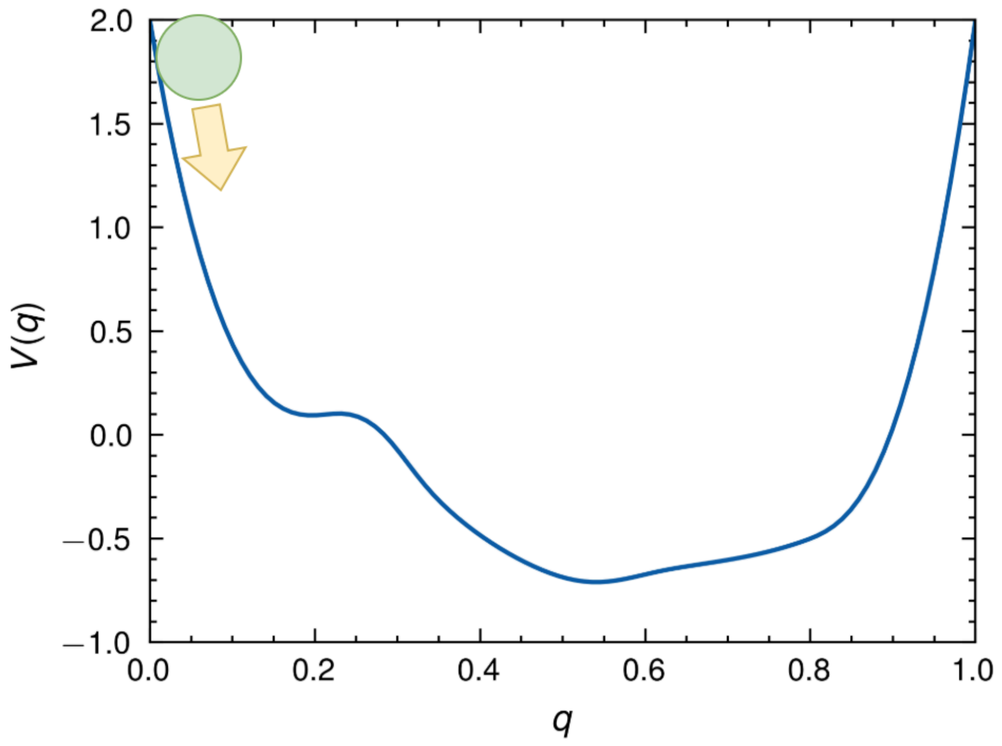


Neural Hamilton

Differences with HNN



HNN



Neural Hamilton

Differences with HNN

Object	HNN	Neural Hamilton
q_0, p_0	Input	Given
$H(q, p)$	Learn	Input
$\dot{q} = \frac{\partial H}{\partial p}, \dot{p} = -\frac{\partial H}{\partial q}$	Given	Learn
$q(t), p(t)$	Output	Output

Figure 8: Comparison of the HNN and Neural Hamilton

Model - TraONet

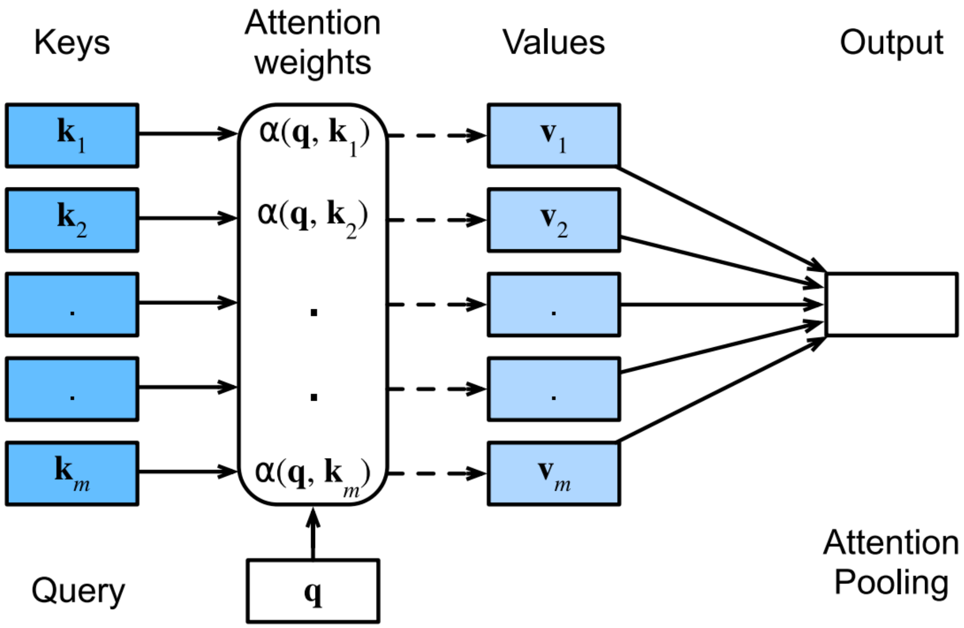


Figure 9: Attention pooling [Zhang et al. (2023)]

- A common strategy for ensuring that the weights sum up to 1 :

$$\alpha(\boldsymbol{q}, \boldsymbol{k}_i) = \frac{\alpha(\boldsymbol{q}, \boldsymbol{k}_i)}{\sum_j \alpha(\boldsymbol{q}, \boldsymbol{k}_j)}$$

- We can pick any function $a(\boldsymbol{q}, \boldsymbol{k})$ and then apply *softmax* to it.

$$\alpha(\boldsymbol{q}, \boldsymbol{k}_i) = \frac{\exp(a(\boldsymbol{q}, \boldsymbol{k}_i))}{\sum_j \exp(a(\boldsymbol{q}, \boldsymbol{k}_j))}$$

- For example, we can use the *dot product* as the function $a(\boldsymbol{q}, \boldsymbol{k})$.

$$a(\boldsymbol{q}, \boldsymbol{k}_i) = \frac{\boldsymbol{q}^T \boldsymbol{k}_i}{\sqrt{d}}$$

How to effectively cover the potential space?

