Should one use an educated or uneducated basis?

Hongkai Zhao
Department of Mathematics
University of California, Irvine

Desired properties

for representation, analysis and understanding:

- Effective: dimension reduction, distinctive, stable, computation complexity, ...

Desired properties

for representation, analysis and understanding:

- Effective: dimension reduction, distinctive, stable, computation complexity, ...
- General?

Desired properties

for representation, analysis and understanding:

- Effective: dimension reduction, distinctive, stable, computation complexity, ...
- General?
- Problem specific?

Desired properties

for representation, analysis and understanding:

- Effective: dimension reduction, distinctive, stable, computation complexity, ...
- General?
- Problem specific?
- Intrinsic dimensionality?

Examples

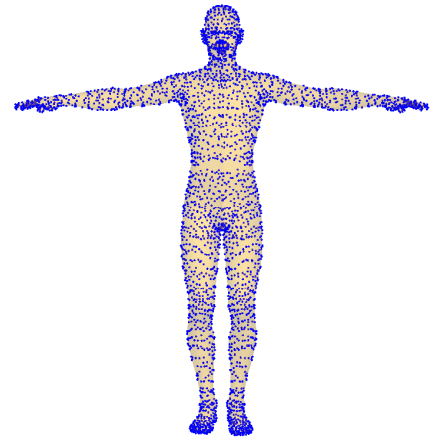
- Multiscale non-rigid point cloud registration based on Laplace-Beltrami eigenmap.
- Hamiltonian Monte Carlo acceleration using surrogate functions with random basis for Bayesian inference.
- Image classification using random basis.
- Learning dominant wave directions for high frequency wave fields.

Non-rigid point cloud registration

joint work with R. Lai

The problem

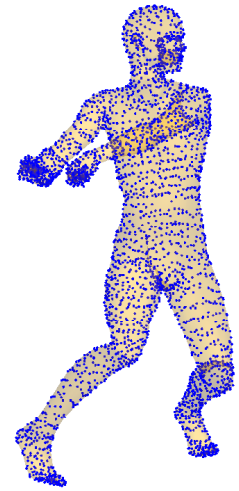
Point cloud is the most basic way for representing geometry and information in 3D and higher dimensions.



How to compare and register two point clouds?

Difficulties:

- unstructured geometric object, no natural basis or parametrization,
- embedding is not unique, e.g., rigid and non-rigid transformation,
- a point cloud primitively embedded in very high dimensional space may posses intrinsic low dimensional structure.



public available data
TOSCA

Laplace-Beltrami (LB) eigenmap

Given a d-dimensional manifold $(\mathcal{M}, g)$, the LB eigen-system $\{\lambda_n, \phi_n\}$ is

$$-\triangle_{\mathcal{M}}\phi_n = \lambda_n\phi_n,\ n = 1, 2, \cdots$$

Assuming the point cloud $\mathcal{X} = \{\mathbf{x}_i \in R^m, i = 1, 2, \ldots, N\}$ is sampled from a d-dimensional manifold $(\mathcal{M}, g)$, define the scale-invariant LB eigenmap (Rustamov'07)

$$\mathcal{L} : \mathbf{x}_i \in R^m \rightarrow \mathbf{p}_i = \left(\frac{\phi_1(\mathbf{x}_i)}{\lambda_1^{d/4}}, \frac{\phi_2(\mathbf{x}_i)}{\lambda_2^{d/4}}, \ldots, \frac{\phi_n(\mathbf{x}_i)}{\lambda_n^{d/4}} \right)^T \in R^n.$$

Nice properties of LB eigenmap

- Intrinsic GPS.

Nice properties of LB eigenmap

- Intrinsic GPS.
- Invariant to scaling and isometric transformation.

Nice properties of LB eigenmap

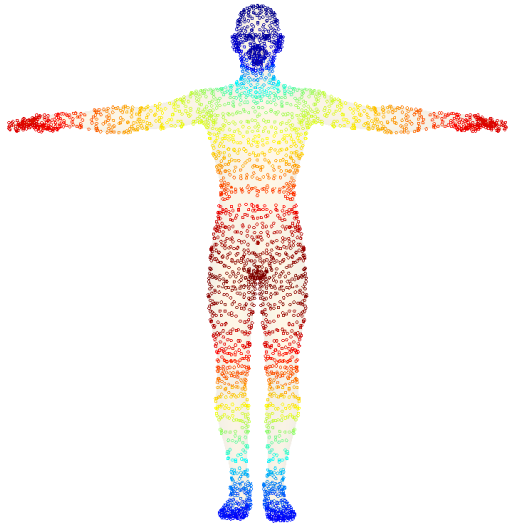
- Intrinsic GPS.
- Invariant to scaling and isometric transformation.
- A natural multiscale characterization with global information.

Nice properties of LB eigenmap

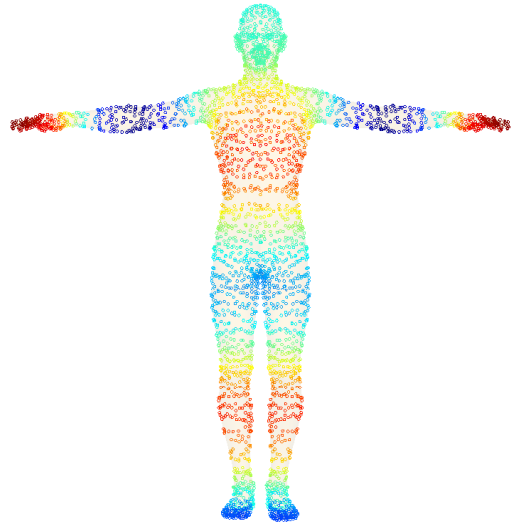
- Intrinsic GPS.
- Invariant to scaling and isometric transformation.
- A natural multiscale characterization with global information.
- A good candidate for nonlinear dimension reduction for point clouds in higher dimensions.

Eigenfunctions of LB operator on point clouds

n = 1

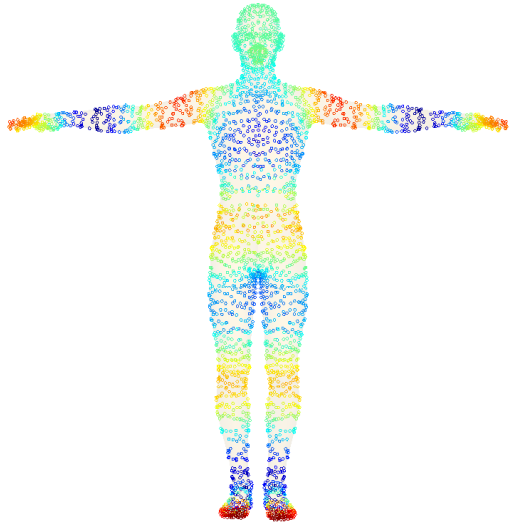


n = 5

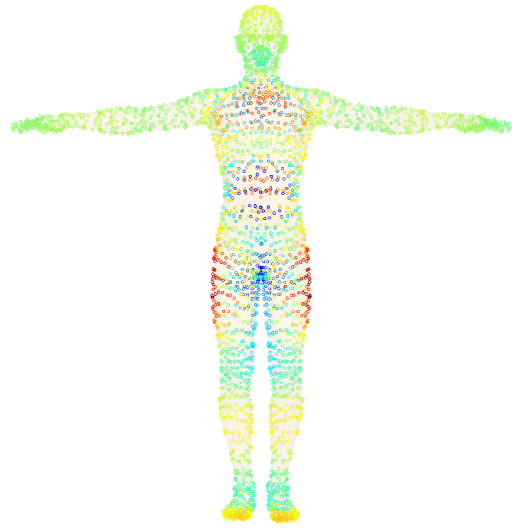


n = 10

n = 15

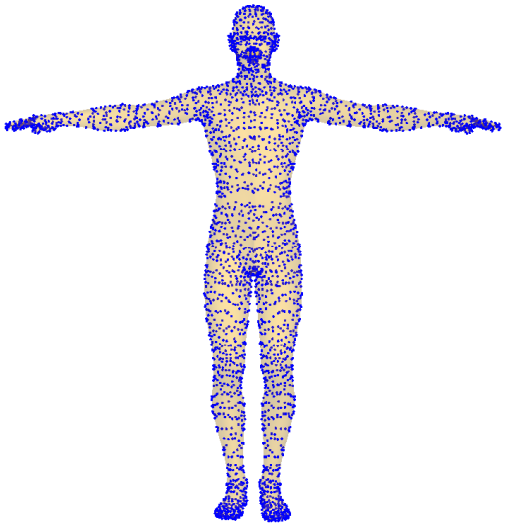


n = 20

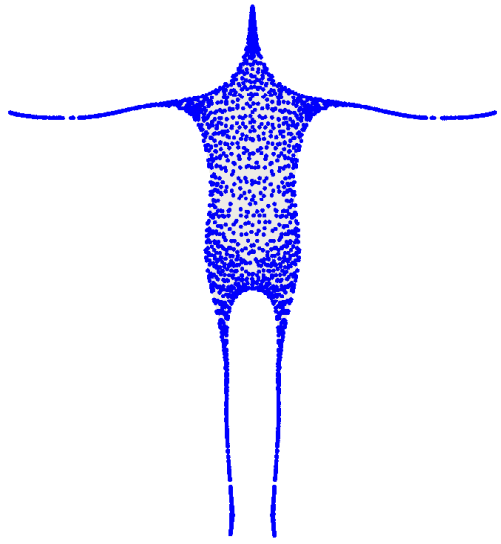


n = 25

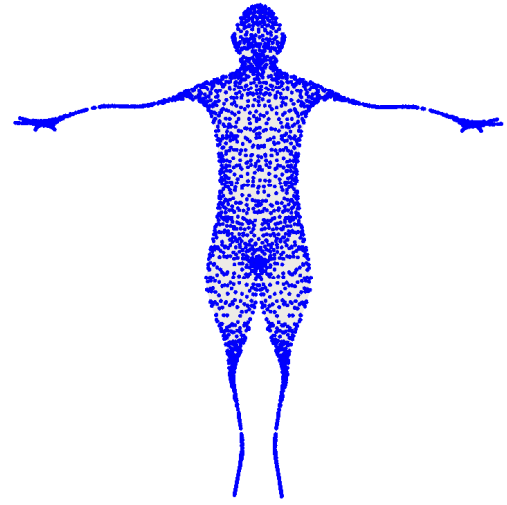
Multiscale representation of point clouds by LB eigen-modes



