

# Perspectives on Probabilistic Graphical Models

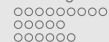
Dong Liu

*Information Science and Engineering  
Department of Intelligent Systems  
KTH - Royal Institute of Technology*

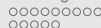


Profile page: <https://firsthandscientist.github.io/>

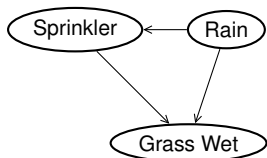
Slide is available at: <https://github.com/FirstHandScientist/phdthesis>



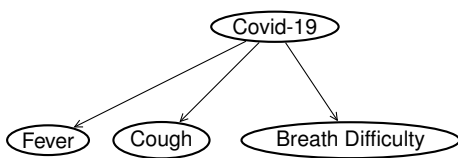
Why are probabilistic graphical models interesting?



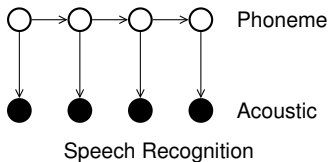
# DIRECTED GRAPH REPRESENTATION



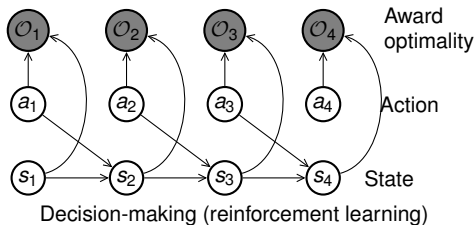
Is the sprinkler working?



Is the person infected by COVID?



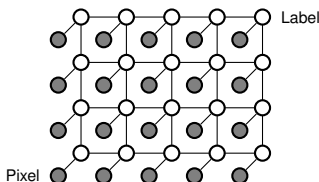
Speech Recognition



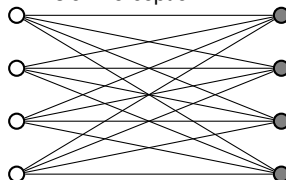
Decision-making (reinforcement learning)

# UNDIRECTED GRAPH REPRESENTATIONS

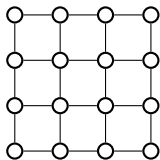
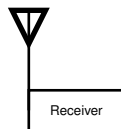
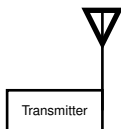
Coco



Vision Perception

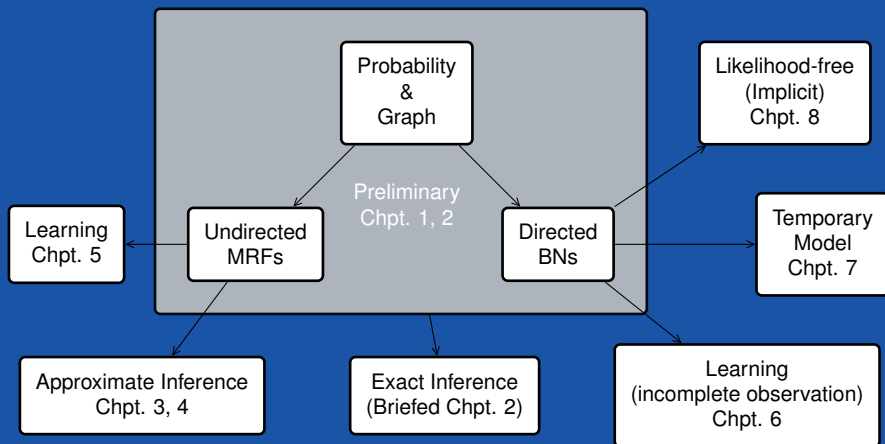


Digital communication

Physics  
Ising or Potts

- Error-control codes
- Computational biology
- Natural language processing
- etc.

# A GUIDE TO THIS DISSERTATION



# WHAT ARE PROBABILISTIC GRAPHICAL MODELS

Informally...

- attributes of our interests in a system  $\rightarrow$  variable nodes
- relationship of these factors  $\rightarrow$  structures of a graph

Intrinsic property: **reasoning with uncertainty**

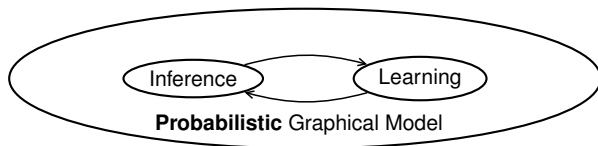
A directed/undirected graph encoding dependencies/independencies of distribution  $p(\mathbf{x}; \theta)$ :

- A BN/Generative model is a directed graph
  - $p(\mathbf{x}; \theta) = \prod_{n=1}^N p(x_n | \mathcal{P}(x_n))$
  - $\mathcal{P}(\cdot)$  are parent nodes
  - the local functions are proper distributions
- An MRF denoted by an undirected graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ 
  - The probability distribution (Gibbs distribution) is  $p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \prod_{a \in \mathcal{I}} \psi_a(\mathbf{x}_a; \theta_a)$
  - $a$  indexes potential functions  $\mathcal{I} = \{\psi_A, \psi_B, \dots, \psi_M\}$
  - $Z(\theta) = \sum_{\mathbf{x}} \prod_a \psi_a(\mathbf{x}_a; \theta_a)$ .

# WHAT TO DO WITH GRAPHICAL MODELS

- The common inference problems:
  - Computing the likelihood of observed data.
  - Computing the marginals distribution  $p(\mathbf{x}_A)$  over particular subset  $A \subset \mathcal{V}$  of nodes
  - Computing the conditional distribution  $p(\mathbf{x}_A|\mathbf{x}_B)$ ,
  - Computing the partition function or the Helmholtz free energy (for MRFs)
- Learning:
  - To model or determine  $p(\mathbf{x}; \theta)$ .

Two key components interacting with each other:

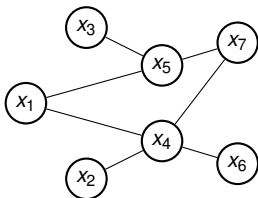


# WHAT IS THE STATE OF $x$ ?

## A TOY EXAMPLE

Assume that we are interested into the state of node  $i$  in an MRF, it can be answered by

- the probability  $p(x_i)$ , or
- an empirical version, a collection of samples  $\{x_i^n\}_{n=1}^N$



What is the state of  $x_4$

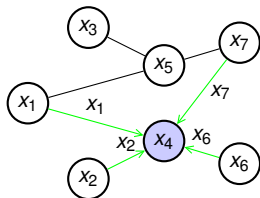


# WHAT IS THE STATE OF $x$ ?

## A TOY EXAMPLE

Assume that we are interested into the state of node  $i$  in an MRF, it can be answered by

- the probability  $p(x_i)$ , or
- an empirical version, a collection of samples  $\{x_i^n\}_{n=1}^N$



Gibbs sampling: let us guess by sampling

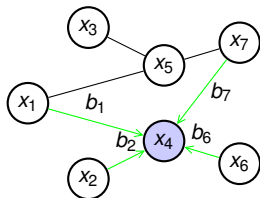
- Sample iteratively:  
 $x_i \sim p(x_i | \mathbf{x}_{-i}) \sim p(x_i, \mathbf{x}_{-i})$
- Queries answered by collecting samples  $\{\mathbf{x}^n\}_1^N$ .

# WHAT IS THE STATE OF $x$ ?

## A TOY EXAMPLE

Assume that we are interested into the state of node  $i$  in an MRF, it can be answered by

- the probability  $p(x_i)$ , or
- an empirical version, a collection of samples  $\{x_i^n\}_{n=1}^N$



Mean Field and BP: *message in form of sample values*  $\rightarrow$  *message in form of belief*

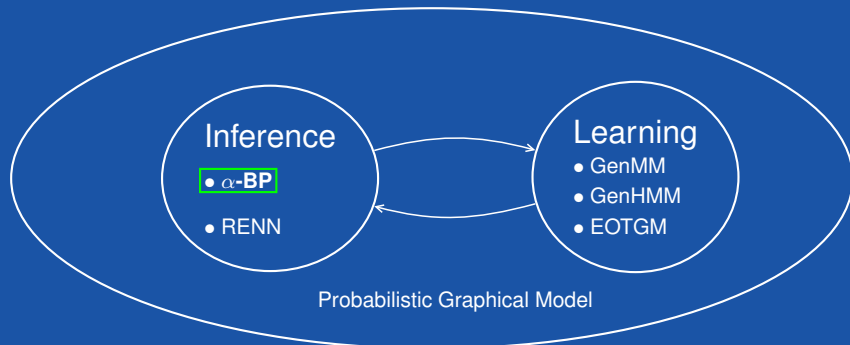
- Propagating beliefs iteratively
- Queries by collected beliefs  $\{b_i\}$ .

---

Intuition from *Gibbs (variational) free energy*

$$F_V(b) = \text{KL}(b(\mathbf{x}) || p(\mathbf{x}; \theta)) - \log Z(\theta)$$

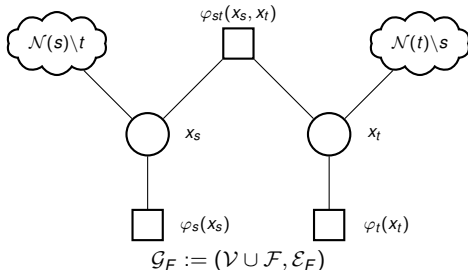
with trial  $b(\mathbf{x})$ . Instance: Bethe free energy.