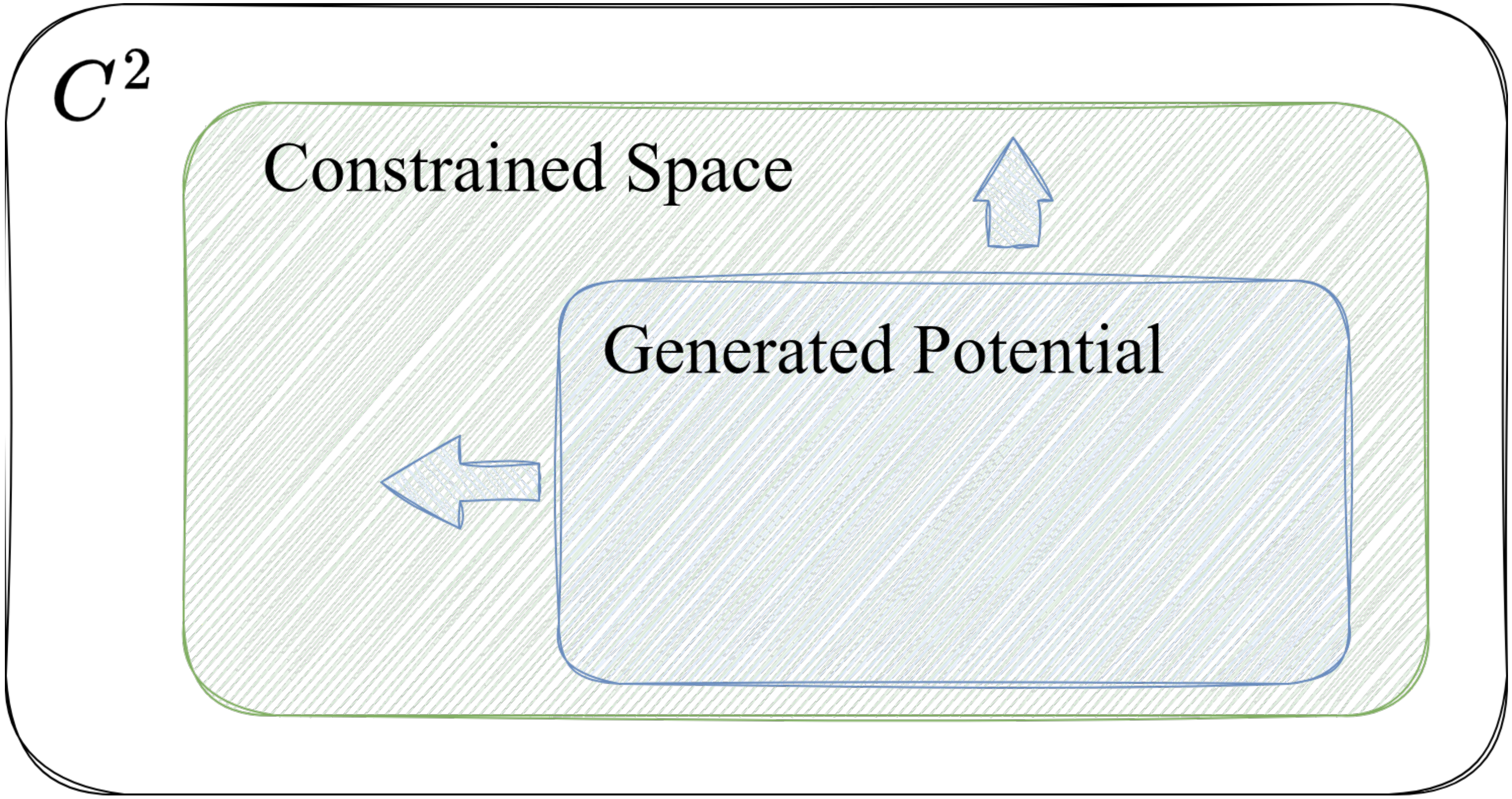


Figure 10: Illustration of the potential function space

How to effectively cover the potential space?

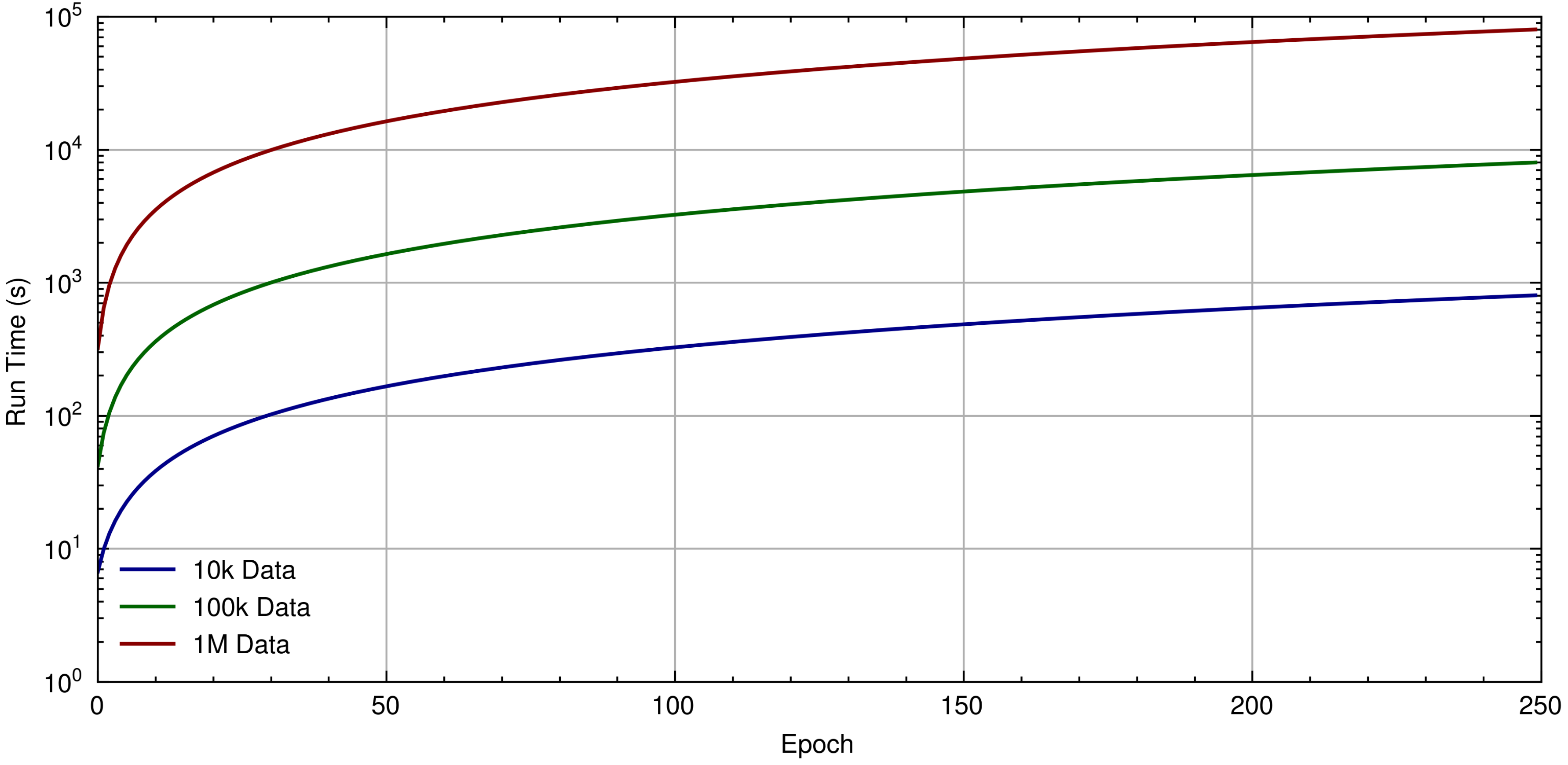


Figure 11: Comparison of the training time among the different numbers of potentials

How to effectively cover the potential space?

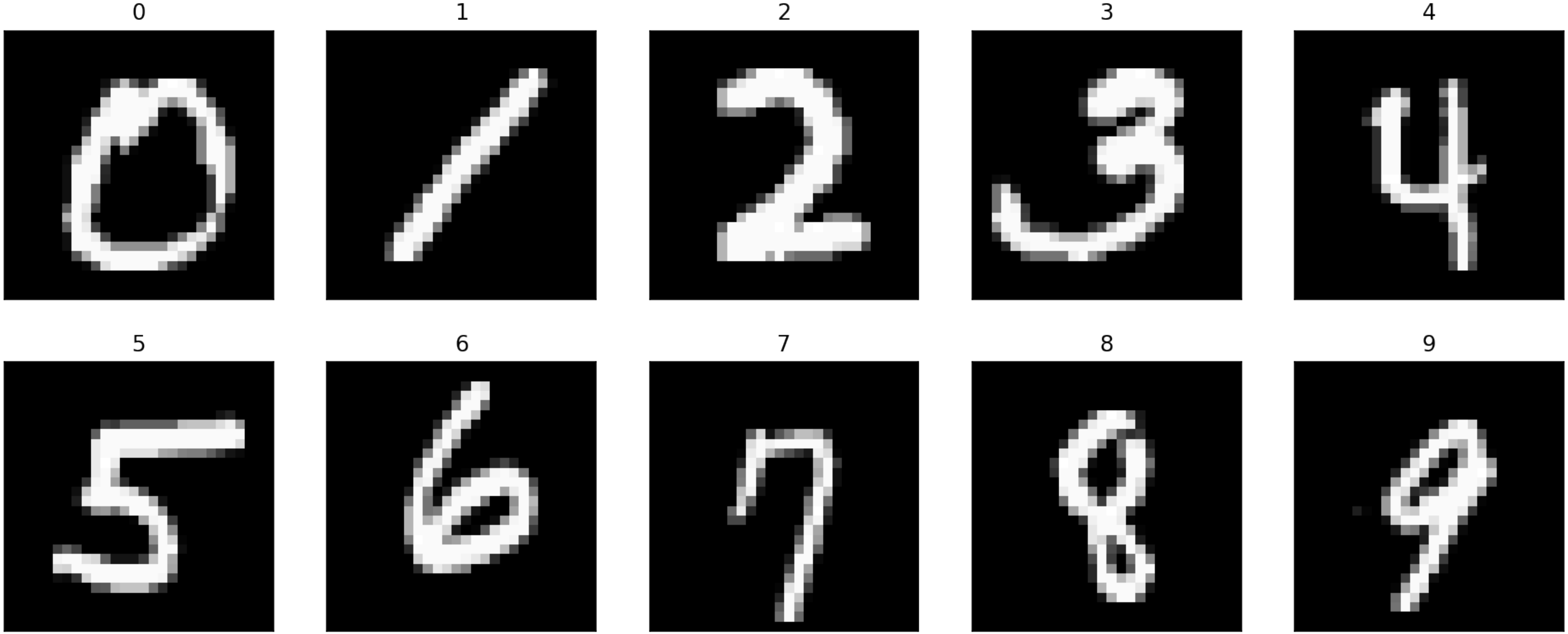


Figure 12: MNIST dataset [LeCun et al., Proceedings of the IEEE (1998)]

How to effectively cover the potential space?