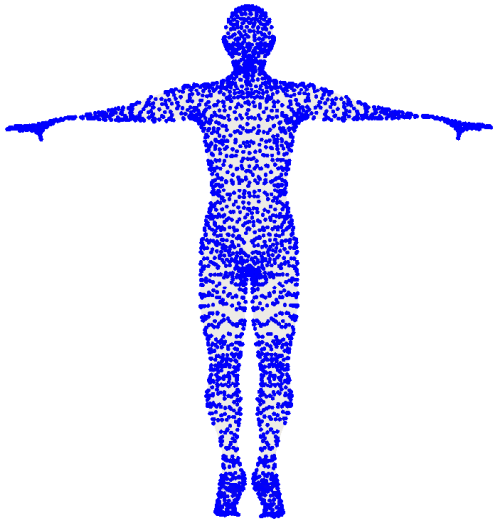
original



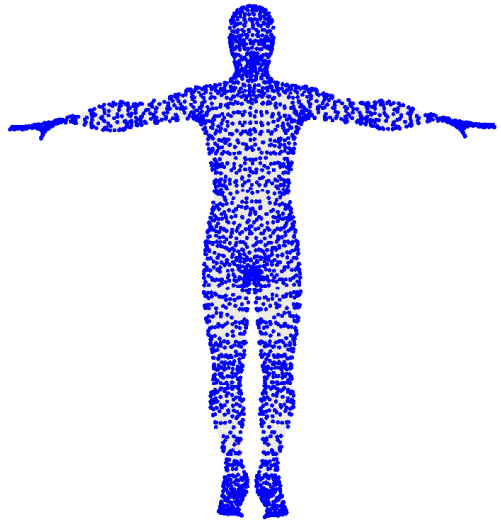
the first 20



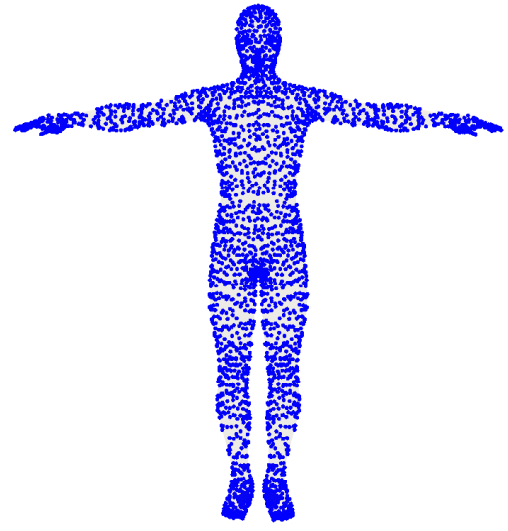
the first 50



the first 100

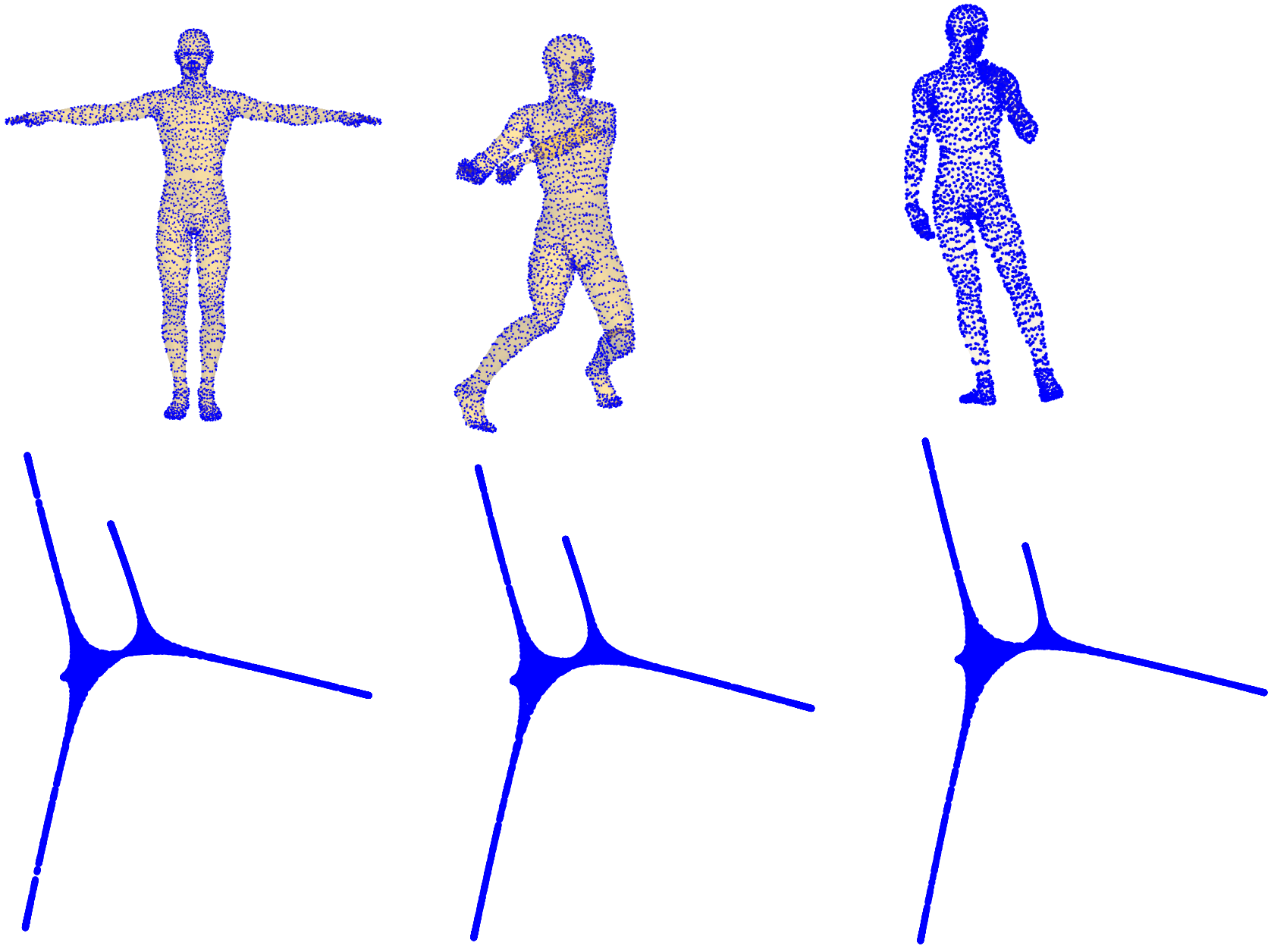


the first 150

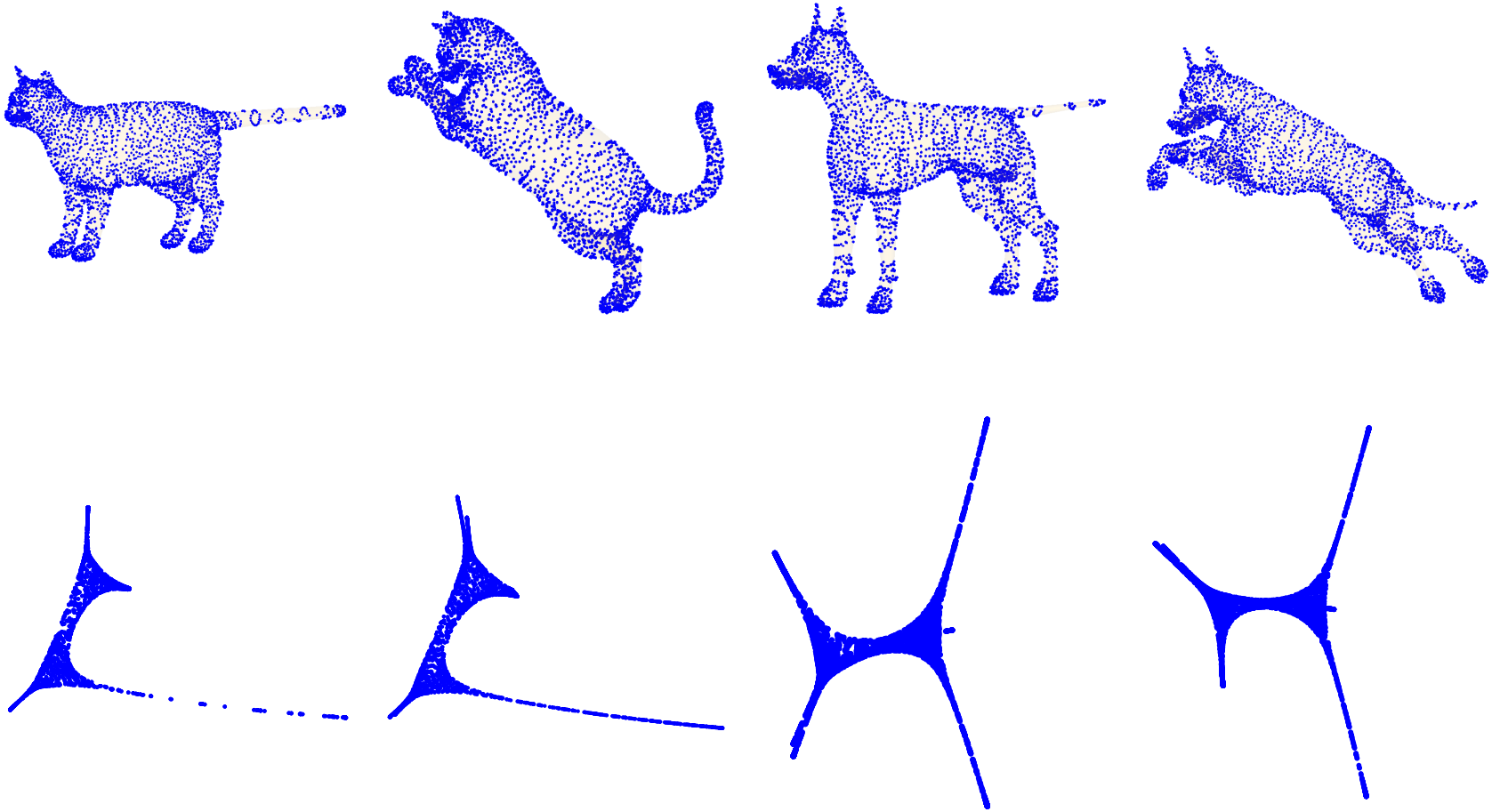


the first 200

Iso-metric invariance of LB eigen-maps

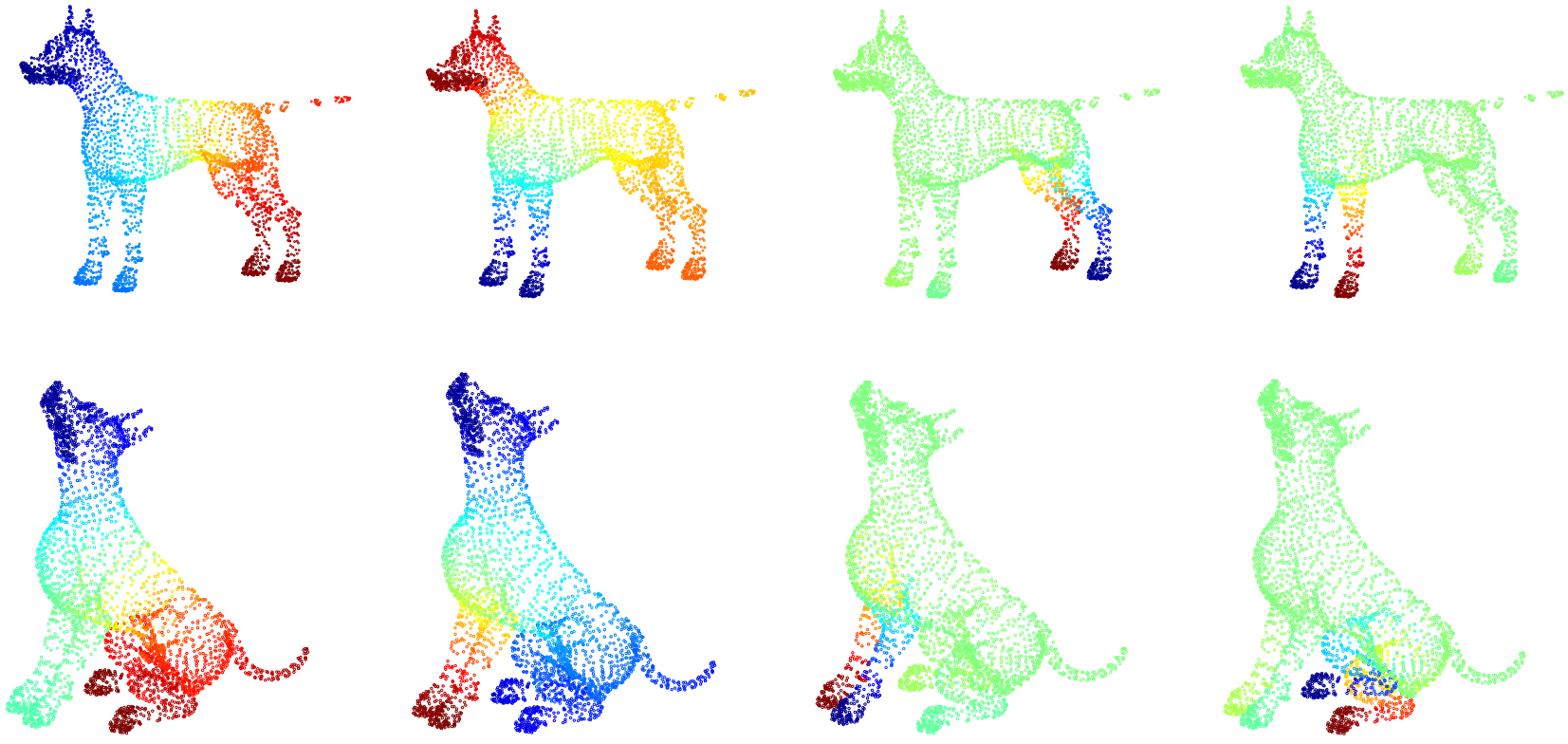


Iso-metric invariance of LB eigen-maps



New issues for LB eigen-map

- possible sign ambiguity of LB eigenfunctions;
- non simple eigenspace;
- possible order switch of LB eigenfunctions.



ϕ_1 ϕ_2 ϕ_3 ϕ_4

Figure: Ambiguities of LB eigen-map.

Key features of our method

- Handle non-rigid transformation using LB eigen-map.
- A registration in distribution sense based on optimal transport.
- Overcome the ambiguity of LB eigen-map by introducing rotation invariant Wasserstein distance in the LB embedding space.
