Inference and Representation
Lecture 10

Joan Bruna
Courant Institute, NYU

Lecture 10 Objectives

- Summary of lecture 9
  - CRFs
  - MLE under exponential family
- Pseudo-likelihood
- MAP Inference
  - Max-Product Algorithm

- Sequence-to-Sequence Models

- Unrolled Message Passing with Graph Neural Networks

Conditional Random Fields (CRF)

- CRFs are undirected graphical models for the conditional distribution $p(Y|X)$ (Y = target variables, X = observed variables).
- A particular case of what we have seen so far:

$$p(Y|X=x) = \frac{1}{Z(x)} \prod_{c \in \mathcal{C}} \phi_c(x_c, y_c) \ ,$$

$$Z(x) = \sum_{y} \prod_{c \in \mathcal{C}} \phi_c(x_c, y_c) \ .$$

- The only difference with unconditional MRF is the normalization:
  - In a CRF, we marginalize only over Y.
  - In a MRF, we marginalize over both X, Y.

MLE under Exponential Family

$$\text{MLE: } \sup_{\theta} \langle \theta, f(x) \rangle - A(\theta)$$

- Q1: Have we seen this function before?

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \overline{f} - \mathbb{E}_{\theta}\{f\}$$

- Q2: How to implement this ascent algorithm in practice?

MLE and Maximum Entropy

- Recall that

$$A^*(\mu) = \sup_{\theta} \langle \theta, f(x) \rangle - A(\theta)$$

  is the convex conjugate of the log-partition function.

- $A^*(\mu) =$ (negative) maximum entropy of any distribution p such that $\mathbb{E}_{X \sim p} f(X) = \overline{f}$. $(\mu = \overline{f})$

MLE and Maximum Entropy

- Recall that

$$A^*(\mu) = \sup_{\theta} \langle \theta, f(x) \rangle - A(\theta)$$

  is the convex conjugate of the log-partition function.

- $A^*(\mu) =$ (negative) maximum entropy of any distribution p such that $\mathbb{E}_{X \sim p} f(X) = \overline{f}$. $(\mu = \overline{f})$

- MLE solution satisfies $\mathbb{E}_{\theta} f = \overline{f}$. (why?)
  - This is denoted as *Moment Matching:* Empirical and model moments match at optimality.

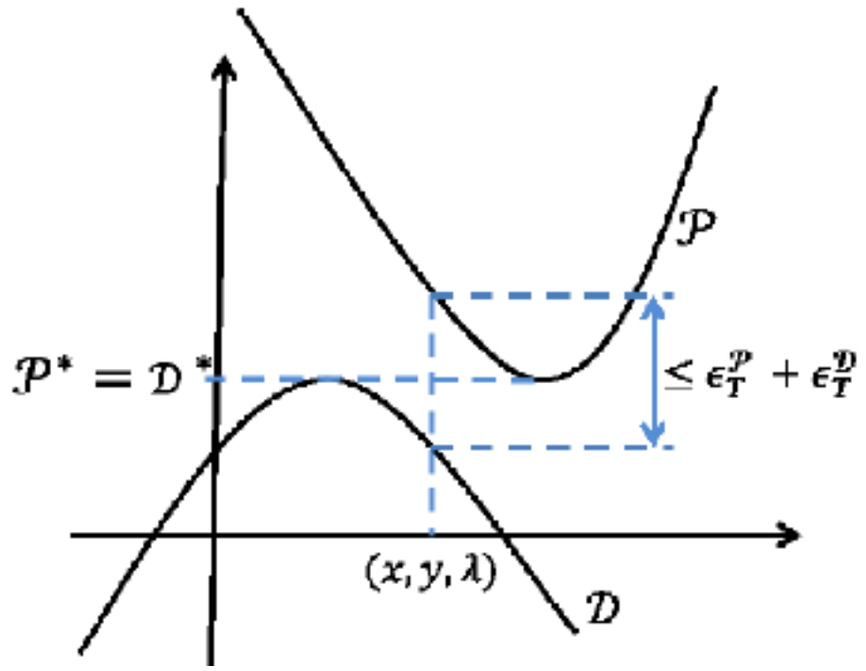
MLE and Maximum Entropy

- This is an instance of convex duality:
  - On the one hand, we assume exponential family and have shown that MLE implies model moments = empirical moments.
  - On the other hand, we assume model moments match empirical moments and shown that Max Ent implies exponential family.

MLE and Maximum Entropy

- This is an instance of convex duality:
  - On the one hand, we assume exponential family and have shown that MLE implies model moments = empirical moments.
  - On the other hand, we assume model moments match empirical moments and shown that Max Ent implies exponential family.
- Moreover, both objectives reach the same value at optimality:

$$A^*(\mu) = \sup_\theta \langle \theta, \mu \rangle - A(\theta) \qquad A(\theta) = \sup_\mu \langle \theta, \mu \rangle - A^*(\mu) \ .$$



Monte-Carlo

- Q2: How to efficiently implement such ascent method?

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \overline{f} - \mathbb{E}_\theta\{f\}$$

Monte-Carlo

- Q2: How to efficiently implement such ascent method?

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \overline{f} - \mathbb{E}_\theta\{f\}$$

- Use MCMC and estimate the gradient from

$$\widehat{\mathbb{E}_\theta f} = \frac{1}{S} \sum_{x^{(s)} \sim p_\theta} f(x^{(s)})$$

- We never need to estimate $A(\theta)$.

- Q: Other alternatives?

Approximating log-partition function

- The original learning objective function was

$$\mathcal{L}(\theta) = \log p(x^1, \ldots, x^L; \theta) = \left\langle \theta \ , \ \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(x_c^l) \right\rangle - A(\theta) \ .$$

- We can consider a tractable approximation of the log-partition function

$$\widetilde{\mathcal{L}}(\theta) = \left\langle \theta \ , \ \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(x_c^l) \right\rangle - \tilde{A}(\theta) \ .$$

- For instance using convex relaxations, ie variational methods.

  - tree-reweighed belief propagation [Wainwright, Wilski & Jakkola] (based on "convexifying" Bethe approximation).
  - log-determinant relaxation [Wainwright & Jordan].

Pseudo-Likelihood

- An alternative route is to replace the objective function $\mathcal{L}(\theta)$ by another one $\bar{\mathcal{L}}(\theta)$, such that they are asymptotically maximized at the same parameter value as $L \to \infty$

Pseudo-Likelihood

- An alternative route is to replace the objective function $\mathcal{L}(\theta)$ by another one $\bar{\mathcal{L}}(\theta)$, such that they are asymptotically maximized at the same parameter value as $L \to \infty$

- The Pseudo-likelihood method (Besag'71) is a consistent estimator if the data is generated by a model p_{θ_0} in our family.

- From Bayes rule, we have

$$p(x;\theta) = \prod_i p(x_i \mid x_1 \ldots, x_{i-1};\theta) \ .$$

Pseudo-Likelihood

- An alternative route is to replace the objective function $\mathcal{L}(\theta)$ by another one $\bar{\mathcal{L}}(\theta)$, such that they are asymptotically maximized at the same parameter value as $L \to \infty$

- The Pseudo-likelihood method (Besag'71) is a consistent estimator if the data is generated by a model p_{θ_0} in our family.

- From Bayes rule, we have

$$p(x;\theta) = \prod_i p(x_i \mid x_1 \ldots, x_{i-1};\theta) \ .$$

- We consider the approximation

$$p(x;\theta) \approx \prod_i p(x_i \mid x_1 \ldots, x_{i-1}, x_{i+1}, \ldots, x_n;\theta) = \prod_i p(x_i \mid x_{-i};\theta)$$

– conditioning over all the other variables (similarly as in Gibbs sampling).

Pseudo-Likelihood

- We replace the likelihood $\mathcal{L}(\theta)$ by the "pseudo-likelihood"

$$\mathcal{L}_P(\theta) = \frac{1}{L} \sum_{l \leq L} \sum_{i=1}^{n} \log p(x_i^l \mid x_{N(i)}^l; \theta) \ .$$

$$N(i) = \text{ Markov Blanket of node } i \ .$$

- Example: pair-wise MRF. Then

$$p(x_i \mid x_{N(i)}; \theta) = \frac{1}{Z(x_{N(i)}; \theta)} \exp \left(\sum_{j \in N(i)} \theta_{ij}(x_i, x_j) \right) \ ,$$

$$Z(x_{N(i)}; \theta) = \sum_{x_i} \exp \left(\sum_{j \in N(i)} \theta_{ij}(x_i, x_j) \right) \ .$$

Pseudo-Likelihood

- This approximation only involves "local" partition functions
  - We need to compute n univariate integrals instead of a single n-dimensional one.
- In the exponential family, the pseudo-likelihood is still concave.
- Although it is asymptotically consistent as $L \to \infty$, this does not mean it is optimal for a fixed L.
- Again, we have a comptutational/statistical tradeoff.

Structured Prediction and MAP

- So far, we have considered fitting in terms of maximizing the empirical likelihood of the model on observed data:

$$\mathcal{L}(\theta) = \log p(x^1, \ldots, x^L; \theta) = \left\langle \theta \ , \ \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(x_c^l) \right\rangle - A(\theta) \ .$$

- In the conditional case, the partial likelihood is

$$\mathcal{L}(\theta) = \log p(\{x, y\}^1, \ldots, \{x, y\}^L; \theta) = \left\langle \theta \ , \ \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(y_c^l, x^l) \right\rangle - L^{-1} \sum_l A(x^l; \theta)$$

- If prediction is done with MAP inference, we have an alternative route.

$$\hat{y}_{\text{MAP}} = \arg\max_y \log p(x, y; \theta) \ ,$$

$$\mathcal{L}_{\text{disc}}(\theta) = L^{-1} \sum_{l \leq L} \mathbf{1}(\hat{y}_{\text{MAP}}^l \neq y^l) \ .$$

Different Learning Objectives

- Fitting models to the data requires choosing the appropriate objective function for the task.

- Density Estimation: We are interested in the full joint distribution, so that later it can be used to extract any marginal or any moment.
- Prediction Task: We know *beforehand* which specific moment (e.g. $\boldsymbol{p(y\ |x)}$).
- *Knowledge* Discovery: We are interested in the graphical structure.

- In particular, in prediction problems, we often need to solve (or approximately solve) MAP inference!

