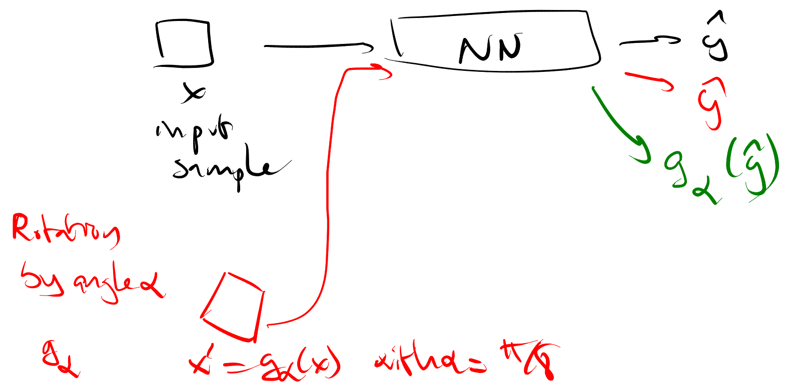
Chapter 5: Incorporating physical knowledge & Deep Learning for physics

I Incorporating priors

Known properties: invariance / equivariance

group of invariance: transforms to be invariant to



ex: image classification

invariant: $NN(g_\alpha(x)) = NN(x)$

equivariant: $NN(g_\alpha(x)) = g_\alpha(NN(x))$

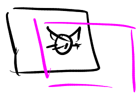
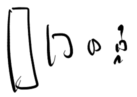
ex: image segmentation

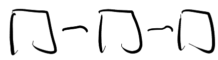
ex: data augmentation

(x, y)

$(g_\alpha(x), y)$ / $(g_\alpha(x), g_\alpha(y))$

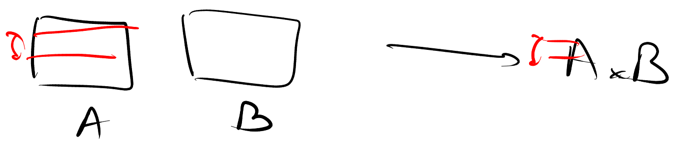
vs: by design

* translation: invariance  convolutions  ← invariant

equivariant " 

ex: U-net

* permutation



1 input ⟶ object

a cloud of points (3D) (pointnet++) ⟶

1 input: a non-ordered list

- point 1
- " 2
- ...

→ a "set"

"Deep Sets"

Features of point 1, 2, 3, ...

individual (independent) processing φ — equivariant

equivariant

stack such stacks

average — invariant

1 stack

$\mu = \frac{1}{N}\sum_i \varphi(x_i)$

Theorem: full expressive power

Original paper: 1 block

↳ can represent any permutation-invariant function (with many enough blocks)

Theorem: IF φ, ψ is expressive enough,

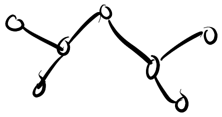
$$\psi\Big(\underbrace{\frac{1}{N}\sum_{i}\varphi(x_i)}_{\text{invariant}}\Big)$$ can represent all permutation-invariant fo

Other ways: attention

	values	keys
x_1	—	—
x_2	—	—
x_3	—	—

$F(x_i, x_j)$ — no ordering involved

- Desired property not formally known:

eg: generate medicat°  molecules non toxic

↳ train a network to learn what "toxic" means
↳ then use it as a training criterion

- Known "mathematical" properties

. normaliz°: eg: output proba distrib°

$\xrightarrow{F}$ [ ] $\in \mathbb{R}$ $\xrightarrow{\text{softmax}}$ [ ] ← $0 \leq \; \leq 1$, $\Sigma = 1$

. div F = 0:

eg: fluid mechanics
incompressible fluid

$x \longrightarrow \boxed{f} \longrightarrow y$ $\quad$ $\mathrm{div}\, y = 0$

$+ \|\mathrm{div}\, y\|^2$

$+ \lambda$ —

$\mathrm{div}\ \mathrm{rot}\ g = 0$ always

$x \longrightarrow \boxed{g} \longrightarrow z \xrightarrow{\text{rot}} y$

Learning equations (evolutions)

