エネルギー保存則など望ましい性質を持つ深層学習の設計について

松原 崇（大阪大学）

深層学習とは

目的汎関数$\mathcal{L}$と関数空間$\mathcal{F}$を
目的に合わせて設計する研究領域である

$$\inf_{f \in \mathcal{F}} \mathcal{L}(f, \{X, Y\})$$

$\mathcal{L}$：ネットワークの入出力を変形する

$\mathcal{F}$：ネットワーク構造で決める

深層学習とは

目的汎関数$\mathcal{L}$と関数空間$\mathcal{F}$を
目的に合わせて設計する研究領域である

$$\inf_{f\in\mathcal{F}} \mathcal{L}(f,\{X,Y\})$$

$\mathcal{L}$：ネットワークの入出力を変形する

$\mathcal{F}$：ネットワーク構造で決める

抽象的な“数学”を現実のタスクに応用可能にする

現実的にはtry-and-errorな調整がいっぱい必要……

深層学習で物理現象をモデル化する

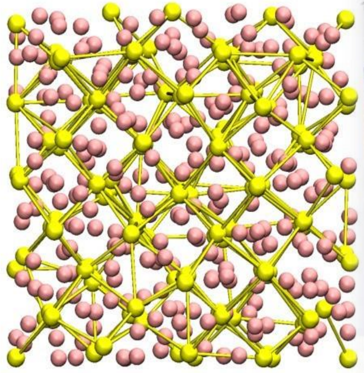
データから物理現象をモデル化し予測

○ミクロにシミュレーションすることは高コスト

○マクロなモデル化はどこかに近似が入る

- 分子動力学法の経験ポテンシャルなど

$$\dot{u} = f(u)$$



物理現象　　物理シミュレーション　　科学的発見や設計

深層学習で物理現象をモデル化する

目的汎関数$\mathcal{L}$は何か？

[Chen+, NeurIPS2017]
[Teshima+, NeurIPSW2020]

万能近似性あり

○微分方程式$\dot{u}=f(u)$で記述する
　→ ベクトル場fを近似する

○離散時刻でサンプルされ，uは既知，$\mathrm{d}u/\mathrm{d}t$は未知
　→ 初期値問題の解とデータの誤差を最小化

システム同定の枠組みが使える

随伴方程式を使って最適化したり

深層学習で物理現象をモデル化する

Neural ODE (NODE) によるモデル化

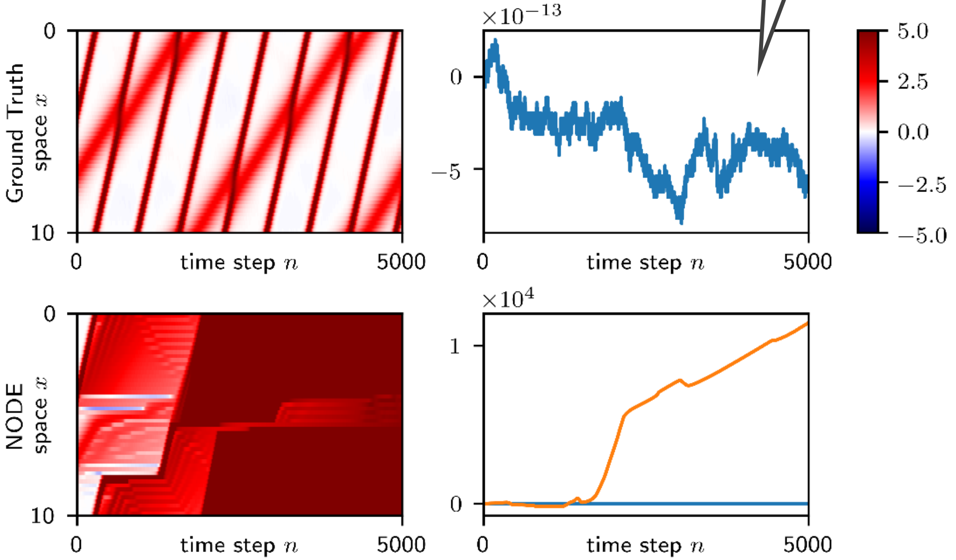
[Chen+, NeurIPS2017]

〇ODE $\dot{u} = f(u)$ のfをニューラルネットワークで近似

KdV方程式

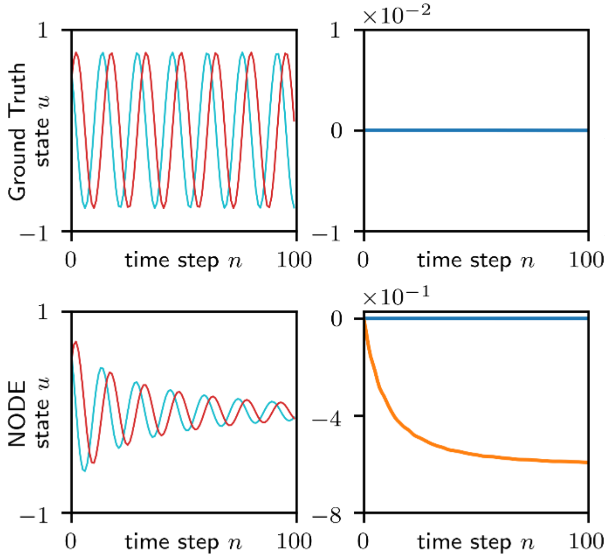
- 途中で崩壊する

エネルギー H



mass-spring system

- エネルギーが減衰する



深層学習で物理現象をモデル化する

関数空間$\mathcal{F}$をハミルトン系に限定

[Greydanus+, NeurIPS2019]
[Chen+, arXiv, 2021]