MAP Inference

- For simplicity, start with unconditional setup:

$$\arg\max_x p(x) \ , \ p(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \phi_c(x_c) \ .$$

- Using energy-based formulation $\phi_c(x_c) = \exp\{\theta_c(x_c)\}$, this is equivalent to

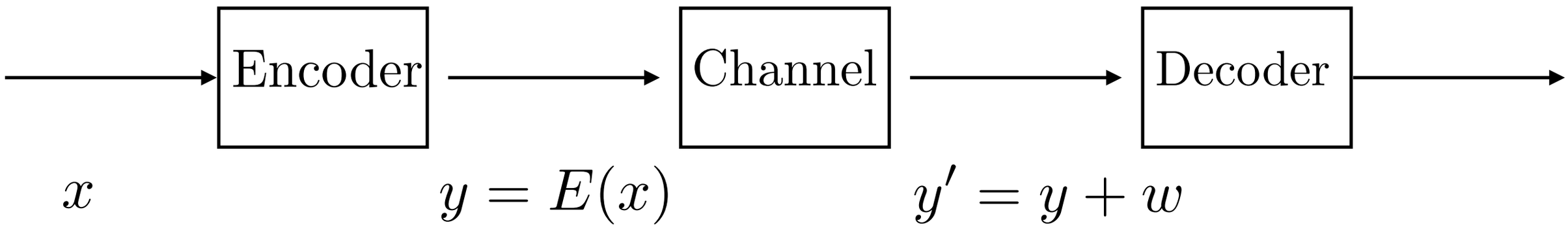
$$\arg\max_x \prod_{c \in \mathcal{C}} \phi_c(x_c) \ , \ \text{(max-product)}$$

$$\arg\max_x \sum_{c \in \mathcal{C}} \theta_c(x_c) \ , \ \text{(max-sum)} \ .$$

Motivation

- Error-Correcting Codes:

$$\hat{x} = \arg\max_x p(x \mid y')$$



- Example using LDPC codes:

$$p(y \mid y') = \frac{1}{Z(y')} \prod_{i=1}^{N} q(y'_i|y_i) \prod_{a=1}^{M} \mathbf{1}(y_{i_1^a} \oplus \cdots \oplus y_{i_{k(a)}^a} = 0) \,.$$

noise distribution

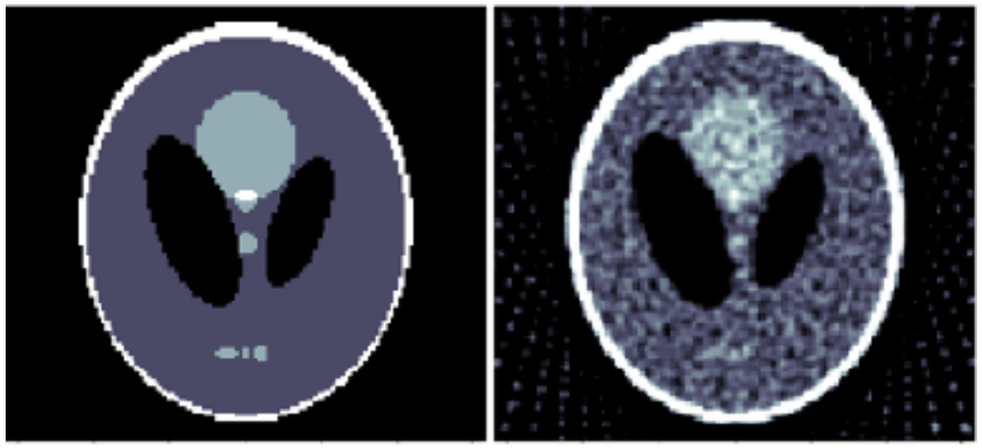
code constraints
(M factors)

- This is an instance of a denoising inverse problem over discrete alphabets.

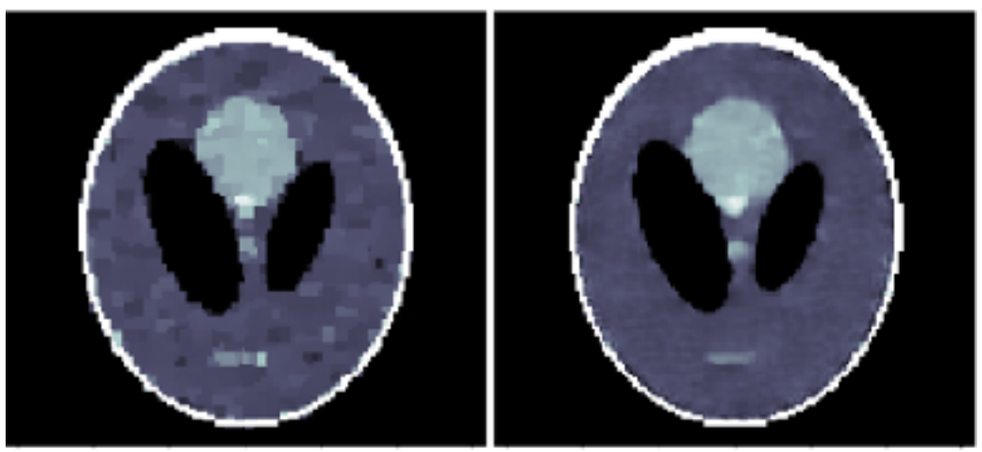
Motivation: Inverse Problems in Signal Processing

- Image Denoising

- Low-Dose Tomography



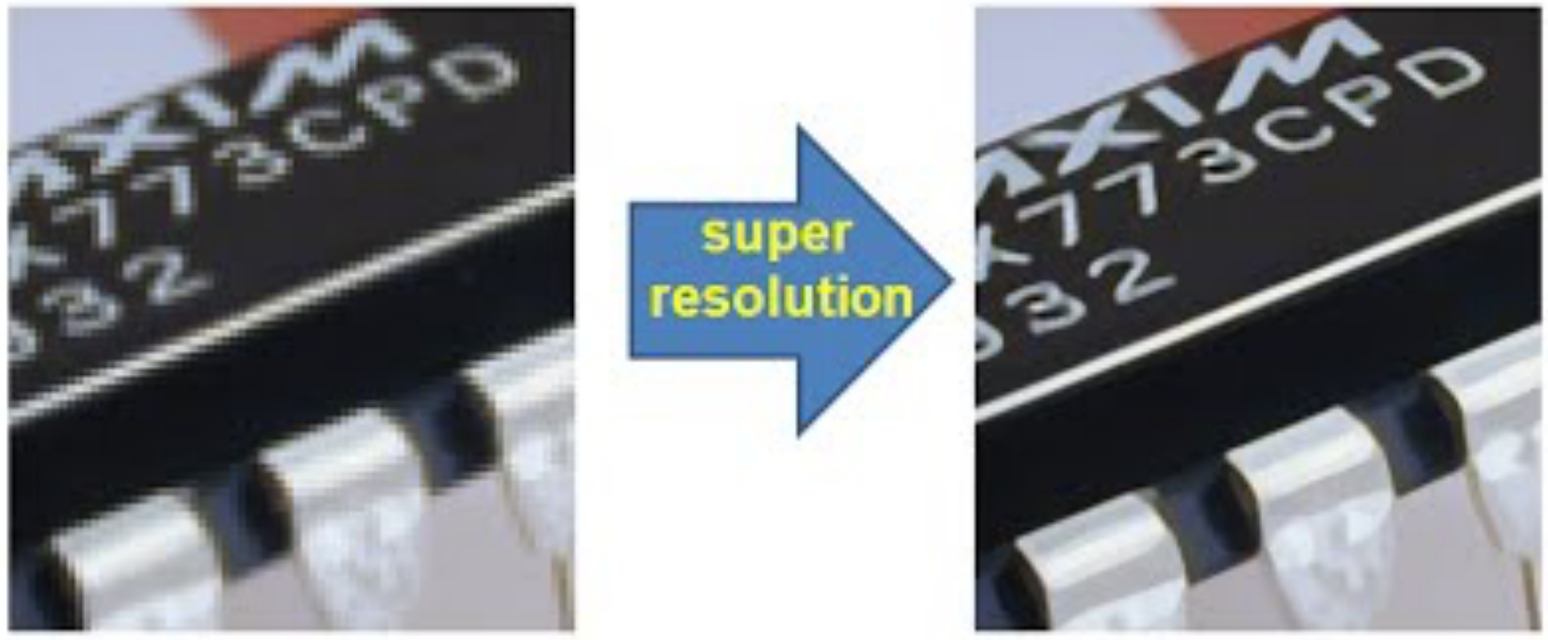
(a) Phantom (b) FBP



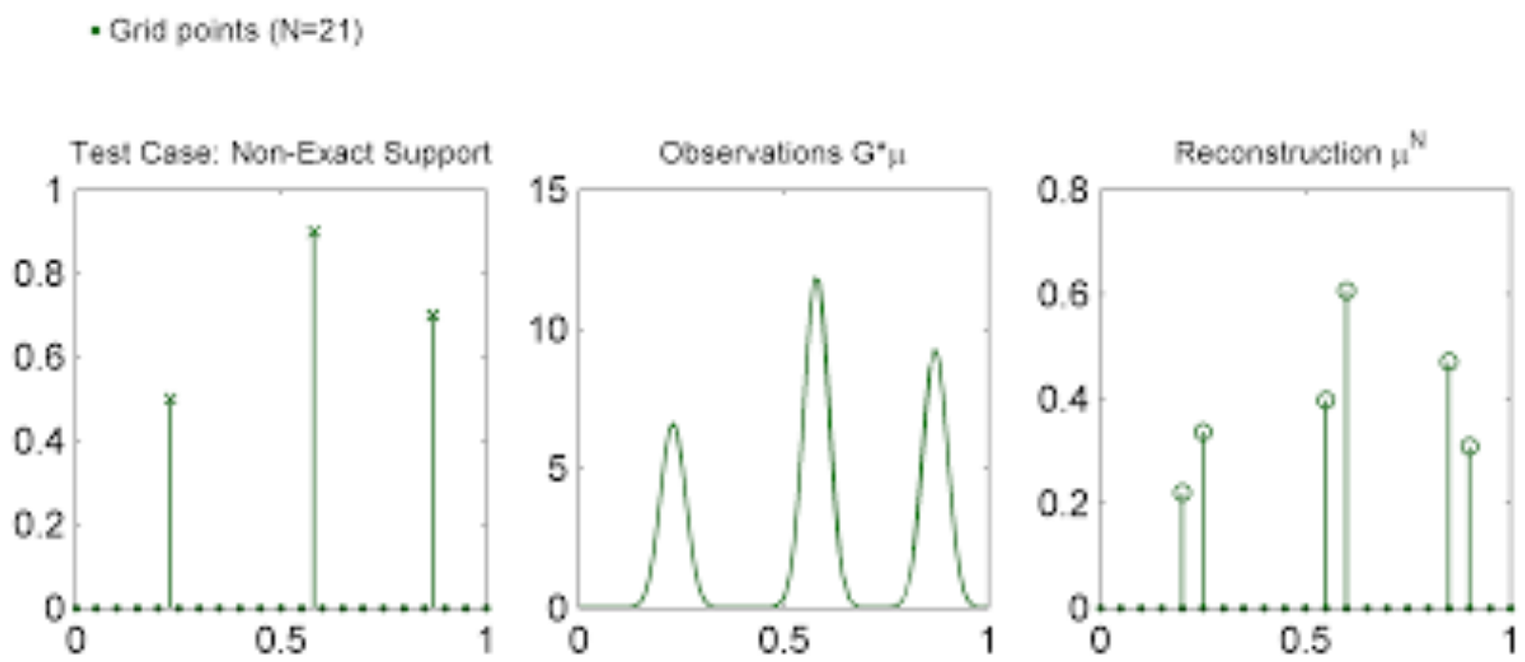
(c) TV (d) Partially learned gradient scheme

