

Factorization in molecular modeling and belief propagation algorithms

Pu Tian^{*,†,‡}

[†]*School of Life Sciences, Jilin University, Changchun, China 130012*

[‡]*School of Artificial Intelligence, Jilin University, Changchun, China 130012*

E-mail: tianpu@jlu.edu.cn

Phone: +86 (0)431 85155287

Abstract

Factorization reduces computational complexity and is therefore an important tool in statistical machine learning of high dimensional systems. Conventional molecular modeling, including molecular dynamics and Monte Carlo simulations of molecular systems, is a large research field based on approximate factorization of molecular interactions. Recently, the local distribution theory was proposed to factorize global joint distribution of a given molecular system into trainable local distributions. Belief propagation algorithms are a family of exact factorization algorithms for trees and are extended to approximate loopy belief propagation algorithms for graphs with loops. Despite the fact that factorization of probability distribution is their common foundation, computational research in molecular systems and machine learning studies utilizing belief propagation algorithms have been carried out independently with respective track of algorithm development. The connection and differences among these factorization algorithms are briefly presented in this perspective, with the hope to intrigue further development in factorization algorithms for physical modeling of complex molecular systems.

15 Introduction to belief propagation algorithm

Marginalization of high dimensional joint distributions is a difficult, and in many cases intractable task. For a joint distribution $p(\mathbf{x})$ with $\mathbf{x} = (x_1, x_2, \dots, x_n)$, assuming each of x_i is on an alphabet \mathcal{X} and has $|\mathcal{X}|$ possible states, a brute force marginalization to a single variable is an operation of $O(|\mathcal{X}|^n)$ computational complexity. Fortunately, factorization can significantly reduce this cost. $\mathbf{x} = (x_1, x_2, \dots, x_n)$ forms an undirected graph with each variable being a node and an edge between all pairs of interacting variables. In graph theory,¹ a clique is a set of nodes that have edges between each pair of nodes in this set, and a maximum clique is a clique that will be destroyed by addition of a new node into this set. For a graph \mathcal{G} with a set of maximum cliques C :

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in C} f(x_c) \quad (1)$$

with $f(x_c)$ being potentials over variable set c and Z being the partition function $\sum_{\mathbf{x}} \prod_{c \in C} f(x_c)$. In a tree, the maximum clique set is the edge set \mathcal{E} , for the specific example with 5 nodes shown in fig. 1:

$$p(\mathbf{x}) \propto f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) f_{2,4}(x_2, x_4) f_{2,5}(x_2, x_5) \quad (2)$$

The cost bound of marginalization goes from $|\mathcal{X}|^5$ to $4 \times |\mathcal{X}|^2$. For marginal probability $p_1(x_1)$:

$$\begin{aligned} p_1(x_1) &\propto \sum_{x_2, x_3, x_4, x_5} f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) f_{2,4}(x_2, x_4) f_{2,5}(x_2, x_5) \\ &= \sum_{x_2, x_3, x_4} f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) f_{2,4}(x_2, x_4) \underbrace{\sum_{x_5} f_{2,5}(x_2, x_5)}_{\triangleq m_5(x_2)} \\ &= \sum_{x_2, x_3} f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) \underbrace{\sum_{x_4} f_{2,4}(x_2, x_4) m_5(x_2)}_{\triangleq m_4(x_2)} \end{aligned}$$

$$\begin{aligned}
&= \sum_{x_2} f_{1,2}(x_1, x_2) m_4(x_2) \underbrace{\sum_{x_3} f_{1,3}(x_1, x_3)}_{\triangleq m_3(x_1)} \\
&= m_3(x_1) \underbrace{\sum_{x_2} f_{1,2}(x_1, x_2) m_4(x_2)}_{\triangleq m_2(x_1)} \\
&\triangleq m_3(x_1) m_2(x_1)
\end{aligned} \tag{3}$$