万能近似性あり

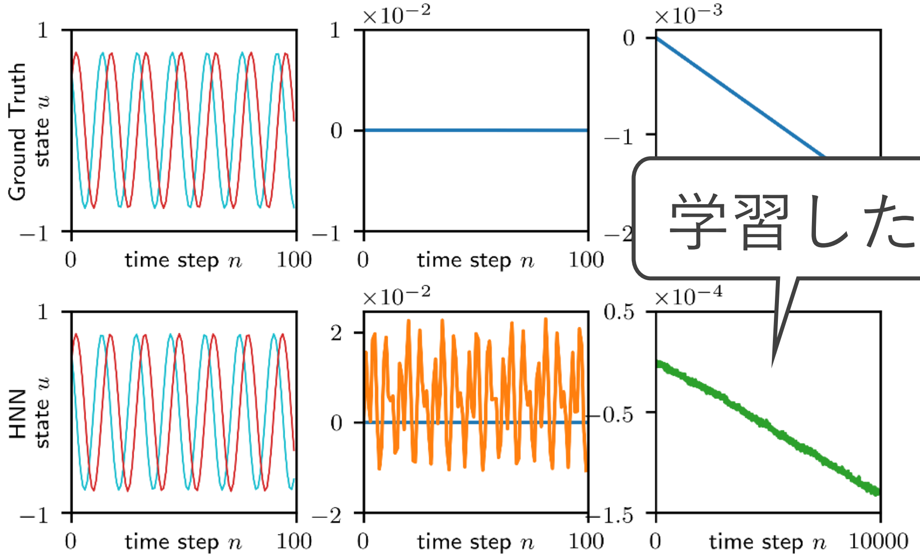
○エネルギー関数 H を近似する

位置 q: $\dot{q} = \nabla_p H(q,p)$

正準運動量 p: $\dot{p} = -\nabla_q H(q,p)$

mass-spring system

- エネルギー関数Hが短期間保存



深層学習で物理現象をモデル化する

Hamiltonian Neural Network [Greydanus+, NeurIPS2019]

○エネルギー関数をニューラルネットワークで近似

○連続時間でエネルギーを保存

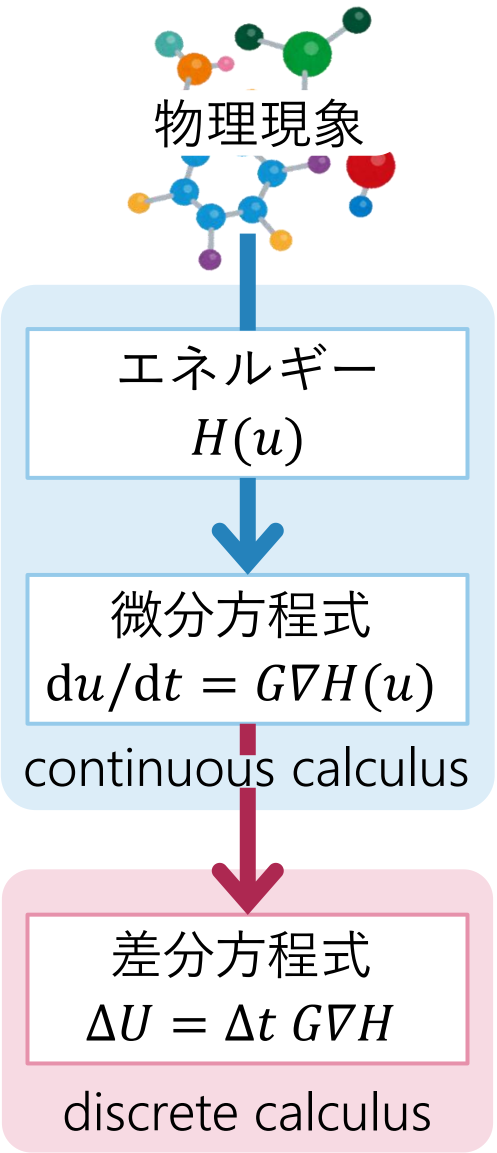
○数値積分するとエネルギーは保存しない

Symplectic HNN [Chen+, ICLR2020]

○シンプレクティック数値積分法を採用

○修正エネルギーを保存(真のエネルギーは振動)

○散逸系には使えない



提案手法

ハミルトン系 → エネルギーに基づくモデル化

○保存系・散逸系の ODEと PDEに適応可能

○一般のシンプレクティック勾配流・リーマン勾配流

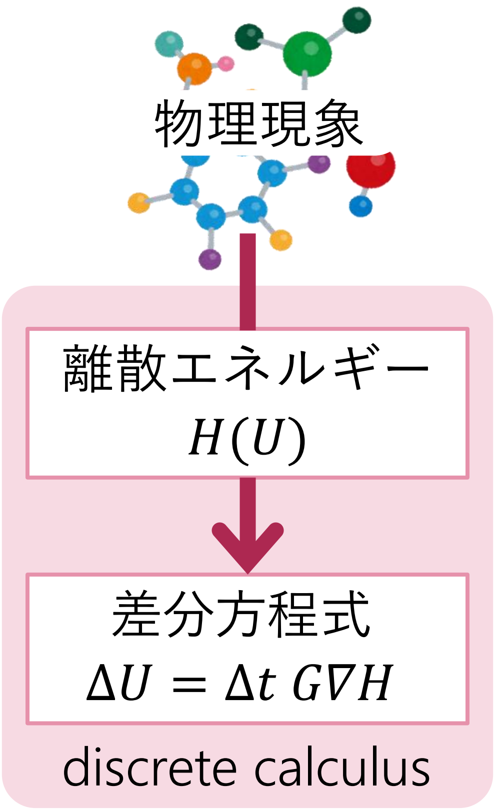
数値積分 → 離散時間モデル化

○離散時間で厳密なエネルギーの保存と散逸

○離散化誤差を起こさない

深層学習に適応可能

○“自動離散微分”を提案



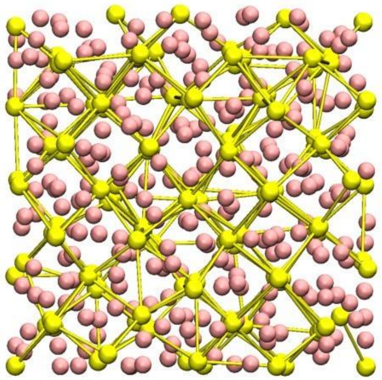
[Matsubara+, NeurIPS2020]