Motivation: Inverse Problems in Signal Processing

- Image Super-Resolution

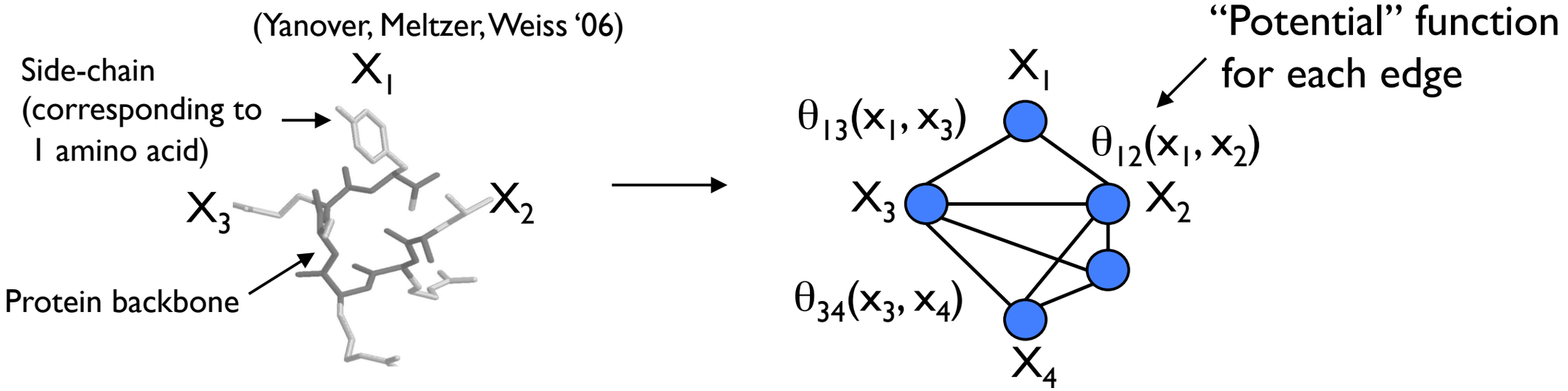


- Sparse Spike Deconvolution



Protein Side-Chain Placement

- Find minimum energy configuration of amino acid side-chains along a fixed carbon backbone:



- Orientations of the side-chains are represented by discretized angles (rotamers).
- Rotamers for nearby amino-acids are energetically coupled (attractive and repulsive forces).

MAP Inference is generally hard

- Local Search
  - Visit variables along certain order, and change state greedily to maximize the value of the assignment.
- *Branch-and-Bound*
  - Exhaustive search over space of assignments, pruning branches that can be provably shown not to contain the MAP assignment.
  - Can use LP relaxations or dual decompositions (later) to obtain upper bounds.
  - Q: How to obtain a lower bound?
- *Branch-and-Cut*
  - Same as branch-and-bound, but spend more resources getting tighter bounds.
  - Adds new constraints to cut off fractional solutions of the LP relaxation.

Approximate MAP with Message-Passing

- Similarly as with BP, we can use a variational framework to perform approximate MAP: pretend the graph is a tree and derive an exact algorithm.

- Instead of marginal probabilities, we consider here *max-marginals*:

$$M_i(x_i^*) = \max\{p(x) \ : \ x_i = x_i^*\} \ .$$
$$x$$

- Q: Why are max-marginals useful to perform MAP inference?
  - If for each $i \in V$ there exists x_i^* such that $M_i(x_i^*) > M_i(\tilde{x}_i)$ for any $\tilde{x}_i \neq x_i^*$, then

$$x^* = (x_1^*, \ldots, x_N^*) \ , \ p(x^*) > p(x) \text{ for } x \neq x^* \ .$$

Approximate MAP with MP

- More generally, we choose an ordering of the variables, e.g. (1..N), and:
  - Compute $M_1(x_1)$ and maximize it: $x_1^* = \arg\max M_1(x_1)$.
  - Condition the model $p(x \mid x_1 = x_1^*)$
  - Compute $M_2(x_2)$ for this new model, fix $x_2^* = \arg\max_{x_2} M_2(x_2)$.
  - Iterate.

- Since Max-marginals are only used to maximize, we only need them up to normalization.

- Q: How to adapt BP to compute those max-marginals?

Approximate MAP with MP

- We adapt message passing update rules to compute max-marginals by replacing sums with maximizations:

$$\eta_{i\to a}^{(k+1)}(x_i) \simeq \prod_{b\in N(i)\setminus\{a\}} m_{b\to i}^{(k)}(x_i) \ ,$$

$$m_{a\to i}^{(k)}(x_i) \simeq \max_x \left\{ \phi_a(x) \prod_{j\in N(a)\setminus\{i\}} \eta_{j\to a}^{(k)}(x_j) \right\} \ .$$

- The Max-marginals are estimated as

$$\eta_i^{(k)}(x_i) \simeq \prod_{a\in N(i)} m_{a\to i}^{(k-1)}(x_i) \ .$$

Max-Product

- Similarly as BP, this algorithm is *exact* on trees:

Theorem: Consider a tree graphical model with diameter t_*. Then
- Independently of initialization, Max-Product updates converge after at most t_* iterations.
- For any node i and $t > t_*$, $\eta_i^{(t)}(x_i) = M_i(x_i)$.

- The equivalent min-sum formulation is called in statistical physics as the *Energy Cavity Method*.
- In the special case of HMMs, the *Viterbi* algorithm uses these updates
  - Heavily used in channel decoding.

Dual Decomposition

- Consider the MAP inference in a pairwise Markov Random Field (max-sum formulation):

$$\mathrm{MAP}(\theta) = \max_x \left\{ \sum_{i \in V} \theta_i(x_i) + \sum_{(ij) \in E} \theta_{ij}(x_i, x_j) \right\} .$$

- We have

$$\mathrm{MAP}(\theta) \leq \sum_{i \in V} \max_{x_i} \theta_i(x_i) + \sum_{(ij) \in E} \max_{x_i, x_j} \theta_{ij}(x_i, x_j) .$$

  - Note that the upper bound can be efficiently evaluated.

- The parametrization θ is not unique: we can reparametrize

$$\tilde{\theta}_i(x_i) = \theta_i(x_i) + f(x_i) \ , \ \tilde{\theta}_{ij}(x_i, x_j) = \theta_{ij}(x_i, x_j) - f(x_i)$$

$$\text{For any } f, \ p_\theta \equiv p_{\tilde{\theta}}.$$

- Q: How to reparametrize to make RHS as tight as possible?

Dual Decomposition

- Let us consider

$$\tilde{\theta}_i(x_i) = \theta_i(x_i) + \sum_{(ij) \in E} \delta_{j \to i}(x_i) \ ,$$

$$\tilde{\theta}_{ij}(x_i, x_j) = \theta_{ij}(x_i, x_j) - \delta_{j \to i}(x_i) - \delta_{i \to j}(x_j) \ .$$

- We verify that

$$\forall \ x \ , \ \sum_i \theta_i(x_i) + \sum_{(ij) \in E} \theta_{(ij)}(x_i, x_j) = \sum_i \tilde{\theta}_i(x_i) + \sum_{(ij) \in E} \tilde{\theta}_{(ij)}(x_i, x_j) \ .$$

- Thus

$$\mathrm{MAP}(\theta) = \mathrm{MAP}(\tilde{\theta}) \leq \sum_{i \in V} \max_{x_i} \tilde{\theta}_i(x_i) + \sum_{(ij) \in E} \max_{x_i, x_j} \tilde{\theta}_{ij}(x_i, x_j) \ .$$

- Goal: Optimize the bound wrt δ .

Dual Decomposition

- We define the following dual objective:

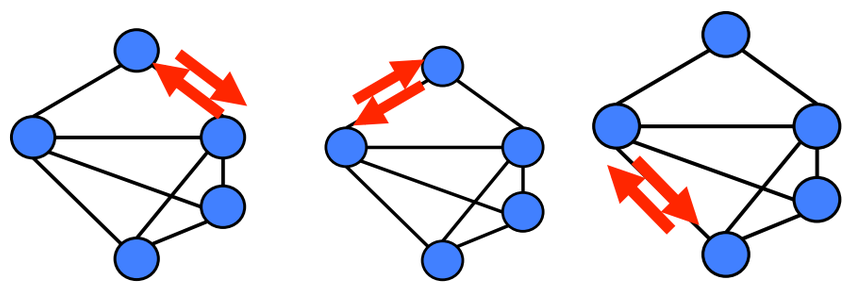
$$L(\delta) = \sum_{i \in V} \max_{x_i} \left(\theta_i(x_i) + \sum_{j:(ij) \in E} \delta_{j \to i}(x_i) \right) + \sum_{(ij) \in E} \max_{x_i, x_j} \left(\theta_{ij}(x_i, x_j) - \delta_{j \to i}(x_i) - \delta_{i \to j}(x_j) \right)$$

$$\text{DUAL-LP}(\theta) = \min_{\delta} L(\delta) \ .$$

- Relaxation of MAP:

$$\forall \ \delta \ , \ \text{MAP}(\theta) \leq \text{DUAL-LP}(\theta) \leq L(\delta) \ .$$

- How to solve this Dual Linear Program?
- An efficient strategy is to use block coordinate descent, which yields a very similar algorithm as max-product:



Max-Product Linear Programming (MPLP)

Input: A set of factors $\theta_i(x_i), \theta_{ij}(x_i, x_j)$

Output: An assignment $x_1, \ldots, x_n$ that approximates the MAP

Algorithm:

- Initialize $\delta_{i\to j}(x_j) = 0, \quad \delta_{j\to i}(x_i) = 0, \quad \forall ij \in E, x_i, x_j$
- Iterate until small enough change in $L(\delta)$:

  For each edge $ij \in E$ (sequentially), perform the updates:

$$\delta_{j\to i}(x_i) \quad = \quad -\frac{1}{2}\delta_i^{-j}(x_i) + \frac{1}{2}\max_{x_j}\left[\theta_{ij}(x_i, x_j) + \delta_j^{-i}(x_j)\right] \quad \forall x_i$$

$$\delta_{i\to j}(x_j) \quad = \quad -\frac{1}{2}\delta_j^{-i}(x_j) + \frac{1}{2}\max_{x_i}\left[\theta_{ij}(x_i, x_j) + \delta_i^{-j}(x_i)\right] \quad \forall x_j$$

  where $\delta_i^{-j}(x_i) = \theta_i(x_i) + \sum_{ik\in E, k\neq j} \delta_{k\to i}(x_i)$

- Return $x_i \in \arg\max_{\hat{x}_i} \tilde{\theta}_i^{\delta}(\hat{x}_i)$

- Algorithm Converges on general pairwise MRFs.
- Extension to arbitrary factor graphs.
- More efficient than tree-reweighed Max-Product.