¹⁶ $p_1(x_1)$ may be subsequently obtained with normalization shown below:

$$\begin{aligned}
\text{¹⁷ } p_1(x_1) &= \frac{m_2(x_1) m_3(x_1)}{\sum_{x' \in \mathcal{X}} m_2(x') m_3(x')}
\end{aligned} \tag{4}$$

Similarly, for marginal probability $p_3(x_3)$:

$$\begin{aligned}
p_3(x_3) &\propto \sum_{x_1, x_2, x_4, x_5} f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) f_{2,4}(x_2, x_4) f_{2,5}(x_2, x_5) \\
&= \sum_{x_1, x_2, x_4} f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) f_{2,4}(x_2, x_4) \underbrace{\sum_{x_5} f_{2,5}(x_2, x_5)}_{\triangleq m_5(x_2)} \\
&= \sum_{x_1, x_2} f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) \underbrace{\sum_{x_4} f_{2,4}(x_2, x_4) m_5(x_2)}_{\triangleq m_4(x_2)} \\
&= \sum_{x_1} f_{1,3}(x_1, x_3) \underbrace{\sum_{x_2} f_{1,2}(x_1, x_2) m_4(x_2)}_{\triangleq m_2(x_1)} \\
&= \sum_{x_1} f_{1,3}(x_1, x_3) m_2(x_1) \\
&\triangleq m_1(x_3)
\end{aligned} \tag{5}$$

$$\begin{aligned}
\text{¹⁸ } p_3(x_3) &= \frac{m_1(x_3)}{\sum_{x' \in \mathcal{X}} m_1(x')}
\end{aligned} \tag{6}$$

In calculation of both marginals $p_1(x_1)$ and $p_3(x_3)$, the following messages are computed:

$$\begin{aligned} m_5(x_2) &= \sum_{x_5} f_{2,5}(x_2, x_5) \\ m_4(x_2) &= \sum_{x_4} f_{2,4}(x_2, x_4) m_5(x_2) \\ m_2(x_1) &= \sum_{x_2} f_{1,2}(x_1, x_2) m_4(x_2) \end{aligned} \quad (7)$$

Computing marginal for each node in a tree takes $n - 1$ message passing, and consequently $n(n - 1)$ message passing are necessary for n nodes when computed respectively (see Fig 1a,b). However, instead of computing marginals for each variable separately, $2(n - 1)$ messages over $n - 1$ edges of a n -node tree maybe computed so that all marginals obtained without repetitive message computation as illustrated in Fig 1c and demonstrated in the equation below. This message passing is the sum-product, or belief propagation algorithm.²⁻⁴

$$m_{i \rightarrow j}(x_j) = \sum_{x_i} f_{i,j}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}(x_i) \quad (8)$$

The marginals for each variable is obtained by:

$$p_i(x_i) \propto \prod_{j \in N(i)} m_{j \rightarrow i}(x_i) \quad (9)$$

message from i to j ($m_{i \rightarrow j}$) is sent once i receives messages from all of its neighbors $N(i)$ except node j . When a unary potential is included for each node, we have:

$$m_{i \rightarrow j}(x_j) = \sum_{x_i} g_i(x_i) f_{i,j}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}(x_i) \quad (10)$$

$$p_i(x_i) \propto g_i(x_i) \prod_{j \in N(i)} m_{j \rightarrow i}(x_i) \quad (11)$$

One may easily compute the maximum probability configuration \mathbf{x}^* by replacing sum operations in equations 8 or 10 and bookkeeping the corresponding state of each variable in each maximum

operation for backtracking. This algorithm is the max-product algorithm.