# ALTERNATIVE VIEW OF BP: $\alpha$ -BP

Input:

- A pairwise Markov random field:  
 $p(\mathbf{x}) \propto \prod_{s \in \mathcal{V}} \varphi_s(x_s) \prod_{(s,t) \in \mathcal{E}} \varphi_{st}(x_s, x_t)$
- A trial distribution:  $q(\mathbf{x}) \propto \prod_{s \in \mathcal{V}} \tilde{\varphi}_s(x_s) \prod_{(s,t) \in \mathcal{E}} \tilde{\varphi}_{st}(x_s, x_t)$   
 with factorization  
 $\tilde{\varphi}_{s,t}(x_s, x_t) := m_{st}(x_t) m_{ts}(x_s)$
- A metric:  $\alpha$ -Divergence



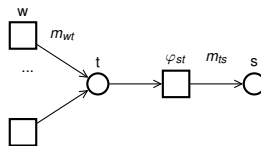
---

Definition of  $\alpha$ -divergence  $\mathcal{D}_\alpha(p \| q) = \frac{\sum_{\mathbf{x}} \alpha p(\mathbf{x}) + (1-\alpha) q(\mathbf{x}) - p(\mathbf{x})^\alpha q(\mathbf{x})^{1-\alpha}}{\alpha(1-\alpha)}$

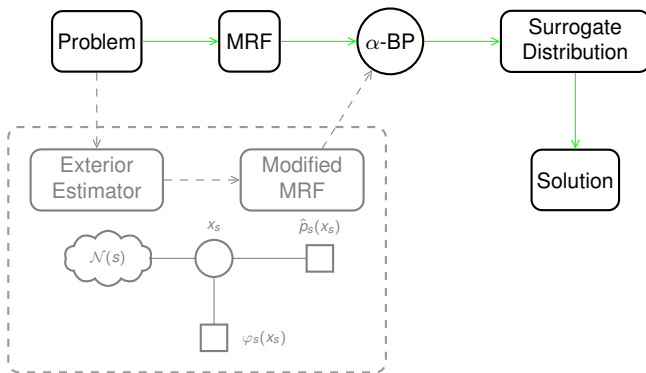
Approximate local minimization:

- Direct minimization:  $\operatorname{argmin}_{\tilde{\varphi}_{ts}^{\text{new}}(x_t, x_s)} \mathcal{D}_{\alpha_{ts}}(q^{\setminus(t,s)}(\mathbf{x}) \varphi_{ts}(x_t, x_s) \| q^{\setminus(t,s)}(\mathbf{x}) \tilde{\varphi}_{ts}^{\text{new}}(x_t, x_s))$
- Local minimization:  $\operatorname{argmin}_{\tilde{\varphi}_{ts}^{\text{new}}(x_t, x_s)} \mathcal{D}_{\alpha_{ts}}(q^{\setminus(t,s)}(\mathbf{x}) \varphi_{ts}(x_t, x_s) \| q^{\setminus(t,s)}(\mathbf{x}) \tilde{\varphi}_{ts}^{\text{new}}(x_t, x_s))$ , say you are updating  
 $\tilde{\varphi}_{ts}^{\text{new}}(x_t, x_s) = m_{ts}^{\text{new}}(x_s) m_{st}(x_t)$

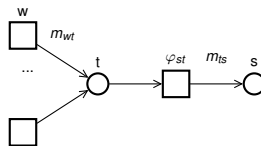
Updating message via  $\alpha$ -BP:



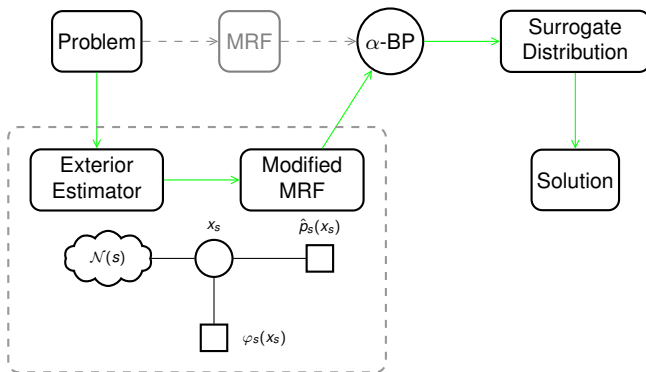
$$\underbrace{m_{ts}^{\text{new}}(x_s)}_{\text{new msg. to s}} \propto \underbrace{m_{ts}(x_s)^{1-\alpha_{ts}}}_{\text{old msg. to s}} \left[ \sum_{x_t} \varphi_{ts}(x_t, x_s)^{\alpha_{ts}} \underbrace{m_{st}(x_t)^{1-\alpha_{ts}}}_{\text{old msg. to t}} \underbrace{\varphi_t(x_t) \prod_{w \in \mathcal{N}(t) \setminus s} m_{wt}(x_t)}_{\text{msg. from variable node t to factor } \varphi_{st}} \right].$$



Updating message via  $\alpha$ -BP:



$$\underbrace{m_{ts}^{\text{new}}(x_s)}_{\text{new msg. to s}} \propto \underbrace{m_{ts}(x_s)^{1-\alpha_{ts}}}_{\text{old msg. to s}} \left[ \sum_{x_t} \varphi_{ts}(x_t, x_s)^{\alpha_{ts}} \underbrace{m_{st}(x_t)^{1-\alpha_{ts}}}_{\text{old msg. to t}} \underbrace{\varphi_t(x_t) \prod_{w \in \mathcal{N}(t) \setminus s} m_{wt}(x_t)}_{\text{msg. from variable node t to factor } \varphi_{st}} \right].$$



# INSIGHTS OF $\alpha$ -BP

## Connection to standard BP

- $\alpha \rightarrow 1$
- $\alpha$ -divergence reduces to KL-divergence
- Update rule of  $\alpha$ -BP reduces to  $m_{ts}^{\text{new}}(x_s) \propto \sum_{x_t} \varphi_{st}(x_s, x_t) \varphi_t(x_t) \prod_{w \in \mathcal{N}(t) \setminus s} m_{wt}(x_t)$ , which is standard BP update rule

## Convergence

For an arbitrary pairwise Markov random field over binary variables, if the largest singular value of matrix  $\mathbf{M}(\alpha, \theta)$  is less than one,  $\alpha$ -BP converges to a fixed point. The associated fixed point is unique.

See Corollary 3.1 for relaxed condition where singular value computation is avoided.

## What does that mean

- You can safely use  $\alpha$ -BP as an alternative to (loopy) BP
- You can use matrix  $\mathbf{M}$  to check if you are guaranteed to get stable solution from  $\alpha$ -BP

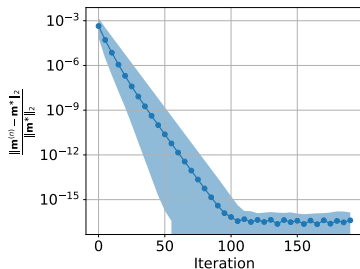
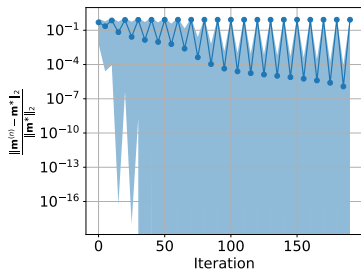
Matrix  $\mathbf{M}(\alpha, \theta)$ , size  $|\vec{\mathcal{E}}| \times |\vec{\mathcal{E}}|$ , indexed by directed edges ( $t \rightarrow s$ ), as

For binary, symmetric log-potentials

$$\begin{aligned} \varphi_{st}(x_s, x_t) &= \exp \{ \theta_{st}(x_s, x_t) \}, \\ \varphi_s(x_s) &= \exp \{ \theta_s(x_s) \}. \end{aligned}$$

$$M_{(t \rightarrow s), (u \rightarrow v)} = \begin{cases} |1 - \alpha_{ts}|, & u = t, v = s, \\ |1 - \alpha_{ts}| \tanh |\alpha_{ts} \theta_{ts}|, & u = s, v = t, \\ \tanh |\alpha_{ts} \theta_{ts}|, & u \in \mathcal{N}(t) \setminus s, v = t, \\ 0, & \text{otherwise.} \end{cases}$$

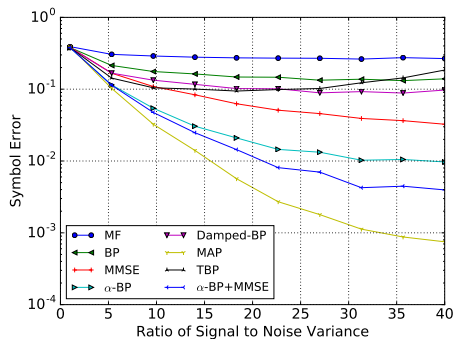
# SOME NUMERICAL RESULTS: CONVERGENCE



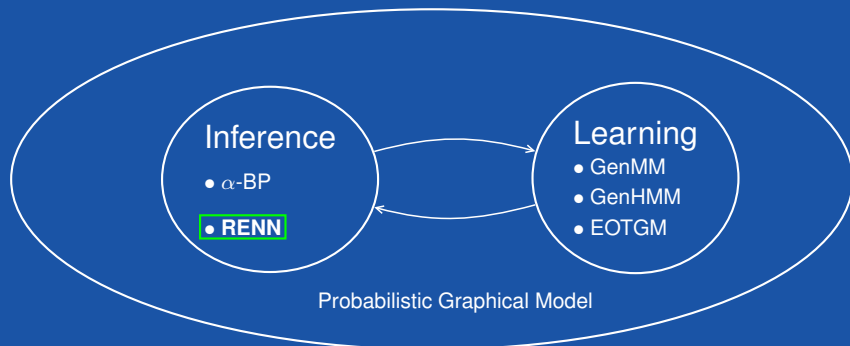
Numerical illustration of convergence, with normalized error  $\frac{\|\mathbf{m}^{(n)} - \mathbf{m}^*\|_2}{\|\mathbf{m}^*\|_2}$  versus the number of iterations. Blue region: range of the normalized error of 100 graph realizations. Curve: mean error of the 100 realized graphs.



# SOME NUMERICAL RESULTS: APPLICATION CASE



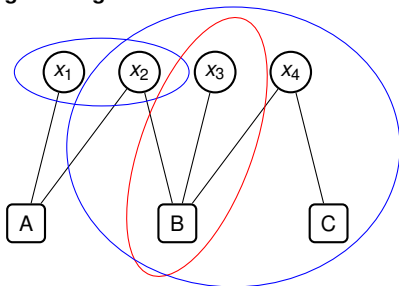
Numerical results of  $\alpha$ -BP: symbol error of MIMO detection.