- Achieve computation efficiency by using the rotation invariant sliced-Wasserstein distance.
- A multiscale implementation that improves both robustness and efficiency.
- Work for other embeddings and point clouds in high dimensions .

Registration of point clouds

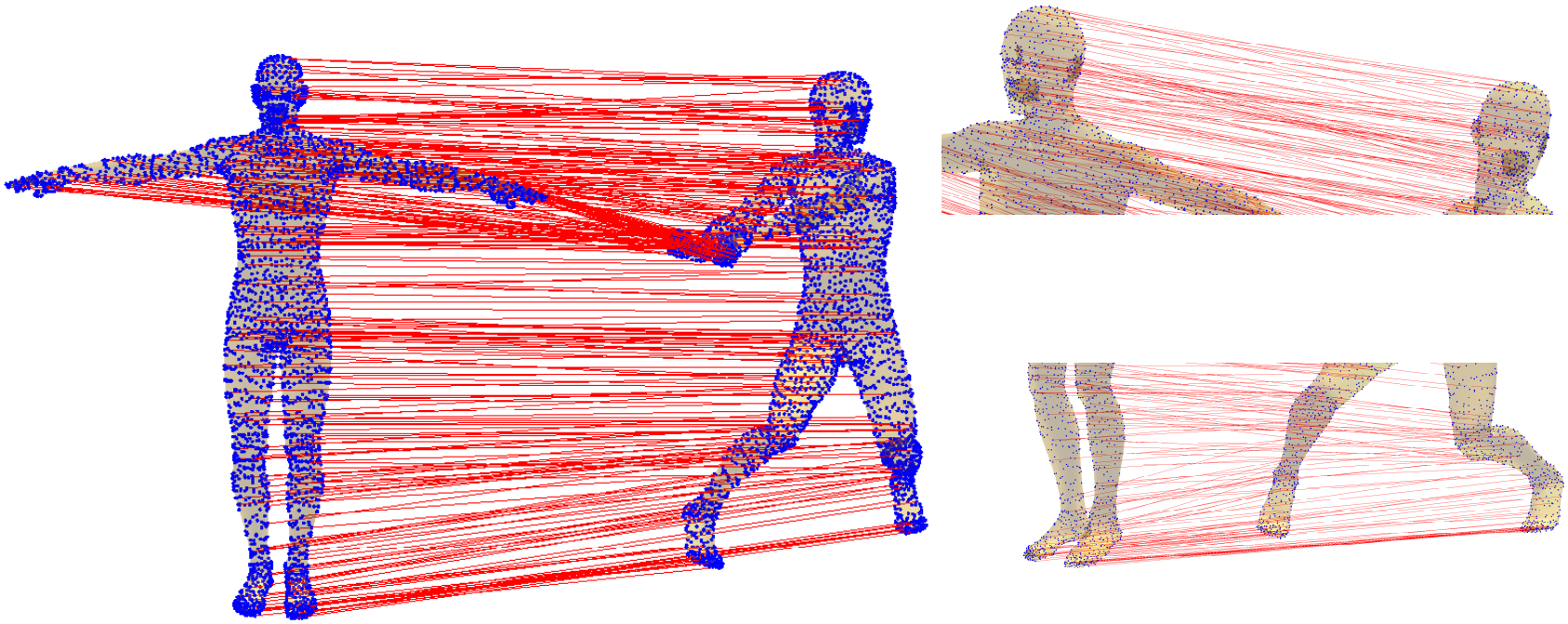
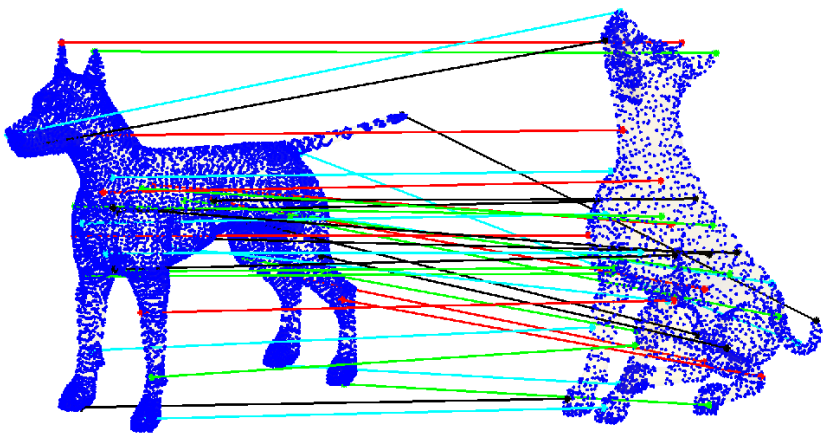
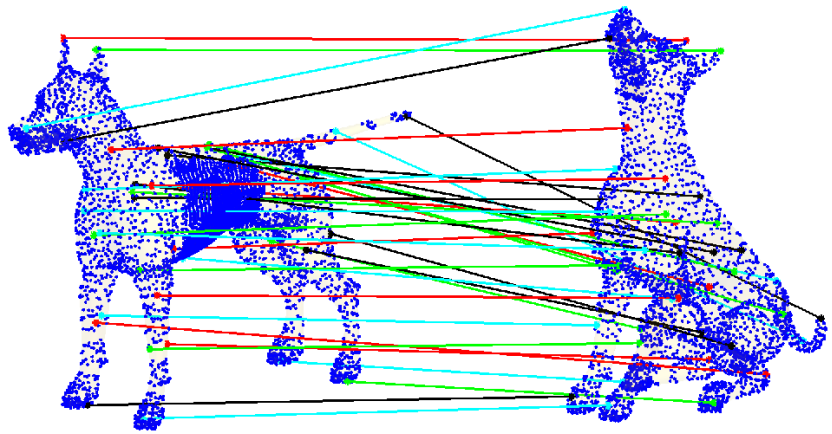


Figure: Registration using the first 5 egienfunctions. Num. of points = 3400. Computation time = 9.03s

Registration of non-uniform data

By defining registration in distribution, one has more flexibilities by introducing appropriate mass/weight to each data set in the optimal transport problem. For example, to deal with non-uniform data one can chose the wright according to the local sampling density.