提案手法: エネルギーに基づくのモデル化

多くの物理系はエネルギーでモデル化できる

$\mathrm{d}u/\mathrm{d}t = G\nabla H(u)$

ODE $u = (q \ \ p)^\top$



PDE $u = (u_1 \ \ldots \ u_n)^\top$

提案手法: エネルギーに基づくのモデル化

多くの物理系はエネルギーでモデル化できる

$\mathrm{d}u/\mathrm{d}t = G\nabla H(u)$

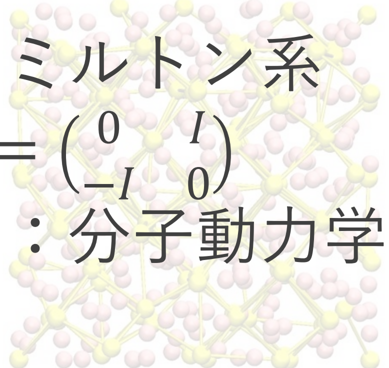
ODE $u = (q \;\; p)^\top$

PDE $u = (u_1 \; \dots \; u_n)^\top$

保存系

G: 歪対称行列

$$\mathrm{d}H/\mathrm{d}t = \partial H/\partial u \cdot \mathrm{d}u/\mathrm{d}t = \nabla H^\top G \nabla H = 0$$



ハミルトン系

$S = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$

例：分子動力学，天体

提案手法: エネルギーに基づくのモデル化

多くの物理系はエネルギーでモデル化できる

$\mathrm{d}u/\mathrm{d}t = G\nabla H(u)$

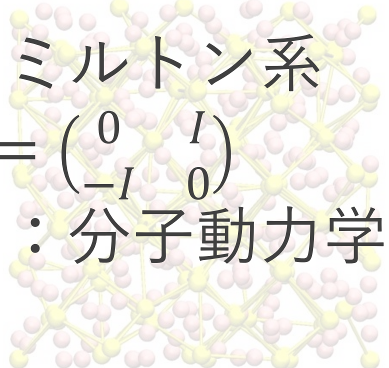
ODE $u = (q \;\; p)^\top$

PDE $u = (u_1 \;\; \ldots \;\; u_n)^\top$

保存系

G: 歪対称行列

$\mathrm{d}H/\mathrm{d}t = \nabla H^\top G \nabla H = 0$



ハミルトン系

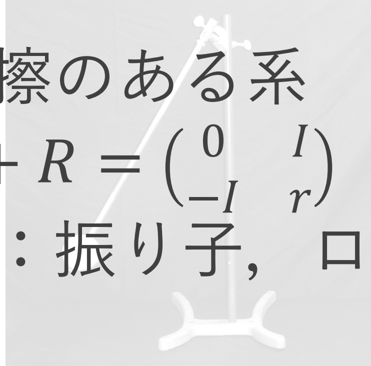
$S = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$

例：分子動力学，天体

散逸系

G: 半負定値行列

$\mathrm{d}H/\mathrm{d}t = \nabla H^\top G \nabla H \leq 0$



摩擦のある系

$S + R = \begin{pmatrix} 0 & I \\ -I & r \end{pmatrix}$

例：振り子，ロボット

提案手法: エネルギーに基づくのモデル化

多くの物理系はエネルギーでモデル化できる

$\mathrm{d}u/\mathrm{dt} = G\nabla H(u)$

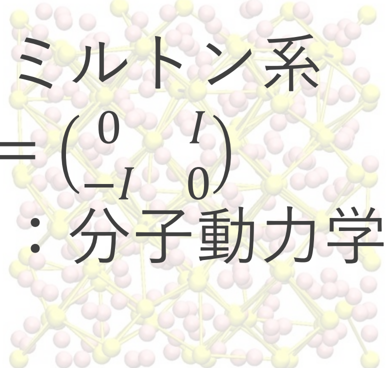
ODE $u = (q \;\; p)^\top$

PDE $u = (u_1 \; \ldots \; u_n)^\top$

保存系

G: 歪対称行列

$\mathrm{d}H/\mathrm{dt} = \nabla H^\top G \nabla H = 0$



ハミルトン系

$S = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$

例：分子動力学，天体

Hamiltonian field theory

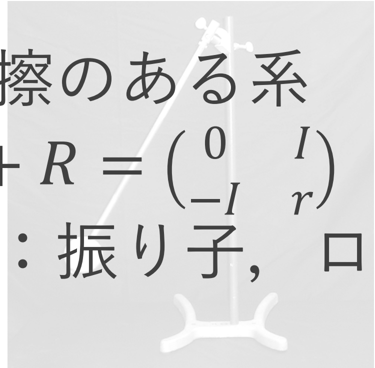
$$D = \frac{1}{2\Delta x} \begin{pmatrix} 0 & 1 & & \cdots & & -1 \\ -1 & 0 & 1 & & & \\ & \vdots & & \ddots & \vdots & \\ & & & -1 & 0 & 1 \\ & & & \cdots & & \\ 1 & & & & -1 & 0 \end{pmatrix}$$

例：波の方程式

散逸系

G: 半負定値行列

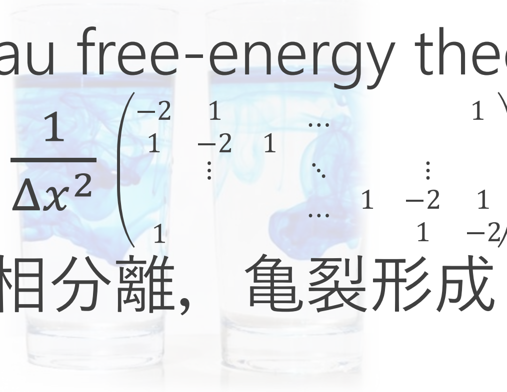
$\mathrm{d}H/\mathrm{dt} = \nabla H^\top G \nabla H \le 0$