# CONTINUING: WHAT IS THE STATE OF $x$ ?

YEDIDIA, FREEMAN, WEISS: A STEP TO GENERALIZATION

**Message among variables & factors** → **message among regions**



- Blue region: valid region
- Red region: invalid region

Generalized belief propagation (GBP) generalizes loopy BP

- usual better approximation than LBP
- higher complexity
- sensitive to scheduling of region messages

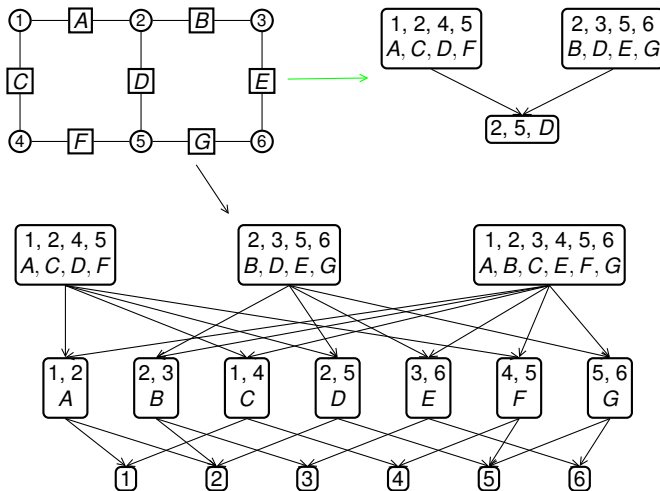
Approximating **variational free energy**  $F_v(b)$  with trial  $b$  including  $\{b_R\}$ .

A *region*  $R$  is a set  $V_R$  of variables nodes and a set  $A_R$  of factor nodes, such that if a factor node ' $a$ ' belongs to  $A_R$ , all the variables nodes neighboring  $a$  are in  $V_R$ .

# A TOY EXAMPLE OF REGION GRAPHS

Factor graph representation of MRF (2-by-3 grid) with factor nodes.

MRF  $\rightarrow$  region graph:



- Clustering nodes
- level/layer-wise
- Hierarchical
- Msg. Scheduling
- ...
- See Section 4.1

# RENN: REGION-BASED ENERGY NEURAL NETWORK

The **region-based free energy** of a region graph is

$$F_R(\mathcal{B}; \theta) = \sum_{R \in \mathcal{R}} \underbrace{c_R}_{\text{Counting number}} \sum_{\mathbf{x}_R} \underbrace{b_R(\mathbf{x}_R)}_{\text{Belief on region R}} \underbrace{(E_R(\mathbf{x}_R; \theta_R) + \ln b_R(\mathbf{x}_R))}_{\text{Region energy}},$$

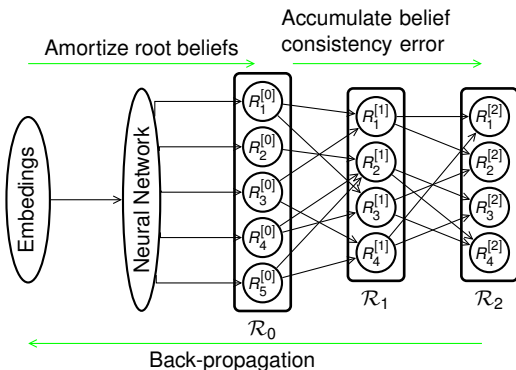
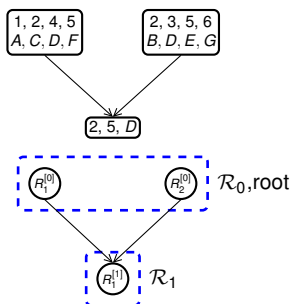
- counting number: balance the contribution of each region
- region energy:  $-\sum_{a \in A_R} \ln \varphi_a(\mathbf{x}_a; \theta_a)$

# RENN: REGION-BASED ENERGY NEURAL NETWORK

The **region-based free energy** of a region graph is

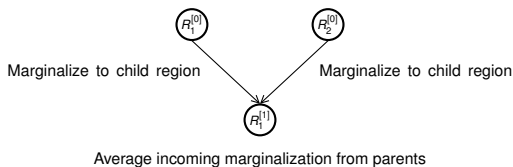
$$F_R(\mathcal{B}; \theta) = \sum_{R \in \mathcal{R}} \underbrace{c_R}_{\text{Counting number}} \sum_{\mathbf{x}_R} \underbrace{b_R(\mathbf{x}_R)}_{\text{Belief on region R}} \underbrace{(E_R(\mathbf{x}_R; \theta_R) + \ln b_R(\mathbf{x}_R))}_{\text{Region energy}},$$

- counting number: balance the contribution of each region
- region energy:  $-\sum_{a \in A_R} \ln \varphi_a(\mathbf{x}_a; \theta_a)$



# RENN

Non-root belief:

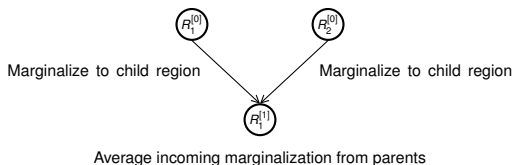


Objective of RENN:

$$\min_{\text{parameter of NN}} \underbrace{\text{region-based free energy}(F_R)}_{\text{Assumulated average of all regions}} + \underbrace{\text{panelty on belief consistency}}_{\text{Recursively computed via levels of region graph}}$$

# RENN

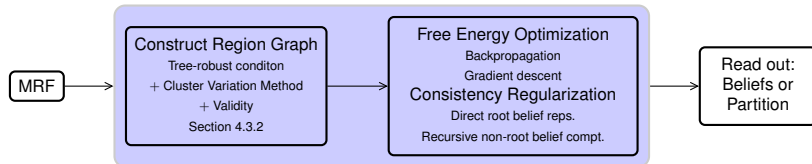
Non-root belief:



Objective of RENN:

$$\min_{\text{parameter of NN}} \underbrace{\text{region-based free energy}(F_R)}_{\text{Assumulated average of all regions}} + \underbrace{\text{panelty on belief consistency}}_{\text{Recursively computed via levels of region graph}}$$

RENN Inference:



RENN



# INSIGHTS OF RENN

## Generalization

Bethe free energy can be recovered from region-based free energy:

- two-level region graph representation
- constraint that each region can contain at most one factor node

Section 4.2.1

## Attributes of RENN

- RENN requires neither sampling technique nor training data (ground-truth marginal probabilities) in performing inference tasks;
- RENN does gradient descent w.r.t. its neural network parameter instead of iterative message-passing, and returns approximation of marginal probabilities and partition estimation in one-shot
- No message propagation, thus no need of message scheduling
- Competitive performance and efficiency

## SOME NUMERICAL COMPARISONS

Ising model:  $p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \exp(\sum_{(i,j) \in \mathcal{E}_F} J_{ij} x_i x_j + \sum_{i \in \mathcal{V}} h_i x_i)$ ,  $\mathbf{x} \in \{-1, 1\}^N$ ,

- $J_{ij}$  is the pairwise log-potential between node  $i$  and  $j$ ,  $J_{ij} \sim \mathcal{N}(0, 1)$
- $h_i$  is the node log-potential for node  $i$ ,  $h_i \sim \mathcal{N}(0, \gamma^2)$

Inference on grid graph ( $\gamma = 0.1$ ).

Metric	$n$	Mean Field	Loopy BP	Damped BP	GBP	Inference Net	RENN
$\ell_1$ error	25	$0.271 \pm 0.051$	$0.086 \pm 0.078$	$0.084 \pm 0.076$	$0.057 \pm 0.024$	$0.111 \pm 0.072$	<b><math>0.049 \pm 0.078</math></b>
	100	$0.283 \pm 0.024$	$0.085 \pm 0.041$	$0.062 \pm 0.024$	$0.064 \pm 0.019$	$0.074 \pm 0.034$	<b><math>0.025 \pm 0.011</math></b>
	225	$0.284 \pm 0.019$	$0.100 \pm 0.025$	$0.076 \pm 0.025$	$0.073 \pm 0.013$	$0.073 \pm 0.012$	<b><math>0.046 \pm 0.011</math></b>
	400	$0.279 \pm 0.014$	$0.110 \pm 0.016$	$0.090 \pm 0.016$	$0.079 \pm 0.009$	$0.083 \pm 0.009$	<b><math>0.061 \pm 0.009</math></b>
Corre- lation $\rho$	25	$0.633 \pm 0.197$	$0.903 \pm 0.114$	$0.905 \pm 0.113$	$0.923 \pm 0.045$	$0.866 \pm 0.117$	<b><math>0.951 \pm 0.112</math></b>
	100	$0.582 \pm 0.112$	$0.827 \pm 0.134$	$0.902 \pm 0.059$	$0.899 \pm 0.043$	$0.903 \pm 0.049$	<b><math>0.983 \pm 0.012</math></b>
	225	$0.580 \pm 0.080$	$0.801 \pm 0.078$	$0.863 \pm 0.088$	$0.869 \pm 0.037$	$0.873 \pm 0.037$	<b><math>0.949 \pm 0.022</math></b>
	400	$0.596 \pm 0.054$	$0.779 \pm 0.059$	$0.822 \pm 0.047$	$0.852 \pm 0.024$	$0.841 \pm 0.028$	<b><math>0.912 \pm 0.025</math></b>
log Z error	25	$2.512 \pm 1.060$	$0.549 \pm 0.373$	$0.557 \pm 0.369$	<b><math>0.169 \pm 0.142</math></b>	$0.762 \pm 0.439$	$0.240 \pm 0.140$
	100	$13.09 \pm 2.156$	$1.650 \pm 1.414$	$1.457 \pm 1.365$	<b><math>0.524 \pm 0.313</math></b>	$2.836 \pm 2.158$	$1.899 \pm 0.495$
	225	$29.93 \pm 4.679$	$3.348 \pm 1.954$	$3.423 \pm 2.157$	<b><math>1.008 \pm 0.653</math></b>	$3.249 \pm 2.058$	$4.344 \pm 0.813$
	400	$51.81 \pm 4.706$	$5.738 \pm 2.107$	$5.873 \pm 2.211$	<b><math>1.750 \pm 0.869</math></b>	$3.953 \pm 2.558$	$7.598 \pm 1.146$

- $\ell_1$  error of beliefs v.s. true
- correlation  $\rho$  between true and approximate marginals,
- log Z error, true v.s. free energy approximation.