Sequence Structured Prediction

- Many tasks require a prediction from sequence to sequence:

Machine Translation

There is a light that never goes out

Il y a une lumière qui ne disparait jamais

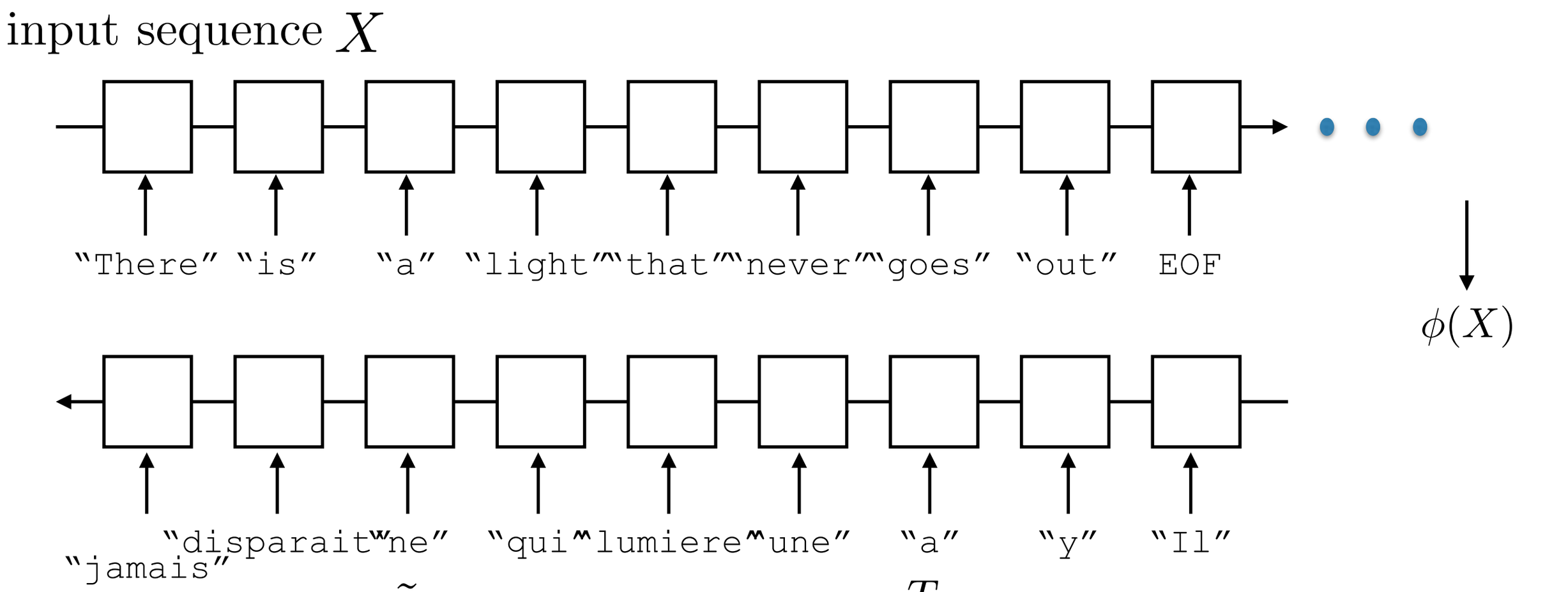
Question Answering

What is the best ramen place in the Bay Area?

Ramen Shop, in Rockridge

Sequence Structured Prediction

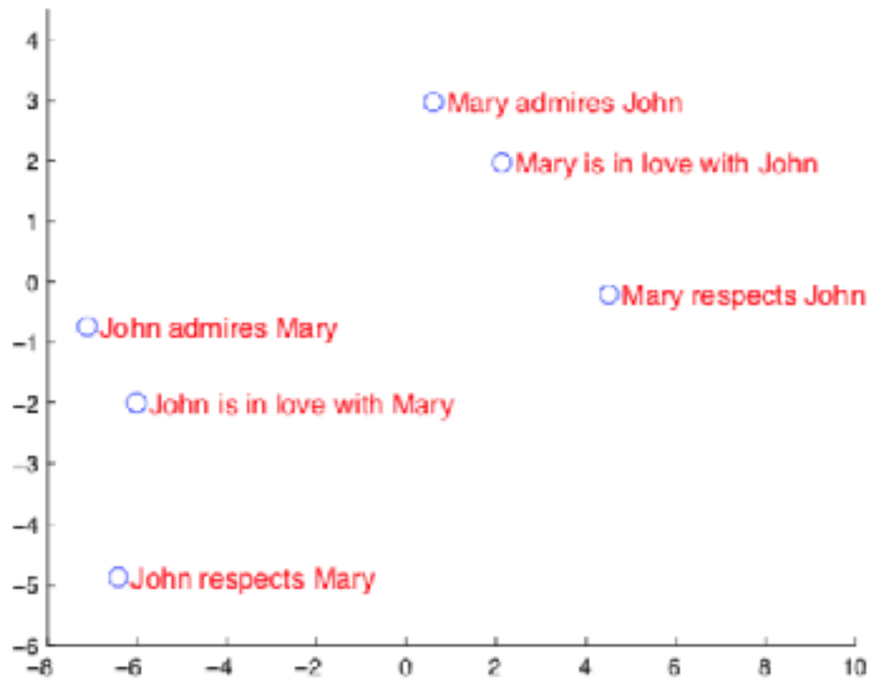
- Conditional model:
  - Input sequence is used to initialize the state of the output decoder.

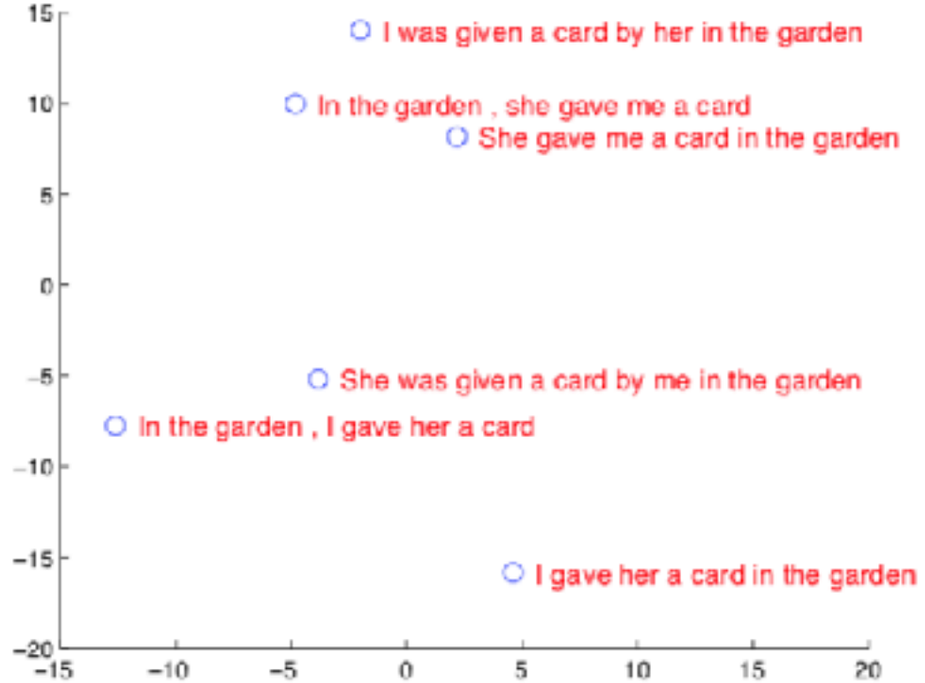


$$p(\tilde{X} \mid X) = \prod_{t=0}^{T} p(\tilde{X}_{t+1} \mid X, \tilde{X}_1, \ldots, \tilde{X}_t)$$

Sequence-to-Sequence

- In [Sutskever, Vinyals & Le'14], authors propose to model these conditional distributions with Long-Short Term Memory networks (LSTMs).
  - Trained with Maximum likelihood (softmax over vocabulary).
  - At Test, beam-search: a greedy sequential strategy for MAP inference.





  - The hidden states learnt by the model capture meaning that is sensitive to word ordering.

Sequence-to-Sequence

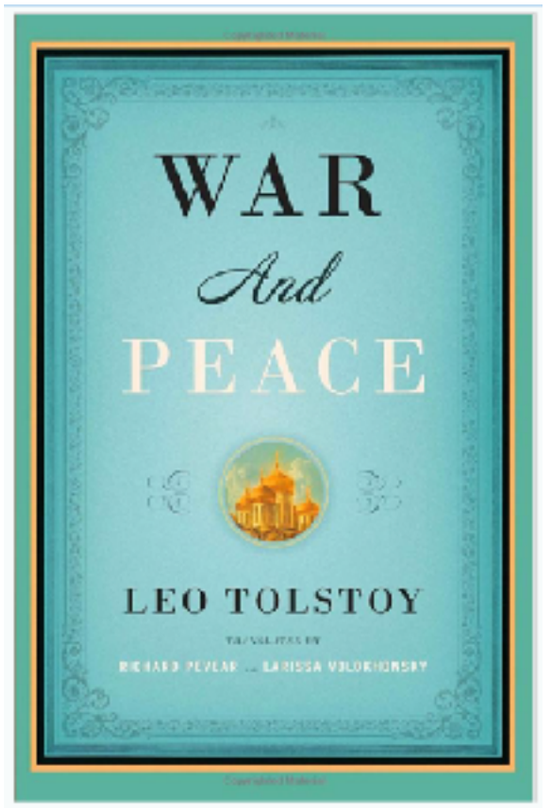
- In [Sutskever, Vinyals & Le'14], authors propose to model these conditional distributions with Long-Short Term Memory networks (LSTMs)

Type	Sentence
Our model	Ulrich UNK , membre du conseil d' administration du constructeur automobile Audi , affirme qu' il s' agit d' une pratique courante depuis des années pour que les téléphones portables puissent être collectés avant les réunions du conseil d' administration afin qu' ils ne soient pas utilisés comme appareils d' écoute à distance .
Truth	Ulrich Hackenberg , membre du conseil d' administration du constructeur automobile Audi , déclare que la collecte des téléphones portables avant les réunions du conseil , afin qu' ils ne puissent pas être utilisés comme appareils d' écoute à distance , est une pratique courante depuis des années .
Our model	" Les téléphones cellulaires , qui sont vraiment une question , non seulement parce qu' ils pourraient potentiellement causer des interférences avec les appareils de navigation , mais nous savons , selon la FCC , qu' ils pourraient interférer avec les tours de téléphone cellulaire lorsqu' ils sont dans l' air " , dit UNK .
Truth	" Les téléphones portables sont véritablement un problème , non seulement parce qu' ils pourraient éventuellement créer des interférences avec les instruments de navigation , mais parce que nous savons , d' après la FCC , qu' ils pourraient perturber les antennes-relais de téléphonie mobile s' ils sont utilisés à bord " , a déclaré Rosenker .

"Attention" Mechanisms

- Limits of sequence-to-sequence model.
  - All the information of the input sequence is contained in the vector

    $\phi(X)$
  - As the length of input increases, we require more information to perform the translation.