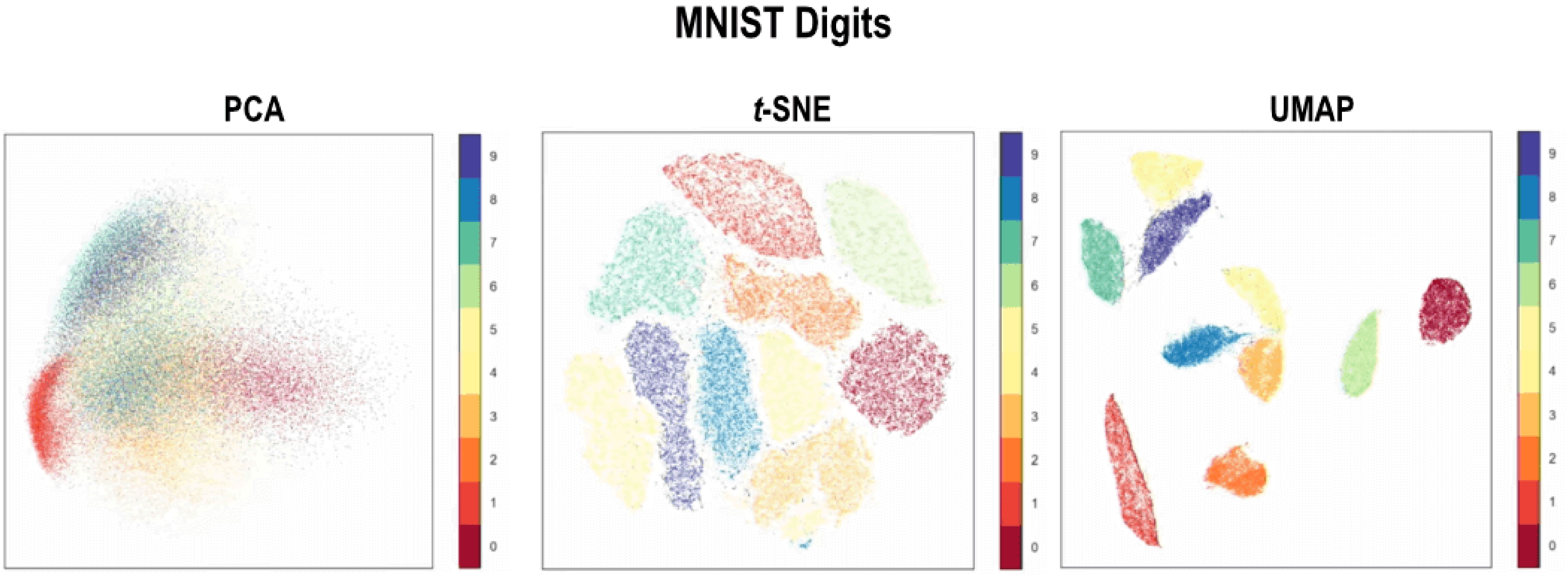


Figure 13: Comparison of the PCA, t-SNE and UMAP [Capershire Meta (2021)]

How to effectively cover the potential space?

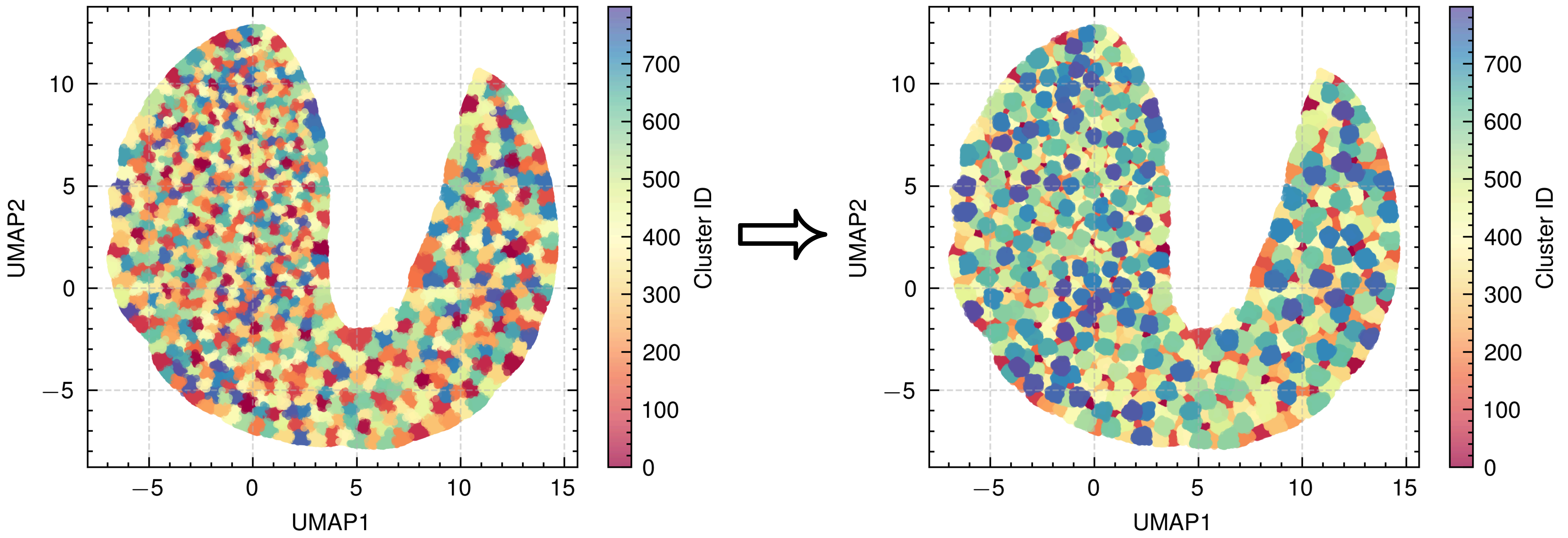


Figure 14: Comparison of the UMAP projection of the 800k dataset (Left) and the sampled 80k dataset (Right)

How to effectively cover the potential space?