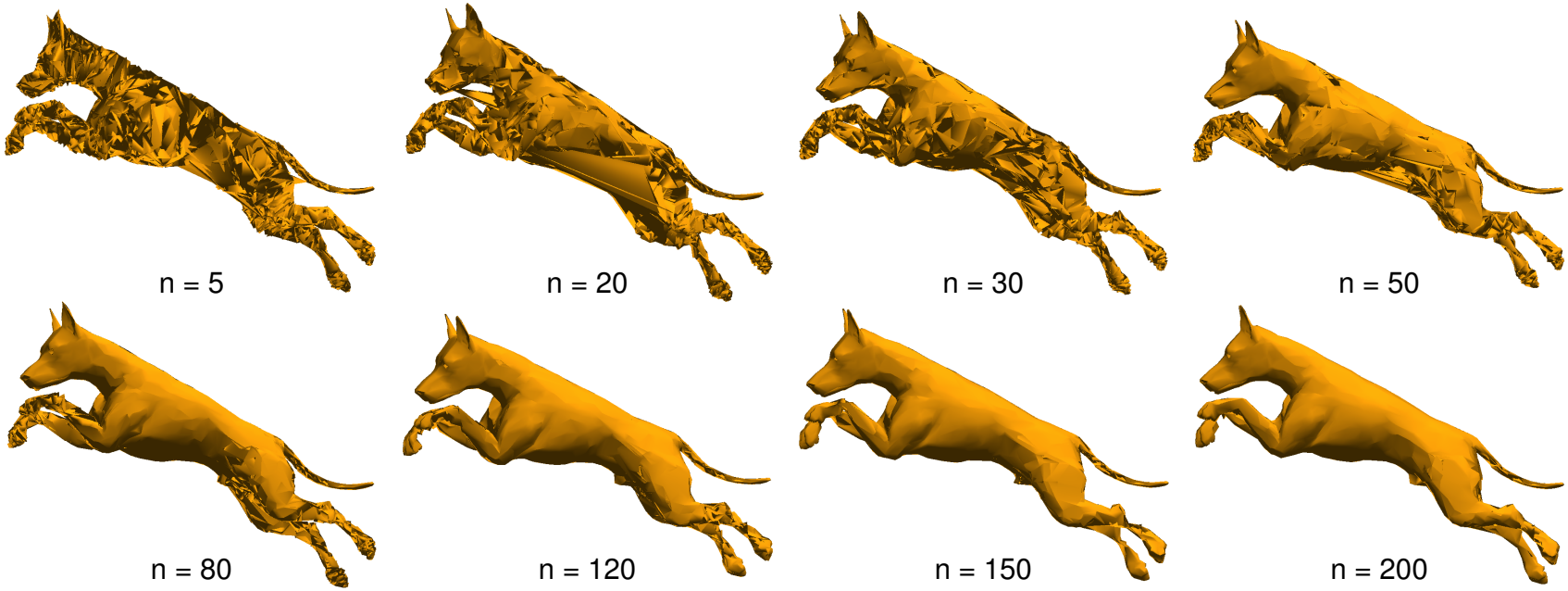




Multiscale registration

Computation time

method	Time (s)							
	$n_1 = 5$	$n_2 = 10$	$n_3 = 20$	$n_4 = 30$	$n_5 = 50$	$n_6 = 80$	$n_7 = 120$	$n_8 = 150$
	$L = 500$	$L = 800$	$L = 1000$	$L = 1500$	$L = 3000$	$L = 6000$	$L = 10000$	$L = 15000$
RSWD	0.61	1.21	1.50	2.26	4.75	9.58	16.50	26.23
empirical	0.45	.70	0.83	1.23	2.30	4.76	7.68	11.67



Efficient and scalable computational models for Bayesian inference

joint work with B. Shahbaba and C. Zhang

Bayesian inference

Given a set of independent observations $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ modeled by an underlying distribution $p(\mathbf{y}|\mathbf{q})$ with parameters $\mathbf{q} = \{q_1, \ldots, q_d\}$, one is interested in sampling from the posterior distribution,

$$p(\mathbf{q}|Y) = \frac{p(Y|\mathbf{q})p(\mathbf{q})}{p(Y)} \propto \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{q}) \cdot p(\mathbf{q}).$$

Given a probabilistic model for the observed data, Bayesian inference properly quantifies uncertainty and reveals the global landscape of the model.

Computational challenge for Bayesian inference

- exact posterior inference in practice is often intractable.
- most approximation models and their algorithms are computationally intensive, e.g., MCMC method, especially for big data and high dimensions.

Computational challenge for Bayesian inference

- exact posterior inference in practice is often intractable.
- most approximation models and their algorithms are computationally intensive, e.g., MCMC method, especially for big data and high dimensions.

The key question: how to explore the underlying structure of the model to propose effective sample states with

- reduced computational cost,
- high acceptance rate,
- fast convergence.

Hamiltonian Monte Carlo (HMC)

HMC introduces a Hamiltonian dynamic system that includes gradient of the potential to generate proposals

$$\frac{d\mathbf{q}}{dt} = \frac{\partial H}{\partial \mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}}$$

where $H(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}, \quad U(\mathbf{q}) = -\sum_{i=1}^{N} \log P(\mathbf{y}_i|\mathbf{q}) - \log P(\mathbf{q}).$

from a joint distribution $P(\mathbf{q}, \mathbf{p}) \propto \exp\left(-U(\mathbf{q}) - \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}\right)$.

Hamiltonian Monte Carlo (HMC)

HMC introduces a Hamiltonian dynamic system that includes gradient of the potential to generate proposals

$$\frac{d\mathbf{q}}{dt} = \frac{\partial H}{\partial \mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}}$$

where $H(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}, \quad U(\mathbf{q}) = -\sum_{i=1}^{N} \log P(\mathbf{y}_i|\mathbf{q}) - \log P(\mathbf{q}).$

from a joint distribution $P(\mathbf{q}, \mathbf{p}) \propto \exp\left(-U(\mathbf{q}) - \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}\right)$.

- HMC reduces the random walk behavior and proposes states with a high probability of acceptance.
- Riemannian HMC uses Fisher information (Hessian) for M to incorporate local geometric structure.

Hamiltonian Monte Carlo (HMC)

HMC introduces a Hamiltonian dynamic system that includes gradient of the potential to generate proposals

$$\frac{d\mathbf{q}}{dt} = \frac{\partial H}{\partial \mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}}$$

where $H(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}, \quad U(\mathbf{q}) = -\sum_{i=1}^{N} \log P(\mathbf{y}_i|\mathbf{q}) - \log P(\mathbf{q}).$

from a joint distribution $P(\mathbf{q}, \mathbf{p}) \propto \exp\left(-U(\mathbf{q}) - \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}\right)$.

- HMC reduces the random walk behavior and proposes states with a high probability of acceptance.
- Riemannian HMC uses Fisher information (Hessian) for M to incorporate local geometric structure.

Computational challenge: repetitive evaluations of U, its derivatives, M^{-1},

- N is large (big data),
- the number of parameter is large (high dimensions),
- the model is complex.

Random network surrogate (RNS) HMC

Key idea: construct a random network surrogate for U that implicitly subsample the data by

- exploring redundancy in the data and regularity in the parameter space
- effectively capturing the collective properties of large datasets
- with scalability, flexibility and efficiency

by using a linear combination of nonlinear basis functions with random parameterization through efficient optimization from training data, i.e.,

$$U(\mathbf{q}) \approx \sum_{i=1}^{s} \beta_i a(\mathbf{q}; \boldsymbol{\gamma}_i),$$

where $a(x)$ is the nonlinear basis function, $\boldsymbol{\gamma}_i$ is the random parameter, and β_i are learned/trained from the available samples.

Remark: Universal approximation is theoretically guaranteed (Cybenko, Hornick, Huang, Rahimi and Recht).

Random network surrogate (RNS) HMC

Key features of our method:

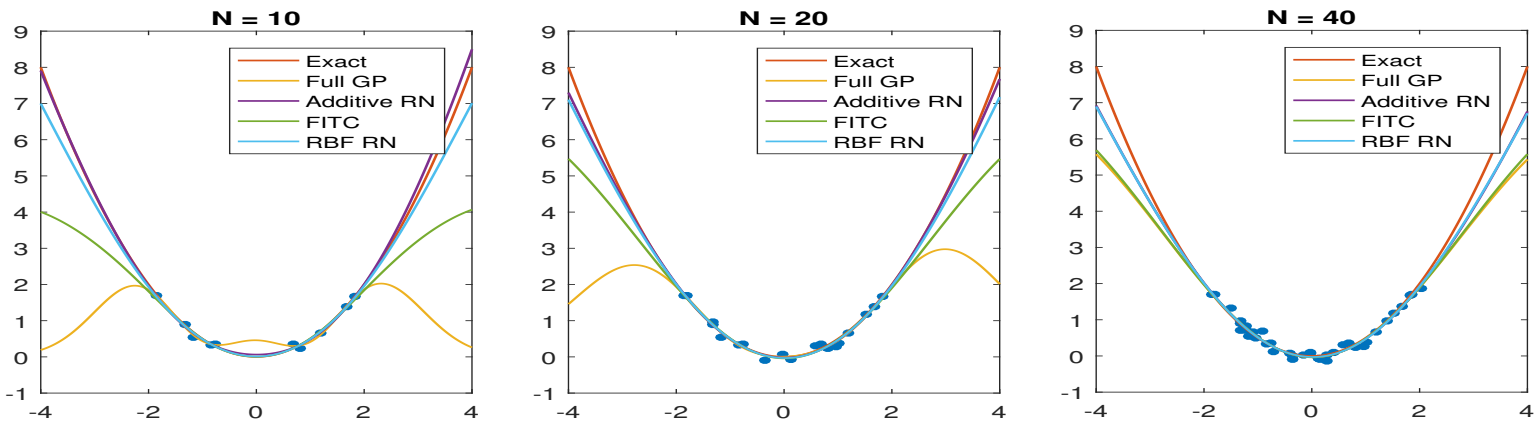
- Exploration phase: efficient optimization to train the surrogate function.
- Exploitation phase: the surrogate function is used to approximate the potential function and its derivatives in the HMC.
- Scalability and flexibility: free form approximation, different choices of basis functions, the number of hidden nodes can be adjusted to achieve desired accuracy and stability.
- Adaptivity: sampling and learning can be coupled on the fly.