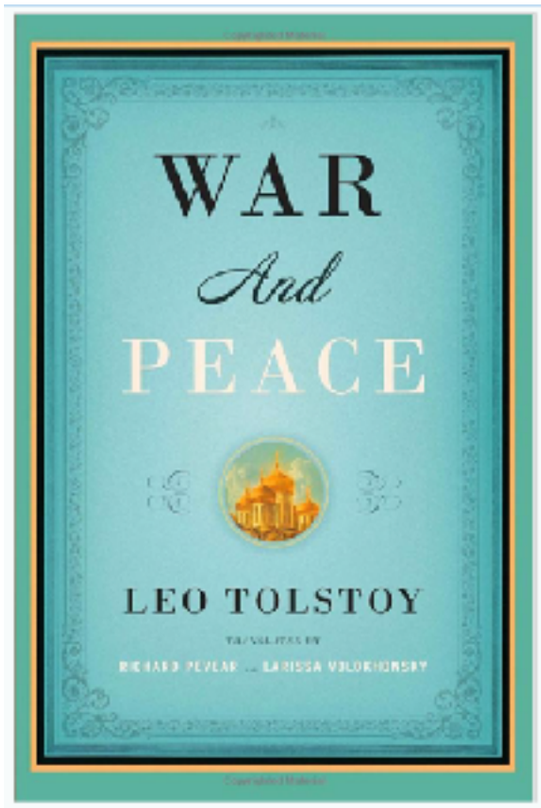


"Attention" Mechanisms

- Limits of sequence-to-sequence model.
  - All the information of the input sequence is contained in the vector

    $$\phi(X)$$

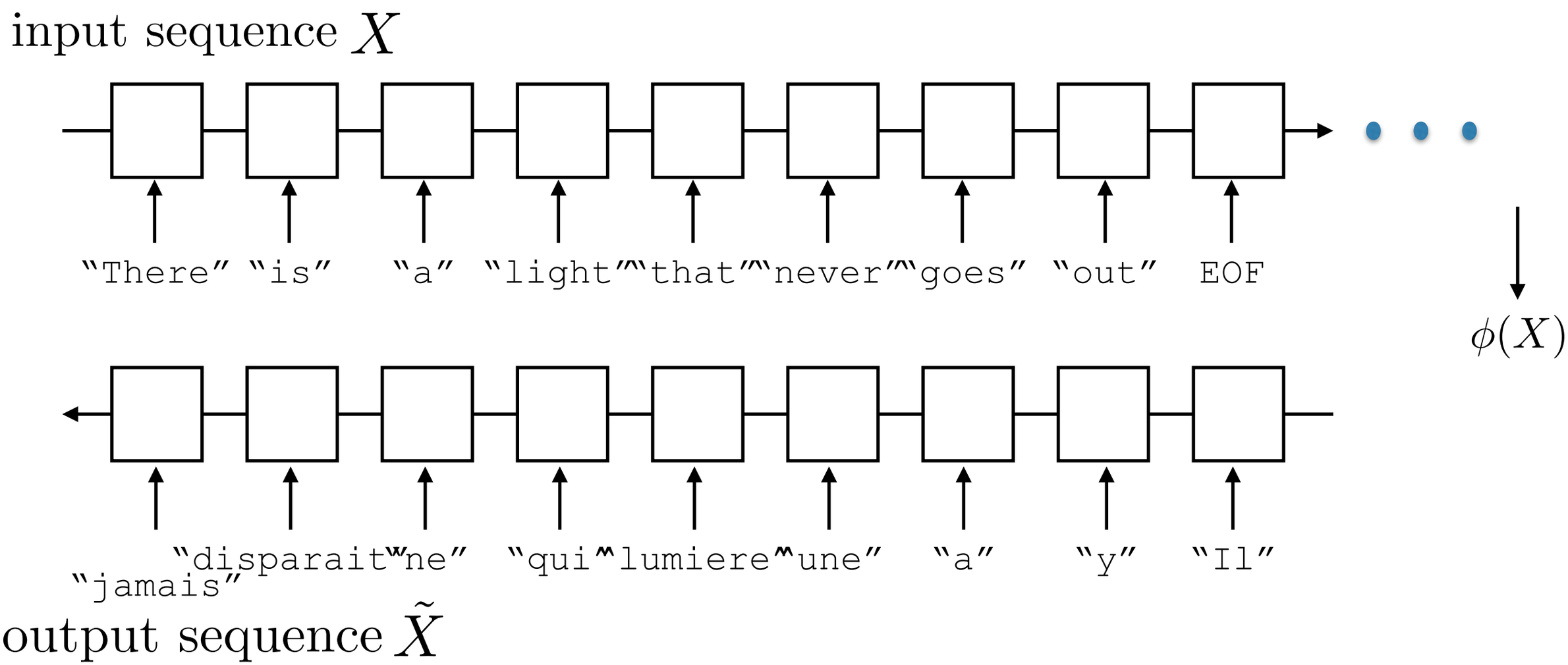
  - As the length of input increases, we require more information to perform the translation.



  - Although the ***global*** amount of information grows, the ***local*** amount of information required to translate does not. How to exploit it?

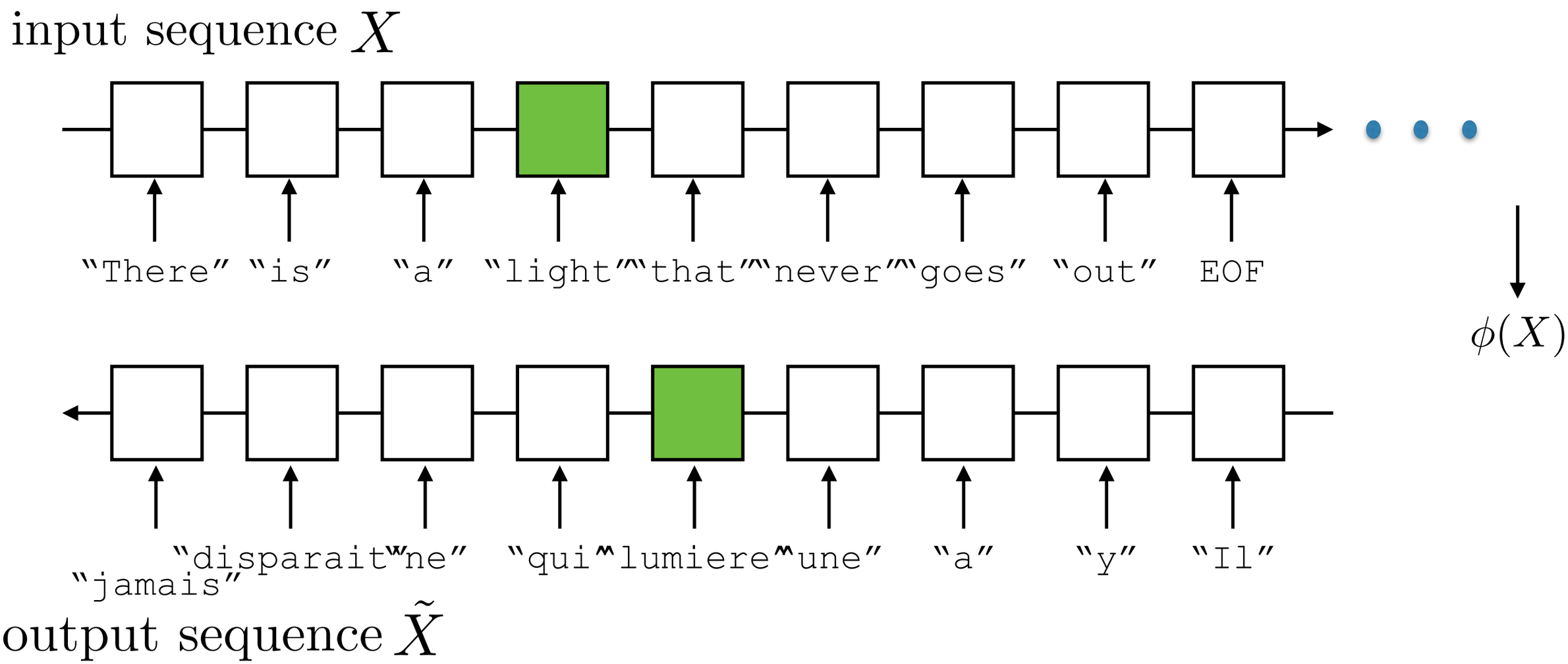
"Attention" Mechanisms

[Badhanu et al.'15]



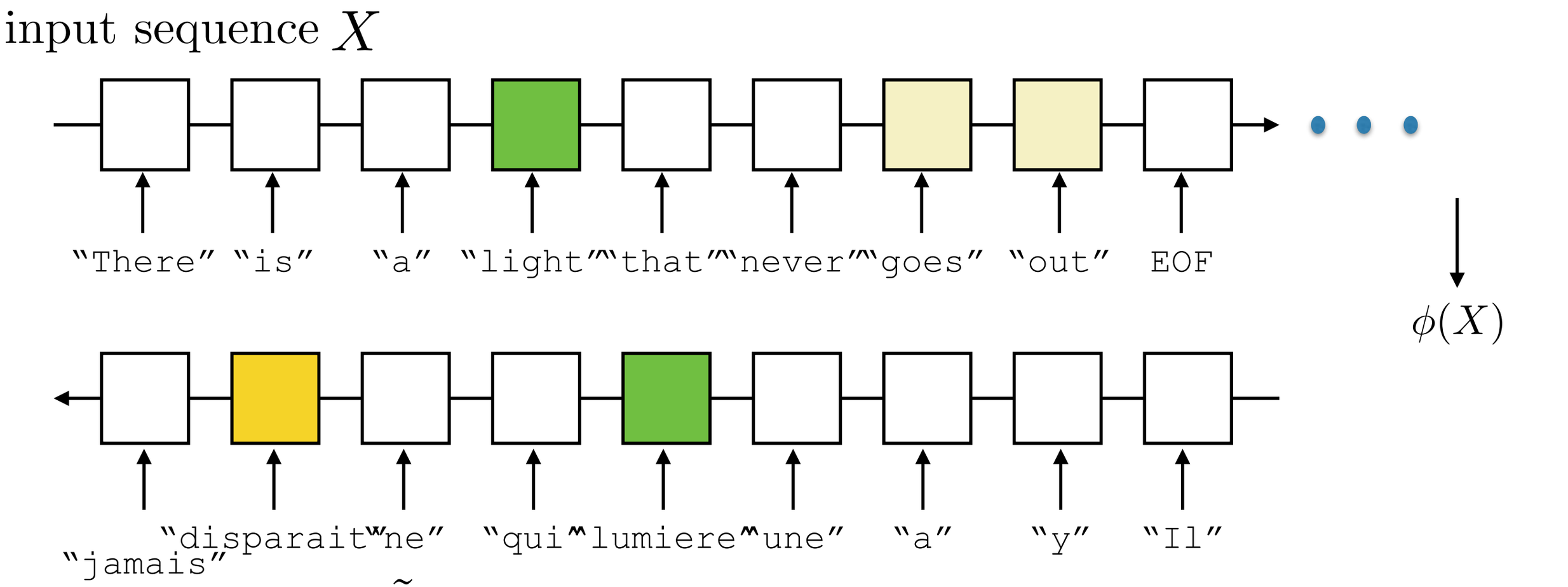
"Attention" Mechanisms

[Badhanu et al.'15]



"Attention" Mechanisms

[Badhanu et al.'15]



output sequence $\tilde{X}$

$$p(\tilde{X} \mid X) = \prod_{t=0}^{T} p(\tilde{X}_{t+1} \mid att(X, \tilde{X}_1, \ldots, \tilde{X}_t), \tilde{X}_1, \ldots, \tilde{X}_t)$$

"Attention" Mechanisms

[Badhanu et al.'15]

- Pros
  - Generalizes to larger input/output sequences.

- Challenges
  - Harder to train
  - How to address larger memories efficiently?
  - Learning where to look?