摩擦のある系

$S + R = \begin{pmatrix} 0 & I \\ -I & r \end{pmatrix}$

例：振り子，ロボット



Landau free-energy theory

$$D_2 = \frac{1}{\Delta x^2} \begin{pmatrix} -2 & 1 & & \cdots & & 1 \\ 1 & -2 & 1 & & & \\ & \vdots & & \ddots & \vdots & \\ & & & 1 & -2 & 1 \\ & & & \cdots & & \\ 1 & & & & 1 & -2 \end{pmatrix}$$

例：相分離，亀裂形成

提案手法: 離散時間モデル化

連続時間システム

$$\mathrm{d}u/\mathrm{dt} = G\nabla H(u)$$

連続時間の微積分

$$\mathrm{d}H/\mathrm{dt} = \nabla H \cdot \mathrm{d}u/\mathrm{dt} \ = \nabla H^\top G \nabla H$$

$\because$連鎖律 $\mathrm{d}H = \nabla H \cdot \mathrm{d}u$

Gが歪対称なら保存系
Gが半負定値なら散逸系

[Furihata+, Chapman and Hall/CRC, 2020]

提案手法: 離散時間モデル化

連続時間システム

$$\mathrm{d}u/\mathrm{dt} = G\nabla H(u)$$

離散時間のシステム

$$\Delta u/\Delta t = \bar{G}\bar{\nabla} H(u_{n+1}, u_n)$$

$$\begin{cases} \Delta u = u_{n+1} - u_n \\ \Delta H = H(u_{n+1}) - H(u_n) \end{cases}$$

連続時間の微積分

$$\mathrm{d}H/\mathrm{dt} = \nabla H \cdot \mathrm{d}u/\mathrm{dt} \ = \nabla H^\top G \nabla H$$

$\because$連鎖律 $\mathrm{d}H = \nabla H \cdot \mathrm{d}u$

Gが歪対称なら保存系
Gが半負定値なら散逸系

[Furihata+, Chapman and Hall/CRC, 2020]

提案手法: 離散時間モデル化

連続時間システム

$$\mathrm{d}u/\mathrm{d}t = G\nabla H(u)$$

離散時間のシステム

$$\Delta u/\Delta t = \bar{G}\bar{\nabla}H(u_{n+1}, u_n)$$

$$\begin{cases} \Delta u = u_{n+1} - u_n \\ \Delta H = H(u_{n+1}) - H(u_n) \end{cases}$$

連続時間の微積分

$$\mathrm{d}H/\mathrm{d}t = \nabla H \cdot \mathrm{d}u/\mathrm{d}t \;= \nabla H^\top G \nabla H$$

$\because$連鎖律 $\mathrm{d}H = \nabla H \cdot \mathrm{d}u$

離散時間の微積分

$$\Delta H/\Delta t = \bar{\nabla}H \cdot \Delta u/\Delta t \;= \bar{\nabla}H^\top \bar{G}\bar{\nabla}H$$

$\because$連鎖律 $\Delta H = \bar{\nabla}H \cdot \Delta u$

$\bar{G}$が歪対称なら保存系
$\bar{G}$が半負定値なら散逸系

[Furihata+, Chapman and Hall/CRC, 2020]

提案手法: 自動離散微分

連続時間の微積分

$\mathrm{d}H(u) = \nabla H(u) \cdot \mathrm{d}u$ 連鎖律

$\mathrm{d}H(ax;u) = a\mathrm{d}H(x;u)$ 線形性

自動微分

出力層から逆伝播

$\nabla(f \circ g)(u) = J_g(u)^\top \nabla f(g(u))$

$J_g = W$ 線形層

$\left(J_g\right)_{kk} = \dfrac{\partial \sigma\left(u^{(k)}\right)}{\partial u^{(k)}}$ 活性化関数

提案手法: 自動離散微分

連続時間の微積分

離散時間の微積分

連鎖律

$$\mathrm{d}H(u) = \nabla H(u) \cdot \mathrm{d}u$$

$$\Delta H(u_{n+1}, u_n) = \bar{\nabla} H(u_{n+1}, u_n) \cdot \Delta u$$

線形性

$$\mathrm{d}H(ax; u) = a\mathrm{d}H(x; u)$$

$$\bar{\mathrm{d}}H(ax; u, v) = a\bar{\mathrm{d}}H(x; u, v)$$

一貫性

$$\bar{\mathrm{d}}H(\cdot; u, u) = \mathrm{d}H(\cdot; u)$$

自動微分

自動離散微分

$$\nabla(f \circ g)(u) = J_g(u)^\top \nabla f(g(u))$$

$$\bar{\nabla}(f \circ g)(u, v) = \bar{J}_g(u, v)^\top \bar{\nabla} f\big(g(u), g(v)\big)$$