Basis function

Our basis function with random orientation, scaling and offset $\{\mathbf{w}, b\}$:

$$a(\mathbf{q}, \gamma) = a(\mathbf{w}^t \cdot \mathbf{q} + b), \quad a(x) = \ln(1 + e^x), \quad \gamma = \{\mathbf{w}, b\}$$

which has a non-local nature and proper behavior at infinity to approximate a potential function.



Comparing different surrogate approximations with an increasing number of observations $N = 10, 20, 40$ on target function $y = x^2/2$. The observation points are nested samples from the standard normal distribution.

Optimization

Given training samples in the parameter space at $\mathbf{q}_i \in R^d, i = 1, \ldots, t$,

- Least square fitting the potential function at training samples.

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{t} \| \sum_{j=1}^{s} \beta_j a(\mathbf{q}_i, \gamma_j) - U(\mathbf{q}_i) \|^2 (+ \frac{\lambda}{2} \|\beta\|^2)$$

- Least square fitting the gradient of the potential function using training samples (score matching).

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{t} \| \sum_{j=1}^{s} \beta_j \nabla_q a(\mathbf{q}_i, \gamma_j) - \nabla_q U(\mathbf{q}_i) \|^2 (+ \frac{\lambda}{2} \|\beta\|^2)$$

- The computation cost is $O(tN + dst + ts^2 + s^3)$, linear in N, the number of data, and in t, the number of training samples.
- The number of hidden nodes, s, can be used to balance accuracy and stability.

Generalization of random network surrogate HMC

If radial basis function centered at training sample points $\mathbf{q}_i, i = 1, \ldots, t$ is used

$$a(\mathbf{q}, \gamma_i) = c^2 exp(-\frac{\|\mathbf{q} - \mathbf{q}_i\|^2}{2l^2}) = K(\mathbf{q}, \mathbf{q}_i), \quad \gamma_i = \{\mathbf{q}_i, l\}$$

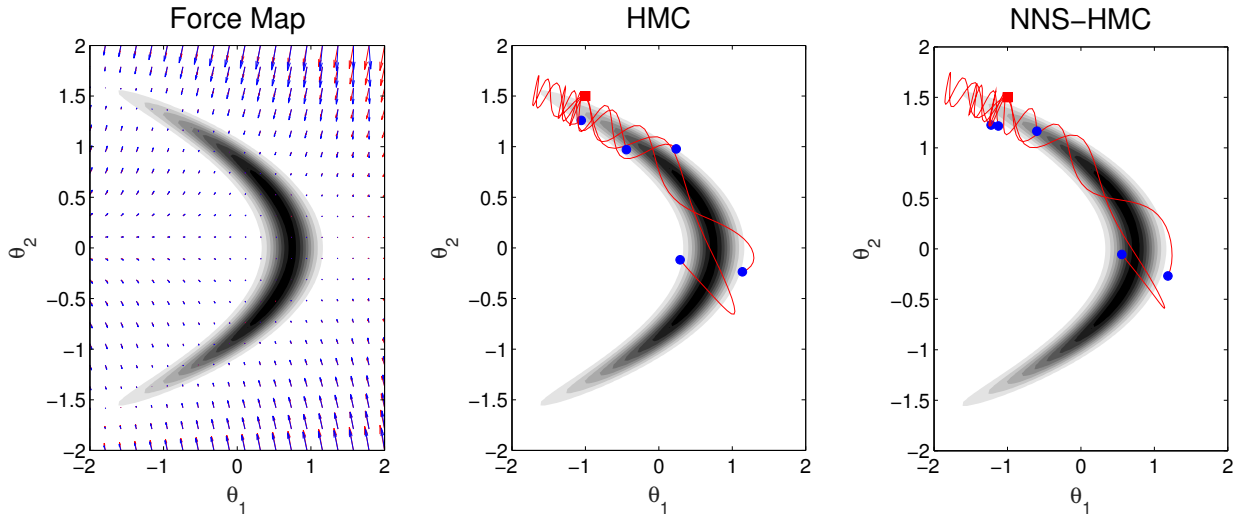
to fit the training data $(\mathbf{q}_i, U(\mathbf{q}_i)), j = 1, \ldots, t$ by the following optimization

$$\min_{\beta} \frac{1}{2} \sum_{j=1}^{t} \| \sum_{i=1}^{t} \beta_j a(\mathbf{q}_i, \gamma_j) - U(\mathbf{q}_i) \|^2 + \frac{1}{2} \sigma^2 \beta^T K_t \beta, \quad K_t = [K(\mathbf{q}_i, \mathbf{q}_j)]_{i,j=1}^{t}$$

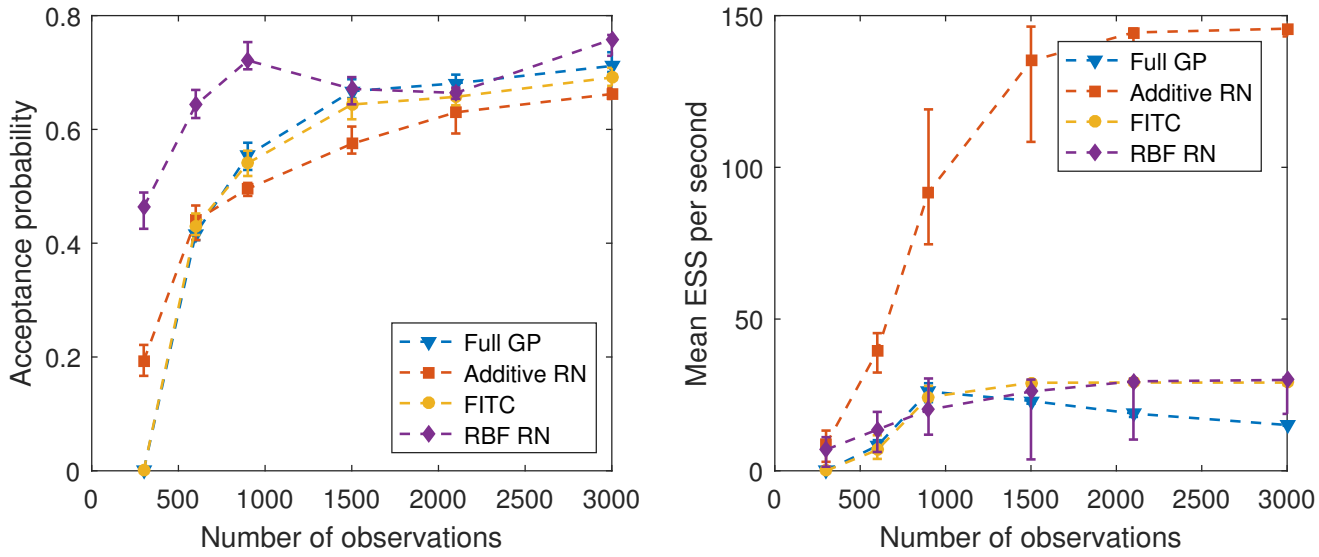
one recovers the popular Gaussian process (GP) model.

Remark: The computation cost is $O(t^3)$ to invert the covariance matrix K_t.

Tests

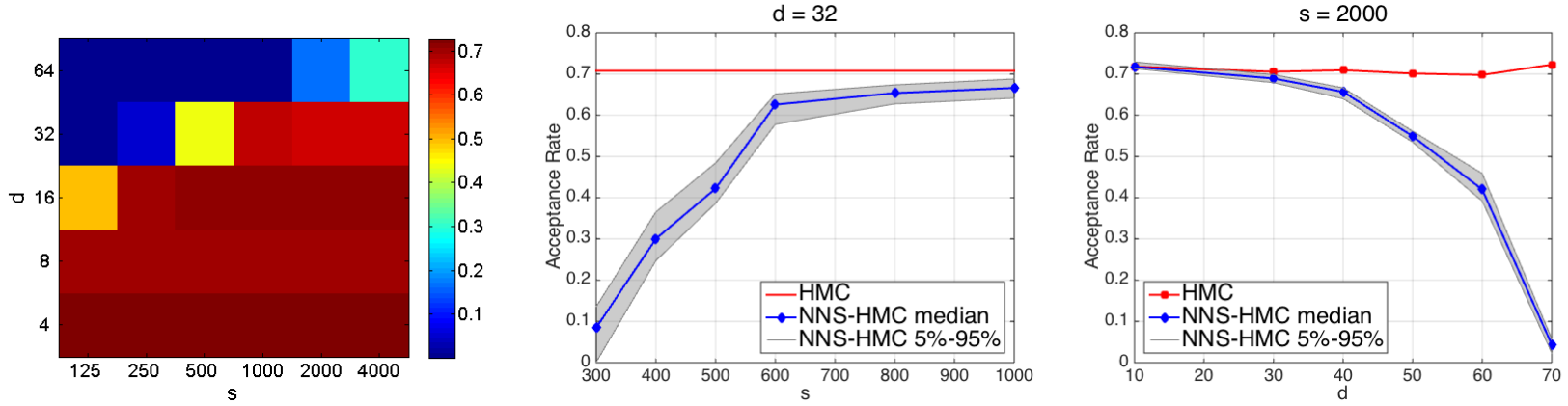


Comparing HMC and NNS-HMC for a 2-dimensional banana-shaped distribution.