Machine Translation

Source	An admitting privilege is the right of a doctor to admit a patient to a hospital or a medical centre to carry out a diagnosis or a procedure, based on his status as a health care worker at a hospital.
Reference	Le privilège d'admission est le droit d'un médecin, en vertu de son statut de membre soignant d'un hôpital, d'admettre un patient dans un hôpital ou un centre médical afin d'y délivrer un diagnostic ou un traitement.
RNNenc-50	Un privilège d'admission est le droit d'un médecin de reconnaître un patient à l'hôpital ou un centre médical d'un diagnostic ou de prendre un diagnostic en fonction de son état de santé.
RNNsearch-50	Un privilège d'admission est le droit d'un médecin d'admettre un patient à un hôpital ou un centre médical pour effectuer un diagnostic ou une procédure, selon son statut de travailleur des soins de santé à l'hôpital.
Google Translate	Un privilège admettre est le droit d'un médecin d'admettre un patient dans un hôpital ou un centre médical pour effectuer un diagnostic ou une procédure, fondée sur sa situation en tant que travailleur de soins de santé dans un hôpital.

[Badhanu et al.,'15]

Unrolling MP with Neural Networks

- We have seen that structured output MAP prediction in a general graphical model is computationally hard:

$$\arg\max_y p(y \mid x) = \arg\max_y p(y, x)$$

- Learning is also hard in general factor graphs:

$$\max_\theta \log p(y \mid x; \theta) \ .$$

- If we could find a fast approximation of MAP solutions, we could train end-to-end using discriminative loss:

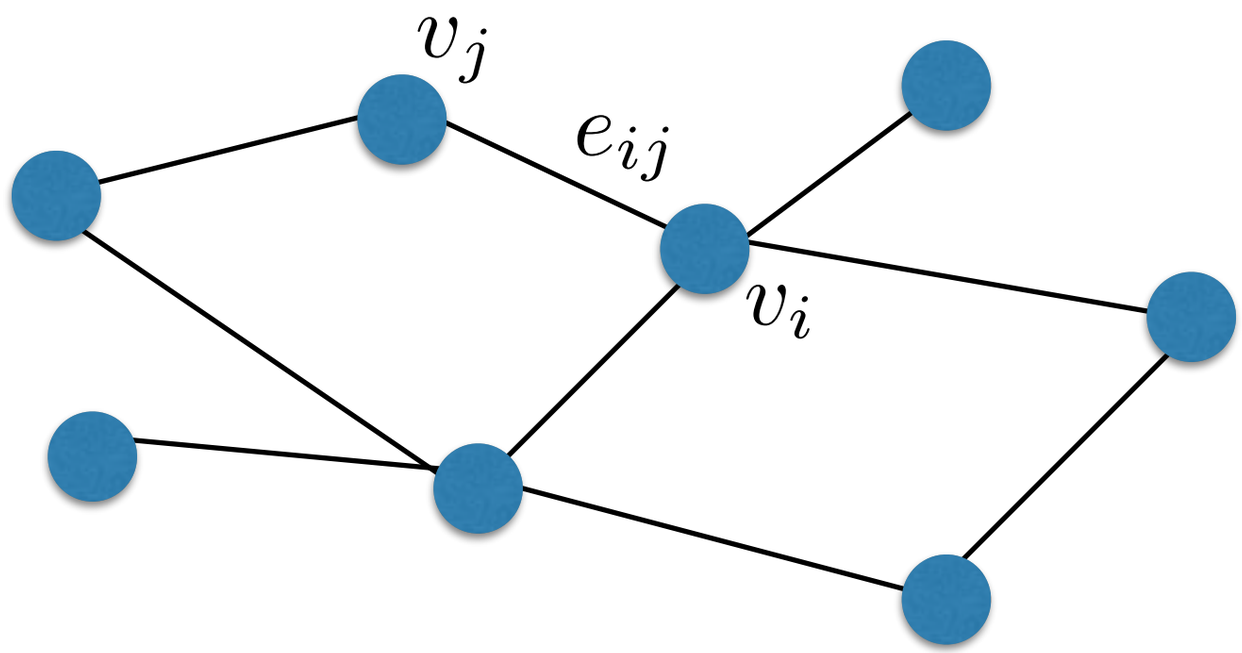
$$\hat{y}_{\text{MAP}} = \arg\max_y \log p(x, y; \theta) \ ,$$

$$\mathcal{L}_{\text{disc}}(\theta) = L^{-1} \sum_{l \leq L} \mathbf{1}(\hat{y}^l_{\text{MAP}} \neq y^l) \ .$$

- Idea: unroll finite number of Message-Passing iterations.

Message Passing and Neural Networks

- This defines a generic class of neural networks operating on graphs.



- Each layer of the network updates latent variables on both nodes and edges of the graph:

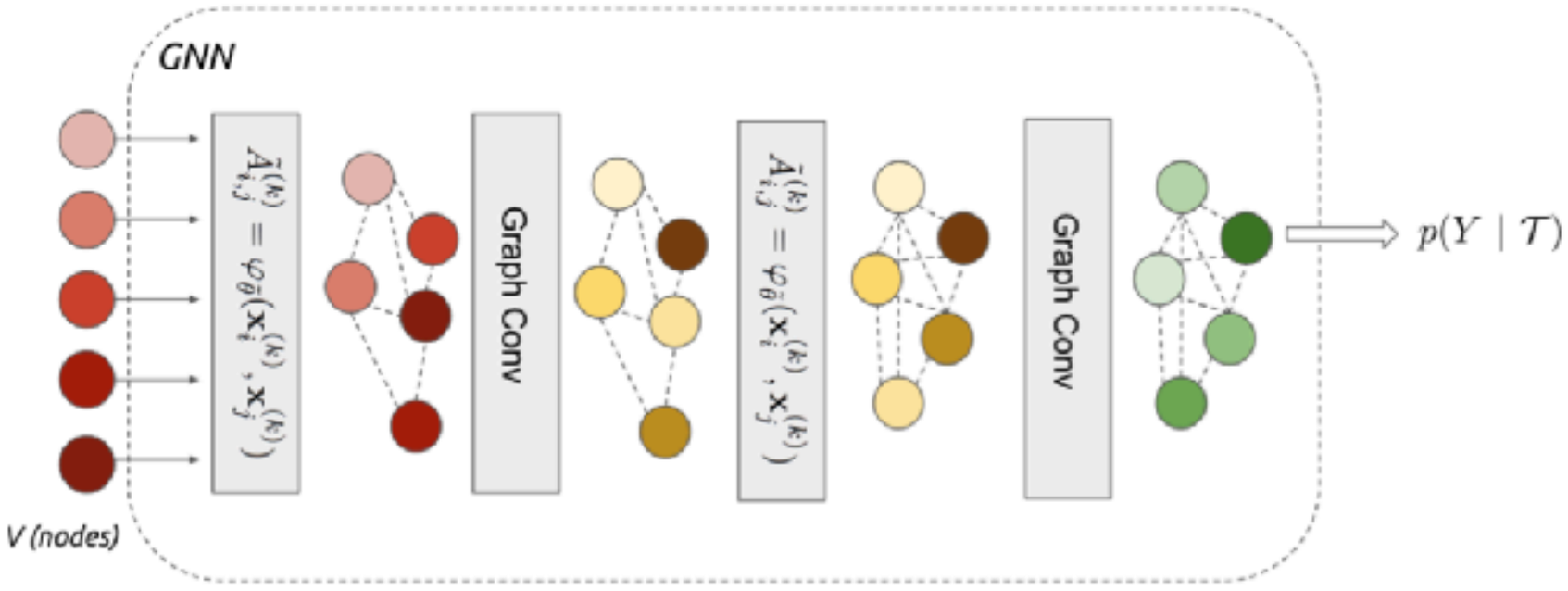
$$h_i^{(k+1)} = f_\theta \left(h_i^{(k)}, \sum_{(ij)\in E} H_{ij}^{(k)} h_j^{(k)} \right) ,$$

$$H_{ij}^{(k+1)} = g_\theta(h_i^{(k+1)}, h_j^{(k+1)}) .$$

f_θ, g_θ: Neural Networks

Neural Message-Passing

- These update equations have the same local structure as Message-Passing algorithms we have previously seen (BP, Max-Product).



- The main difference is that now, the MAP estimate is replaced with a parameterized estimate $\hat{Y}_\theta$.
  - We can train the parameters θ with gradient descent!
- These models are also "natural" generalization of CNNs on graphs.

Neural Networks and Inference

- More generally, we can view many iterative inference procedures as specific neural networks with "infinite", recursive layers:

$$\min_{\theta} E(\theta) \;\leftrightarrow\; \theta^{(t+1)} = \mathcal{G}_t(\theta^{(t)}) \;\leftrightarrow\; \Phi = \ldots \mathcal{G}_T \circ \mathcal{G}_{T-1} \cdots \circ \mathcal{G}_1$$

- We can always consider a finite-time, data-driven version of such inference by:
  - Truncating to T iterations.
  - Training the parameters of the inference algorithms directly on the loss we care about.
  - Pros: Typically leads to better computational/statistical tradeoffs.
  - Cons: lack of theoretical guarantees.

LISTA [Gregor and LeCun,'10]

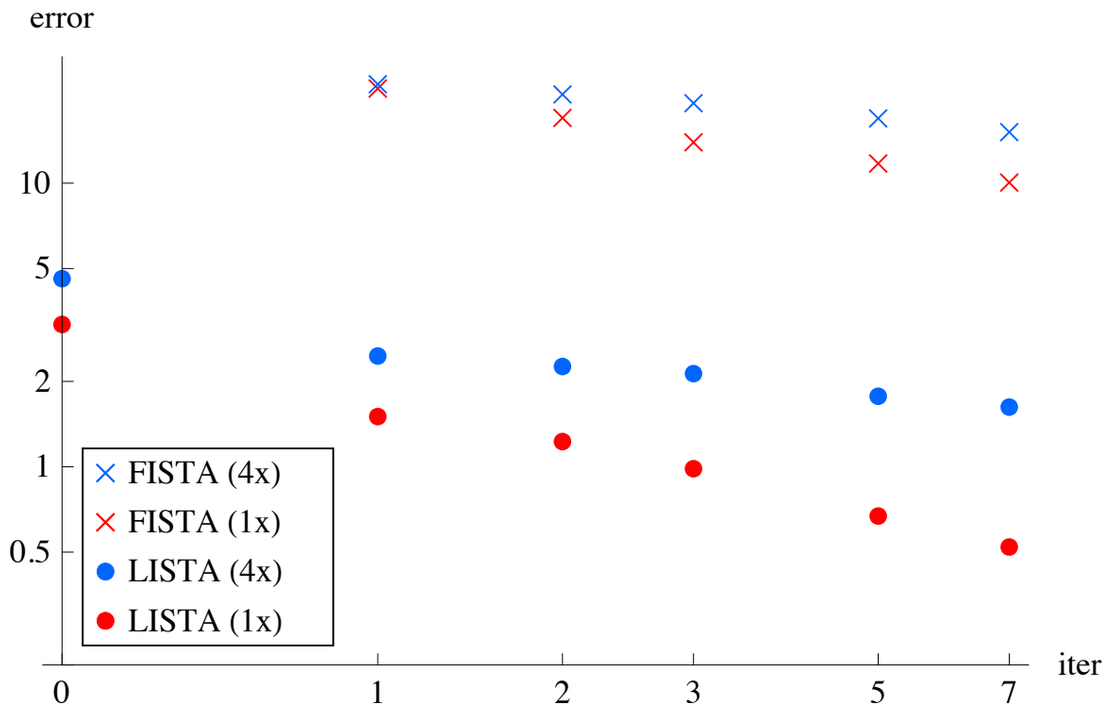
- Explicit Sparse encoder trained to predict the output of the Lasso:

$$\min_{W,S} \frac{1}{n} \sum_{i \leq n} \|\Phi(x_i) - F(x_i, W, S)\|^2$$

LISTA [Gregor and LeCun,'10]