線形層

$$J_g = W$$

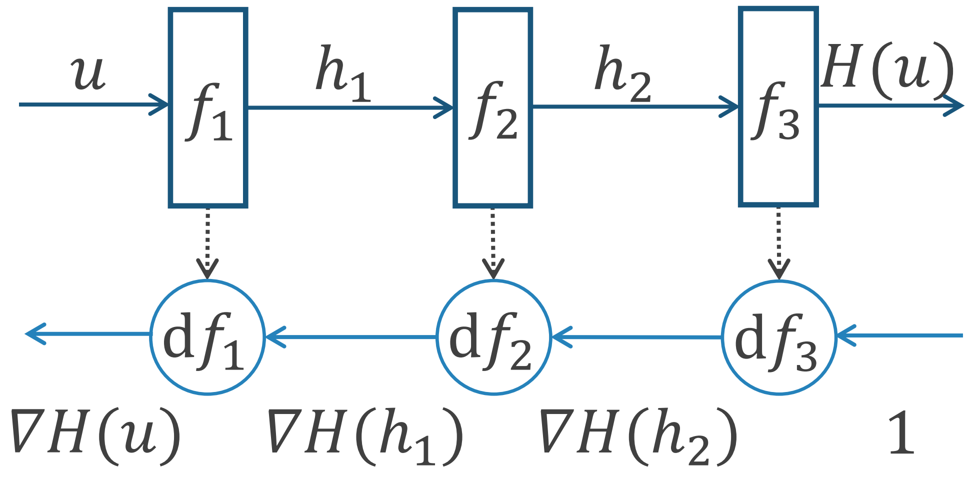
$$\bar{J}_g = W$$

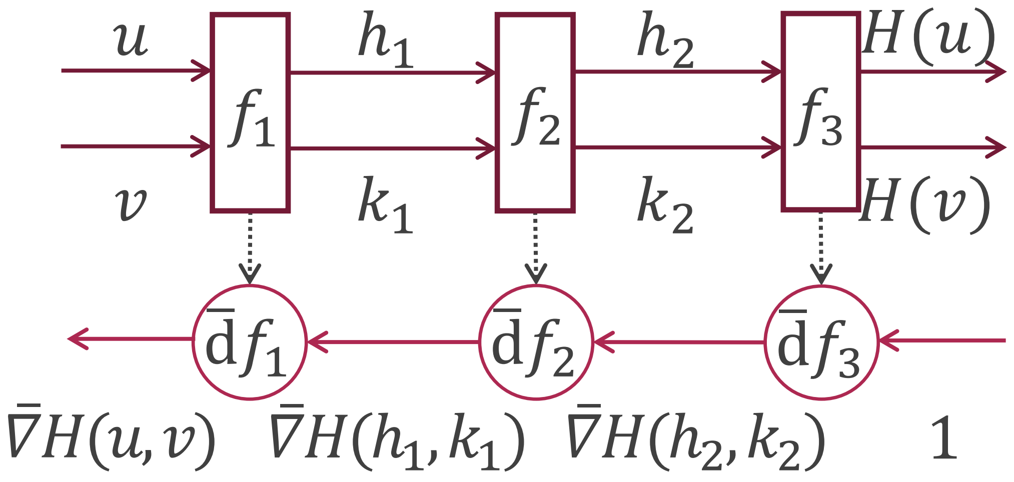
活性化関数

$$\left(J_g\right)_{kk} = \frac{\partial \sigma(u^{(k)})}{\partial u^{(k)}}$$

$$\left(\bar{J}_g\right)_{kk} = \frac{\sigma(u^{(k)}) - \sigma(v^{(k)})}{u^{(k)} - v^{(k)}}$$

提案手法: 自動離散微分





自動微分

$$\nabla(f \circ g)(u) = J_g(u)^\top \nabla f(g(u))$$

自動離散微分

$$\bar{\nabla}(f \circ g)(u,v) = \bar{J}_g(u,v)^\top \bar{\nabla} f\big(g(u), g(v)\big)$$

線形層

$$J_g = W \qquad \bar{J}_g = W$$

活性化関数

$$\left(J_g\right)_{kk} = \frac{\partial \sigma(u^{(k)})}{\partial u^{(k)}} \qquad \left(\bar{J}_g\right)_{kk} = \frac{\sigma(u^{(k)}) - \sigma(v^{(k)})}{u^{(k)} - v^{(k)}}$$

提案手法: 自動離散微分

離散時間システム

$$\Delta u/\Delta t = \bar{G}\bar{\nabla}H(u_{n+1}, u_n)$$

$$\begin{cases} \Delta u = u_{n+1} - u_n \\ \Delta H = H(u_{n+1}) - H(u_n) \end{cases}$$

離散時間の微積分

$$\Delta H/\Delta t = \bar{\nabla}H \cdot \Delta u/\Delta t \ = \bar{\nabla}H^\top \bar{G}\bar{\nabla}H$$

∵連鎖律 $\Delta H = \bar{\nabla}H \cdot \Delta u$

自動離散微分

$$\bar{\nabla}(f \circ g)(u, v) = \bar{J}_g(u, v)^\top \bar{\nabla}f\big(g(u), g(v)\big)$$

$$\bar{J}_g = W$$

$$\left(\bar{J}_g\right)_{kk} = \frac{\sigma\left(u^{(k)}\right) - \sigma\left(v^{(k)}\right)}{u^{(k)} - v^{(k)}}$$