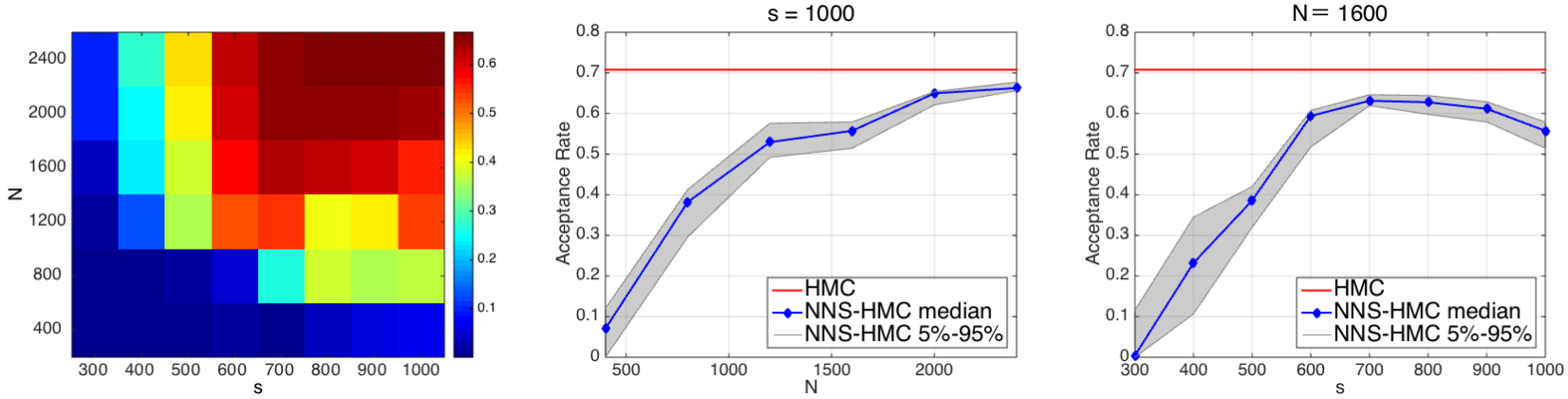


Comparison of efficiency on a 32 dimensional Gaussian target whose covariance matrix has one eigenvalue of 1, and all others are 0.01. Each algorithm is run ten times. The medians and 80% error bars are plotted.

Tests



Acceptance probability of surrogate induced Hamiltonian flow on simulated logistic regression models for different number of parameters, d, and hidden neurons, s.



Acceptance probability of the surrogate induced Hamiltonian flow based on a simulated logistic regression models with dimension $d = 32$.

An inverse problem for elliptic PDE

$$\nabla_{\mathbf{x}} \cdot (c(\mathbf{x}, \theta) \nabla_{\mathbf{x}} u(\mathbf{x}, \theta)) = 0, \quad \mathbf{x} = (x_1, x_2) \in [0, 1]^2$$

with boundary conditions A log-Gaussian process prior for $c(\mathbf{x})$ with covariance kernel:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{2l^2}\right), \quad \sigma = 1, \ l = 0.2.$$

The diffusivity field is approximated by Karhunen-Loéve (K-L) expansion:

$$c(\mathbf{x}, \theta) \approx \exp\left(\sum_{i=1}^{d} \theta_i \sqrt{\lambda_i} v_i(\mathbf{x})\right)$$

where λ_i and $v_i(\mathbf{x})$ are the eigenvalues and eigenfunctions of the integral operator defined by the kernel C.

An inverse problem for elliptic PDE

$$\nabla_{\mathbf{x}} \cdot (c(\mathbf{x}, \theta) \nabla_{\mathbf{x}} u(\mathbf{x}, \theta)) = 0, \quad \mathbf{x} = (x_1, x_2) \in [0, 1]^2$$

with boundary conditions A log-Gaussian process prior for $c(\mathbf{x})$ with covariance kernel:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{2l^2}\right), \quad \sigma = 1, \ l = 0.2.$$

The diffusivity field is approximated by Karhunen-Loéve (K-L) expansion:

$$c(\mathbf{x}, \theta) \approx \exp\left(\sum_{i=1}^{d} \theta_i \sqrt{\lambda_i} v_i(\mathbf{x})\right)$$

where λ_i and $v_i(\mathbf{x})$ are the eigenvalues and eigenfunctions of the integral operator defined by the kernel C.

The problem: inference for parameters θ_i endowed with independent standard normal priors, $\theta_i \sim \mathcal{N}(0, 0.5^2)$.

The experiment: The K-L expansion is truncated at $d = 20$. Data are generated by adding independent Gaussian noise to observations of the solution field,

An inverse problem for elliptic PDE

$$\nabla_{\mathbf{x}} \cdot (c(\mathbf{x}, \theta) \nabla_{\mathbf{x}} u(\mathbf{x}, \theta)) = 0, \quad \mathbf{x} = (x_1, x_2) \in [0, 1]^2$$

with boundary conditions A log-Gaussian process prior for $c(\mathbf{x})$ with covariance kernel:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{2l^2}\right), \quad \sigma = 1, \ l = 0.2.$$

The diffusivity field is approximated by Karhunen-Loéve (K-L) expansion:

$$c(\mathbf{x}, \theta) \approx \exp\left(\sum_{i=1}^{d} \theta_i \sqrt{\lambda_i} v_i(\mathbf{x})\right)$$

where λ_i and $v_i(\mathbf{x})$ are the eigenvalues and eigenfunctions of the integral operator defined by the kernel C.

The problem: inference for parameters θ_i endowed with independent standard normal priors, $\theta_i \sim \mathcal{N}(0, 0.5^2)$.

The experiment: The K-L expansion is truncated at $d = 20$. Data are generated by adding independent Gaussian noise to observations of the solution field,

Remark. This is a complex probability model where a PDE is involved. Each evaluation of the potential function and its derivatives in HMC is extremely expensive.

Results

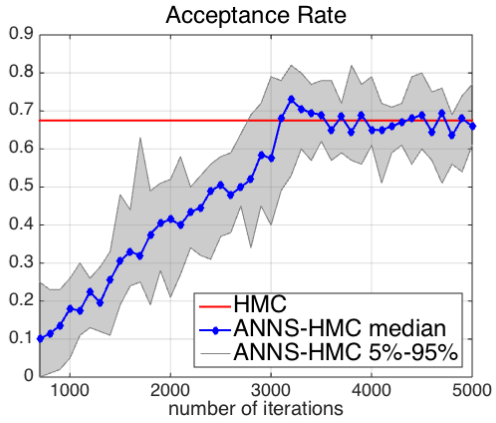
Experiment	Method	AP	ESS	s/Iter	$\min(\mathrm{ESS})/s$	speedup
LR (Simulation)	HMC	0.76	(4351,5000,5000)	0.061	14.17	1
	RMHMC	0.80	(1182,1496,1655)	3.794	0.06	0.004
$s = 2000$	RNS-HMC	0.76	(4449,4999,5000)	0.007	123.56	**8.72**
LR (Bank Marketing)	HMC	0.70	(2005,2454,3368)	0.061	6.52	1
	RMHMC	0.92	(1769,2128,2428)	0.631	0.56	0.09
$s = 1000$	RNS-HMC	0.70	(1761,2358,3378)	0.007	52.22	**8.01**
LR (a9a 60 dimension)	HMC	0.72	(1996,2959,3564)	0.033	11.96	1
	RMHMC	0.82	(5000,5000,5000)	3.492	0.29	0.02
$s = 2500$	RNS-HMC	0.68	(1835,2650,3203)	0.005	81.80	**6.84**
Elliptic PDE	HMC	0.91	(4533,5000,5000)	0.775	1.17	1
	RMHMC	0.80	(5000,5000,5000)	4.388	0.23	0.20
$s = 1000$	RNS-HMC	0.75	(2306,3034,3516)	0.066	7.10	**6.07**

Comparing algorithms using logistic regression models and an elliptic PDE inverse problem. For each method, we provide the acceptance probability (AP), the CPU time (s) for each iteration and the time-normalized ESS.

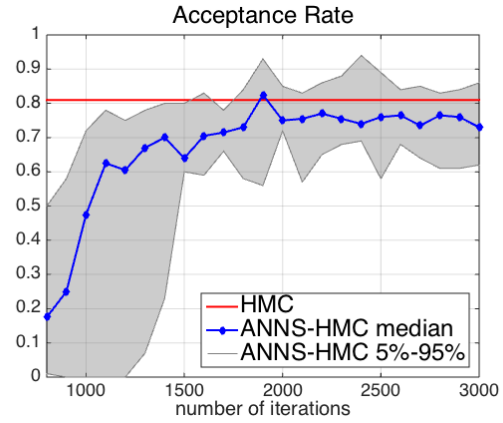
Adaptive random network surrogate HMC

- Idea: Generate the samples and training the network surrogate simultaneously.
- Catch: Need to have adaption rate of the surrogate function $c_t \to 0$ with iteration t of Markov chain to guarantee ergodicity and convergence (Roberts and Rosenthal 06).
- At the initial stage when samples are few, local quadratic (Laplace) approximation of the potential function.
- Efficient update of the random network surrogate with new samples based on Sherman-Morrison-Woodbury formula.

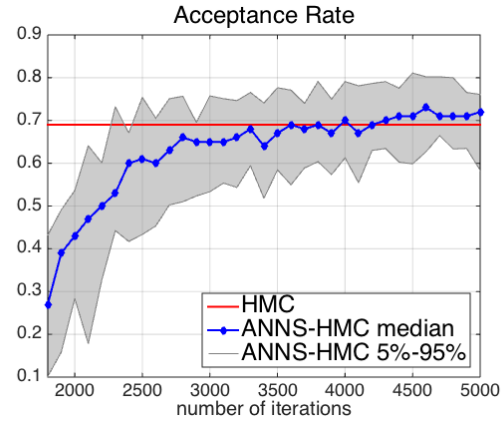
Test results



(a) Simulated Data

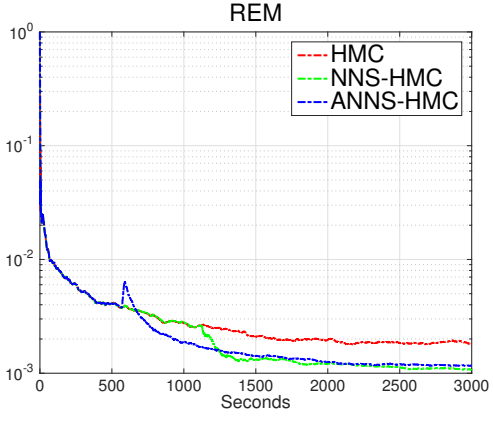


(b) Bank Market

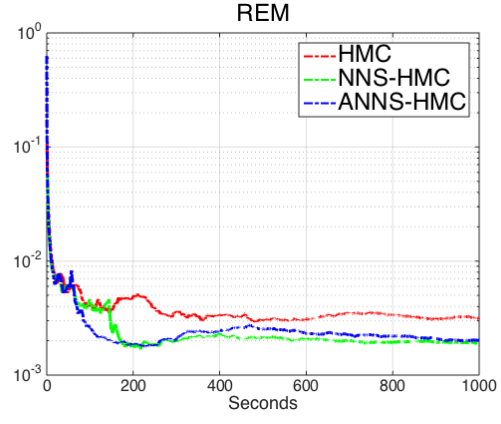


(c) Elliptic PDE

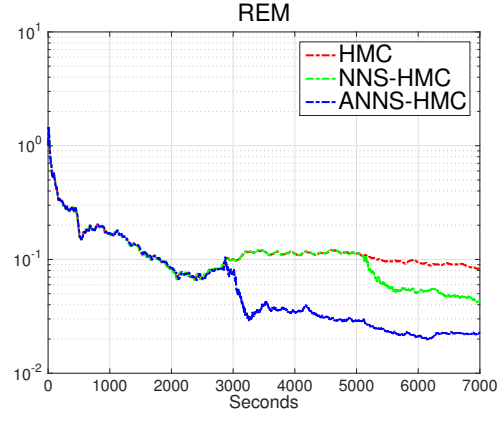
Median acceptance rate of ANNS-HMC along with the corresponding 90% interval (shaded area). The red line shows the average acceptance rate of standard HMC.



