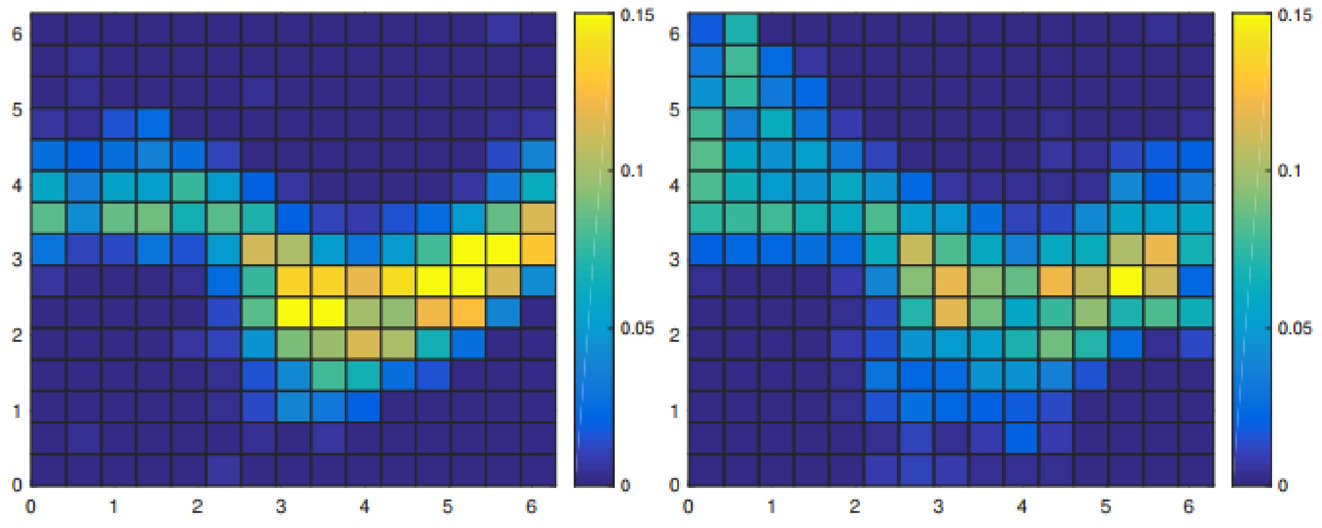
background

具体问题：求解invariant measure



(a) Ground truth invariant measure (b) Predicted invariant measure

目前的解法： 交互粒子方法（Interacting Particle Method，IPM）是一种用于求解偏微分方程（Partial Differential Equation，PDE）的数值解法。它属于粒子方法（Particle Method）的一种。

粒子方法是一类基于离散粒子的数值计算方法，它将连续的物理问题离散化为离散的粒子集合，并通过模拟粒子之间的相互作用来近似求解连续问题。交互粒子方法是粒子方法的一种变体，它通过模拟粒子之间的相互作用来模拟和求解偏微分方程。

在交互粒子方法中，粒子之间的相互作用被建模为力、能量或其他物理量的交换。通过模拟粒子之间的相互作用，交互粒子方法可以近似求解各种类型的偏微分方程，包括椭圆型、抛物型和双曲型方程。

交互粒子方法在求解偏微分方程时具有一些优点。首先，它可以处理复杂的几何形状和边界条件，因为粒子可以自由移动和交互。其次，交互粒子方法可以自适应地调整粒子的密度和分布，以适应问题的特性和解的变化。此外，交互粒子方法还可以处理高维问题和多尺度问题。

然而，交互粒子方法也存在一些挑战和限制。例如，粒子之间的相互作用需要进行精确建模，并且计算复杂度较高。此外，粒子方法在处理高精度和长时间尺度问题时可能面临数值稳定性和计算效率的挑战。