目的関数

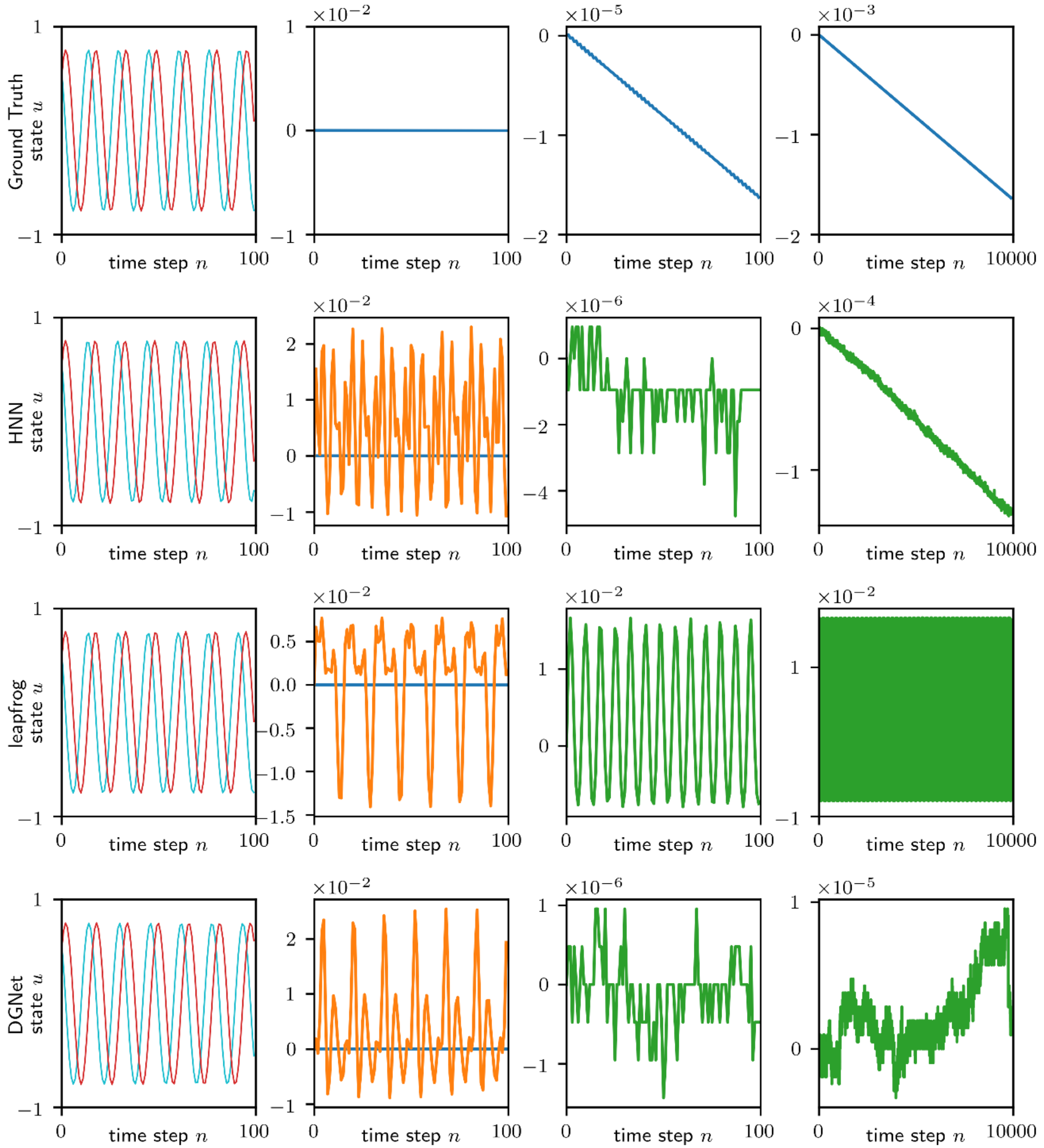
$$\frac{1}{N}\|\Delta u/\Delta t - \bar{G}\bar{\nabla}H(u_{n+1}, u_n)\|^2$$

○離散時間でモデル化されているので離散化誤差がない

○学習は非常に速い(予測は陰的)