Junction tree algorithm, loopy and gaussian belief propagation

In graphs where one or more loop(s) exist(s), the above mentioned belief propagation algorithm does not ensure correct calculation of marginals, and max product algorithm does not ensure obtaining maximum probability configuration. Exact belief propagation maybe realized for general graph by constructing chordal graph through triangulation, which involves adding edges to a graph such that no chordless loop of length greater than 4 exists (see Fig. 2 for a schematic illustration). Maximum cliques in a triangulated graph can be joined to form a junction tree. The most important property of such a tree is the junction tree property, which ensures that if a node x_i is in two maximum cliques c_j and c_k , then it must be a member of every maximum clique that is on the path from c_j to c_k .^{5,6} Computational complexity of a junction tree is determined by the maximum tree width, which is the size of the largest maximum clique. Many different chordal graphs with different tree width may be generated from a given graph with loops, a number of algorithms are available to perform triangulation and construct chordal graphs.⁷⁻¹⁰ However, no algorithm is able to ensure finding the best triangulation scheme corresponding to the (or one of if multiple chordal graphs with the same tree width exist) smallest tree width.

Despite there is no assurance that exact marginals would be obtained by belief propagation algorithm for graphs with loops, we may carry out such message passing computations anyway, and the algorithm is termed loopy belief propagation accordingly. Turbo code¹¹ is one outstanding example of utilizing loopy belief propagation for telecommunication encoding and decoding. From the parallel version¹² of the belief propagation algorithm (see equation 10) shown below:

$$m_{i \rightarrow j}^{t+1} \propto \sum_{x_i} g_i(x_i) f_{i,j}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}^t(x_i) \quad (12)$$

Messages are usually initialized to be 1 and normalized at each time step. If loopy belief propagation converges, then equation 12 suggests that the converged message would be a fixed point of

the messaging updating function. It was proved that the fixed point of belief propagation message updates result in marginals that are some local extrema of the Bethe variational problem.¹³ The details of both Bethe approximation and other treatment of loopy belief propagation is well beyond the scope this perspective and readers are referred to references elsewhere.^{14,15}

Belief propagation was developed for discrete variables initially.² The summation in message computation becomes integration for continuous variable, which is usually intractable for arbitrary distributions. Fortunately, gaussian distributions may be both analytically integrated and resulting in gaussians. This great property may be utilized in belief propagation to treat continuous variables as gaussians. In cases where such naive approximation is not sufficient, linearization with first order Taylor expansion may be utilized to improve accuracy.¹⁶

Factorization in conventional molecular modeling

Conventional molecular simulation is a mature methodology with a wide variety of tools, which are routinely utilized in many research fields including but not limited to chemistry, biology and materials science with more than 30,000 relevant publications each year.¹⁷ The foundation of molecular simulation is a parameter set termed force fields, which represent interaction energy between/among various molecular degrees of freedom as a function of relevant coordinates. Molecular simulation is deemed by many of its practitioners as an independent research field from, while being in fact a well defined form of, statistical machine learning. The force fields parameterization is the learning stage and the simulation and sampling constitute the inference process.

The total energy of a given molecular configuration $\mathbf{x} = (x_1, x_2, \dots, x_n)$ for a simple Lennard-Jones¹⁸ system is usually as shown below:

$$E_{total}(\mathbf{x}) = \sum_{i-j, r_{ij} \leq r_{cut}} 4\epsilon \left(\left(\frac{r_{ij}}{\sigma} \right)^{12} - \left(\frac{r_{ij}}{\sigma} \right)^6 \right) \quad (13)$$

This equation is fundamentally a factorization as shown below:

$$p(\mathbf{x}) \propto \text{Exp}(-E_{\text{total}}) = \prod_{i-j, r_{ij} \leq r_{\text{cut}}} \text{Exp}(-f_{i-j, r_{ij} \leq r_{\text{cut}}}(r_{ij}(x_i, x_j)))$$

$$f_{i-j} = 4\epsilon \left(\left(\frac{r_{ij}}{\sigma} \right)^{12} - \left(\frac{r_{ij}}{\sigma} \right)^6 \right) \quad (14)$$