(a) Simulated Data



(b) Bank Market



(c) Elliptic PDE

Relative error of mean as a function of running time.

Neural Response based Extreme Learning Machine for Image Classification

joint work with H. Li

Key elements of our method

- A simple and efficient image classification method based on multilayer feature mapping and extreme learning machine (ELM).
- Multilayer feature mapping based on random basis and max-pooling.
- SIFT preprocessing is used to introduce more invariance properties.
- At the ELM learning stage, elastic-net regularization is used.

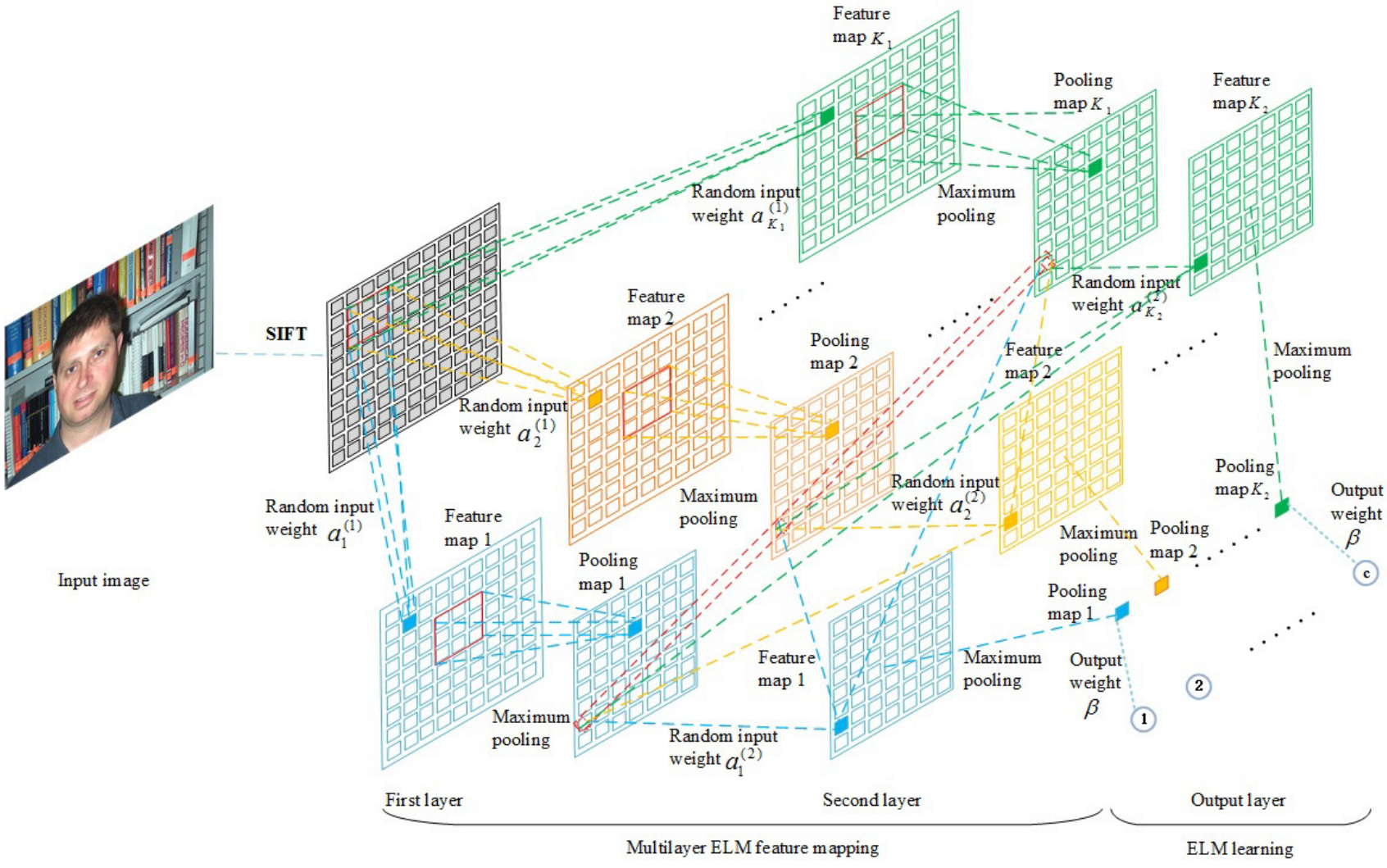
Extreme learning machine

For single layer ELM with K hidden nodes with activation function $\sigma(x)$, one tries to design output weights $\beta_j, j = 1, 2, \ldots, K$ to fit the output f on given data $\mathbf{x}_i, i = 1, 2, ...,$

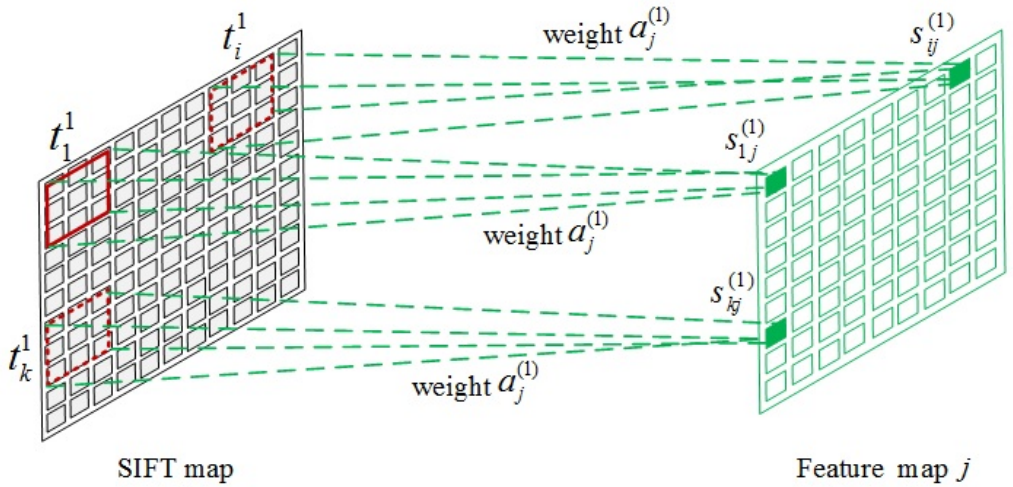
$$f(\mathbf{x}_i) = \sum_{j=1}^{K} \beta_j \sigma(\mathbf{a}_j^T \mathbf{x}_i + b_j)$$

- $\mathbf{a}_j, b_j$ are randomly chosen $\Rightarrow$ universal approximation theorem of random basis (Barron, Huang, Rahimi and Recht) + bless of dimensions.
- Linear model $\Rightarrow$ computationally efficient.
- Sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$ is used in our application.

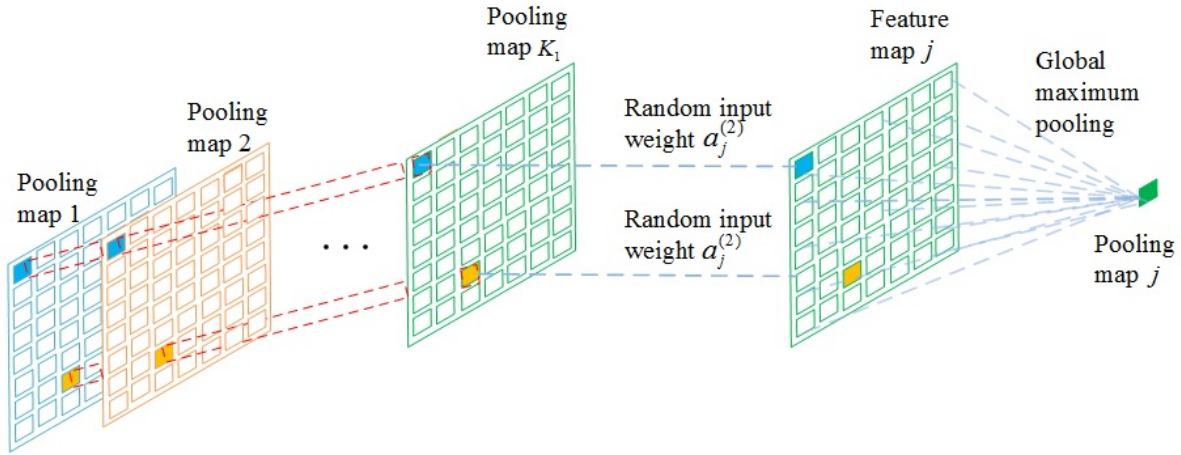
Our algorithm



Random feature map and maxpooling



first layer feature map



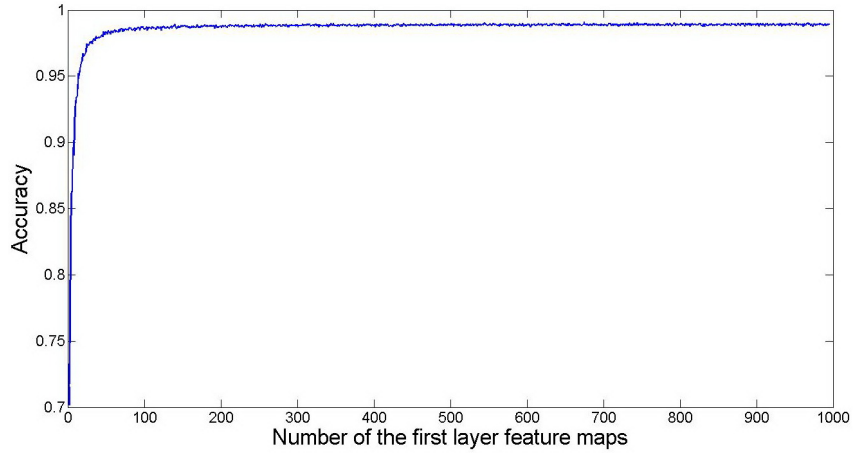
second layer feature map and max pooling