综上所述，交互粒子方法是一种用于求解偏微分方程的数值解法，它通过模拟粒子之间的相互作用来近似求解连续问题。它在处理复杂几何形状、自适应调整和多尺度问题方面具有优势，但也面临一些挑战和限制。

motivation

1. IPM从uniform到invariant measure的收敛时间长
2. DNN在生成分布上很有优势，可以数据驱动，拟合复杂问题, 训练好的端到端模型推理复杂度低。
3. 改进?
   - 如何将IPM和DNN结合起来?
   - 先用DNN(DPM)生成类似的invariant measure再用IPM精确求解

methodology(eg.2D)

data:

- uniform distribution, X:[H,W,C]
- hyper-parameters:physical parameter η, 对应微分方程的一些参数
- target distribution (generated by IPM), Y:[H,W,C]

model:2D->d=2

- input ${(x_{i,r}, η_r)}_i \in \mathbb{R^{d+p}}$, $i = 1 \cdots N$, 坐标点padding物理参数
- output $f_\theta(x;η)$, $r = 1 \cdots N_{dict}$
- hypernetwork: input η, output w,b of layer_k_2

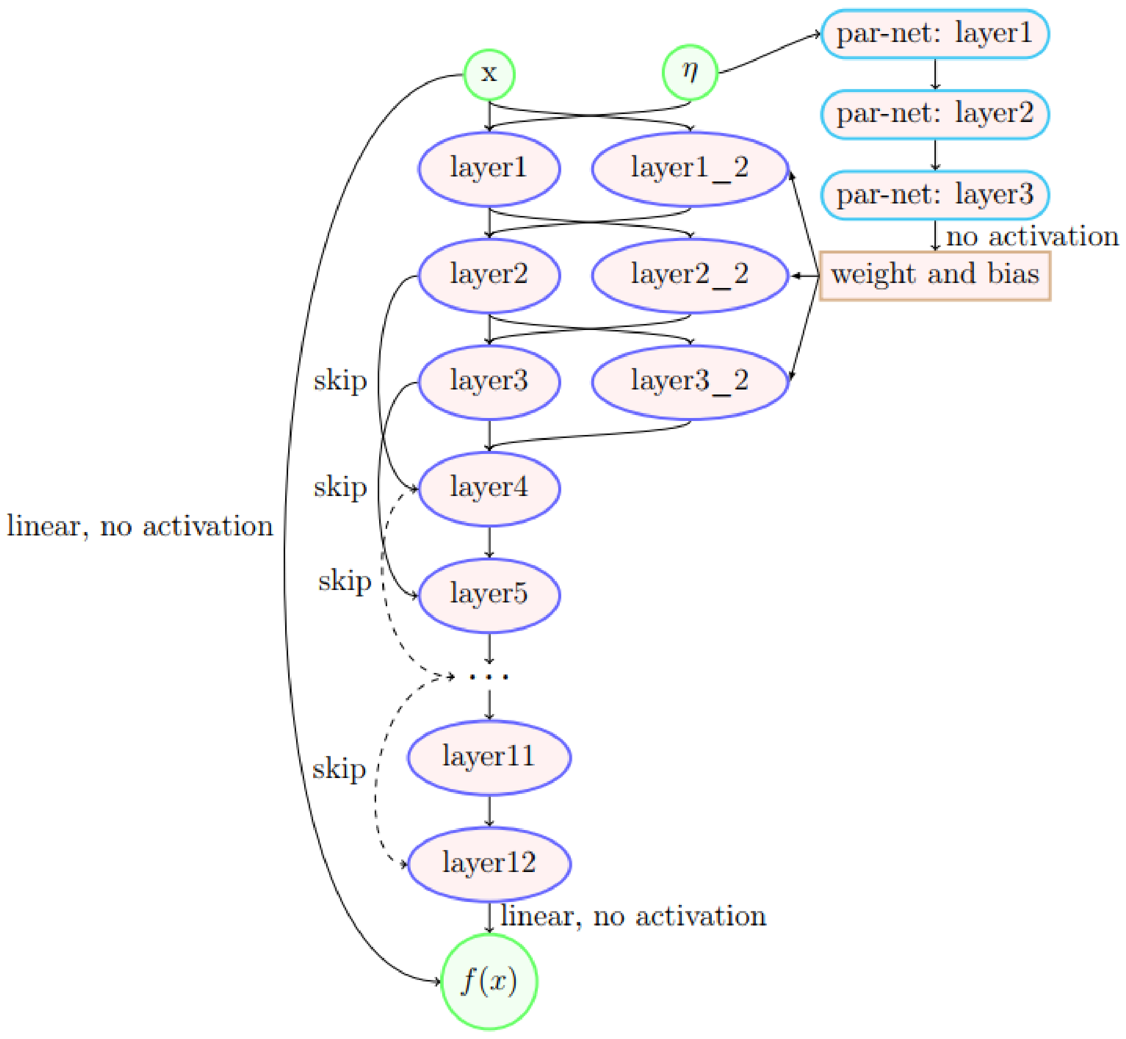


Figure 1: Layout of the proposed deep network where η is physical parameter input for learning its implicit dependence in the output $f(x)$.

Optimization: solve: subproblem LP

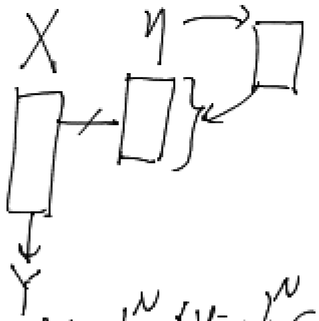
Algorithm 1: Random Pivot Search

```
Result: Given transition matrix $\gamma$, randomly search $M$ rows/columns with
        largest elements ("pivots").
Randomly pick $i_1$ as the first row;
find $j_1$ such that $\gamma_{i_1 j_1}$ is the largest among $\{\gamma_{i_1 1}, \cdots, \gamma_{i_1 N}\}$;
for $k \leftarrow 2$ to $M$ do
    find $i_k$ such that $\gamma_{i_k j_{k-1}}$ is the largest among $\{\gamma_{i' j_{k-1}}\}_{i' \in \{1,\cdots,N\} \backslash \{i_1,\cdots,i_{k-1}\}}$;
    find $j_k$ such that $\gamma_{i_k j_k}$ is the largest among $\{\gamma_{i_k j'}\}_{j' \in \{1,\cdots,N\} \backslash \{j_1,\cdots,j_{k-1}\}}$;
end
```

Algorithm 2: DeepParticle Learning

```
Randomly initialize weight parameters $\theta$ in network $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$;
repeat
    for physical parameter set $r \leftarrow 0$ to $n_\eta$ do