実験結果

mass-spring system
（保存系のODE）



3層のニューラルネットワーク
活性化関数はtanh

実験結果

畳み込み1層＋全結合2層＋Gの畳み込み

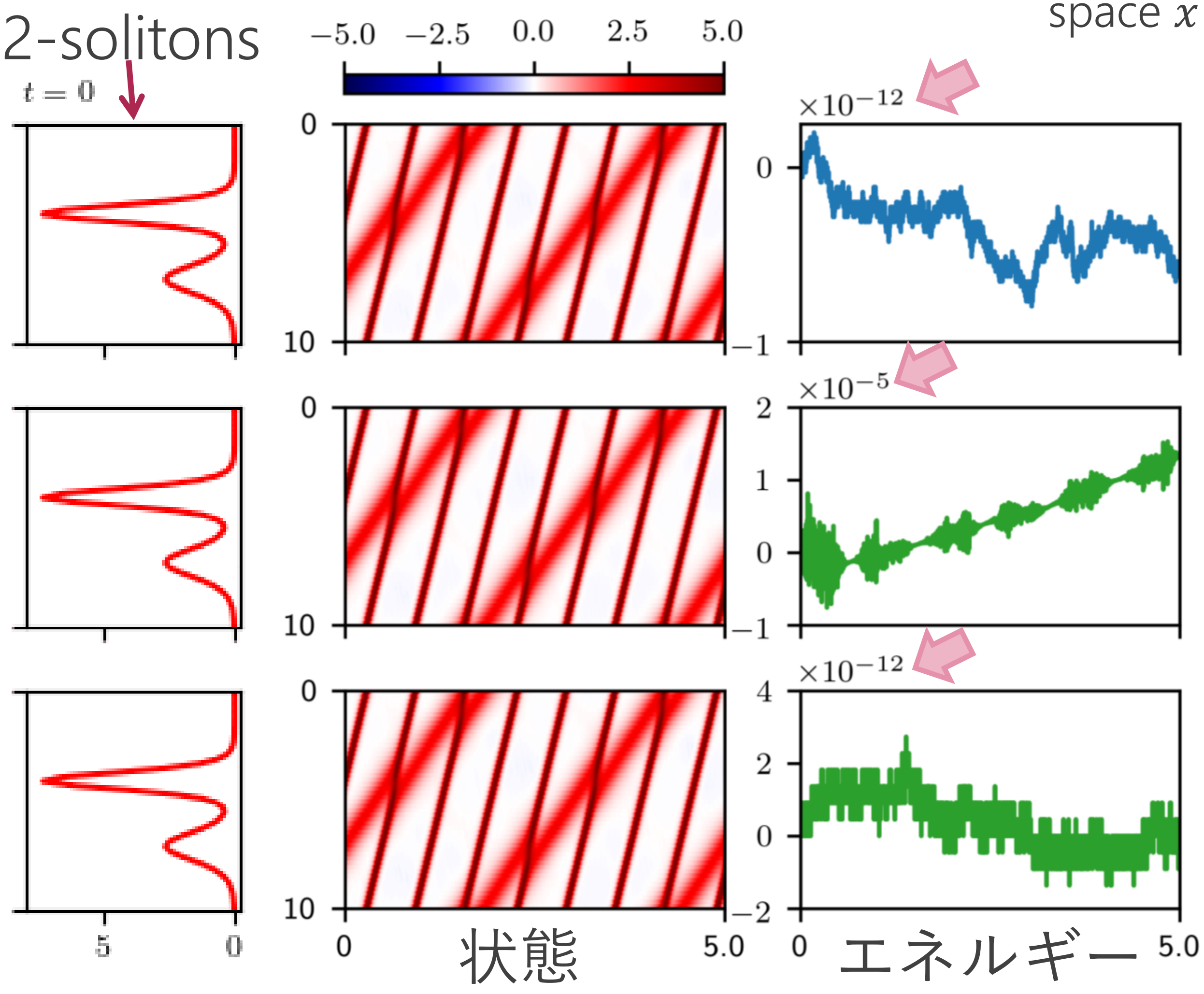
KdV方程式
（保存系のPDE）

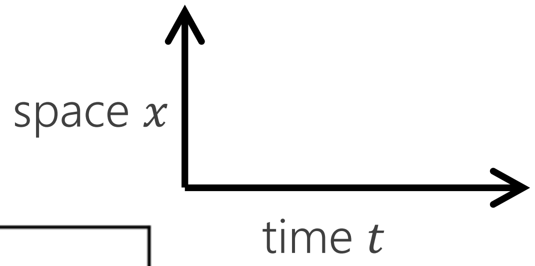
教師データ

$$\dot{u} = G\nabla H(u)$$

$$\frac{\Delta u}{\Delta t} = G\bar{\nabla} H(u_{n+1}, u_n)$$

(proposed)





実験結果

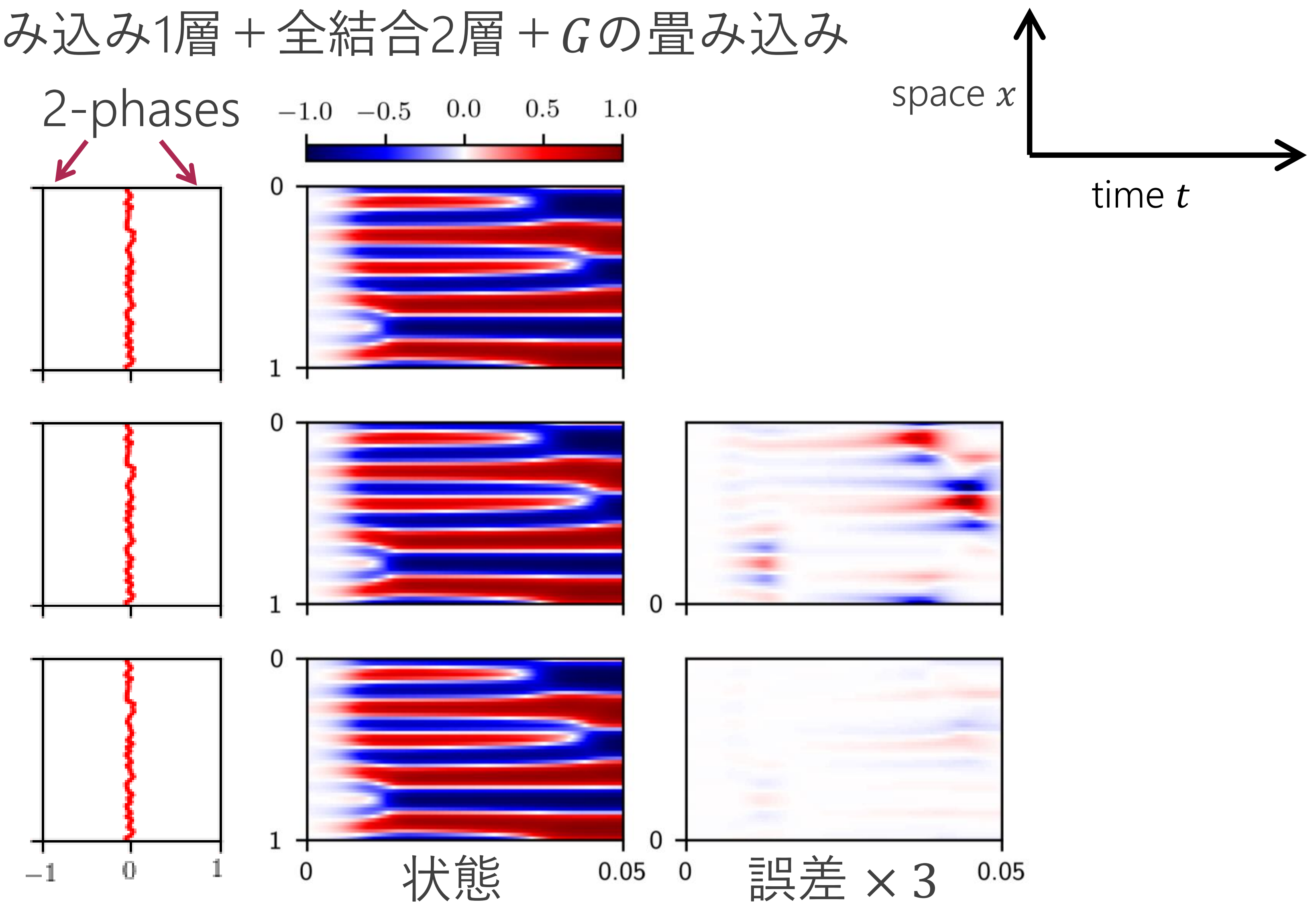
畳み込み1層＋全結合2層＋Gの畳み込み

Cahn-Hilliard方程式
（散逸系のPDE）

教師データ

$\dot{u} = G\nabla H(u)$

$\frac{\Delta u}{\Delta t} = G\bar{\nabla} H(u_{n+1}, u_n)$
(proposed)