- Explicit Sparse encoder trained to predict the output of the Lasso:

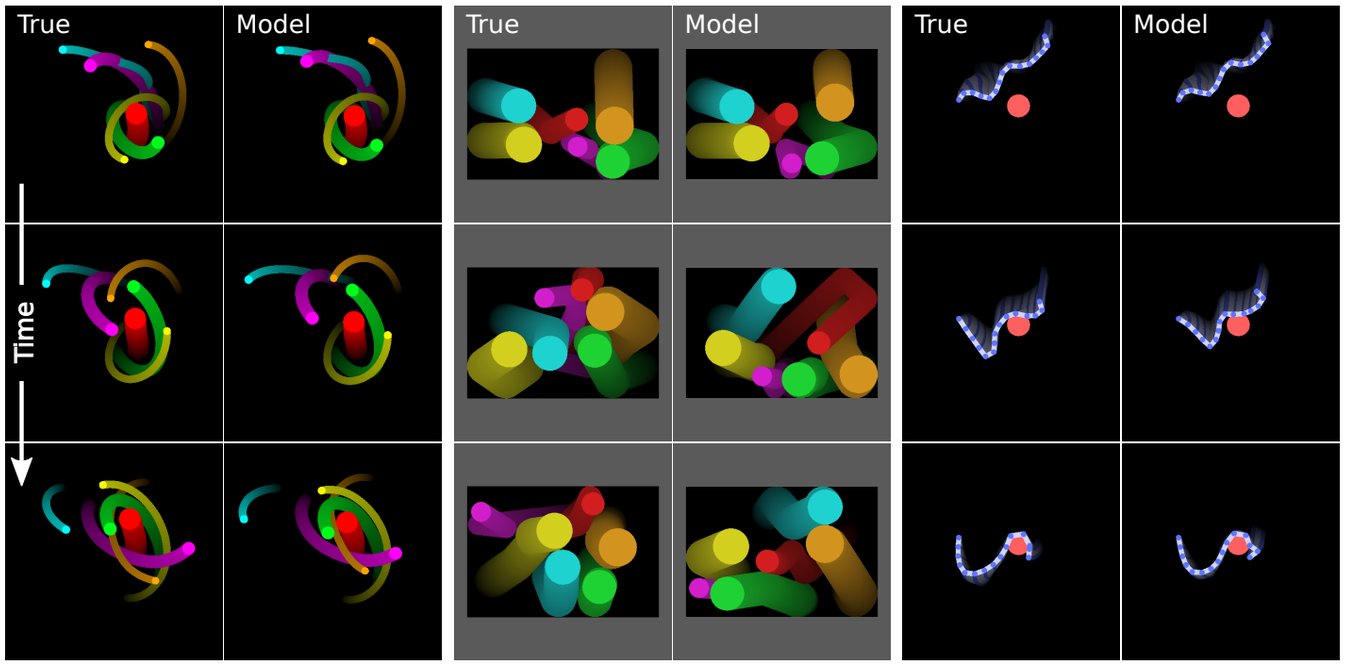
$$\min_{W,S} \frac{1}{n} \sum_{i \leq n} \|\Phi(x_i) - F(x_i, W, S)\|^2$$



- LISTA adapts to the data distribution and produces much faster approximate sparse codes.

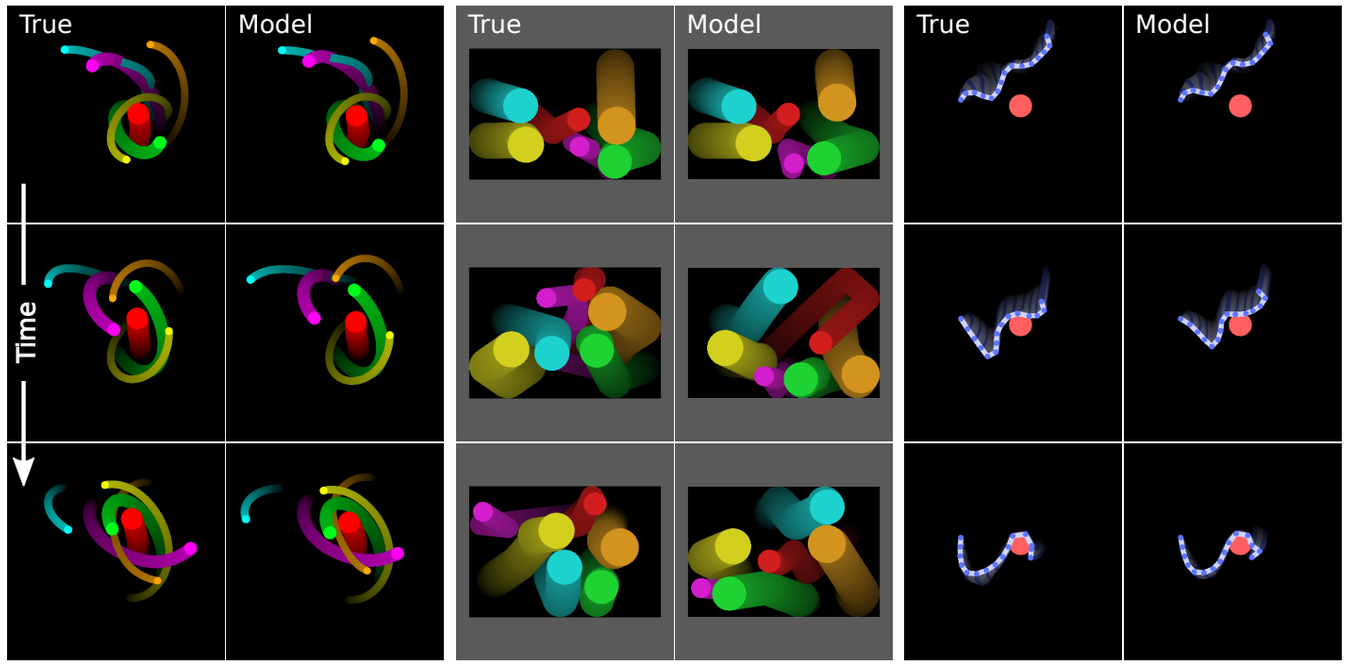
Applications

- N-body Dynamical Systems prediction [Battaglia et al.][Chang et al.'17]
  - Uses graph determined by N-particles and gated GNNs.
  - Also consider edge features.

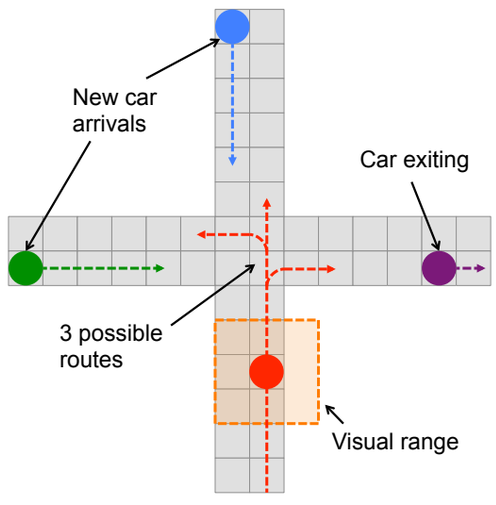


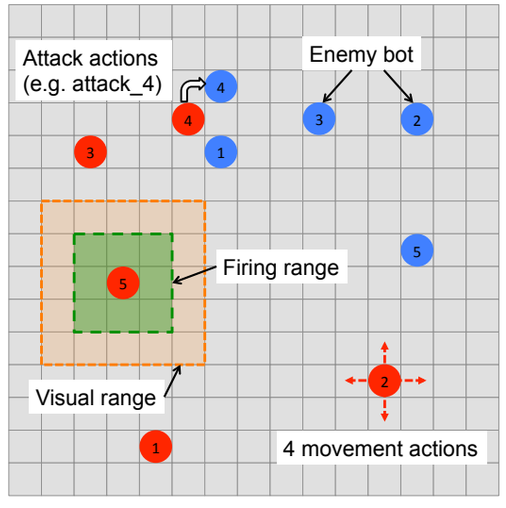
Applications

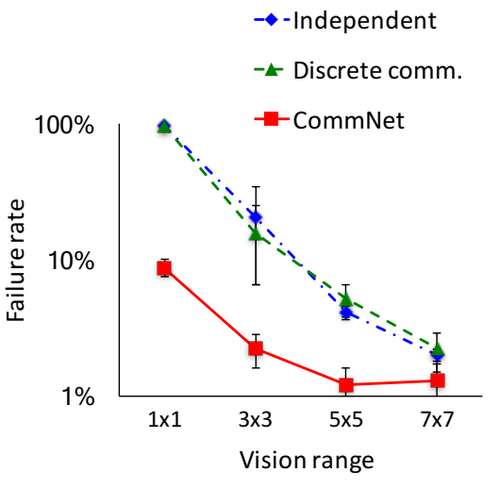
- N-body Dynamical Systems prediction [Battaglia et al.][Chang et al.'17]
  - Uses graph determined by N-particles and gated GNNs.
  - Also consider edge features.



- Learning to Communicate [Sukhbaatar, Szlam, Fergus. '16]
  - Multi-agent Cooperative Environments.
  - Information is propagated in the agent network using a GNN.

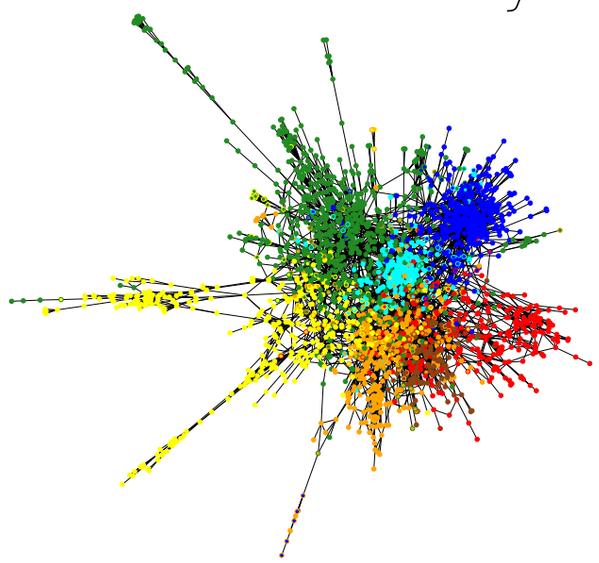






Applications

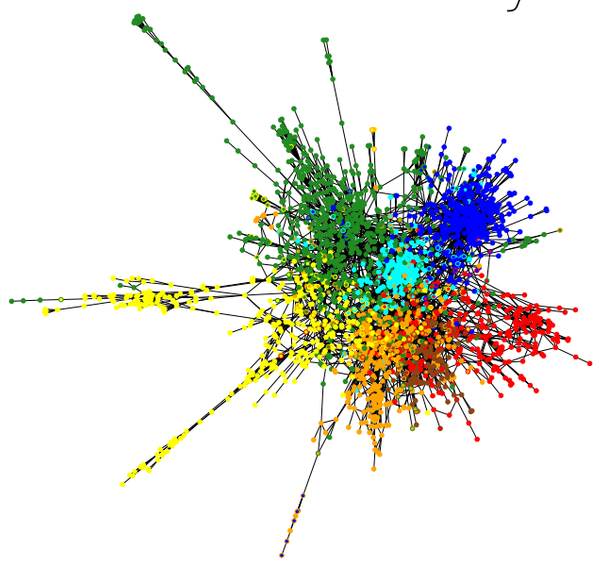
- Semi-Supervised Learning [Kipf & Welling,'17].
  - simplified Laplacian network.
  - state-of-the-art accuracy and efficient.