Let us denote each particle as a node, and the corresponding interaction between a pair of particles as an edge, then each factor representing an interacting pair of particles. This is a factorization that has a very similar form as that in the belief propagation (equation 1). However, there are three major differences if examined carefully. Firstly and most importantly, in equation 1, the graph itself is static with fixed number of edges. However, for Lennard-Jones particles, edges are dynamic in two aspects. One is that as the system propagate with time, edges broke as some pairs of particles drift from within to beyond cutoff and new ones form as some other drift from beyond to within cutoff. The other is that for pairs within cutoff, their interactions change as a function of distance. Secondly, in equation 1, each factor is over a maximum clique in which all pairs of nodes are connected by an edge. For Lennard-Jones particles, when the cut-off for direct interaction is taken as 3σ , all particles within a sphere of radius of cutoff are certainly within a single clique as shown in Fig. 3a. Factorization into pairwise terms is apparently an approximation, the resulting error is not easy to quantify for arbitrarily complex interactions. Certainly exact number of particles in each maximum clique can be different as Lennard-Jones particles usually do not form a strict single crystal and local packing continuously change. Thirdly, nodes in equation 1 have discrete states, while properties of Lennard-Jones particles (e.g. position and momentum) are continuous.

For more general molecular systems, energy contributions are more complex^{19–21} and usually calculated as shown below:

$$E_{\text{total}} = E_{\text{bonding}} + E_{\text{bending}} + E_{\text{dihedral}} + E_{\text{vdW}} + E_{\text{Electro}} \quad (15)$$

$$= \sum_{\text{bond}_i} E_{\text{bond}_i} + \sum_{\text{bend}_j} E_{\text{bend}_j} + \sum_{\text{dihedral}_k} E_{\text{dihedral}_k} + \sum_{\text{VdW}_l} E_{\text{VdW}_l} + \sum_{\text{esi}} E_{\text{ES}_{esi}} \quad (16)$$

Here i, j, k, l, esi are indices for bonding, bending, dihedral, Van de Waals and electrostatic interactions respectively. Take exponentiation of this equation:

$$\begin{aligned} \text{Exp}(-E_{total}) = & \prod_{bond_i} \text{Exp}(-E_{bond_i}) \prod_{bend_j} \text{Exp}(-E_{bend_j}) \\ & \times \prod_{dihedral_k} \text{Exp}(-E_{dihedral_k}) \prod_{VdW_l} \text{Exp}(-E_{VdW_l}) \prod_{esi} \text{Exp}(-E_{ES_{esi}}) \end{aligned} \quad (17)$$

Bonded terms, including that accounting for bonds, bends and dihedrals/impropers, are fundamentally local. Van der Waals interactions are usually modeled by LJ potentials. So similar to simple LJ systems, there is approximation of representing interactions in maximum cliques by assumed independent local interactions, challenges of continuous variables with non-gaussian distributions and dynamic edges. More importantly, there are usually partial charges of various magnitude in atoms and resulting long-range electrostatic interactions. Should such interactions to be treated exactly, then factorization is not possible and a molecular system need to be treated as one single maximum clique. The exponential complexity with respect to the size of the maximum clique renders exact treatment intractable in many realistic molecular systems. So treating long range interactions as being independent from local ones is an essential approximation in computational molecular science. In reality, naive pairwise calculation of electrostatic interaction diverges and usually some form of Ewald summation is utilized to ensure convergence.^{22,23} Fortunately, in molecular simulations, treating long-range interactions as being independent from local interactions has been widely tested with great successes.²⁴ Additionally, maximum cliques usually form loops in condensed molecular systems, and molecular degrees of freedom usually do not have gaussian distributions. Therefore, exact inference is not tractable even if only local interactions contribute to molecular correlations.