実験結果

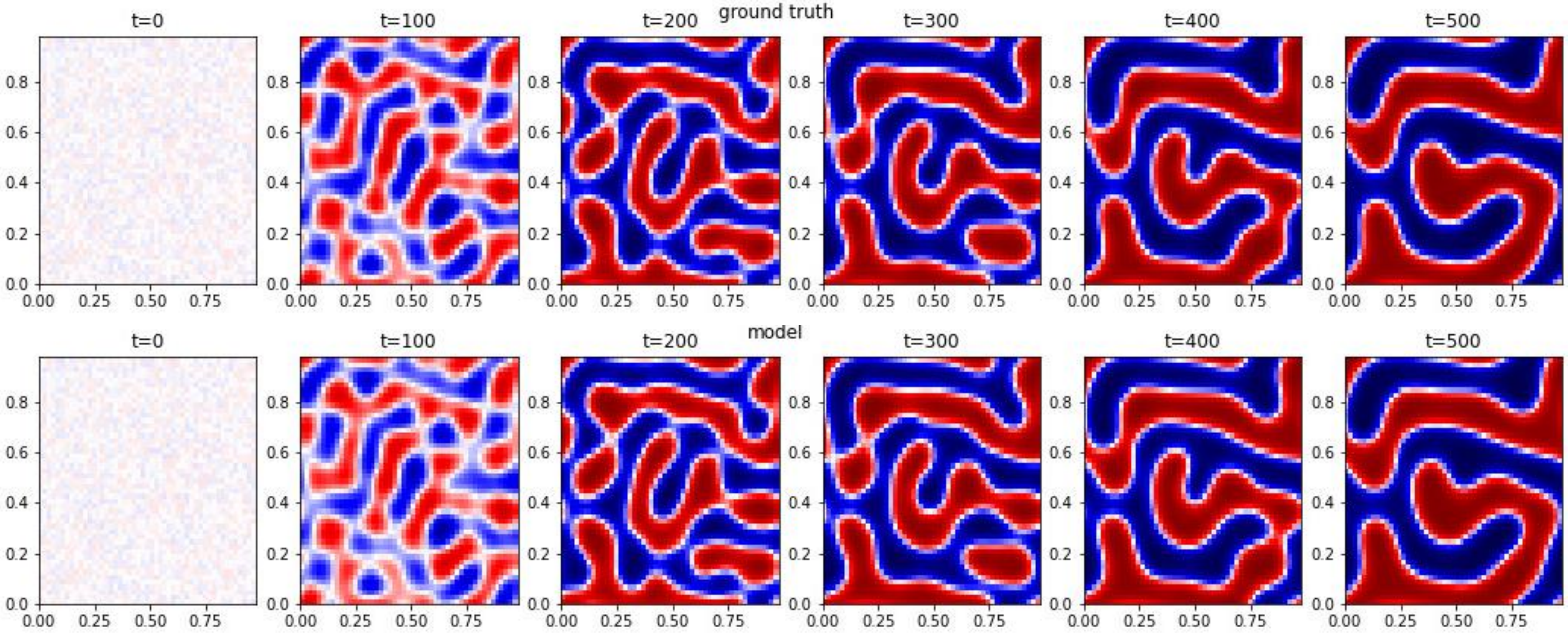
二乗誤差（スケール調整済み）

手法	KdV方程式		Cahn-Hilliard方程式	
	エネルギー H	状態 u_n	エネルギー H	状態 u_n
$\dot{u} = f(u)$	>1000	>1000	>1000	913.96
$\dot{u} = G\nabla H(u)$	3.01	0.34	4.89	0.80
$\Delta u/\Delta t = G\bar{\nabla}H(u_{n+1}, u_n)$	**1.60**	**0.25**	**0.34**	**0.07**

4次の解法であるドルマン＝プリンス法よりも精度が高い

実験結果

2D Cahn-Hilliard方程式
（数値積分ではGPUのメモリに乗らなかった）



実験結果

2D Cahn-Hilliard方程式
（Gを学習した）

$$G=\begin{pmatrix} 2.450 & 1.339 & 2.448 \\ 1.328 & -15.148 & 1.326 \\ 2.447 & 1.338 & 2.452 \end{pmatrix}$$

- 合計がゼロ
  →質量が保存される
- 負定値行列である
  →エネルギー散逸性がある
- 回転・反転対称
  →等方性がある

学習結果を解析すれば系の性質も分かる

提案手法のまとめ

ハミルトン系 → エネルギーに基づくモデル化

○保存系・散逸系の ODEと PDEに適応可能

○一般のシンプレクティック勾配流・リーマン勾配流

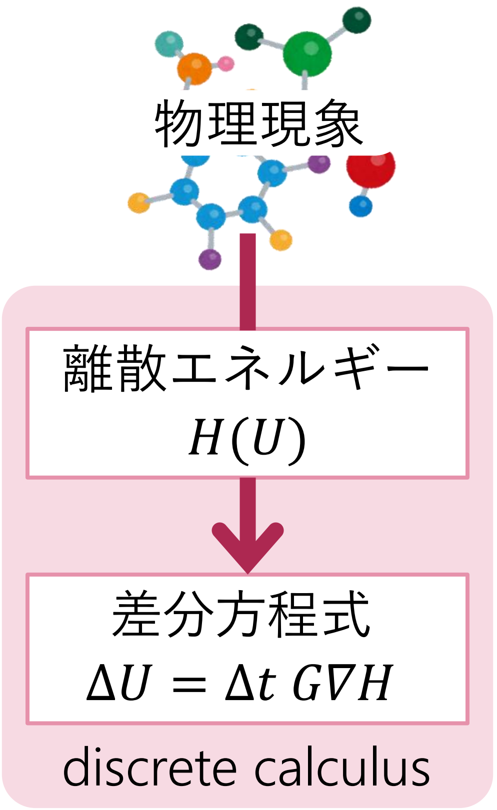
数値積分 → 離散時間モデル化

○離散時間で厳密なエネルギーの保存と散逸

○離散化誤差を起こさない

深層学習に適応可能

○"自動離散微分"を提案



[Matsubara+, NeurIPS2020]

深層学習とは

目的汎関数$\mathcal{L}$と関数空間$\mathcal{F}$を
目的に合わせて設計する研究領域である

$$\inf_{f\in\mathcal{F}} \mathcal{L}(f,\{X,Y\})$$

$\mathcal{L}$：離散時間システムを近似する

$\mathcal{F}$：エネルギーに基づき(離散)勾配を使う
偏微分方程式は3×3カーネル1回で十分
微分可能で有界な活性化関数が良い

深層学習とは

目的汎関数$\mathcal{L}$と関数空間$\mathcal{F}$を
目的に合わせて設計する研究領域である

$$\inf_{f \in \mathcal{F}} \mathcal{L}(f, \{X, Y\})$$

$\mathcal{L}$：？

$\mathcal{F}$：？