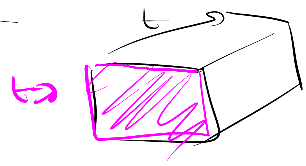
. "data assimilation": know the formula, up to a few parameters

$$\frac{\partial F}{\partial t} = \alpha \Delta_x F + \beta\, \mathrm{div}\, F \cdot \varphi \quad + \text{noise}$$

. know the shape of the solution

Physically-Informed Neural Networks   PINNs

- know the equation
- domain $\Omega \times T$



operator

$$\frac{\partial F}{\partial t} = -L(F, t, x)$$

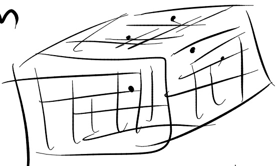
$$\frac{\partial F(x)}{\partial t} = \cdots \quad (F(x)) \quad (\nabla_x F(x)) \quad \Delta_x F(x)$$

or

boundary conditions

or forcing terms

- discretize the domain
- " the differential operators



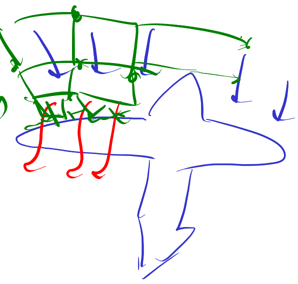
⇒ searching for a tensor F

- training / optimization criterion : $\left\| \frac{\partial F}{\partial t} + L(F, t, x) \right\|^2$

$$+ \alpha \| F(t=0) - F_0 \|^2 + \beta \| F(t, x=\partial\Omega) - F_{\partial\Omega}(t) \|^2$$

or no discretization :

e.g. fluid mechanics



low resol.

high resol.

→ graph-NN

or

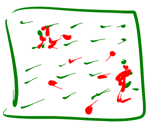
- no mesh
- input the location & time x t

$x \in \Omega$

$t \in [0, T]$

$x \to$, $t \to$ $\boxed{F}$ $\to F(x, t)$

uniform sampling



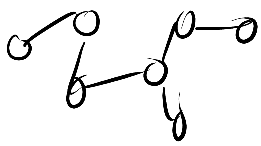
pseudo-uniform

* Other type of physical knowledge
  - stationary distribution
  - ergodic

you know : Boltzmann distribution

$$p(\text{conformation } c) \propto e^{-E(c)}$$

molecules   folding in 3D = conformation ⇕ chemical properts



close-form formula

$$\sum_{\text{nodes}} (\ldots) + \sum_{\text{pair nodes}} (\ldots) \quad \text{physics}$$

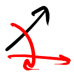
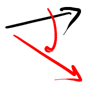
physical knowledge

* Back to invariances / equivariances by design

Rotations

Simplistic

$\vec{v} \xrightarrow{F} F(\vec{v})$

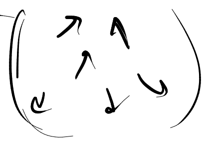
 equivariant? 

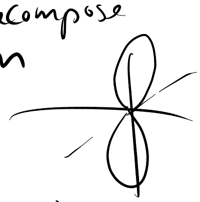
$\vec{v} \to \|\vec{v}\| \to F(\|\vec{v}\|)$ invariant

direction $\to \frac{\vec{v}}{\|\vec{v}\|}$ identity (equivariant)

$y = F(\|\vec{v}\|)\,\frac{\vec{v}}{\|\vec{v}\|}$

Clebsch-Gordan coefficients (theoretical physics)

input  $\xrightarrow{\text{equivariant}}$

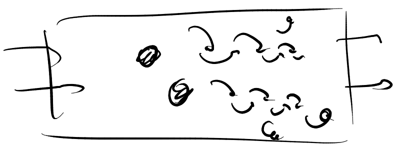
→ decompose on  → ⋮ ⟹ how to mix while satisfying rotat° equivariant → $\sum \dots c_{l,n\dots}$

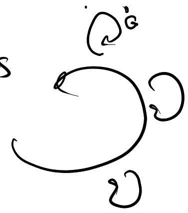
spherical harmonics

material: glass → liquid / solid

⟹ rotat°-equiv networks (Welling)