Factorization in the local distribution theory

Just as in the case of repetitive local computation exist in the calculation of marginals as demonstrated in equations 3, 5 and 7. Ubiquitous repetitive local sampling (RLS) exist in molecular simulations.^{17,25,26} From a graph perspective, such RLS is fundamentally associated with the intrinsic dynamic property of graphs describing molecular systems, in contrast to conventional static graph as studied by sum-product algorithm. The local distribution theory (LDT) was developed to address RLS and tremendously increase computational efficiency without reducing accuracy/resolution.^{17,26} LDT is fundamentally a new factorization scheme for facilitating statistical machine learning for dynamic graphs in high dimensional space as shown in the following:

$$\begin{aligned}
 P(\Phi, \mathbf{x}) &= Q(\Phi, R) \\
 &= \frac{Q(\Phi, R)}{\prod_{i=1}^m q(\Phi, r_i)} \prod_{i=1}^m q(\Phi, r_i) \\
 &\approx \prod_{i=1}^m q(\Phi, r_i) \exp(-\sum F_{MED}(\Phi, R)) \exp(-\sum F_{LR}(\Phi, R))
 \end{aligned} \tag{18}$$

Here, $\Phi = (\phi_1, \phi_2, \dots, \phi_k)$ is relevant thermodynamic and environmental variables, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ are molecular coordinates, $R = (r_1, r_2, \dots, r_m)$ ($m \leq n, m = n$ is preferred) are dynamic local regions, each of which represent a dynamic collection of coordinates ($r_i = \mathcal{M}(x_{i1}, x_{i2}, \dots, x_{il})$) for molecular degrees of freedom. \mathcal{M} is a translation, rotation and permutation invariant coordinate transformation matrix that transforms from global to local coordinates. $\prod_{i=1}^m q(\Phi, r_i)$ is the product of local distributions and the global correlation factor $\frac{Q(\Phi, R)}{\prod_{i=1}^m q(\Phi, r_i)}$ is approximately factorized into mediated ($\sum F_{MED}(\Phi, R)$) and long range ($\sum F_{LR}(\Phi, R)$) interactions. The learning stage of LDT is to train local distributions, which cache statistical significance of various molecular configurations within local spaces defined by a preset cutoff from the origin of a local coordinates originating from a given molecular DOF. Such local distributions, defined by fully trainable neural network parameters in combination with selected neural network architectures, store both energetic and

entropic contributions from extensive local sampling data of either experimental or computational origin. By caching and repetitively utilizing such local distributions, LDT eliminates RLS to a great extent and therefore saves computational resources in sampling dynamic graphs, similarly as belief propagation algorithm saves computation in static graphs. In the inference stage, local distributions are assembled under sampling constraints due to mediated interactions and correction of direct long range interactions if exist.

As illustrated in Fig.3b, in contrast to conventional molecular simulations, in which each pair of interaction (or an edge in graph representing molecular system) is included as an independent factor (Fig. 3a), both local interactions confined within cutoff spherical space and mediated interactions are jointly considered in LDT. Independence assumptions are therefore significantly weaker and accordingly, improvement of accuracy is expected simultaneously with tremendously increased efficiency due to substitution of RLS by local distributions. Therefore, LDT has great potential to address the tradeoff dilemma between accuracy and efficiency in more than half century of molecular simulations. However, this does come with a significant one-time price to pay, which is to accurately fitting local distributions. This is fundamentally a density estimation task in high dimensional system, which is an open problem with many feasible approaches with respective limitation.^{27,28}

Distinction and potential synergy among different factorization schemes

The fundamental difference between factorization in belief propagation algorithm and that in molecular modeling of physical systems is that the former deals with a static graph (V, E) while the latter addresses a dynamic one $(V, E_{dynamic})$. The variation of edges in molecular systems is continuous but may be simplified as binary as in the case of protein contact map.² Specifically, as the distance between two atoms changes over time, so is the corresponding factor potential. In belief propagation, an isolated maximum clique has a fixed factor potential, the computation is to fig-