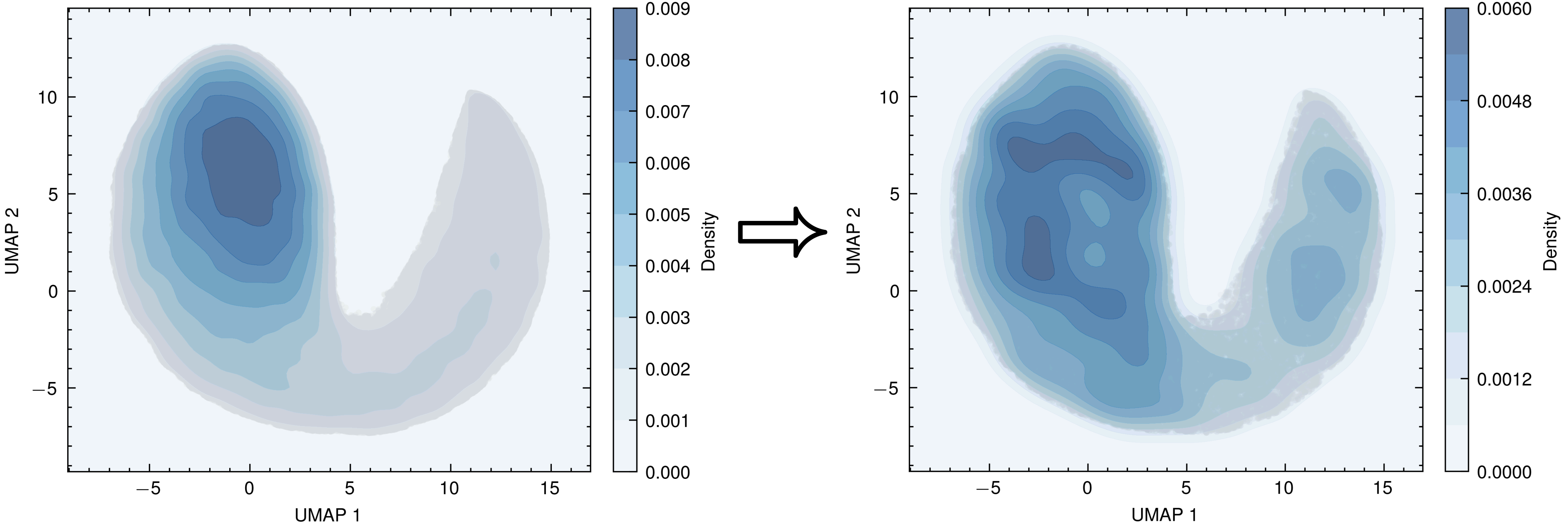


Figure 15: Comparison of the UMAP projection of the 800k dataset (Left) and the sampled 80k dataset (Right)

How to effectively cover the potential space?

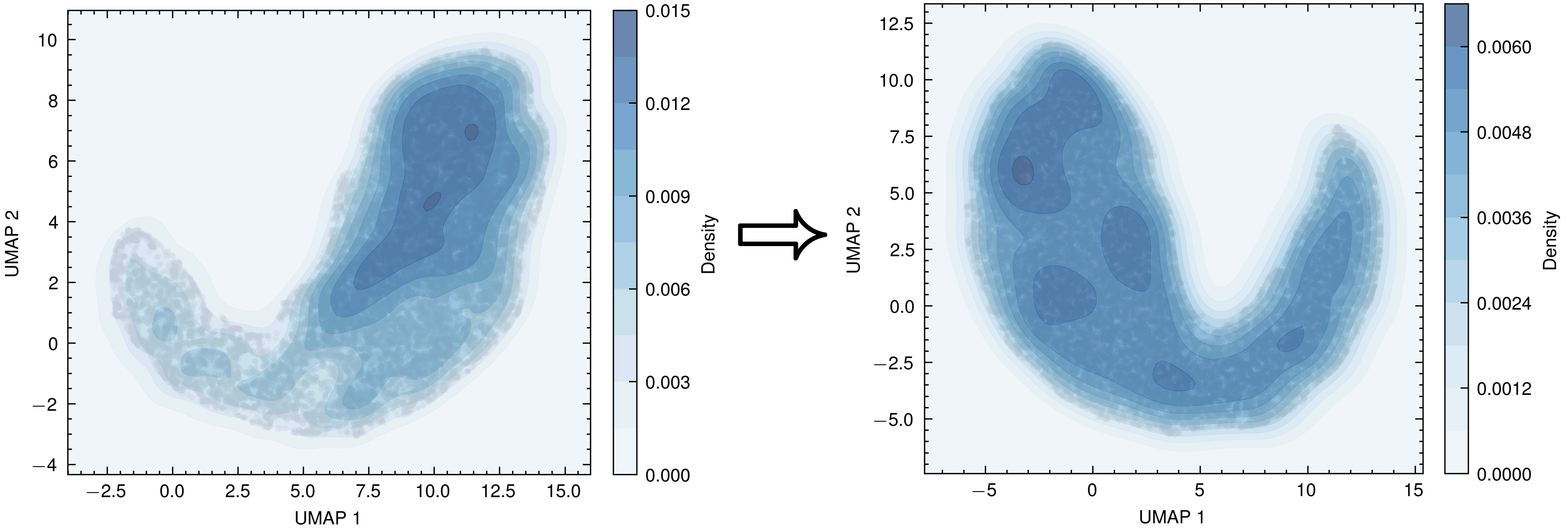
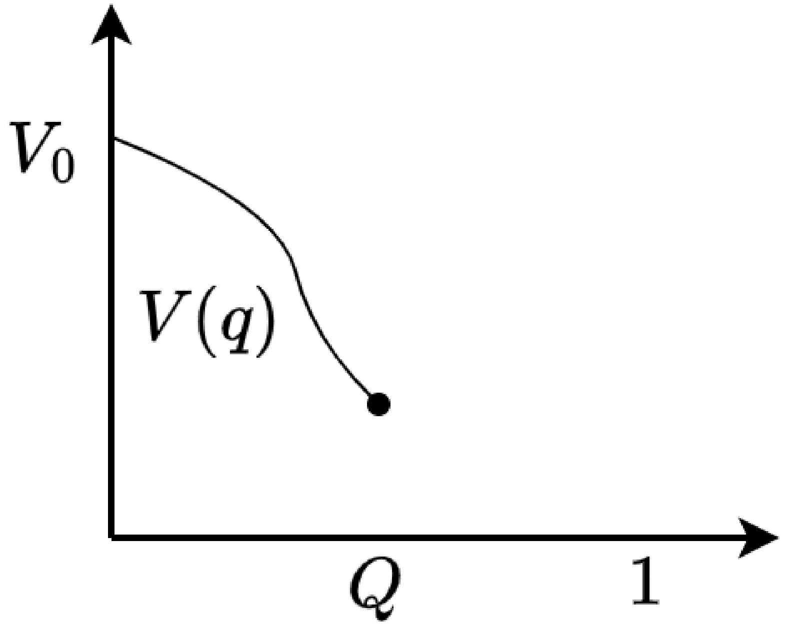


Figure 16: Comparison of the UMAP projections of the 20k dataset w/o filtering (Left) and w/ filtering (Right)

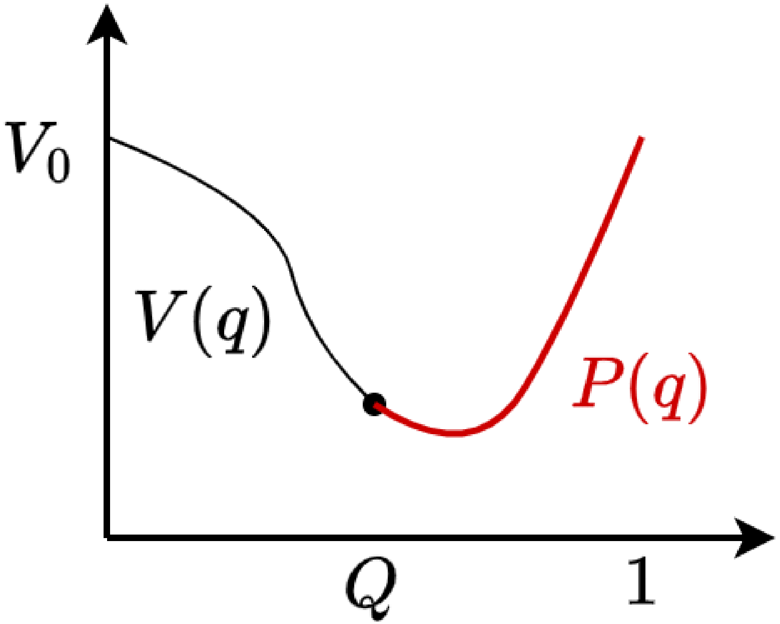
How about unbounded?

- Consider a monotonically decreasing C^2 potential $V(q)$ defined on $[0, Q]$, where $0 < Q < 1$ and $V(0) = V_0$



How about unbounded?