# SOME NUMERICAL COMPARISONS

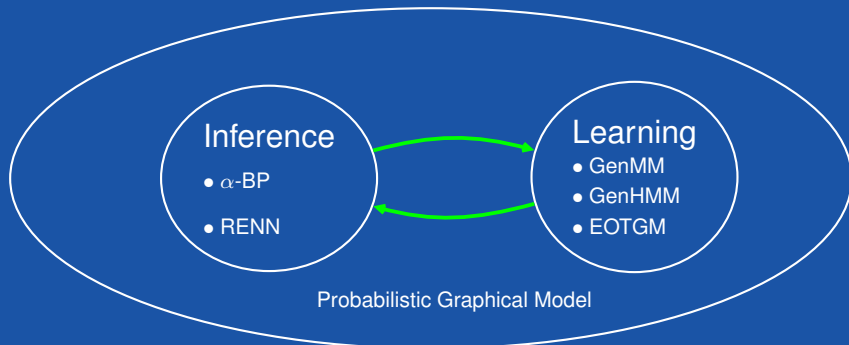
## RICHER COMPARISONS

Inference on grid and complete graphs.

		Metric		Mean Field	Loopy BP	Damped BP	GBP	Inference Net	RENN
High Temp -erature	Complete graph N=16	$\ell_1$ -	$\gamma = 1$	$0.273 \pm 0.086$	$0.239 \pm 0.059$	$0.239 \pm 0.059$	$0.260 \pm 0.086$	$0.249 \pm 0.067$	<b>0.181</b> $\pm 0.092$
		error	$\gamma = 4$	$0.197 \pm 0.049$	$0.181 \pm 0.035$	$0.180 \pm 0.034$	$0.210 \pm 0.070$	$0.174 \pm 0.030$	<b>0.125</b> $\pm 0.050$
	$J_{ij} \sim \mathcal{N}(0, 1)$ $h_i \sim \mathcal{N}(0, \gamma^2)$	log Z	$\gamma = 1$	$20.66 \pm 5.451$	$178.7 \pm 22.18$	$178.9 \pm 21.88$	$153.3 \pm 25.29$	$213.6 \pm 12.75$	<b>14.41</b> $\pm 4.135$
		error	$\gamma = 4$	<b>10.74</b> $\pm 7.385$	$565.7 \pm 73.33$	$566.1 \pm 73.13$	$106.0 \pm 54.43$	$588.3 \pm 62.58$	$14.72 \pm 4.155$
Low Temp -erature	Grid graph N=100	$\ell_1$	5	$0.257 \pm 0.065$	$0.115 \pm 0.071$	$0.120 \pm 0.073$	$0.250 \pm 0.024$	$0.164 \pm 0.036$	<b>0.100</b> $\pm 0.046$
		error	15	$0.328 \pm 0.068$	$0.228 \pm 0.088$	$0.267 \pm 0.147$	$0.303 \pm 0.026$	$0.279 \pm 0.024$	<b>0.207</b> $\pm 0.054$
	$J_{ij} \sim \mathcal{U}(-u, u)$ $h_i \sim \mathcal{U}(-1, 1)$	log Z	5	$42.65 \pm 17.86$	$7.346 \pm 7.744$	<b>5.444</b> $\pm 4.811$	$8.369 \pm 7.401$	$65.60 \pm 8.786$	$11.34 \pm 4.724$
		error	15	$164.9 \pm 56.07$	$58.40 \pm 41.36$	$101.9 \pm 54.31$	<b>23.10</b> $\pm 15.06$	$224.3 \pm 25.52$	$78.85 \pm 15.08$

Average consumed time per inference instance (unit: second)

	Mean Field	Loopy BP	Damped BP	GBP	Inference Net	RENN
Grid $\mathcal{G}$ , $N = 400$ , $J_{ij} \sim \mathcal{N}(0, 1)$ , $h_i \sim \mathcal{N}(0, 0.1^2)$	9.897	425.0	328.3	286.3	74.41	101.0
Complete $\mathcal{G}$ , $N = 16$ , $J_{ij} \sim \mathcal{N}(0, 1)$ , $h_i \sim \mathcal{N}(0, 1)$	0.457	0.777	1.285	14.29	12.45	16.16
Grid $\mathcal{G}$ , $N = 100$ , $J_{ij} \sim \mathcal{U}(-15, 15)$ , $h_i \sim \mathcal{U}(-1, 1)$	2.314	253.3	229.3	53.72	103.4	79.38
Complete $\mathcal{G}$ , $N = 9$ , $J_{ij} \sim \mathcal{U}(-15, 15)$ , $h_i \sim \mathcal{U}(-1, 1)$	0.502	15.86	18.23	3.213	17.21	7.857



# INFERENCE ROUTINE IN LEARNING

What is  $\theta$  in  $p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \prod_a \psi_a(\mathbf{x}_a; \theta_a)$ ?

A direct view:

$$\max_{\theta} \log p(\mathbf{x}; \theta) = \max_{\theta} \sum_a \log \psi_a(\mathbf{x}_a; \theta_a) \quad \underbrace{-\log Z(\theta)}_{\text{can be est. by min } F_V},$$

An alternative view:

$$\frac{\partial \log p(\mathbf{x}; \theta)}{\partial \theta_a} = \frac{\partial \log \varphi_a(\mathbf{x}_a; \theta_a)}{\partial \theta_a} - \underbrace{\mathbb{E}_{p(\mathbf{x}_a; \theta)} \left[ \frac{\partial \log \varphi_a(\mathbf{x}_a; \theta_a)}{\partial \theta_a} \right]}_{\text{can be est. by beliefs}}.$$

Remark:

- This essentially requires estimation of partition function or marginal probabilities.
- Stationary points translate into moment matching.

# INFERENCE ROUTINE IN LEARNING

What is  $\theta$  in  $p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \prod_a \psi_a(\mathbf{x}_a; \theta_a)$ ?

A direct view:

$$\max_{\theta} \log p(\mathbf{x}; \theta) = \max_{\theta} \sum_a \log \psi_a(\mathbf{x}_a; \theta_a) \quad \underbrace{-\log Z(\theta)}_{\text{can be est. by min } F_V},$$

An alternative view:

$$\frac{\partial \log p(\mathbf{x}; \theta)}{\partial \theta_a} = \frac{\partial \log \varphi_a(\mathbf{x}_a; \theta_a)}{\partial \theta_a} - \underbrace{\mathbb{E}_{p(\mathbf{x}_a; \theta)} \left[ \frac{\partial \log \varphi_a(\mathbf{x}_a; \theta_a)}{\partial \theta_a} \right]}_{\text{can be est. by beliefs}}.$$

Remark:

- This essen
- Stationary



- Two modules are not necessarily coupled
- Each module may be replaced by another algorithm while the other one remains.

bilities.

# LEARNING MRFs

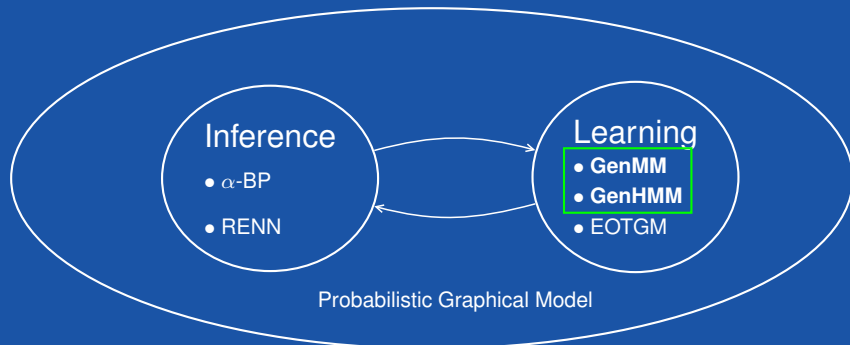
WHAT IS  $\theta$  IN  $p(\mathbf{x}; \theta)$ ?

Table of negative log-likelihood of learned MRFs

N	True	Exact	Mean Field	Loopy BP	Damped BP	GBP	Inference Net	RENN
Grid Graph								
25	9.000	9.004	9.811	9.139	9.196	10.56	9.252	<b>9.048</b>
100	19.34	19.38	23.48	19.92	20.02	28.61	20. 29	<b>19.76</b>
225	63.90	63.97	69.01	66.44	66.25	92.62	68.15	<b>64.79</b>
Complete Graph								
9	3.276	3.286	9.558	5.201	5.880	10.06	5.262	<b>3.414</b>
16	4.883	4.934	28.74	13.64	18.95	24.45	13.77	<b>5.178</b>

Average consumed time per epoch (unit: second) for two MRF learning cases.