ure out how interaction with neighboring cliques influence distribution of internal states of nodes with respective factor potentials being constant. Belief propagation algorithms reduces repetitive local computation in redistribution of marginals for internal node states by nodes in surrounding cliques. LDT reduces repetitive local sampling of dynamic edges. These distinct ways of factorization may be combined in different ways to facilitate statistical inference of high dimensional molecular physical systems. The first step of utilizing both factorization is definition of nodes with internal states. For example, in full quantum mechanical treatment, one may define each atom as a node with rich internal electronic states. For proteins, residues may be defined as distinct nodes with internal rotameric states. Node internal states may be approximately discretely or continuously represented. Apparently, definition of nodes is very flexible and requires significant effort to explore with given type of molecular systems. For a given definition of nodes, at each instant, we may perform loopy belief propagation to search for the maximum likelihood global configuration (or a number of configurations with the most statistical weights) of internal node states, followed by sampling of different edge configuration with LDT and properly selected global sampling strategy. It is likely that definition of nodes, selection of various approximate loopy belief propagation algorithms and selection of global sampling strategy may interact. Exploring different combinations of these efficient algorithms are likely fertile grounds for development of next generation molecular modeling methodology.

Conclusions and prospects

Belief propagation is a family of algorithms in a large class of factorization methodology called probabilistic graphical models with focus on static graphs. These algorithms are mainly developed by informational scientist and applied in coding, communication and machine learning. Conventional molecular simulations and the local distribution theory, are fundamentally factorization with focus on dynamic graphs representing complex molecular systems, and are mainly developed computational chemists and physicists. Both are extensive research fields with very rich contents and

numerous important contributions from many scientists. Comprehensive summary of these two fields is apparently well beyond the scope of this work. This perspective is intended to bridge this discipline gap, and intrigue interest for development of next generation algorithms that assimilate the best of both fields.

Acknowledgement

I thank Professor Zhonghan Hu for comments on the local distribution theory when the manuscript was in preparation.

References

- (1) Trudeau, R. J. *Introduction to graph theory*; Dover publications, 1994.
- (2) Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. *SIAM Review* **1990**, 32, 704–707.
- (3) Kschischang, F.; Frey, B.; Loeliger, H.-A. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* **2001**, 47, 498–519.
- (4) Koller, D.; Friedman, N. *Probabilistic Graphical Models, Principles and Techniques*; MIT Press, 2009.
- (5) Lauritzen, S. L.; Spiegelhalter, D. J. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *Journal of the Royal Statistical Society. Series B (Methodological)* **1988**, 50, 157–224, Full publication date: 1988.
- (6) Jensen, F.; Lauritzen, S.; Olesen, K. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly* **1990**, 4, 269–282.

- (7) Gogate, V.; Dechter, R. A Complete Anytime Algorithm for Treewidth. Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence. Arlington, Virginia, USA, 2004; p 201?208.
- (8) E.H., B.; H.L., B. In *Algorithmic Aspects in Information and Management.*; SW., C., C.K., P., Eds.; Lecture Notes in Computer Science, vol 4041.; Springer, 2006.
- (9) Ottosen, T. J.; Vomlel, J. All roads lead to Rome—New search methods for the optimal triangulation problem. *International Journal of Approximate Reasoning* **2012**, 53, 1350–1366, Fifth European Workshop on Probabilistic Graphical Models (PGM-2010).
- (10) Li, C.; Ueno, M. An extended depth-first search algorithm for optimal triangulation of Bayesian networks. *International Journal of Approximate Reasoning* **2017**, 80, 294–312.
- (11) Berrou, C.; Glavieux, A. *Wiley Encyclopedia of Telecommunications*; American Cancer Society, 2003.
- (12) Gonzalez, J.; Low, Y.; Guestrin, C. Parallel Splash Belief Propagation. *Journal of Machine Learning Research* **2009**, 1, 1,48.
- (13) Yedidia, J. S.; Freeman, W. T.; Weiss, Y. Generalized belief propagation. NIPS'00: Proceedings of the 13th International Conference on Neural Information Processing System. 2000; pp 668–674.
- (14) Kumar, M. P.; Torr, P. H. S. *Computer Vision ? ECCV 2006*; Lecture Notes in Computer Science, volume 3954; Springer, pp 451–463.
- (15) Chen, S. Y.; Tong, H.; Wang, Z.; Liu, S.; Li, M.; Zhang, B. Improved Generalized Belief Propagation for Vision Processing. *Mathematical Problems in Engineering* **2010**, 2011, 416963.
- (16) Ortiz, J.; Evans, T.; Davison, A. J. A visual introduction to Gaussian Belief Propagation. *arXiv preprint arXiv:2107.02308* **2021**,