- Consider a monotonically decreasing C^2 potential $V(q)$ defined on $[0, Q]$, where $0 < Q < 1$ and $V(0) = V_0$



- A new C^2 function $P(q)$ on $[Q, 1]$ such that

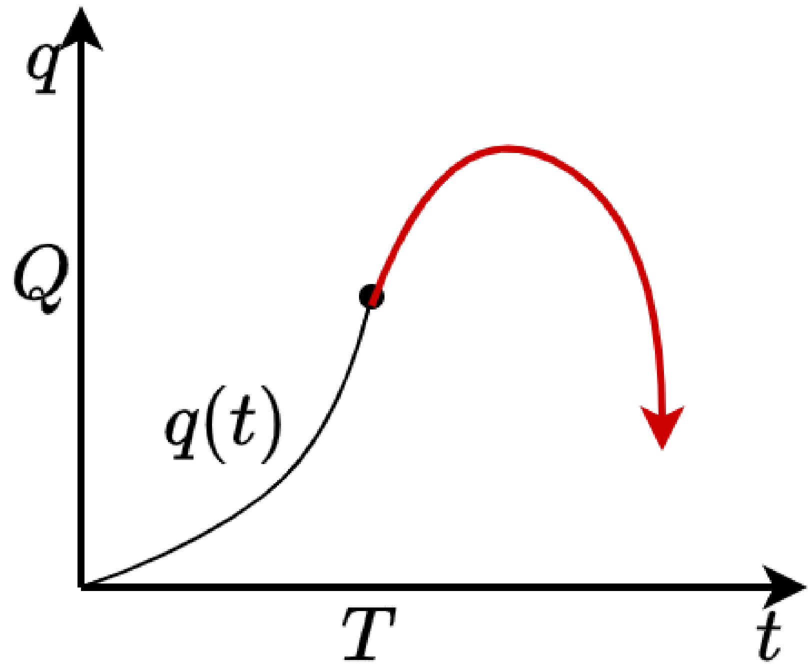
$$P(1) = V_0$$
$$P(Q) = V(Q)$$
$$P'(Q) = V'(Q)$$
$$P''(Q) = V''(Q)$$
$$P(q) < V_0 \quad \text{for} \quad Q < q < 1$$

- Then we can define a new C^2 potential function $\tilde{V}(q)$ as

$$\tilde{V}(q) = \begin{cases} V(q) \text{ if } 0 \leq q \leq Q \\ P(q) \text{ if } Q < q \leq 1 \end{cases}$$

How about unbounded?

- Input new potential function $\tilde{V}(q)$ into the model, then we can get $q(t)$ and $p(t)$



- To extract the relevant dynamics, we determine the time T
- Since $H = \frac{p^2}{2} + V(q) = V_0$, from Hamilton's equation,

$$\frac{\mathrm{d}q}{\mathrm{d}t} = \frac{\partial H}{\partial p} = p = \sqrt{2(V_0 - V(q))}$$

$$\Rightarrow \int_0^T \mathrm{d}t = \int_0^Q \frac{\mathrm{d}q}{\sqrt{2(V_0 - V(q))}}$$

$$\Rightarrow T = \int_0^Q \frac{\mathrm{d}q}{\sqrt{2(V_0 - V(q))}}$$

- Take $q(t)$ and $p(t)$ upto time T

Example: Free-Fall

- Consider a free fall potential: $V(q) = -4(q - 0.5), \quad (0 \leq q \leq 0.5)$ [Answer: $q(t) = 2t^2, p(t) = 4t$]
  - From the previous conditions, we can find a cubic function $P(q) = 32q^3 - 48q^2 + 20q - 2$
  - Obtain the time $T = \int_0^{\frac{1}{2}} \frac{\mathrm{d}q}{\sqrt{2(2 - V(q))}} = \int_0^{\frac{1}{2}} \frac{\mathrm{d}q}{\sqrt{8q}} = 0.5$

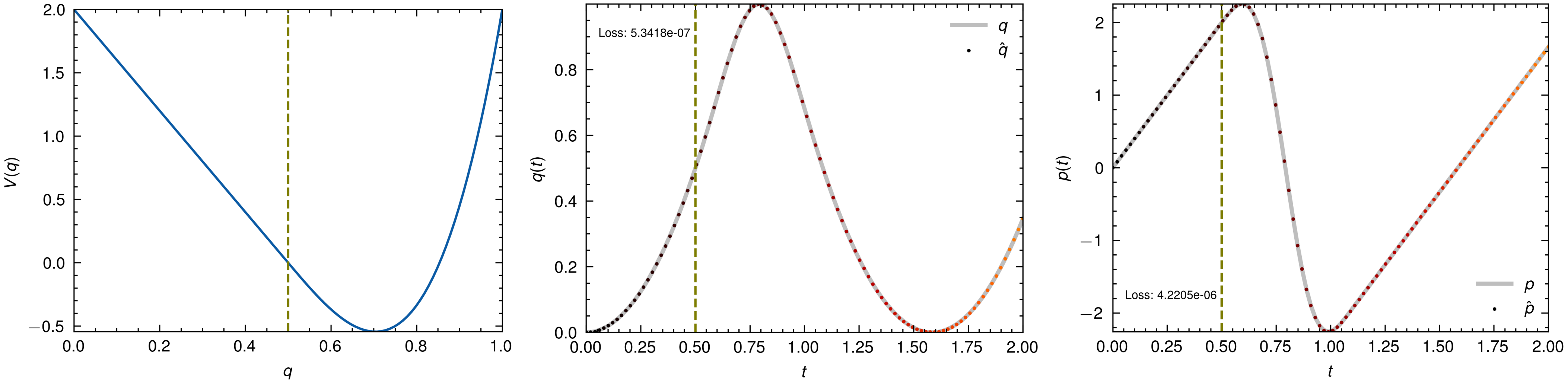
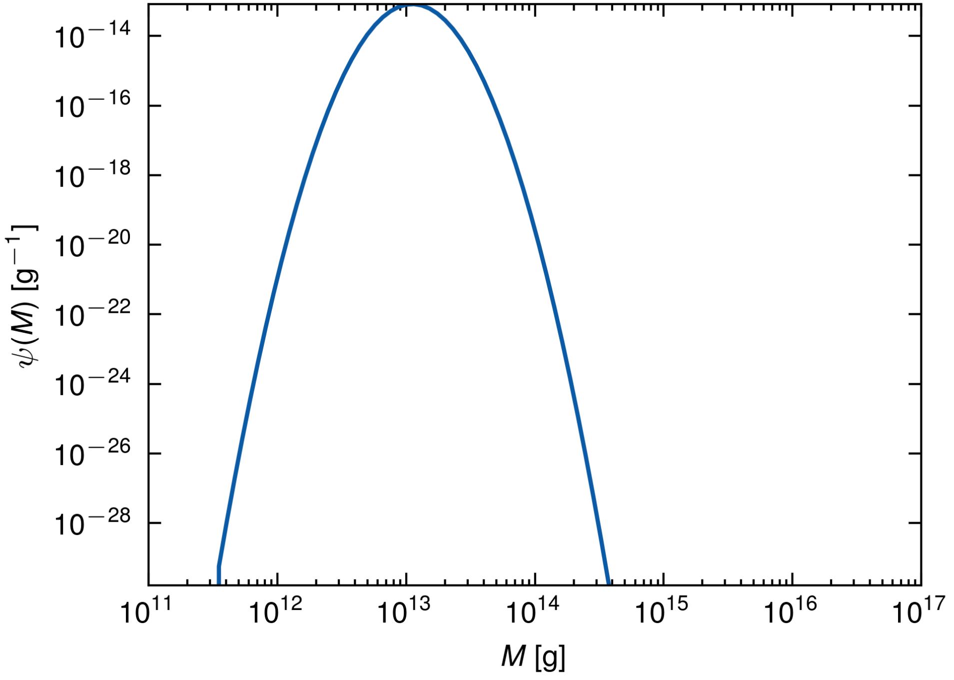


Figure 17: (Left) New potential function $\tilde{V}(q)$, (Middle) $q(t)$, (Right) $p(t)$; Olive dashed line marks the relevant area.