	Mean Field	Loopy BP	Damped BP	GBP	Inference Net	RENN
Grid $\mathcal{G}$ , $N=225$	40.09	335.1	525.1	12.37	19.49	16.03
Complete $\mathcal{G}$ , $N=16$	2.499	12.40	5.431	1.387	0.882	2.262





# INCOMPLETE OBSERVATION

Partial observation:  $\mathbf{x} = [ \underbrace{\mathbf{x}_U}_{\text{Unobserved}}, \underbrace{\mathbf{x}_O}_{\text{Observed}} ]$

$$l(\mathbf{x}_O; \theta) = \log \sum_{\mathbf{x}_U} p(\mathbf{x}_U, \mathbf{x}_O; \theta) = \log \underbrace{Z(\mathbf{x}_O; \theta)}_{\sum_{\mathbf{x}_U} \tilde{p}(\mathbf{x}; \theta)} - \log Z(\theta),$$

Connect Free Energy to Evidence Lower Bounder:

$$l(\mathbf{x}_O; \theta) \geq - \underbrace{F_V(q(\mathbf{x}_U|\mathbf{x}_O))}_{\text{Variational Free Energy}} - \log Z(\theta)$$

$$\begin{aligned} &= \mathbb{E}_{q(\mathbf{x}_U|\mathbf{x}_O)} \left[ \log \frac{p(\mathbf{x}_U, \mathbf{x}_O; \theta)}{q(\mathbf{x}_U|\mathbf{x}_O)} \right] \\ &= \underbrace{\mathbb{E}_{q(\mathbf{x}_U|\mathbf{x}_O)} [\log p(\mathbf{x}_U, \mathbf{x}_O; \theta)] + H(q(\mathbf{x}_U|\mathbf{x}_O))}_{\text{Evidence Lower Bound } F(q, \theta)} \end{aligned}$$

Intuition of maximizing  $F(q, \theta)$

- Maximizing (incomplete) likelihood
- Minimizing free energy

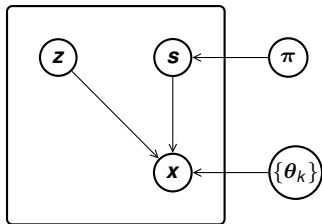
This gives the EM as a coordinate ascent method:

$$\text{E step : } q^{(t+1)} = \underset{q}{\operatorname{argmax}} F(q, \theta^{(t)}),$$

$$\text{M step : } \theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} F(q^{(t+1)}, \theta).$$

# GENERATOR MIXED MODEL

## EQUIPPING EM WITH NORMALIZING FLOWS



- Ideal case: The underline true  $p^*(\mathbf{x})$  is in hypothesis space  $\mathcal{H}$ , i.e.  $p^*(\mathbf{x}) \in \mathcal{H}$ .
- Out of reach: Test  $p^*(\mathbf{x}) \stackrel{?}{\in} \mathcal{H}$
- Luckily, what is at our hands is:

$\mathcal{H}$  is large  $\rightarrow$  candidate  $p(\mathbf{x}; \theta)$  is flexible

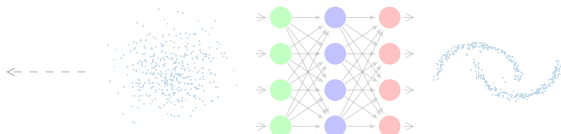
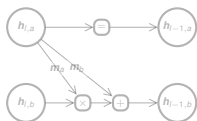
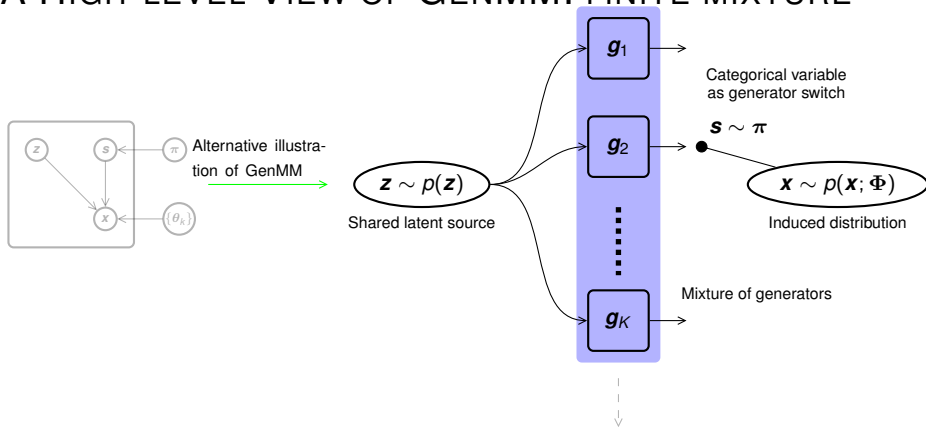
This brings up the finite **mixture** models.

$$p(\mathbf{x}; \Theta) = \sum_{k=1}^K \pi_k p_k(\mathbf{x}) = \sum_{k=1}^K \pi_k p(\underbrace{\mathbf{g}(\mathbf{z}; \theta_k)}_{\text{Variable change via generator } \mathbf{g}})$$

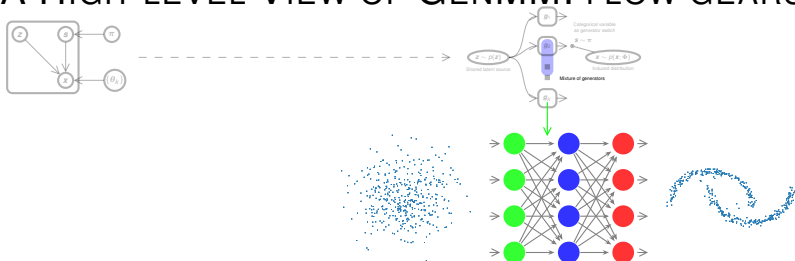
What to expect from GenMM:

- Flexible and expressive model, enlarging hyperspace  $\mathcal{H}$
- Tractable likelihood
- Compatible with typical statistical models
- Compatible with NN tools/frameworks
- Scale to high-dimensional structured data
- Efficient in sampling (data generation)
- ...

# A HIGH-LEVEL VIEW OF GENMM: FINITE MIXTURE



# A HIGH-LEVEL VIEW OF GENMM: FLOW GEARS



When the  $k$ -th generator is selected, i.e.,  $s_k = 1$  and  $s_{k'} = 0$  for  $k' \neq k$ , say  $\tilde{\mathbf{x}} = \mathbf{x}|_{s_k=1}$ . By following the [change of variable rule](#)

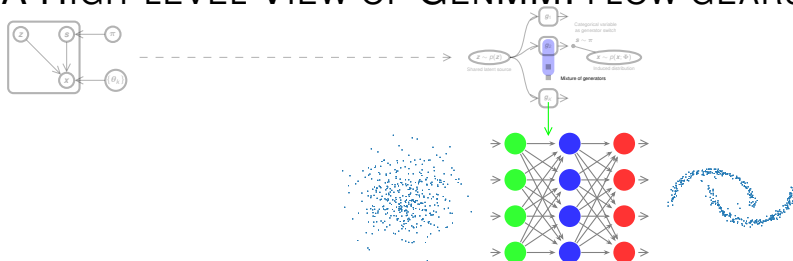


$$\underbrace{p(\tilde{\mathbf{x}})|_{\tilde{\mathbf{x}}=\tilde{\mathbf{g}}(\mathbf{z})}}_{\text{Induced distribution}} = \underbrace{p(\mathbf{z})}_{\substack{\text{Assumed known distribution} \\ \text{easy to sample}}} \cdot \underbrace{\left| \det \left( \frac{\partial \mathbf{z}}{\partial \tilde{\mathbf{x}}} \right) \right|}_{\substack{\text{Computational load} \\ \text{depends on the mapping}}}$$


A toy example:

$$\text{Gaussian linear transform: } Z \sim N(0, 1) \xrightarrow{X=\sigma \cdot Z+\mu} X \sim N(\mu, \sigma)$$

# A HIGH-LEVEL VIEW OF GENMM: FLOW GEARS



When the  $k$ -th generator is selected, i.e.,  $s_k = 1$  and  $s_{k'} = 0$  for  $k' \neq k$ , say  $\tilde{\mathbf{x}} = \mathbf{x}|_{s_k=1}$ . By following the [change of variable rule](#)

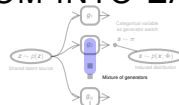
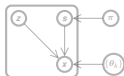


$$\underbrace{p(\tilde{\mathbf{x}})|_{\tilde{\mathbf{x}}=\tilde{\mathbf{g}}(z)}}_{\text{Induced distribution}} = \underbrace{p(z)}_{\substack{\text{Assumed known distribution} \\ \text{easy to sample}}} \cdot \underbrace{\left| \det \left( \frac{\partial \mathbf{z}}{\partial \tilde{\mathbf{x}}} \right) \right|}_{\substack{\text{Computational load} \\ \text{depends on the mapping}}}$$

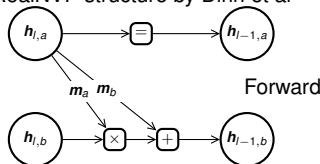
Powering it with a  $L$ -layer neural network implementation:

$$\mathbf{z} = \mathbf{h}_0 \begin{matrix} \xrightarrow{\tilde{\mathbf{g}}_1} \\ \xleftarrow{\tilde{\mathbf{f}}_1} \end{matrix} \mathbf{h}_1 \begin{matrix} \xrightarrow{\tilde{\mathbf{g}}_2} \\ \xleftarrow{\tilde{\mathbf{f}}_2} \end{matrix} \dots \begin{matrix} \xrightarrow{\tilde{\mathbf{g}}_L} \\ \xleftarrow{\tilde{\mathbf{f}}_L} \end{matrix} \mathbf{x} = \mathbf{h}_L$$

# A HIGH-LEVEL VIEW OF GENMM: ZOOM INTO LAYER

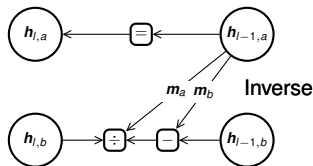


RealNVP structure by Dinh et al



Forward

$$h_{l-1} = \begin{bmatrix} h_{l-1,a} \\ h_{l-1,b} \end{bmatrix} = \begin{bmatrix} h_{l,a} \\ m_a(h_{l,a}) \odot h_{l,b} + m_b(h_{l,a}) \end{bmatrix}$$



Inverse

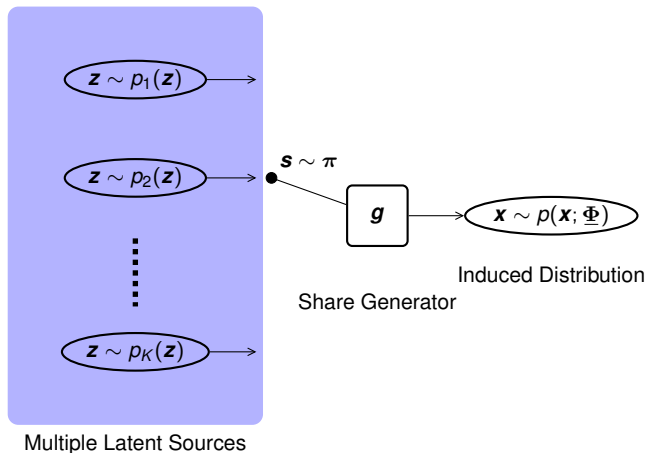
$$h_l = \begin{bmatrix} h_{l,a} \\ h_{l,b} \end{bmatrix} = \begin{bmatrix} h_{l-1,a} \\ (h_{l-1,b} - m_b(h_{l-1,a})) \oslash m_a(h_{l-1,a}) \end{bmatrix}$$