- (17) TIAN, P. The repetitive local sampling and the local distribution theory. *ChemRxiv* **2021**,
- (18) Wang, X.; Ramírez-Hinestrosa, S.; Dobnikar, J.; Frenkel, D. The Lennard-Jones potential: when (not) to use it. *Phys. Chem. Chem. Phys.* **2020**, *22*, 10624–10633.
- (19) Brooks, B. R. et al. CHARMM: The biomolecular simulation program. *Journal of Computational Chemistry* **2009**, *30*, 1545–1614.
- (20) Case, D. A.; Cheatham III, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz Jr., K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber biomolecular simulation programs. *Journal of Computational Chemistry* **2005**, *26*, 1668–1688.
- (21) Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. C. GROMACS: Fast, flexible, and free. *Journal of Computational Chemistry* **2005**, *26*, 1701–1718.
- (22) Toukmaji, A. Y.; Board, J. A. Ewald summation techniques in perspective: a survey. *Computer Physics Communications* **1996**, *95*, 73–92.
- (23) Pan, C.; Hu, Z. Rigorous Error Bounds for Ewald Summation of Electrostatics at Planar Interfaces. *Journal of Chemical Theory and Computation* **2014**, *10*, 534–542.
- (24) French, R. H. et al. Long range interactions in nanoscale science. *Rev. Mod. Phys.* **2010**, *82*, 1887–1944.
- (25) Cao, X.; Tian, P. Molecular free energy optimization on a computational graph. *RSC Adv.* **2021**, *11*, 12929–12937.
- (26) Cao, X.; Tian, P. “Dividing and Conquering” and “Caching” in Molecular Modeling. *International Journal of Molecular Sciences* **2021**, *22*.
- (27) Wang, Z.; Scott, D. W. Nonparametric density estimation for high-dimensional data—Algorithms and applications. *WIREs Computational Statistics* **2019**, *11*, e1461.

- ²⁵¹ (28) Liu, Q.; Xu, J.; Jiang, R.; Wong, W. H. Density estimation using deep generative neural
²⁵² networks. *Proceedings of the National Academy of Sciences* **2021**, *118*.

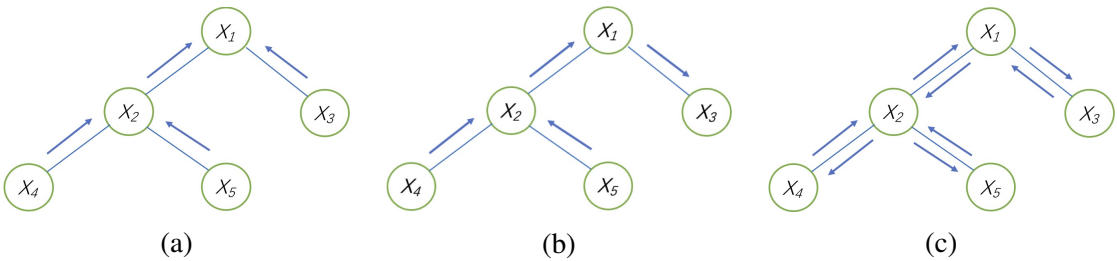


Figure 1: A schematic illustration of marginalization in examples. (a)equation 3 for marginal $p_1(x_1)$, (b) equation 5 for marginal $p_3(x_3)$ and (c)equation 8 for belief propagation.