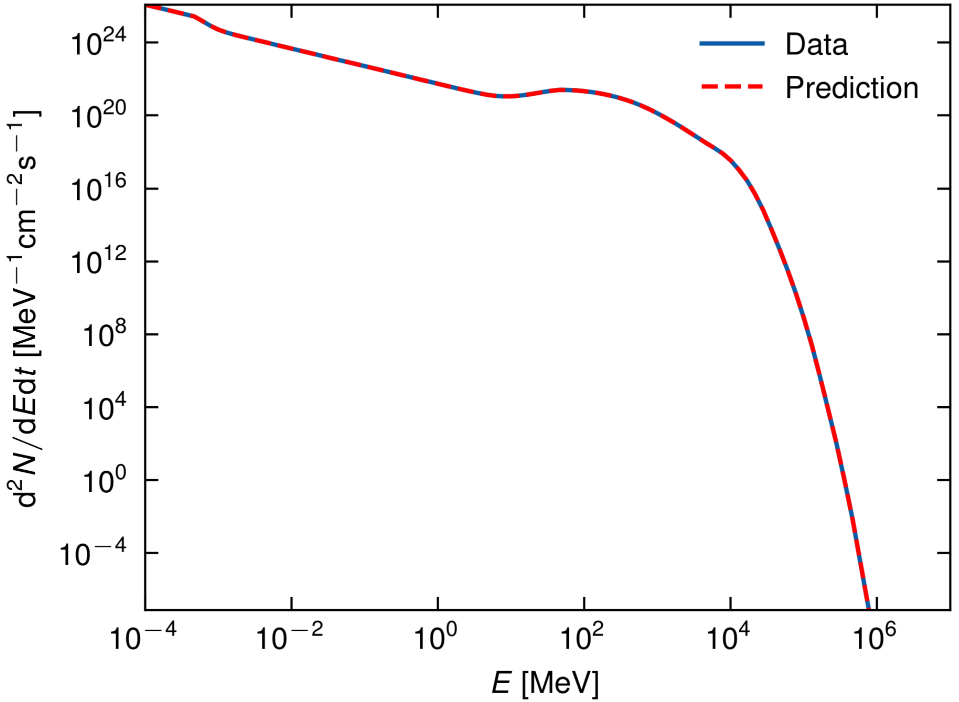
Neural Hawking Operator (Results: *Log-Normal*)

$$\psi(M)$$



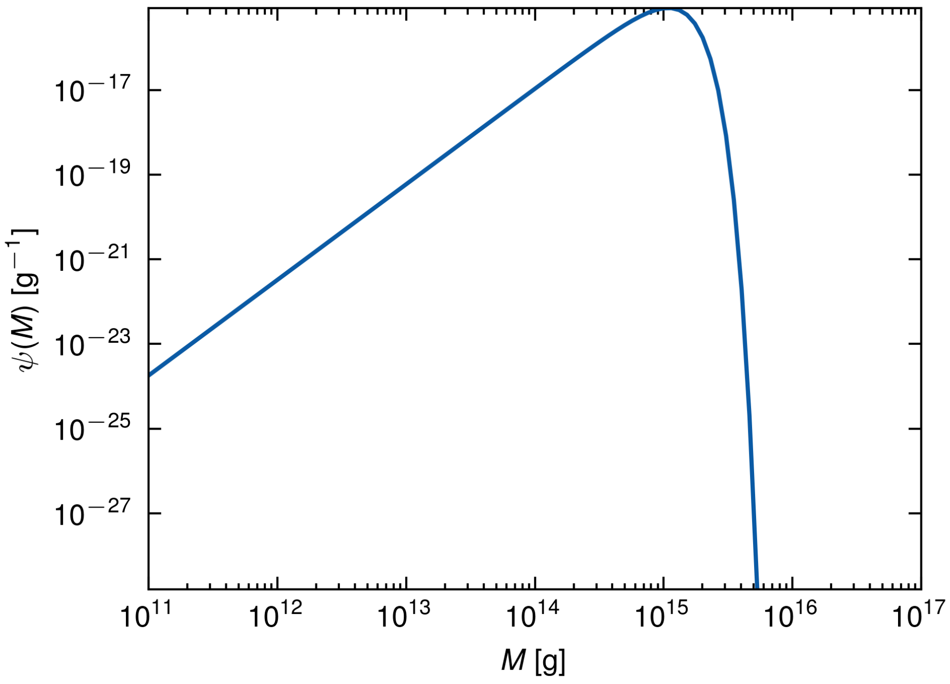
$$\hat{\mathfrak{H}}$$

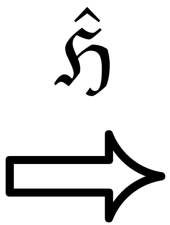
$$\frac{\mathrm{d}^2 N_\gamma^{\mathrm{total}}}{\mathrm{d}E_\gamma \mathrm{d}t}$$



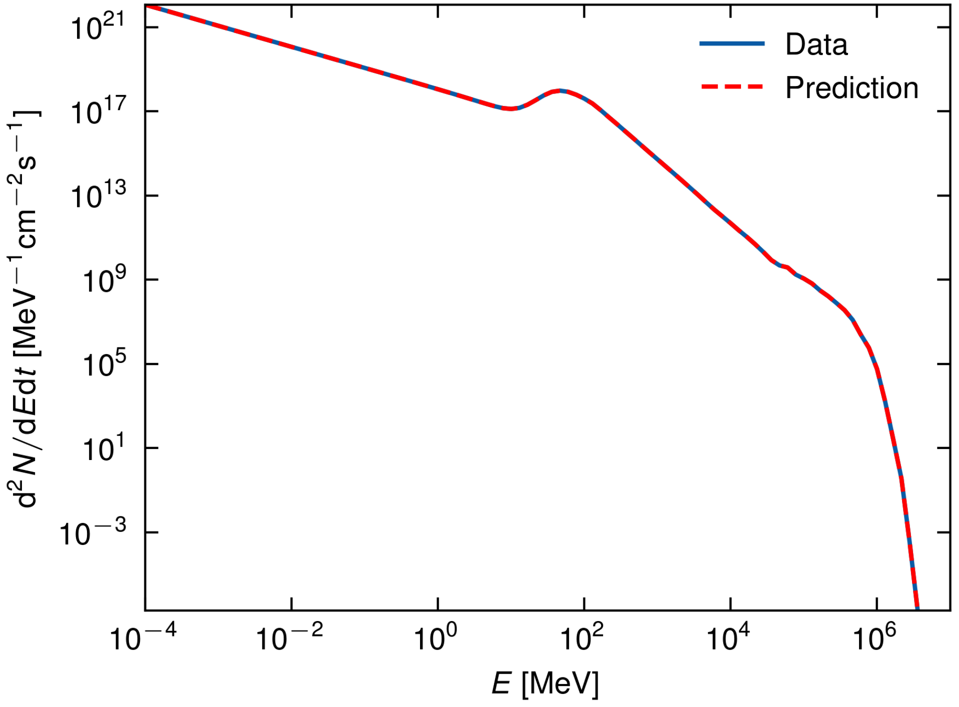
Neural Hawking Operator (Results: *Critical Collapse*)

$$\psi(M)$$



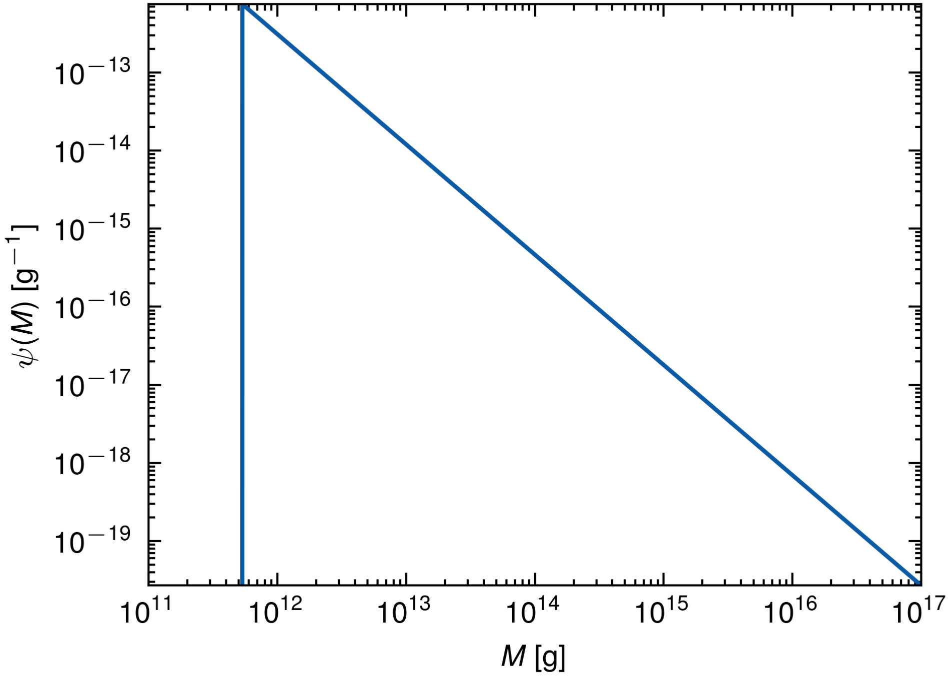


$$\frac{\mathrm{d}^2 N_\gamma^{\mathrm{total}}}{\mathrm{d}E_\gamma \mathrm{d}t}$$