        randomly select $\{x_{i,r}\}$, $\{y_{j,r}\}$, $i, j = 1 : N$ from i.i.d. samples of input and
         target distribution with respect to physical parameter $\eta_r$;
        $\gamma_{ij,r} = 1/N$;
    end
    if not the first training mini-batch then
        for physical parameter set $r \leftarrow 0$ to $n_\eta$ do
            $P_r = \sum_{i,j=1}^{N} |f_\theta(x_{i,r}, \eta_r) - y_{j,r}|^2 \gamma_{ij,r}$;
            while $||\gamma_r||_{fro} < tol$ do
                randomly (or by Alg.1) choose $\{i_{k,r}\}_{k=1}^{M}$, $\{j_{l,r}\}_{l=1}^{M}$ from
                 $\{1, 2, \cdots, N\}$ without replacement;
                solve the linear programming sub-problem (10)-(11) to get $\gamma_r^*$;
                update $\{\gamma_{i_{k,r} j_{l,r}}\}_{k,l=1}^{M}$ with $\{\gamma^*_{i_{k,r} j_{l,r}}\}_{k,l=1}^{M}$.
            end
        end
    end
    repeat
        $P = \sum_{r=1}^{N_r} \sum_{i,j=1}^{N} |f_\theta(x_{i,r}, \eta_r) - y_{j,r}|^2 \gamma_{ij,r}$;
        $\theta \leftarrow \theta - \delta_1 \nabla_\theta P$, $\delta_1$ is the learning step size;
        repeat
            for physical parameter set $r \leftarrow 0$ to $n_\eta$ do
                $P_r = \sum_{i,j=1}^{N} |f_\theta(x_{i,r}, \eta_r) - y_{j,r}|^2 \gamma_{ij,r}$;
                randomly (or by Alg.1) choose $\{i_{k,r}\}_{k=1}^{M}$, $\{j_{l,r}\}_{l=1}^{M}$ from
                 $\{1, 2, \cdots, N\}$ without replacement;
                solve the linear programming sub-problem (10)-(11) to get $\gamma_r^*$;
                update $\{\gamma_{i_{k,r} j_{l,r}}\}_{k,l=1}^{M}$ with $\{\gamma^*_{i_{k,r} j_{l,r}}\}_{k,l=1}^{M}$.
            end
        until given linear programming steps, $N_{LP}$;
    until given steps for each training mini-batch;
until given number of training mini-batches, $N_{dict}$;
Return
```