Pick up one layer to zoom in

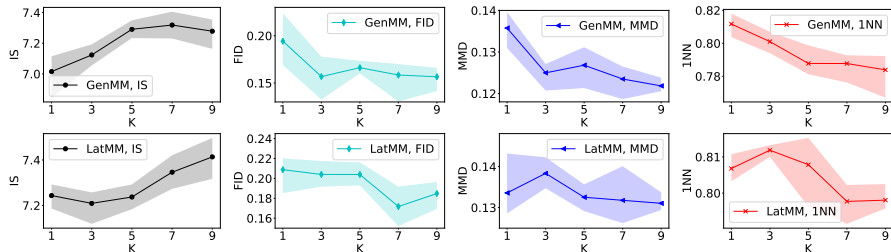
- $\odot$  denotes element-wise product,  $\oslash$  denotes element-wise division
- Mapping  $m_a$ ,  $m_b$  can be as complex as possible and not necessary invertible
- Same computation complexity of forward and inverse mapping
- Triangular matrix of Jacobian

Alternative arch. on market: Auto-regressive flow, Glow, ODE, etc.

# LATMM: ALTERNATIVE MIXTURE



# SEMANTIC SCORES AND EXAMPLES

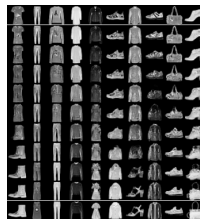
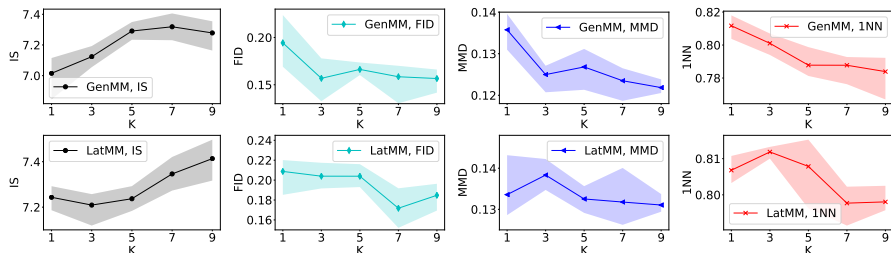


IS, FID, MMD and 1NN of GenMM and LatMM for MNIST dataset.

- IS: Inception Score (large is good)
- FID: Frechet Inception Distance (small is good)
- MMD: Maximum Mean Discrepancy (small is good)
- 1NN: 1-Nearest Neighbor (the closer to 0.5 the better)



# SEMANTIC SCORES AND EXAMPLES



Generated samples by GenMM and LatMM.

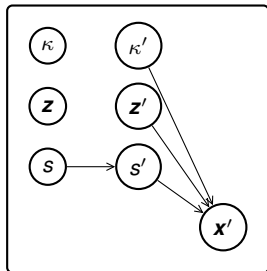
Interpolation in latent space

# APPLICATION TO CLASSIFICATION TASKS

Application to classification with maximum likelihood. Test Accuracy Table of GenMM for Classification Task

Dataset	K=1	K=2	K=3	K=4	K=10	K=20	State Of Art
Letter	0.9459	0.9513	0.9578	0.9581	0.9657	<b>0.9674</b>	0.9582
Satimage	0.8900	0.8975	0.9045	0.9085	0.9105	<b>0.9160</b>	0.9090
Norb	0.9184	0.9257	0.9406	0.9459	0.9538	<b>0.9542</b>	0.8920

# GENHMM: BRING THE CONCEPT INTO HMM

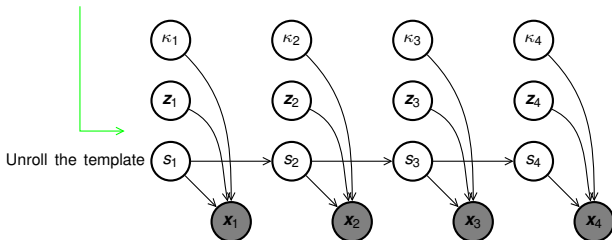


At time  $t$ , the probabilistic model of a state  $s \in \mathcal{S}$  is then given by

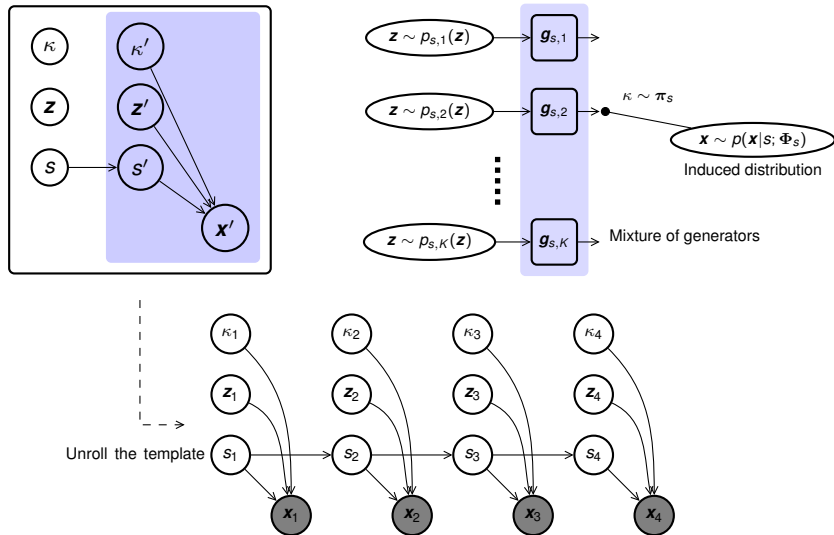
$$p(\mathbf{x}|s; \Phi_s) = \sum_{\kappa=1}^K \pi_{s,\kappa} p(\mathbf{x}|s, \kappa; \theta_{s,\kappa}),$$

where

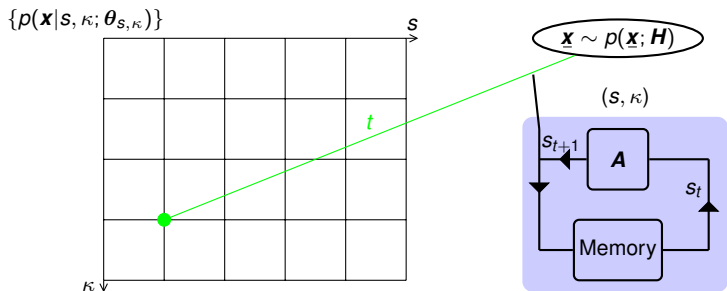
- $\pi_{s,\kappa} = p(\kappa|s; \mathbf{H})$ , naturally  $\sum_{\kappa=1}^K \pi_{s,\kappa} = 1$
- $p(\mathbf{x}|s, \kappa; \theta_{s,\kappa})$  is induced by the  $k$ th generator  $\mathbf{g}_k(\mathbf{z}) = \mathbf{g}(\mathbf{z}; \theta_k)$



# GENHMM: BRING THE CONCEPT INTO HMM



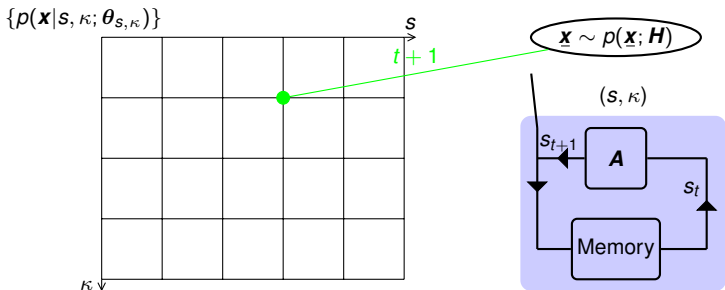
# ALTERNATIVE VIEW OF GENHMM



Hypothesis set of GenHMM as  $\mathcal{H} := \{\mathbf{H} | \mathbf{H} = \{\mathcal{S}, \mathbf{q}, \mathbf{A}, p(\mathbf{x}|\mathbf{s}; \Phi_{\mathbf{s}})\}\}$

- $\mathcal{S}$ : the set of hidden states of  $\mathbf{H}$ .
- $\mathbf{q} = [q_1, \dots, q_{|\mathcal{S}|}]^T$ : the initial state distributions of  $\mathbf{H}$ .  $q_i = p(s_1 = i; \mathbf{H})$ .
- $\mathbf{A}$ : the transition matrix of states in  $\mathbf{H}$ .
- $\Phi = \{\Phi_{\mathbf{s}} | \mathbf{s} \in \mathcal{S}\}$ .

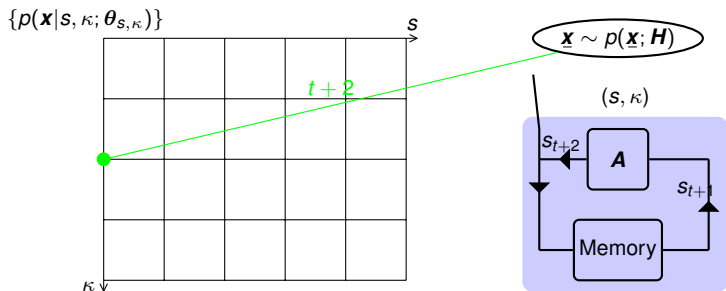
# ALTERNATIVE VIEW OF GENHMM



Hypothesis set of GenHMM as  $\mathcal{H} := \{\mathbf{H} | \mathbf{H} = \{\mathcal{S}, \mathbf{q}, \mathbf{A}, p(\mathbf{x}|\mathbf{s}; \boldsymbol{\Phi}_{\mathbf{s}})\}\}$

- $\mathcal{S}$ : the set of hidden states of  $\mathbf{H}$ .
- $\mathbf{q} = [q_1, \dots, q_{|\mathcal{S}|}]^T$ : the initial state distributions of  $\mathbf{H}$ .  $q_i = p(s_1 = i; \mathbf{H})$ .
- $\mathbf{A}$ : the transition matrix of states in  $\mathbf{H}$ .
- $\boldsymbol{\Phi} = \{\boldsymbol{\Phi}_{\mathbf{s}} | \mathbf{s} \in \mathcal{S}\}$ .