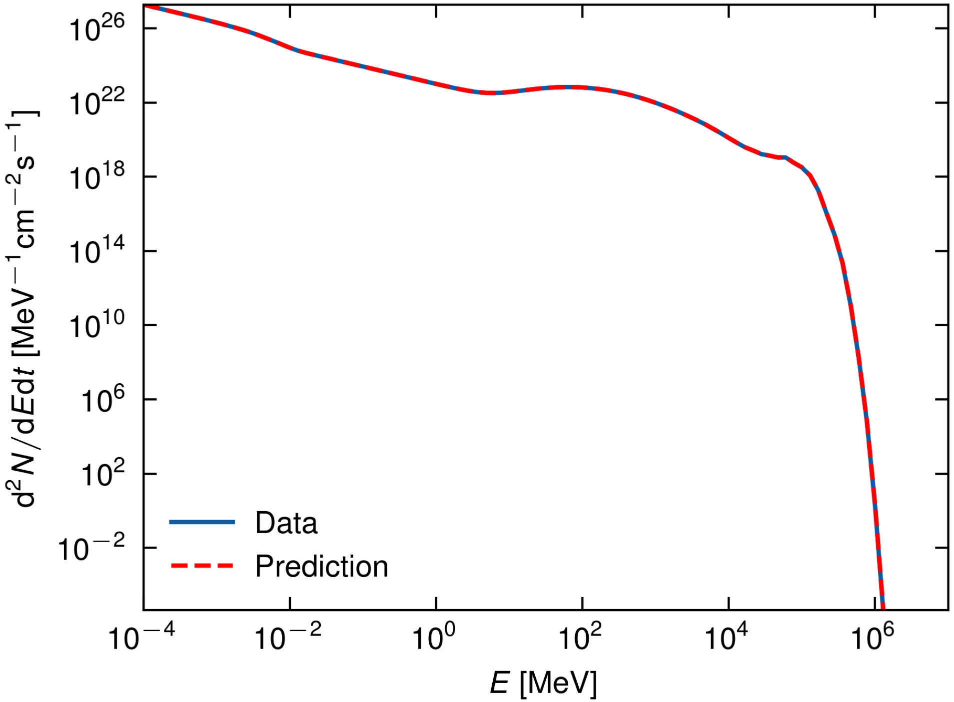
Neural Hawking Operator (Results: *Power-Law*)

$$\psi(M)$$



$$\hat{\mathfrak{H}} \Rightarrow$$

$$\frac{\mathrm{d}^2 N_\gamma^{\mathrm{total}}}{\mathrm{d}E_\gamma \mathrm{d}t}$$



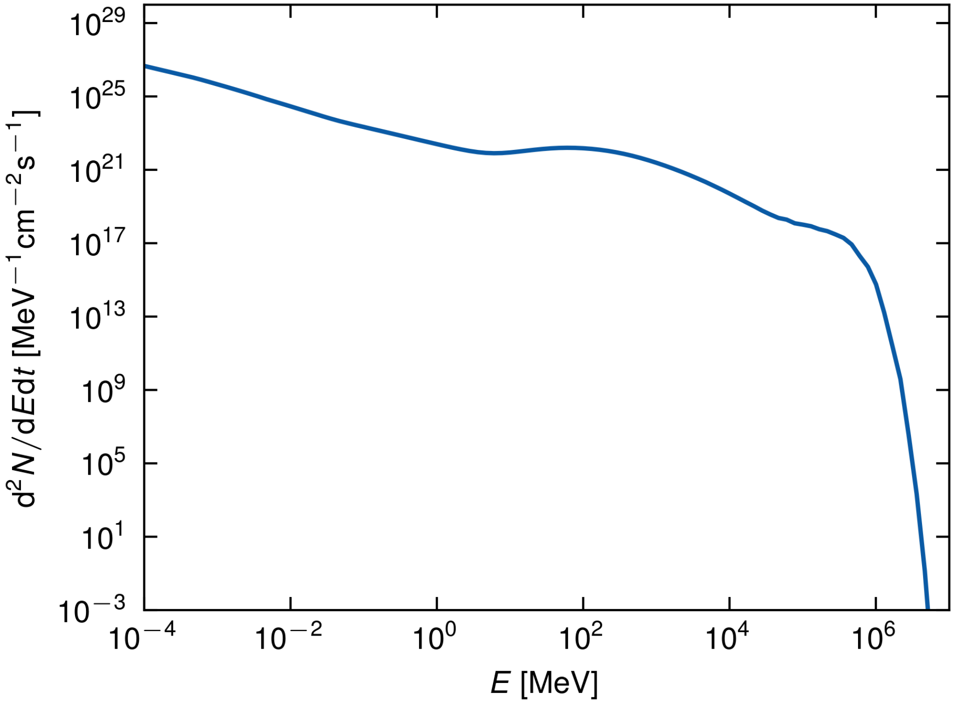
Neural Hawking Operator (Results: *Execution Time*)

Method	Direct Simulation			Operator Inference (Ours)		
	BlackHawk (Seq.)	BlackHawk (Est.)	Numerical (Hybrid)	DeepONet	TraONet	MambONet
Time (s)	2.3010×10^5	7.1960×10^3	3.5576×10^3	$\mathbf{4.5743 \times 10^{-1}}$	1.3504×10^0	6.3986×10^{-1}

Table 2: Comparison of total execution times to compute 100,000 PBH secondary spectra. The **BlackHawk (Est.)** time is an ideal parallel extrapolation from a single-thread measurement. The **Numerical (Hybrid)** is our custom parallelized code. All benchmarks were performed on the hardware specified in the text. **Bold** indicates the best performance.

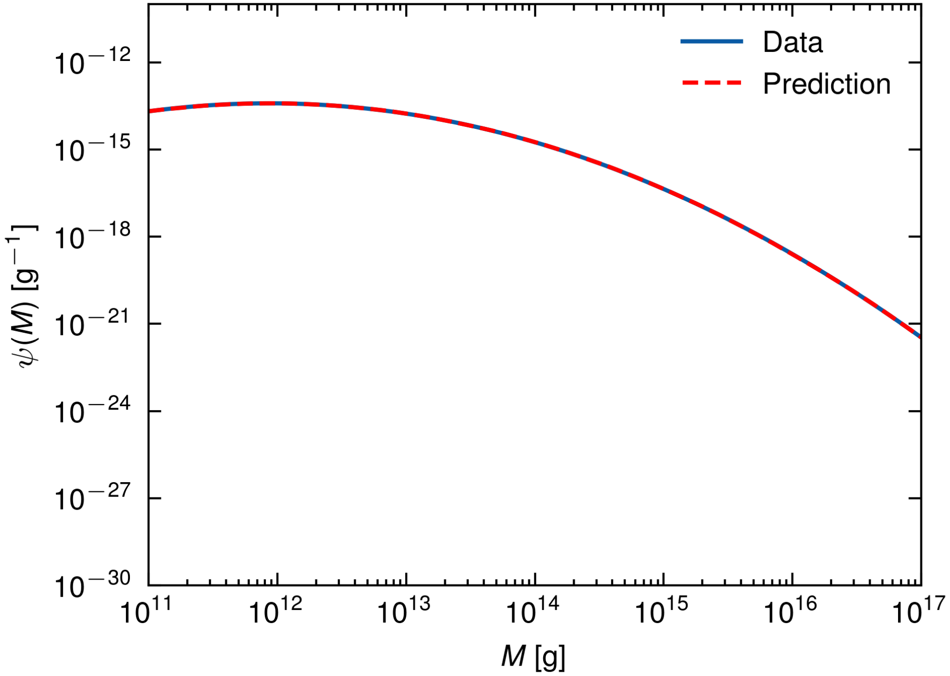
Neural Inverse Hawking Operator (*Log-Normal*)

$$\frac{\mathrm{d}^2 N_\gamma^{\mathrm{total}}}{\mathrm{d}E_\gamma \mathrm{d}t}$$