Method	**Citeseer**	**Cora**	**Pubmed**	**NELL**
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

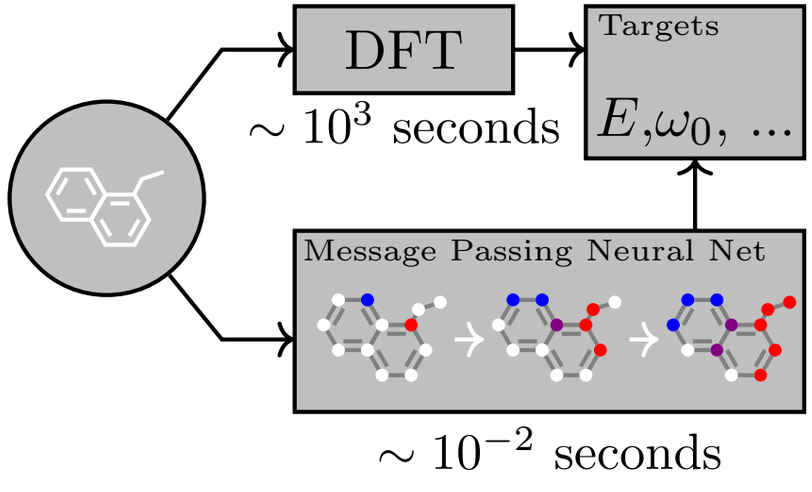
Applications

- Semi-Supervised Learning [Kipf & Welling,'17].
  - simplified Laplacian network.
  - state-of-the-art accuracy and efficient.



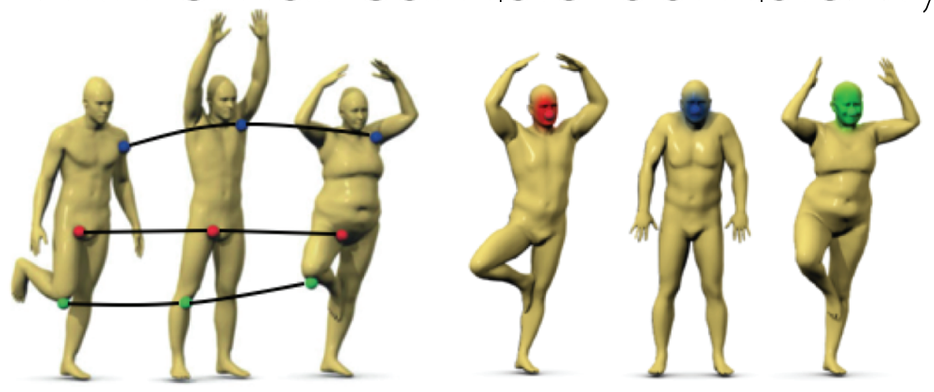
Method	**Citeseer**	**Cora**	**Pubmed**	**NELL**
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

- Quantum Chemistry with GNNs
  - [Gilmer et al.'17][Duvenand et al.'15]
  - State-of-the-art molecular prediction with computational efficiency.

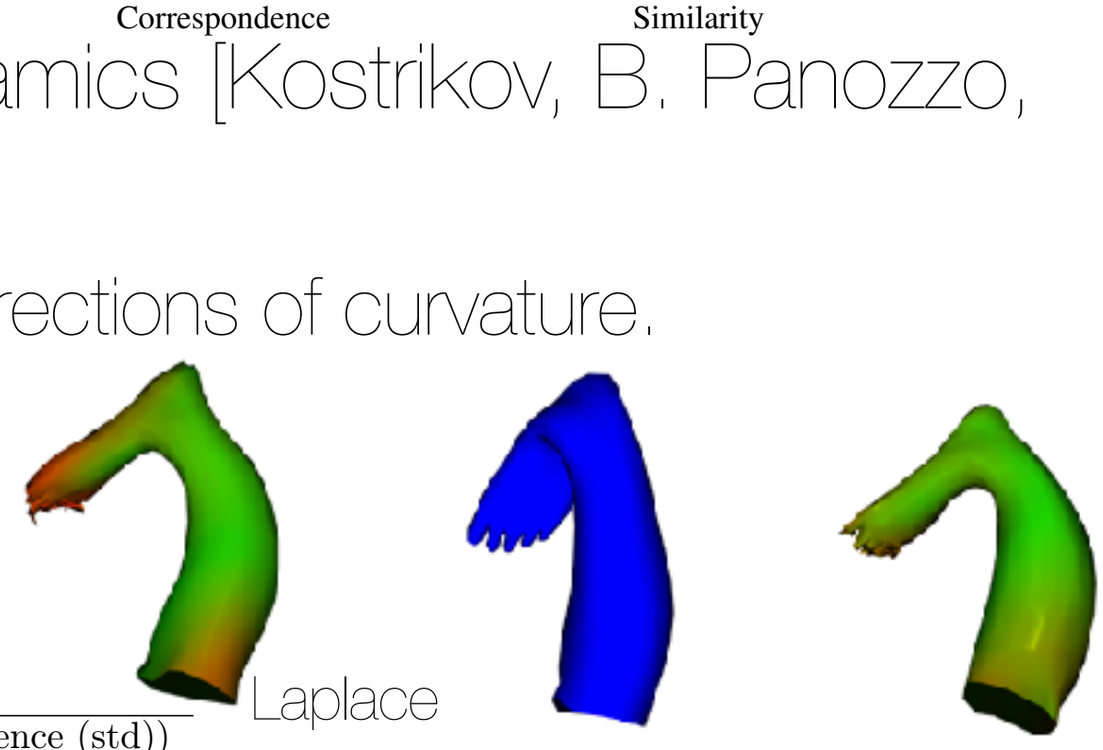


Applications

- Graphics: Shape Correspondence [Bronstein et al.'15,'16].
  - Intrinsic representation using generalized Laplacian network.
  - State-of-the-art correspondence results with small sample complexity.



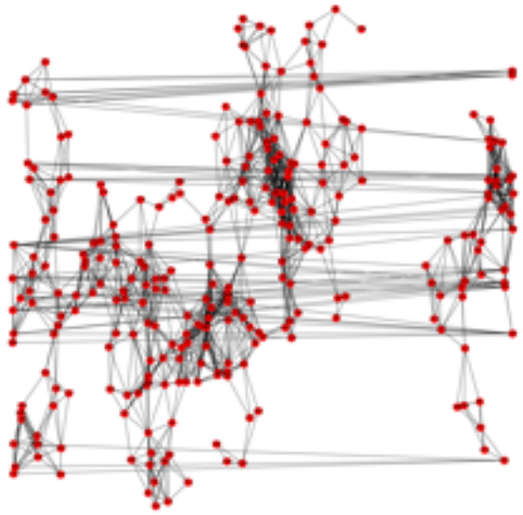
- Graphics: Prediction of non-rigid dynamics [Kostrikov, B. Panozzo, Zorin,'17]
  - Use Dirac operator in order to exploit directions of curvature.
  - Same sample complexity as Laplace.
  - Used for generative modeling.

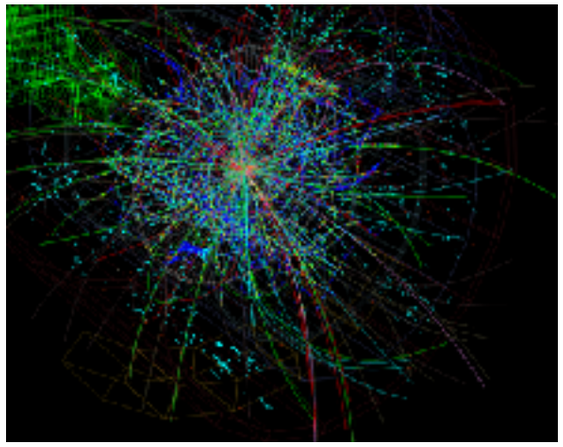


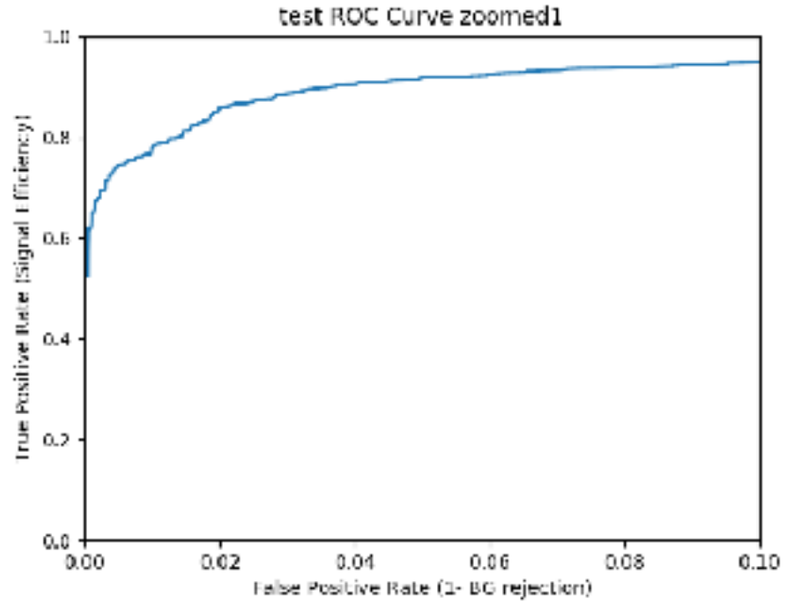
Model	Receptive field	Number of parameters	Smooth L1-loss (mean per sequence (std))
MLP	1	519672	64.56 (0.62)
AvgPool	-	1018872	23.64 (0.21)
Laplace	16	1018872	17.34 (0.52)
Dirac	8	1018872	16.84 (0.16)

Applications

- High-Energy Physics
  - LHC Anomaly Detection [G.Rochette, K.Cramner, G.Louppe & LBL]
    - ❖ Graph determined by energy deposits:







    - ❖ Kernel models physical interactions, but we can learn them too!
  - IceCube Neutrino Detection [P. Bizeul, LBL]
    - ❖ Here Graph is fixed (but irregular).