II Dynamical systems

eg: fluid mechanics 

Navier-Stokes  "Fractal"

- predict evolution in time

$$\frac{\partial F}{\partial t}(x) = g(F, x, t)$$

$$g(F_t(x), \nabla_x F_t(x), \dots)$$

Naive way: $x(t) \xrightarrow{F} x(t+1) \xrightarrow{F} x(t+2) \to \dots$

Recurrent network

$$x_{t+1} = F(x_t)$$

$$x_{t+1} = x_t + h(x_t)$$ prior: $F \simeq Id$

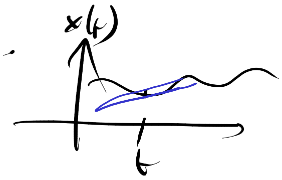
$$x_{t+dt} = x_t + h(x_t)$$ learn h specific for dt

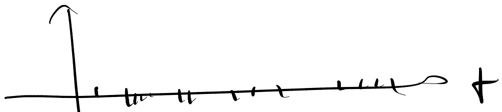
Linear approximation: $$x_{t+dt} = x_t + dt\,h(x_t) + O(dt^2)$$

$t \quad t+dt \quad t+2dt \quad t$

not specific to dt anymore

Exact time discretization: $x_{t+dt} = x_t + dt\, h(x_t) + \frac{1}{2} dt^2 g(x_t) + \dots$

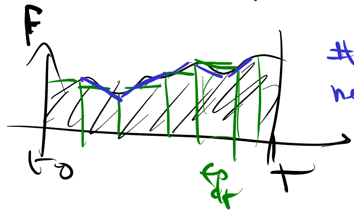




⇒ how to integrate

$$\frac{\partial x}{\partial t} = f(x,t) \qquad x(T) = \int_{t=0}^{T} \frac{\partial x}{\partial t}\, dt + x(0) = \int_{t=0}^{T} f(x,t)\, dt + x(0)$$

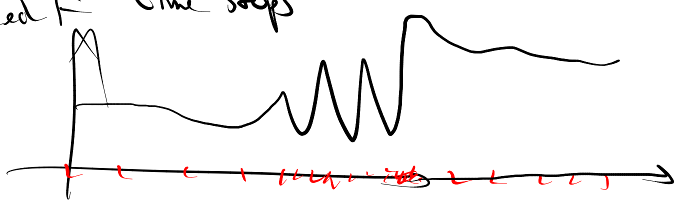
speed
(infinitesimal change)



time # steps needed (to reach given accuracy)

≪ naive discr.

doing better?
classical solvers for integration
Runge–Kutta



Train a NN to predict the speed f:

$$\frac{\partial x}{\partial t} = f(x(t))$$

$x \longrightarrow \boxed{f} \longrightarrow \frac{\partial x}{\partial t}$

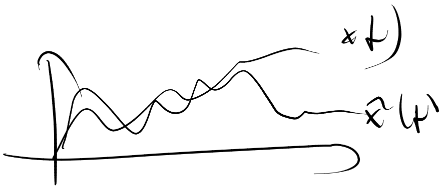
Neural ODE (ODE-net)

call a classical integration solver S

Loss: samples with true trajectory

$$\sum_t \| \hat{x}(t) - x(t) \|^2$$

or $x(T)$



$$\hat{x}(t) = S(f_\theta, x_0, t_0, t)$$

back-prop? $\frac{\partial S}{\partial \theta}$?