# ALTERNATIVE VIEW OF GENHMM



Hypothesis set of GenHMM as  $\mathcal{H} := \{H | H = \{S, \mathbf{q}, \mathbf{A}, p(\mathbf{x}|s; \Phi_s)\}\}$

- $S$ : the set of hidden states of  $H$ .
- $\mathbf{q} = [q_1, \dots, q_{|S|}]^T$ : the initial state distributions of  $H$ .  $q_i = p(s_1 = i; H)$ .
- $\mathbf{A}$ : the transition matrix of states in  $H$ .
- $\Phi = \{\Phi_s | s \in S\}$ .

# LEARNING INTUITION

With empirical distribution  $\hat{p}(\underline{\mathbf{x}}) = \frac{1}{N} \sum_n \delta_{\underline{\mathbf{x}}^{(n)}}(\underline{\mathbf{x}})$ , learning of GenHMM boils down to

$$\min_{\mathbf{H} \in \mathcal{H}} \text{KL}(\hat{p}(\underline{\mathbf{x}}) \| p(\underline{\mathbf{x}}; \mathbf{H}))$$

Then problem becomes maximum likelihood estimation

$$\hat{\mathbf{H}} = \arg \max_{\mathbf{H} \in \mathcal{H}} \log \prod_i p(\underline{\mathbf{x}}^i; \mathbf{H}),$$

E-step: the  $\mathcal{Q}$  function

$$\mathcal{Q}(\mathbf{H}; \mathbf{H}^{\text{old}}) = \underbrace{\mathbb{E}_{\hat{p}(\underline{\mathbf{x}}), p(\underline{\mathbf{s}}, \underline{\kappa} | \underline{\mathbf{x}}; \mathbf{H}^{\text{old}})}}_{\substack{\text{w.r.t. distribution } \hat{p}(\underline{\mathbf{x}}) \\ \text{and } p(\underline{\mathbf{s}}, \underline{\kappa} | \underline{\mathbf{x}}; \mathbf{H}^{\text{old}})}} \underbrace{[\log p(\underline{\mathbf{x}}, \underline{\mathbf{s}}, \underline{\kappa}; \mathbf{H})]}_{\substack{\text{complete lklh.} \\ \text{factorize over time}}},$$

M-step: the maximization step

$$\max_{\mathbf{H}} \mathcal{Q}(\mathbf{H}; \mathbf{H}^{\text{old}}).$$

The M-step can be reformulated as

$$\max_{\mathbf{H}} \mathcal{Q}(\mathbf{H}; \mathbf{H}^{\text{old}}) = \underbrace{\max_{\mathbf{q}} \mathcal{Q}(\mathbf{q}; \mathbf{H}^{\text{old}})}_{\text{Initial State}} + \underbrace{\max_{\mathbf{A}} \mathcal{Q}(\mathbf{A}; \mathbf{H}^{\text{old}})}_{\text{Transition}} + \underbrace{\max_{\Phi} \mathcal{Q}(\Phi; \mathbf{H}^{\text{old}})}_{\text{Generators}}, \quad (1)$$

Maximizing a lower bound of  $\log \prod_i p(\underline{\mathbf{x}}^{(i)}; \mathbf{H})$ .



# LEARNING INTUITION

With empirical distribution  $\hat{p}(\underline{\mathbf{x}}) = \frac{1}{N} \sum_n \delta_{\underline{\mathbf{x}}^{(n)}}(\underline{\mathbf{x}})$ , learning of GenHMM boils down to

$$\min_{\mathbf{H} \in \mathcal{H}} \text{KL}(\hat{p}(\underline{\mathbf{x}}) \| p(\underline{\mathbf{x}}; \mathbf{H}))$$

Then problem becomes maximum likelihood estimation

$$\hat{\mathbf{H}} = \arg \max_{\mathbf{H}} \log \prod_i p(\underline{\mathbf{x}}^i; \mathbf{H}),$$

E-step: the

- $F = \mathcal{Q} + \text{Entropy}$
- E-step require inference (message-passing)
- No optimality in M-step (NN generators, bash-size gradient descent).
- Still, guaranteed non-decreasing lklh. (c.f. Proposition 7.1)

M-step: the maximization step

$$\max_{\mathbf{H}} \mathcal{Q}(\mathbf{H}; \mathbf{H}^{\text{old}}).$$

The M-step can be reformulated as

$$\max_{\mathbf{H}} \mathcal{Q}(\mathbf{H}; \mathbf{H}^{\text{old}}) = \underbrace{\max_{\mathbf{q}} \mathcal{Q}(\mathbf{q}; \mathbf{H}^{\text{old}})}_{\text{Initial State}} + \underbrace{\max_{\mathbf{A}} \mathcal{Q}(\mathbf{A}; \mathbf{H}^{\text{old}})}_{\text{Transition}} + \underbrace{\max_{\Phi} \mathcal{Q}(\Phi; \mathbf{H}^{\text{old}})}_{\text{Generators}}, \quad (1)$$

Maximizing a lower bound of  $\log \prod_i p(\underline{\mathbf{x}}^{(i)}; \mathbf{H})$ .

# APPLICATION OF GENHMM

Speech Recognition:

Configuration GenHMM in experiments on TIMIT

Latent distribution $p_{s,\kappa}(\mathbf{z})$ $s \in \mathcal{S}, \kappa = 1, 2, \dots, K$	Standard Gaussian
Number of flow blocks	4
Non-linear mapping $\mathbf{m}_a, \mathbf{m}_b$	Multiple layer perception 3 layers and with hidden dimension 24

Phoneme classification / recognition

Model	Criterion	K=1	K=3	K=5
GMM-HMM linear variable change	Accuracy	62.3	68.0	68.7
	Precision	67.9	72.6	73.0
	F1	63.7	69.1	69.7
GenHMM non-linear variable change	Accuracy	76.7	<b>77.7</b>	77.7
	Precision	76.9	<b>78.1</b>	78.0
	F1	76.1	<b>77.1</b>	77.0

Robustness to perturbation of noise.

Model	Criterion	White Noise SNR			
		15dB	20dB	25dB	30dB
GMM-HMM	Accuracy	36.6	44.2	50.8	57.1
	Precision	59.2	64.2	68.4	70.6
	F1	39.9	47.7	53.9	59.9
GenHMM	Accuracy	52.4	62.0	69.7	<b>74.3</b>
	Precision	60.0	65.9	71.7	<b>74.8</b>
	F1	52.5	62.0	69.3	<b>73.5</b>

Application to sepsis detection for infants, c.f. see Section 7.5.

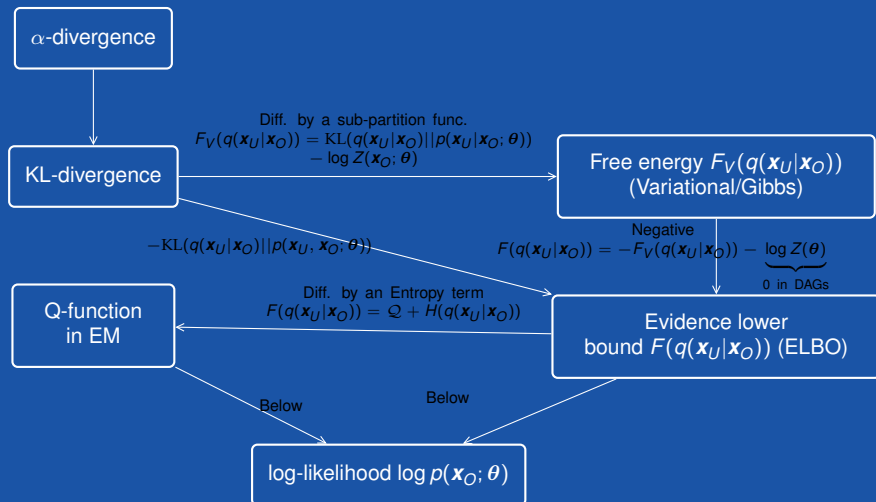
- generative training + discriminative training
- innovative feature inspired from acoustic signal feature

# REMARK ON GENMM/GENHMM

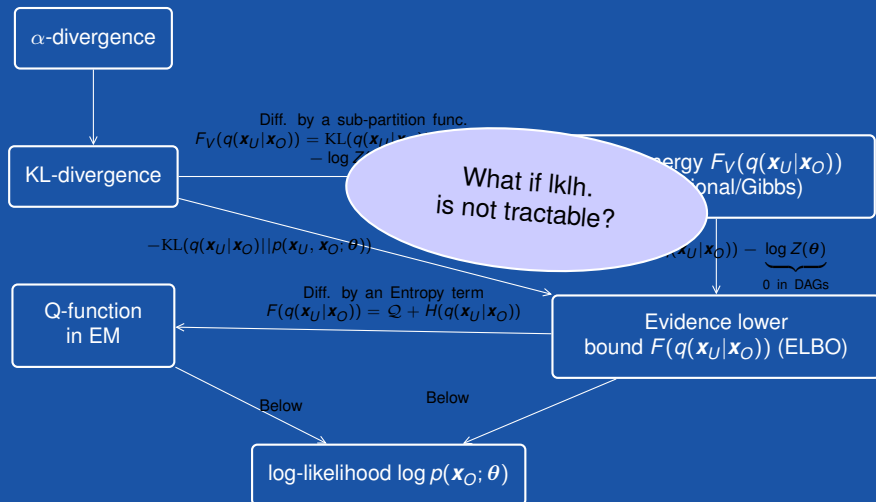
## Attributes

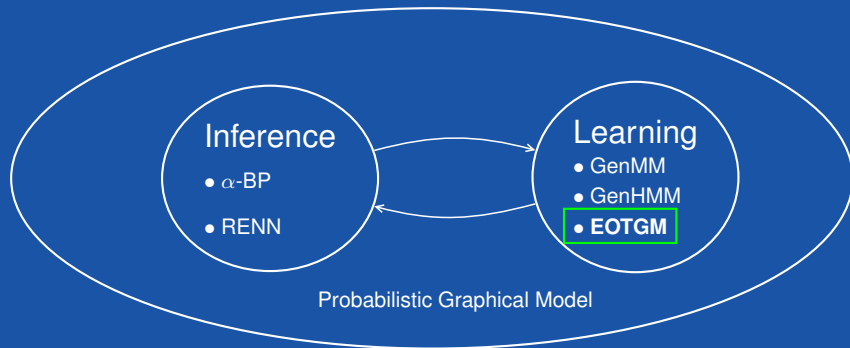
- Free dimension for flexibility: number of mixture + complexity of functional form of neural networks
- Compatible with classic statistic methods and neural network techniques (error back-propagation, optimizer)
- Embed batch-gradient descent into M-step
- Lack of closed-form update rule and generator changes at each gradient step. We tackle by maintaining old and new generators in EM steps