$$W_p(\mu,\nu) := \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{Y \times Y} dist(y', y)^p \, d\gamma(y', y) \right)^{\frac{1}{p}}$$



① $p = 2$
② $dist = \|\cdot\|_2^2$
③ $\int \to \sum^N \quad d\gamma \to \frac{1}{N}\gamma$
④ $y' = f(x)$, $X = Y = R^d$
⑤ $r = 1 \cdots N_{dict}$ i.i.d. samples of $\{x_{i,r}\}_1^N, \{y_{i,r}\}_1^N \subset R^d$
physical parameter: $\eta_r \in R^p$ } $f_\theta(x_{i,r}; \eta_r)$
trainable parameter: θ
mean loss of $n_\eta = N_{dict}$ sets of train data

$$L(f) = \hat{W}^2(f_\theta) := \frac{1}{N n_\eta} \sum_{r=1}^{n_\eta} \left(\inf_{\gamma_r \in \Gamma^N} \sum_{i,j=1}^{N} |f_\theta(x_{i,r}; \eta_r) - y_{j,r}|^2 \gamma_{ij,r} \right)$$

Γ^N: set of doubly stochastic matrix: $\gamma = [\gamma_{ij}]_{N \times N}$

☆ $\gamma_{ij} \geq 0 \quad \forall j \; \sum_{i=1}^{N} \gamma_{ij} = 1 \quad \forall i \; \sum_{j=1}^{N} \gamma_{ij} = 1$

OPT: LP on bounded convex set Γ^N

$$\min \; P(\theta, \{\gamma_r\}) := \sum_{r=1}^{n_\eta} \sum_{i,j=1}^{N} (|f_\theta(x_{i,r}; \eta_r) - y_{j,r}|^2 \gamma_{ij,r})$$

s.t. $\gamma \in \Gamma^N$

Γ^N: possible relation of $f(x)$ and y $\backsim$ $p(f(x), y)$

Iterative algorithm Solve LP : $\longrightarrow$ find opt γ

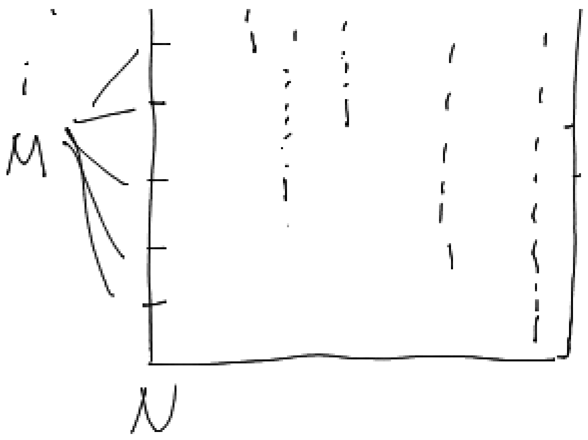
Subproblem: new $\gamma^* \leftarrow$ solve sub LP of γ :

$$\min \; C(\gamma^*) := \sum_{k,l=1}^{M} |f_\theta(x_{i_k}) - y_{j_l}|^2 \gamma^*_{i_k, j_l}$$