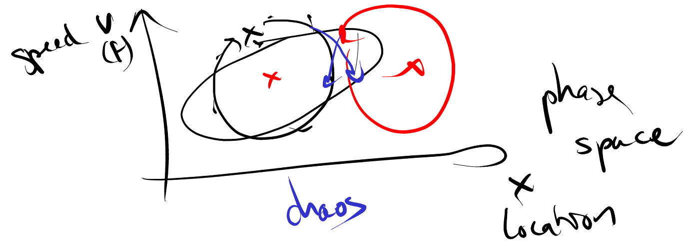
$$\frac{d\hat{x}(t)}{d\theta} = S\left(\frac{\partial \hat{x}(t)}{\partial \theta_t}, \dots\right)$$

adjoint method

$$\hat{x}(t) = \hat{x}(t-dt) + dt\, f_{\theta_t}$$

Learnability of dynamical systems

- difficulty of a ML task $\propto$ expon(intrinsic dim of data)
- fluid mechanics: multi-scale : nearly an infinite # of degrees of freedom (Kolmogorov)
  ↳ same law (at first order) at all scales
  → learn
- chaotic / non chaotic



butterfly effect

Lyapunov exponents
↳ quantifying noise amplification

right statistics

Koopman – Von Neumann

Any dynamic system can be rewritten as a linear dynamic system under the right representation

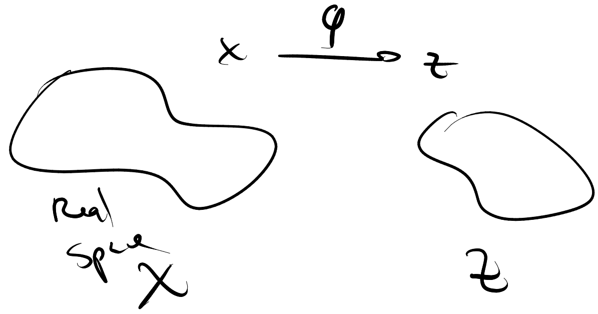
arbitrarily complex

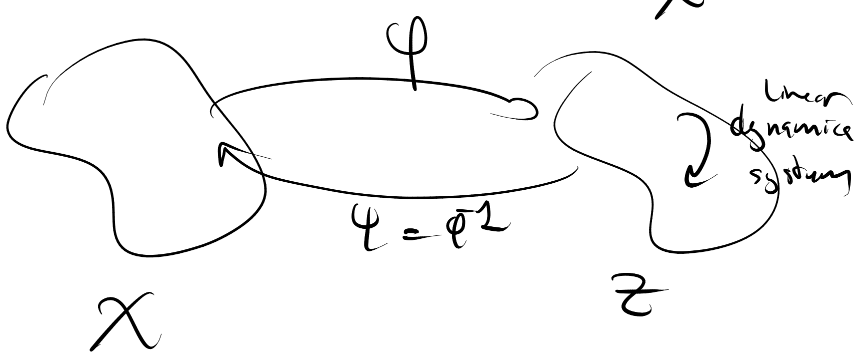
$$\frac{\partial x}{\partial t} = f(x)$$

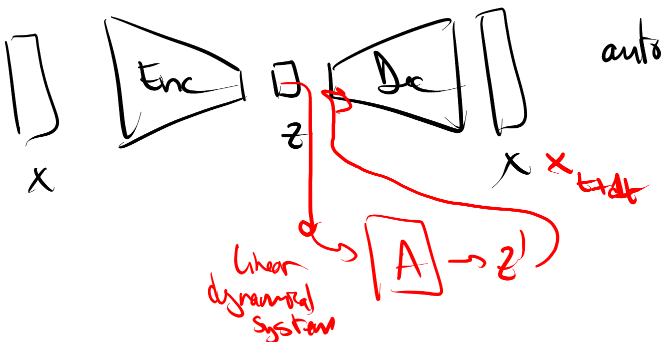
$$\frac{\partial z}{\partial t} = A z$$

matrix

$$\exists \varphi, \quad \frac{\partial z}{\partial t} = A z$$







auto-encoder

$$\varphi^{-1}(A^k \varphi(x)) = x_{t+kdt}$$

Given Enc/Dec : best A,
(φ, ψ) best linear system : linear regression → "closeform" solution A^*

λ compute how much dynamics in z are linear

$$\inf_{\varphi} \mathbb{E}_x \left[\| A^* \varphi(x_t) - \varphi(x_{t+dt}) \| \right]$$

$$\inf_{\varphi} \sup_{A} \mathbb{E}_x \left[\| A^* \varphi(x_t) - \varphi(x_{t+dt}) \|^2 \right]$$