# WHAT HAVE WE BEEN TALKING ABOUT?

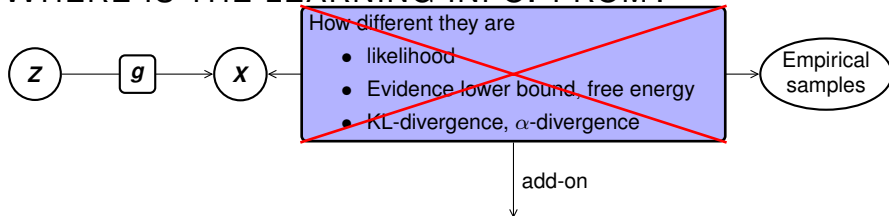


# WHAT HAVE WE BEEN TALKING ABOUT?





# WHERE IS THE LEARNING INFO. FROM?



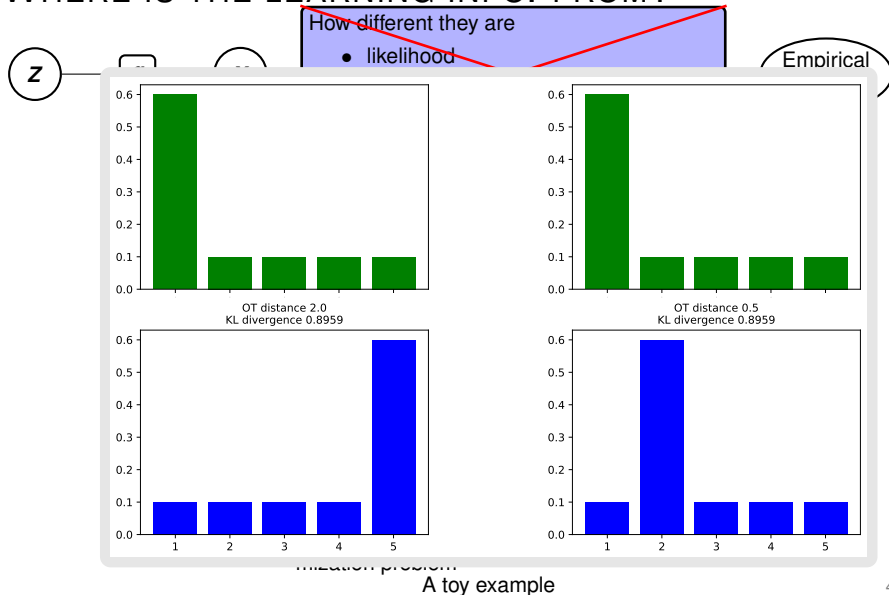
Optimal transport (OT): moving mass from a dist. to another

$$T(p^*, p) = \min_{\pi \in \Pi(p^*, p)} \left\langle \underbrace{\pi}_{\text{marginalize to } p^*, p}, \underbrace{M}_{\substack{\text{cost matrix} \\ \text{pair-wise sample difference}}} \right\rangle,$$

Key attributes:

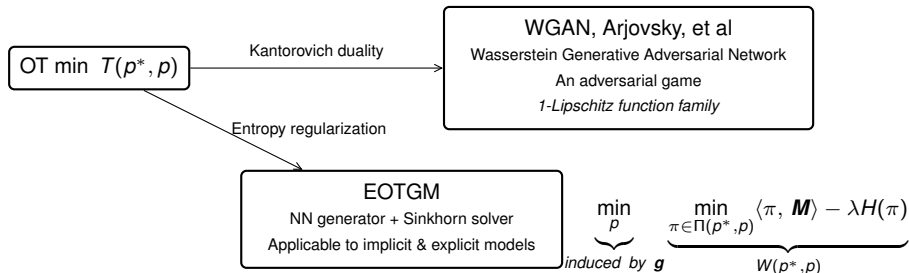
- Doesnot require tractible lklh.
- Learning gradient info. from sample comparison
- High complexity, each evaluation is sovling an optimization problem

# WHERE IS THE LEARNING INFO. FROM?





# EOTGM: EOT BASED GENERATIVE MODEL



Alternatively scale the rows & columns of matrix  $e^{-\mathbf{M}/\lambda}$  (Sinkhorn & Knopp) gives 'soft' solution

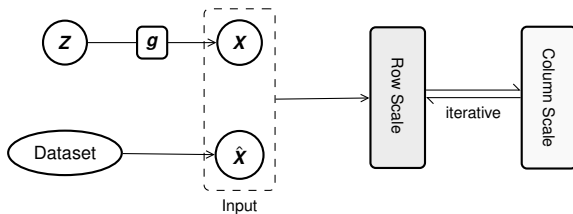
- joint distribution  $\pi^*$
- subgradient  $\beta^*$

which provides the gradient information for adjusting  $\mathbf{g}$ .

# EOTGM AND EOTGAN

## EOTGM

generated v.s. empirical sample comparison

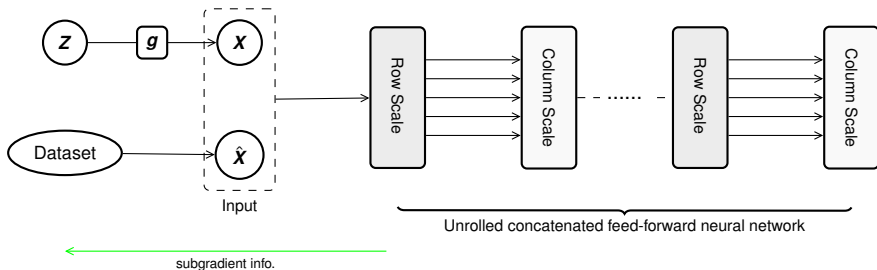


subgradient info.

# EOTGM AND EOTGAN

## EOTGM

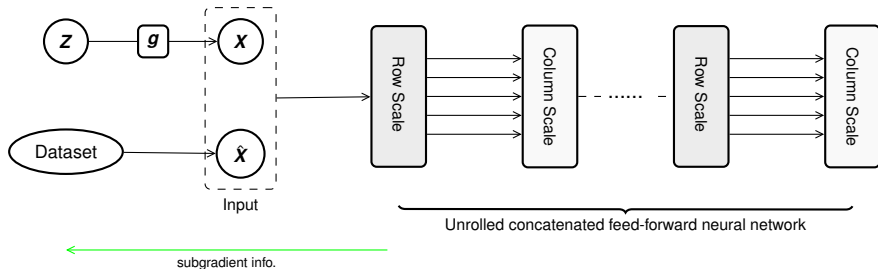
generated v.s. empirical sample comparison



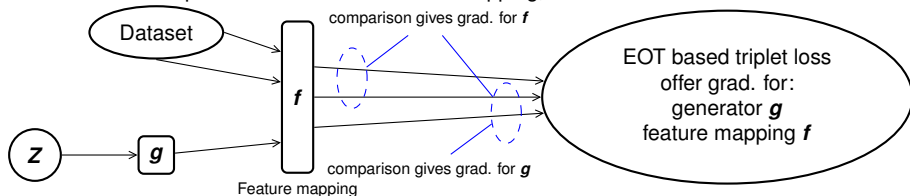
# EOTGM AND EOTGAN

## EOTGM

generated v.s. empirical sample comparison



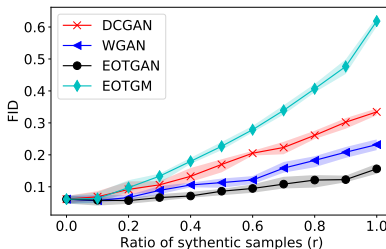
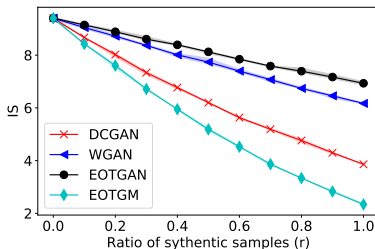
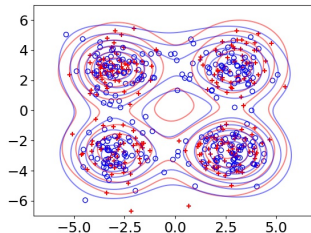
EOTGAN: learn implicit distribution with feature mapping, c.f. Section 8.2.2



# NUMERICAL

→ Toy distribution learning (target at 4-mixture Gaussians) using EOTGM. Real samples (red '+') and contour (red curve), versus generated samples (blue 'o') and contour (blue curve) by  $g$ .

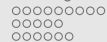
↓ Comparison of IS and FID (on MNIST) versus mixing ratio  $r$ .



# SUMMARY

## Wrap-up

- Inference with message-passing and analysis
- Inference with free energy minimization by neural networks
- Inference .. Learning: their interactions
- Neural network generators in EM for more flexible modeling; A Further step into temporal models
- A bonus modeling method for likelihood-free learning



Thank you for your attention.  
Q&A.