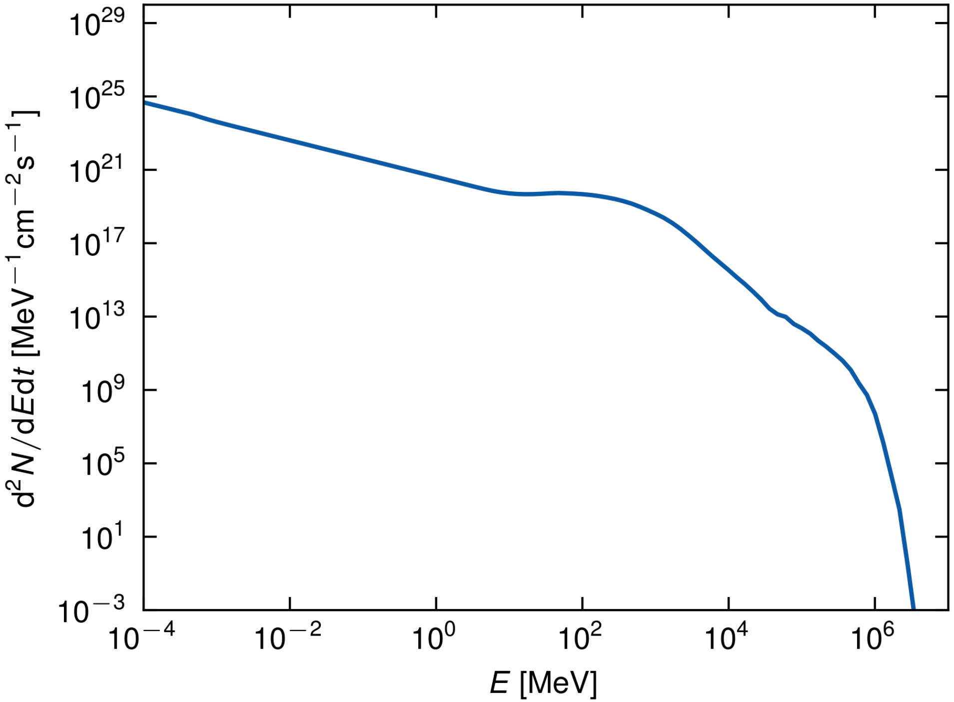
$\hat{\mathfrak{H}}^{-1}$

$$\psi(M)$$



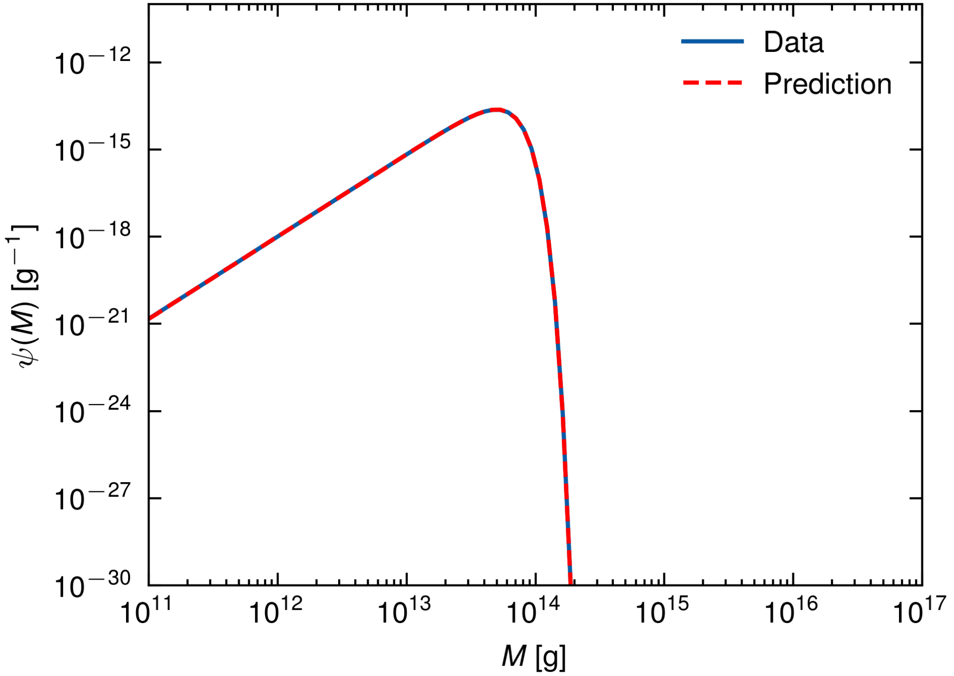
Neural Inverse Hawking Operator (*Critical Collapse*)

$$\frac{\mathrm{d}^2 N_\gamma^{\mathrm{total}}}{\mathrm{d}E_\gamma \mathrm{d}t}$$



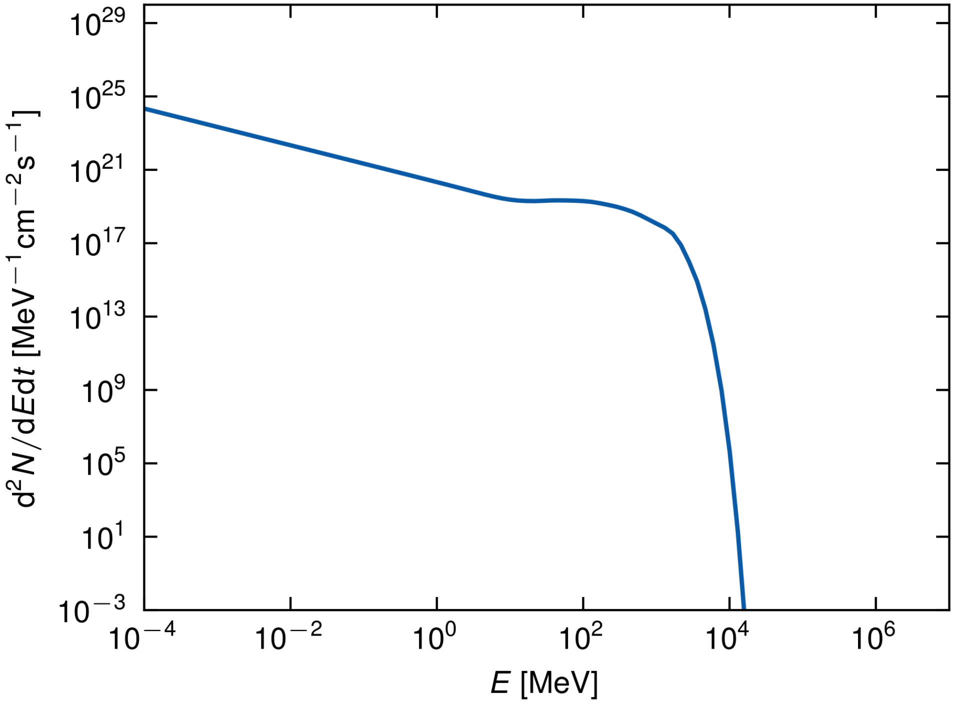
$$\hat{\mathfrak{H}}^{-1}$$

$$\psi(M)$$



Neural Inverse Hawking Operator (*Power-Law*)

$$\frac{\mathrm{d}^2 N_\gamma^{\mathrm{total}}}{\mathrm{d}E_\gamma \mathrm{d}t}$$



$$\hat{\mathfrak{H}}^{-1}$$

$$\psi(M)$$