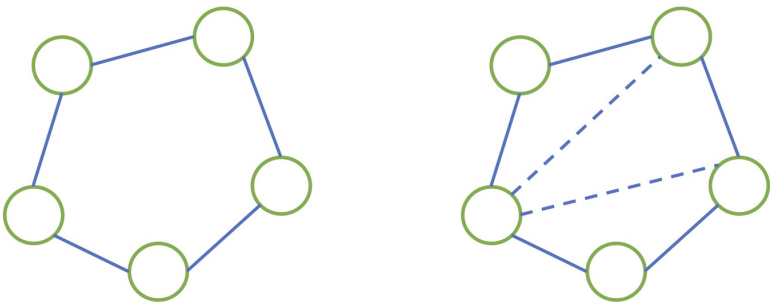


Figure 2: A graph with a loop (left) triangulated with addition of two edges and become chordal (right).

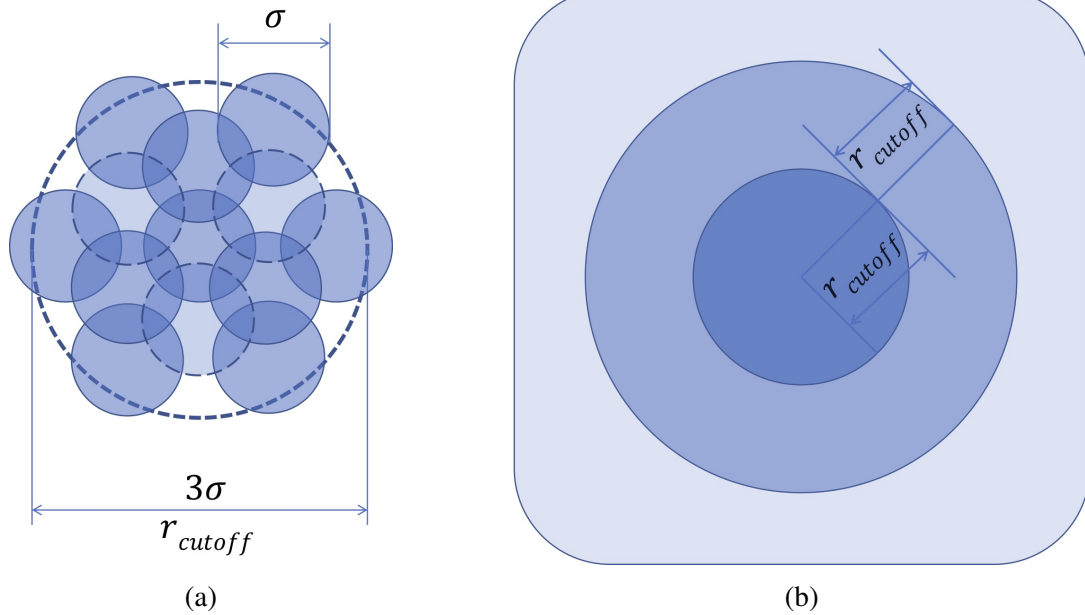


Figure 3: (a) An example of 13 Lennard-Jones particles of diameter σ enclosed within a sphere of diameter 3σ (r_{cutoff}). Three particles partially covered by other particles are shown with dashed circles. These 13 particles form a clique with 78 edges, all should be represented by a single factor/potential for exact inference while in conventional molecular simulation, 78 edges correspond to 78 factors. (b) A schematic illustration of LDT theory in which each spherical space of radius r_{cutoff} is represented as a single factor, a central particle experiences mediated interactions from particles beyond r_{cutoff} but within $2r_{cutoff}$, only interactions further than $2r_{cutoff}$ are treated as long-range interactions.