Test results

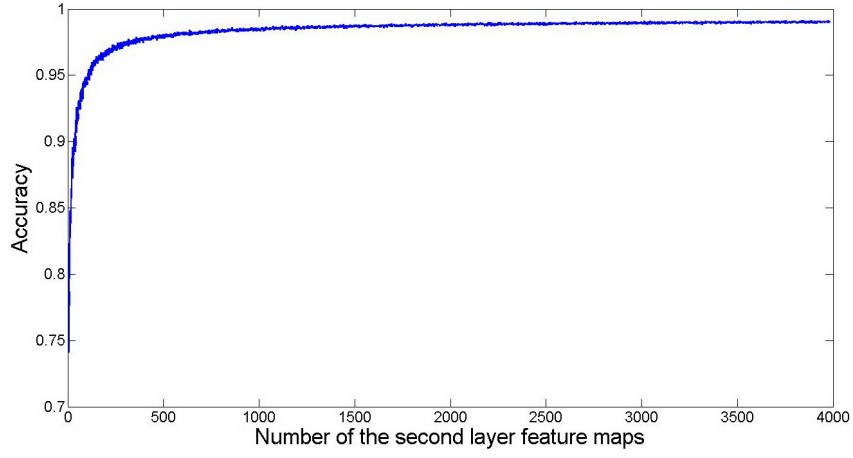
MNIST data set

Methods	Accuracy (%)	Time (s)
CNN	98.92	14,127.68
DBN	98.87	20,580.00
SAE	98.60	36,448.40
DBM	99.05	68,246.00
SDAE	98.72	37,786.03
ScSPM	98.58	13,287.79
NR model	96.43	33,619.54
SNR	96.71	>24h
Classical ELM	95.98	18.33
ML-ELM	99.03	149.47
ELM-LRF	98.35	3,172.86
NR-ELM	**99.18**	622.72

Classification results



accuracy vs the number of first layer feature maps.



accuracy vs the number of second layer feature maps.

Caltech face database.

Images with random rotation angles.

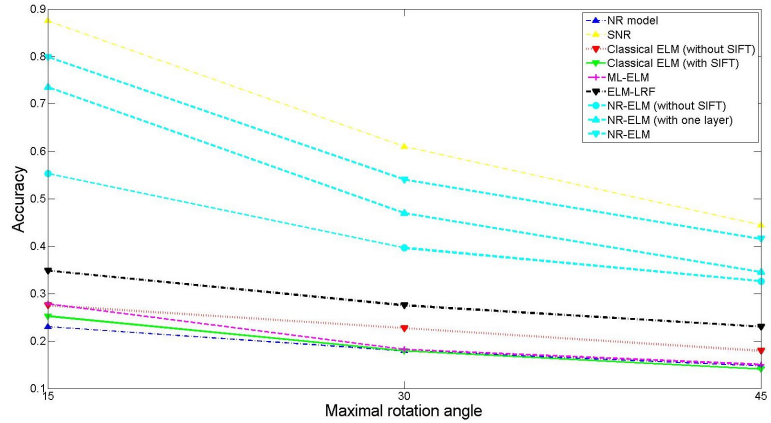
Classification results on the Caltech face database

Methods	Accuracy (%)	Time (s)
CNN	39.62	42,397.84
DBN	26.52	899.14
SAE	42.81	597.13
ScSPM	80.32	1006.40
NR	26.96	1,552.26
SNR	91.79	3,744.15
Classical ELM	38.34	0.49
ML-ELM	37.70	91.07
ELM-LRF	48.88	236.98
NR-ELM (one layer)	90.42	74.68
NR-ELM	**93.93**	144.86

The effect of using SIFT.

Methods	Accuracy (%)	Time (s)
Classical ELM (w/o SIFT)	38.34	0.49
Classical ELM (with SIFT)	37.70	42.63
NR-ELM (w/o SIFT)	69.97	887.10
NR-ELM (with SIFT)	**93.93**	144.86

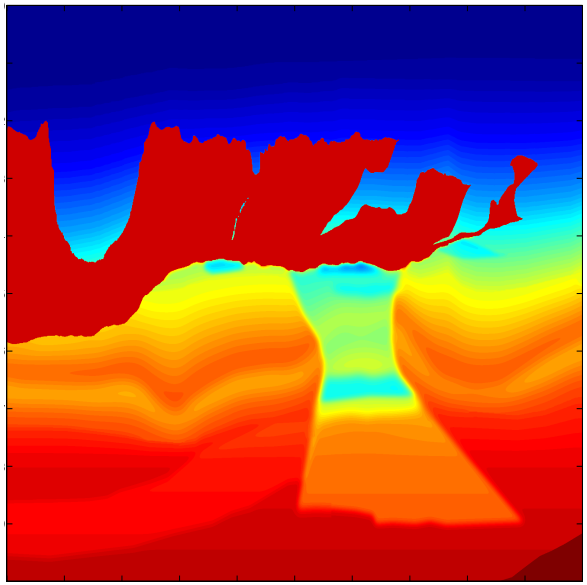
comparisons



Learning dominant wave directions for high frequency wave fields

joint work with J. Fun, J. Qian and L. Zepeda-Núñez

Helmholtz equation



wave speed: $c(\mathbf{x})$
refraction index: $n(\mathbf{x}) = \frac{1}{c(\mathbf{x})}$
frequency: ω
wavelength: $\lambda = \frac{2\pi c}{\omega}$
wave number: $k = \frac{1}{\lambda}$

Helmholtz equation

$$(\Delta + \omega^2 n^2(\mathbf{x}))u(\mathbf{x}) = f(\mathbf{x})$$

+ b. c.

Applications:

- Wave propagation
- Inverse problems
- Imaging
- Non-destructive testing

Helmholtz equation (HE) in high frequency regime

$$\Delta_x u(\mathbf{x}) + \omega^2 n^2(\mathbf{x}) u(\mathbf{x}) = f(\mathbf{x}) \quad , \quad \mathbf{x} \in \Omega \subset R^d$$

When domain size $\gg \lambda$ or high resolution is needed, $\lambda \ll 1$, $\Rightarrow$ high frequency regime: $\omega \gg 1$ ($\lambda \ll 1$), HE is notoriously difficult to solve.

- Solution is highly oscillatory.
- Waves propagate in all directions.
- Waves are strongly scattered by the medium.
- True degrees of freedom are large $\geq O(\omega^d)$.

Motivations

Design better basis functions by learning/probing the medium.

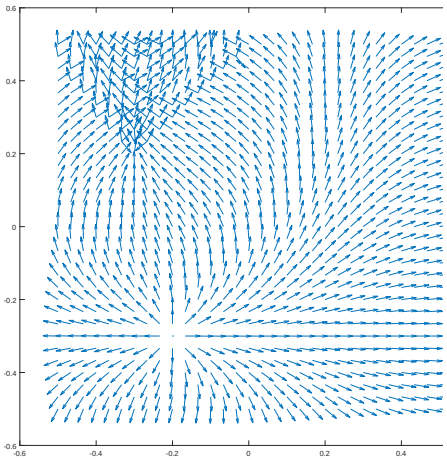
- Learn local dominant wave front directions, which depends on both the medium and source.
- Incorporate the information into basis functions to capture local oscillation pattern.
- Both local dominant wave front directions and the wave field can be updated and improved iteratively.

Ultimate Goal: overall complexity $O(\omega^d \log \omega)$.

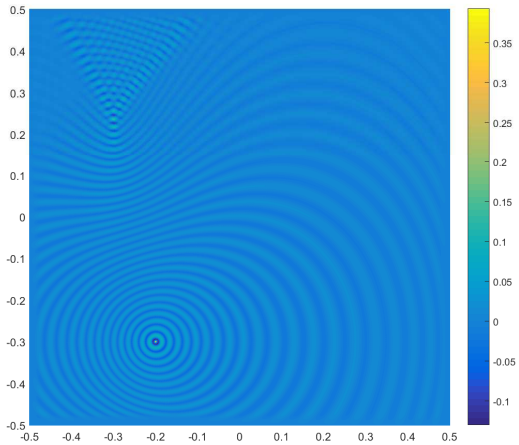
Our problem specific approach

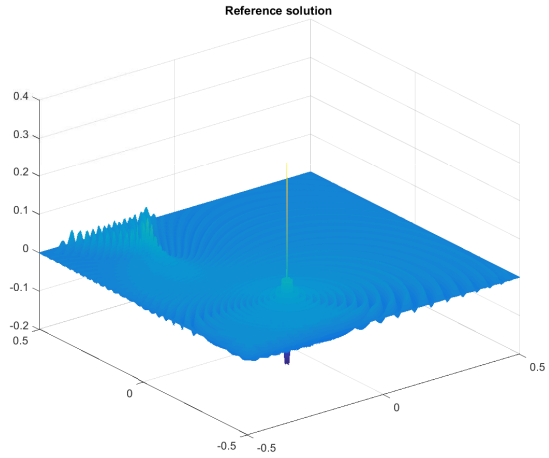
- Probe the medium using a low frequency wave.
- Process the low frequency wave field to learn the ray field.
- Local ray field information is incorporated into the basis functions to improve both efficiency and stability for computing the high frequency wave field.
- Learning is specific to the medium and source distribution.
- The ray field and wave field can be improved iteratively.

Example



ray field





wave field