$i_k, j_l \in \{1 \cdots N\}$
$M \ll N$
without replacement

s.t.
$$\begin{cases} \sum_{k=1}^{M} \gamma^*_{i_k, j_l} = \sum_{k=1}^{M} \gamma_{i_k, j_l} & \forall l = 1 \cdots M \\ \sum_{l=1}^{M} \gamma^*_{i_k, j_l} = \sum_{l=1}^{M} \gamma_{i_k, j_l} & \forall k = 1 \cdots M \\ \gamma^*_{i_k, j_l} \geq 0 & \forall k, l = 1 \cdots M \end{cases}$$

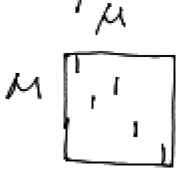
$\gamma_{i_{k,r}\, j_{k_l,r}, r = 1}$



▨ $\times n_\eta$

$p_n \downarrow \Leftrightarrow p \downarrow$

$\to$ sub problems



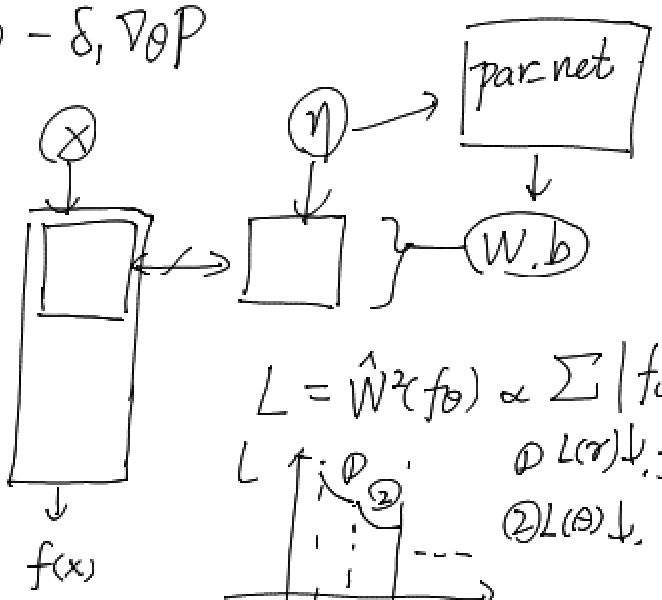
$\Gamma^* \to$ permutation matrices

e.g. $\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}$

$\gamma_{ij} = \delta_{j,\pi(i)} \in \{0,1\}$.

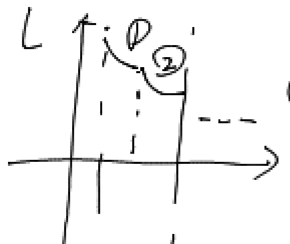
Fixed γ: update $\theta \to$ opt f_θ

$\left\{ \begin{array}{l} \text{given LP steps: } N_{LP} \quad \text{repeat } \gamma^* \leftarrow \gamma \;\; N_{LP} \text{ times} \\ \theta \leftarrow \theta - \delta_1 \nabla_\theta P \end{array} \right.$



$\square$ update by $-\nabla P$

$L = \hat{W}^2(f_\theta) \propto \sum |f_\theta(x) - y|^2 \gamma$



① $L(\gamma)\downarrow$, fix θ

② $L(\theta)\downarrow$, fix γ

layer i $\quad W_i : H \times W \quad b_i : H \times 1$

① $\eta \to$ par_net $\to \{W_i, b_i\}_i$

(i) $\eta_r \in R^{d_\eta} \to [B, H, W, D]$ $\left\{ \begin{array}{l} B = bsz = n_\eta \times N \\ H = col(W_i) \\ W = rol(W_i) \\ D = d_\eta \end{array} \right.$

(ii) W_r: $[W, D, k_1]$ layer 1 $\Longrightarrow [B, H, W, K_1]$

$\{ [W, K_1, k_2]$ layer 2 $\Rightarrow [B, H, W, k_2]$

$[W, K_2, 3]$ layer 3 $\Rightarrow [B, H, W, K_3]$ $\quad K_3 = 1$

② layer: 1~12 $\quad K_i = 20 \quad g(\cdot) = \sigma(\cdot) \quad \underbrace{W_i}$ or (b_i when $W=1$)

12: